Towards More Efficient, Robust, Instance-adaptive, and Generalizable Online Learning

WANG, Zhiyong

A Thesis Submitted in Partial Fulfilment
of the Requirements for the Degree of
Doctor of Philosophy
in
Computer Science and Engineering

The Chinese University of Hong Kong
April 2025

Thesis Assessment Committee

Professor FARNIA Farzan (Chair)

Professor LUI Chi Shing John (Thesis Supervisor)

Professor YU Bei (Committee Member)

Professor LUO Xiapu Daniel (External Examiner)

Abstract of thesis entitled:

Towards More Efficient, Robust, Instance-adaptive, and Generalizable Online Learning

Submitted by WANG, Zhiyong
for the degree of Doctor of Philosophy
at The Chinese University of Hong Kong in April 2025

The primary goal of my research is to develop provably efficient and practical algorithms for data-driven online sequential decision-making under uncertainty. My work focuses on reinforcement learning (RL), multi-armed bandits, and their applications, including recommendation systems, computer networks, video analytics, and large language models (LLMs). Online learning methods, such as bandits and RL, have demonstrated remarkable success—ranging from outperforming human players in complex games like Atari and Go to advancing robotics, recommendation systems, and fine-tuning LLMs.

Despite these successes, many established algorithms rely on idealized models that can fail under model misspecifications or adversarial perturbations, particularly in settings where accurate prior knowledge of the underlying model class is unavailable or where malicious users operate within dynamic systems. These challenges are pervasive in real-world applications, where robust and adaptive solutions are critical. Furthermore, while worst-

case guarantees provide theoretical reliability, they often fail to capture instance-dependent performance, which can lead to more efficient and practical solutions. Another key challenge lies in generalizing to new, unseen environments, a crucial requirement for deploying these methods in dynamic and unpredictable settings. To address these important issues, my research aims to address these limitations by driving the field toward

more efficient, robust, instance-adaptive, and generalizable online learning.

Towards this end, I focus on developing more efficient, robust, instance-adaptive, and generalizable for both general reinforcement learning (RL) and bandits.

1. Efficient, Instance-adaptive, and Generalizable Reinforcement Learning: Reinforcement Learning (RL) has achieved significant breakthroughs in various applications, from game playing to autonomous systems. However, two major challenges persist in the field: developing algorithms that provide efficient, instance-adaptive performance guarantees and ensuring that these algorithms can generalize effectively to new, unseen environments. Current state-of-the-art RL methods often rely on worst-case performance analyses, which can be overly conservative and fail to leverage the specific structure of individual problems. Additionally, many RL algorithms struggle with generalization, particularly in offline settings where the agent must perform well in environments that differ from the training data. Addressing these challenges is crucial for advancing RL theory and enabling its application in more complex and dynamic real-world scenarios.

I proved that surprisingly standard model-based RL approaches can achieve horizon-free and variance-dependent regret bounds [1], contributing significantly to the RL theory community by identifying the simplest approach for achieving such tight bounds in large-scale online and offline RL problems with general function approximation. Furthermore, I made substantial contributions to the area of zero-shot generalization in offline RL [2]. Previous empirical works [3] demonstrated that standard offline RL algorithms struggle to generalize to new, unseen environments. I initiated the first theoretical analysis in this area, identifying the causes of these failures and proposing provably efficient offline RL algorithms that address these challenges, which are supported by significant improvements in large-scale experiments over prior methods. Details are as follows.

- **Minimalist Approach to Horizon-Free and Second-Order Bounds** [1]: We are the first to identify the minimalist algorithms and analyses to achieve horizon-free and instance-dependent (second-order) bounds for both online and offline RL with general function approximations. "Horizon-free" implies that our bounds do not depend polynomially on the Markov Decision Process horizon. Our second-order bounds scale with the variances of the policy returns, which can be small when the system is nearly deterministic or the optimal policy has small values. These bounds offer significant tight and instance-dependent theoretical guarantees for efficient RL. *This work was recently selected as a reference in Cornell's CS 6789: Foundations of Reinforcement Learning*

course.

- **Zero-Shot Generalization in Offline RL** [2]: We studied offline RL with zero-shot generalization (ZSG), where the agent accesses an offline dataset from various environments and aims to perform well on unseen test environments without further interaction. We proposed novel frameworks to find near-optimal policies with ZSG, providing both nearly optimal theoretical guarantees (tight upper bounds of suboptimality gaps) and empirical validations (significant outperformance over previous offline RL methods on the real-world Procgen dataset). Our frameworks represent a significant advancement in understanding and enhancing generalization in offline reinforcement learning.

2. Efficient, Robust, and Instance-adaptive Multi-armed Bandits: Multi-armed bandits (MAB) are fundamental tools for sequential decision-making under uncertainty, with widespread applications in areas such as recommendation systems, online advertising, and user engagement. However, existing algorithms often struggle in real-world scenarios involving model misspecifications, adversarial corruptions, or dynamic environments, making the development of efficient and robust methods a critical research direction.

I have made significant contributions towards more efficient, robust, and instance-adaptive multi-armed bandit algorithms, particularly for real-world applications like recommendation systems. I designed provably efficient large-scale bandit algorithms that are robust to model misspecifications [4], adversarial corruptions [5],

and preference feedback [6], scenarios where classic algorithms often fail. Motivated by real-world applications of bot detection proposed by my collaborators at Adobe Research, I also pioneered the study of online detection of malicious users in dynamic systems [5], an important but open research area. Additionally, I have contributed to the field of conversational contextual bandits, which are widely applied to conversational recommendation systems [7, 8, 9]. Details are as follows.

- **Robust Clustering in Bandits** [4]: Clustering of bandits (CB) utilizes similarities over user preferences and has shown significant success in large-scale recommender systems. We addressed the limitations of existing CB algorithms that require well-specified linear user models. We developed robust CB algorithms accommodating inaccurate user preference estimations and erroneous clustering due to model misspecifications. Our algorithms achieve tight regret upper bounds matching the lower bounds up to logarithmic factors and have significant empirical improvements in real-world datasets for recommendation systems.

- **Online Malicious User Detection** [5]: Recognizing challenges such as click fraud, fake reviews, and bot detection, we introduced a novel online learning problem named LOCUD. Our work is the first to address the dual objectives of performing online detection of malicious users and minimizing regret by learning and leveraging unknown user relations inferred from disrupted behaviors. We proposed general frameworks that demonstrate both strong theoretical guarantees

—nearly optimal regret bounds and tight detection accuracy—and robust experimental performance, achieving high rewards in recommender systems and online detection accuracy comparable to state-of-the-art offline deep learning-based methods.

- **Conversational Contextual Bandits** [7, 8, 9]: We explored conversational contextual bandits, which accelerate learning in recommendation systems by eliciting user preferences through occasional queries for explicit feedback on key terms. We proposed the ConLinUCB framework, achieving better incorporation of arm-level and key-term-level feedback. Our algorithms, ConLinUCB-BS and ConLinUCB-MCR, achieve state-of-the-art performance both theoretically and empirically (up to 54% improvement in learning accuracy and up to 72% improvement in computational efficiency over previous SOTA methods) [7]. We also studied various different settings and proposed corresponding robust algorithms for conversational bandits [8, 9].

- **Variance-Adaptive Regret in Non-Stationary Bandits** [10]: We investigated non-stationary stochastic linear bandits with evolving reward distributions. We proposed algorithms that utilize the variance of the reward distribution, introducing Restarted WeightedOFUL$^+$ and Restarted SAVE$^+$, which achieve variance-dependent bounds. When the total variance V_K is smaller than the total round K, our algorithms outperform previous state-of-the-art results.

- **Clustering of Bandits with Preference Feedback** [6]: We introduce the first "clustering of dueling bandit algorithms" to enable collaborative decision-making based on preference feedback. We propose two novel algorithms: (1) Clustering of Linear Dueling Bandits (COLDB) which models the user reward functions as linear functions of the context vectors, and (2) Clustering of Neural Dueling Bandits (CONDB) which uses a neural network to model complex, non-linear user reward functions. Both algorithms are supported by rigorous theoretical analyses, demonstrating that user collaboration leads to improved regret bounds. Extensive empirical evaluations on synthetic and real-world datasets further validate the effectiveness of our methods.
- **Other Collaborative Works** [11, 12, 13, 14, 15]: I also contributed to projects on applications of bandits in computer networks for adaptive congestion control [11], theory of combinatorial bandits [13], federated bandits for recommendation systems [14], and the safety study of adversarial attacks on bandits [15], broadening the impact of online learning methods in these domains.

論文題目 ： 更高效、更穩健、具實例自適應性與泛化能力的線上學習
作者 ： 王智勇
學校 ： 香港中文大學
學系 ： 計算機科學與工程學系
修讀学位 ： 哲學博士
摘要 ：

我的研究主要目標是設計具理論保證且實用的演算法，用於不確定性下的數據驅動線上序列決策。我專注於強化學習（Reinforcement Learning, RL）、多臂老虎機（Multi-armed Bandits, 多臂老虎機）以及其在推薦系統、電腦網路、影像分析與大型語言模型（Large Language Models, LLMs）等應用領域。線上學習方法（如多臂老虎機與 RL）已在眾多場景中展現驚人成效——從在 Atari 與圍棋等複雜遊戲中超越人類玩家，到推動機器人技術、推薦系統及大型語言模型的精調等應用。

儘管已有這些成功，許多現有演算法仍依賴理想化的模型假設，這些假設在模型錯置（misspecification）或對手擾動下可能失效，特別是在難以獲得正確模型先驗知識的環境或在惡意使用者存在的動態系統中。這些挑戰在真實應用中普遍存在，因此穩健與自適應的解決方案至關重要。此外，儘管最壞情形理論保證具備理論可靠性，但其往往無法反映樣本實例的特徵，從而限制了效率與實用性。另一個關鍵挑戰是如何對新環境進行泛化，這是將這些方法部署於動態與不可預測情境中的必要

條件。為解決上述挑戰，我的研究目標是推動該領域邁向：

更高效、更穩健、具實例自適應性與泛化能力的線上學習

因此，我針對強化學習與多臂老虎機兩大類問題，設計具備上述四種性質的演算法。

1. **高效、自適應並具泛化能力的強化學習：**強化學習（RL）在從遊戲到自動化系統的多項應用中取得突破性成果。但現階段仍面臨兩大挑戰：一是設計具實例自適應性與高效率的演算法，二是確保這些演算法能有效泛化至新環境。現有 RL 方法多仰賴最壞情形分析，往往過於保守，無法有效利用具體問題的結構特性。此外，許多 RL 演算法在離線設定下難以泛化，導致在測試環境表現不佳。解決這些問題對於推動 RL 理論與實踐至關重要。

我證明，即便是標準的模型式 RL 方法，也能實現無視時間尺度（horizon-free）與變異數相關（variance-dependent）的遺憾界限（regret bound）[1]。此成果為 RL 理論社群帶來重要貢獻，證明即使是最簡的演算法，在大規模的線上與離線 RL 問題下也能達到緊緻的界限。此外，我也對離線 RL 中的零樣本泛化問題（zero-shot generalization）做出重要貢獻 [2]。過去的實證研究指出標準 RL 難以泛化 [3]，我首次從理論角度分析此問題，並提出具理論保證且在大型實驗中顯著優於現有方法的解決方案。具體成果如下：

- **簡約方法實現無視時間與次序界限** [1]：我們首度識別出一種極簡分析框架與演算法，能針對具有函數逼近的一般 RL 問題實現無 horizon 與實例自適應的次序界限（second-order bounds）。無 horizon 意味著界限與 MDP 的時間長度無關，而次序界限會依據策略回報的變異程度縮放，特

別當系統近乎確定性或最優策略回報值很小時界限更緊。此成果已被康乃爾大學 CS 6789 課程指定為參考資料。

- **離線強化學習的零樣本泛化** [2]：我們研究了離線 RL 中的零樣本泛化（ZSG）問題，代理只從不同環境的資料集中學習，需在未知測試環境中直接執行。我們提出新的架構與理論保證（子最優性差距的上限），並在大型真實資料集（如 Procgen）上顯著優於前人方法。

2. 高效、穩健且自適應的多臂老虎機方法：多臂老虎機問題是處理不確定性下序列決策的核心方法，廣泛應用於推薦系統、線上廣告與使用者互動分析等領域。但現有演算法在面對模型錯置、對手擾動與非平穩環境時表現不佳，因此設計穩健且有效的演算法尤為關鍵。

我針對上述問題提出多項具理論保證且兼具實用性的多臂老虎機演算法，特別針對推薦系統的應用。我設計了可處理模型錯置 [4]、對手破壞 [5] 及偏好回饋 [6] 等場景下仍具有效率的演算法，並首度探討動態系統中惡意使用者的即時偵測 [5]，這是現實中重要但未充分研究的問題。我亦參與了會話式多臂老虎機問題的研究，廣泛應用於交互式推薦系統中 [7, 8, 9]。具體成果如下：

- **多臂老虎機聚類的穩健性** [4]：我們研究了多臂老虎機聚類（Clustering of Bandits, CB）方法，該方法透過使用者偏好間的相似性來提升推薦效果。我們提出能容忍模型錯置與錯誤聚類的穩健演算法，並給出對應理論遺憾上限與實驗驗證。

- **線上惡意使用者偵測** [5]：面對點擊詐欺、假評論與機器人等行為，我們提出 LOCUD 問題並首次處理雙重目標：一

是識別惡意使用者，二是降低決策遺憾。我們提出具理論與實驗保證的通用架構，在推薦系統中同時達到高獎勵與接近離線深度學習方法的偵測精度。

- **會話式多臂老虎機** [7, 8, 9]：我們研究會話式多臂老虎機問題，透過詢問使用者關鍵字以加速偏好學習。我們提出 ConLinUCB 架構，整合 arm 與關鍵詞回饋，提出 ConLinUCB-BS 與 ConLinUCB-MCR 等方法，在學習精度與運算效率上顯著優於現有方法。

- **變異數適應的非平穩多臂老虎機** [10]：我們研究非平穩線性多臂老虎機問題，提出能利用獎勵變異特性的演算法，包括 Restarted WeightedOFUL$^+$ 與 Restarted SAVE$^+$，在變異總量 $V_K < K$ 的情況下顯著優於前人結果。

- **具偏好回饋的多臂老虎機聚類** [6]：我們首度研究具偏好回饋的多臂老虎機聚類，提出 COLDB（線性偏好）與 CONDB（神經網路偏好）兩大架構，並證明在合作決策下可獲得更小遺憾，並在合成與真實資料集上有顯著實驗成果。

- **其他合作研究** [11, 12, 13, 14, 15]：我也參與多個應用場景的多臂老虎機研究，包括電腦網路擁塞控制 [11]、組合多臂老虎機理論 [13]、聯邦式多臂老虎機推薦 [14] 與安全性分析 [15]，擴展了線上學習方法的應用廣度。

Acknowledgement

First and foremost, I would like to express my deepest gratitude to my advisor, **Professor John C.S. Lui**, for his invaluable guidance and unwavering support throughout my Ph.D. journey. John has always respected and encouraged my decisions, especially during the most challenging moments over the past four years—when I struggled to find research ideas, encountered repeated setbacks, or dealt with paper rejections. His patience and trust gave me the strength and confidence to persevere. To me, he is not only a mentor, but also a true friend. He taught me how to think critically, how to conduct meaningful research, and, more importantly, how to face adversity with resilience. I was especially touched during the final year of my Ph.D., when I was applying for postdoctoral positions: John kindly offered to be CC'd on every inquiry email I sent and followed up with personalized recommendation letters—over a hundred in total. His dedication and generosity moved me deeply. Words cannot fully express my appreciation—thank you, John, for everything.

I am also deeply grateful to **Professor Shuai Li** from Shanghai Jiao Tong University, who guided me during the early stages of my Ph.D. journey. She generously mentored me through my first three research projects, from which I learned a great deal. I

have been deeply impressed by her dedication and strong motivation. She instilled in me the discipline and rigor essential for beginning a research career and helped me build a solid foundation in the field of bandits. Her encouragement and support during challenging times in my research meant a great deal to me. Thank you, Prof. Li, for your mentorship, patience, and generosity.

My sincere thanks go to **Professor Wen Sun** from Cornell University, who graciously hosted me during my research visit. I had the pleasure of meeting him at NeurIPS 2023, and he generously welcomed me to join his group. During my time at Cornell, I benefited tremendously from his expertise in reinforcement learning. His weekly discussions and insightful feedback were instrumental to my development as a researcher. Thank you, Wen, for the enriching experience and your kind mentorship.

I would also like to extend my heartfelt thanks to **Professor Dongruo Zhou** from Indiana University, a close collaborator and dear mentor. We worked together on several projects, and I was continually inspired by his brilliance, dedication, and insightful thinking. More than a collaborator, Dongruo has been like an older brother to me—offering not only technical guidance, but also thoughtful career advice and warm encouragement. Thank you, Dongruo, for your friendship, support, and the many lessons you' ve taught me.

I am thankful to **Dr. Wei Chen** and **Dr. Siwei Wang** from Microsoft Research Asia, who mentored me during my internship. Their guidance broadened my research vision and taught me how to identify and tackle impactful problems. Thank you for your

mentorship and for sharing your deep insights and experience.

I am also grateful to my other close collaborators: **Prof. Zhongxiang Dai** (CUHK Shenzhen), **Dr. Tong Yu** (Adobe Research), and **Prof. Jiancheng Ye** (Macau University of Science and Technology). Working with them has been both intellectually rewarding and personally enjoyable. Through our collaborations, I gained not only valuable research experience but also fresh perspectives and renewed motivation. I sincerely appreciate their support, generosity, and enthusiasm for research.

I would also like to thank my thesis committee members—**Prof. Farzan Farnia**, **Prof. Bei Yu**, and **Prof. Xiapu Daniel Luo**—for their time, insightful feedback, and constructive suggestions. I am truly grateful for their contributions to my academic development.

This journey would not have been the same without the camaraderie and companionship of my fellow lab members at CUHK and Cornell: **Xutong Liu**, **Zhuohua Li**, **Xuchuang Wang**, **Jincheng Wang**, **Shiyuan Zheng**, **Maoli Liu**, **Chengchang Liu**, **Xudong Liu**, **Xiangxiang Dai**, **Bin Luo**, **Zeyu Zhang**, **Ziyi Han**, **Dian Shen**, **Fang Kong**, **Bo Sun**, **Yuwen Huang**, **Qiang Zhao**, **Jianhao He**, **Runzhe Wu**, **Yiyi Zhang**, **Nico Espinosa Dice**, **Zhaolin Gao**, **Jinyan Su**, **Yiding Chen**, **Rebecca Liu**, **Yann Hicke**, and **Owen Oertell**. Thank you all for the great memories and support throughout these years. Special thanks to **Xutong** for his generous help during the early stages of my research, to **Xiangxiang** for our enjoyable and productive collaborations, and to **Runzhe** for accompanying me to dinners and helping me move into my new apartment during my stay at

Cornell.

To my beloved family—thank you for your unconditional love and unwavering support. I am especially grateful to my parents, who have always been my strongest pillars. Your sacrifices, belief in me, and endless encouragement have carried me through every step of this journey. I owe everything to you.

Lastly, and most dearly, I want to thank my girlfriend, **Ms. Yiwen Liu**. You are my soulmate and my greatest source of strength. Through all the highs and lows of my Ph.D. journey, you have stood by my side with love, patience, and unwavering belief. I will always cherish the moments when I felt defeated and you gently reminded me, "That's OK, nothing will change my love for you." Your presence brings light into my life, and your support reminds me that no matter how hard the road may be, I am never alone. Thank you for everything.

This thesis is dedicated to my beloved parents and my beloved girl.

Contents

List of Figures

List of Tables

List of Publications

Papers in Submission (* denotes equal contribution)

[1] **Towards Zero-Shot Generalization in Offline Reinforcement Learning,**
Zhiyong Wang, Chen Yang, John C.S. Lui, Dongruo Zhou,
Adaptive Learning in Complex Environments TTIC Workshop, 2024.
ICML 2024 Workshop: Aligning Reinforcement Learning Experimentalists and Theorists.
TTIC Summer Workshop 2024: Data-Driven Decision Processes: From Theory to Practice.
In submission.

[2] **Online Clustering of Dueling Bandits,**
Zhiyong Wang, Jiahang Sun, Mingze Kong, Jize Xie, Qinghua Hu, John C.S. Lui, Zhongxiang Dai,
In submission.

[3] **In-Context Federated Learning: A Collaborative Approach for Iterative Answer Refinement,**
Ruhan Wang*, Zhiyong Wang*, Chengkai Huang*, Rui Wang, Tong Yu, Lina Yao, John C.S. Lui, Dongruo Zhou,
In submission.

[4] **Large Language Model-Enhanced Multi-Armed Bandits,**
Jiahang Sun*, Zhiyong Wang*, Runhan Yang*, Chenjun Xiao, John C.S. Lui, Zhongxiang Dai,
Accepted in ICLR 2025 Workshop on Reasoning and Planning for Large Language Models
In submission.

[5] **Meta-Prompt Optimization for LLM-Based Sequential Decision Making,**
Mingze Kong, Zhiyong Wang, Yao Shu, Zhongxiang Dai,
Accepted in ICLR 2025 Workshop on Reasoning and Planning for Large Language Models
In submission.

[6] **Federated Linear Dueling Bandits,**
Xuhan Huang, Yan Hu, Zhiyan Li, Zhiyong Wang, Benyou Wang, Zhongxiang Dai,
In submission.

[7] **Cascading Bandits Robust to Adversarial Corruptions,**
Jize Xie, Cheng Chen, Zhiyong Wang, Shuai Li,
In submission.

PUBLICATIONS (* denotes equal contribution, # denotes corresponding author)

[1] **Model-based RL as a Minimalist Approach to Horizon-Free and Second-Order Bounds,**
Zhiyong Wang, Dongruo Zhou, John C.S. Lui, Wen Sun.

Selected as a course reference paper for CS 6789: Foundations of Reinforcement Learning at Cornell University.

Accepted in the Thirteenth International Conference on Learning Representations (ICLR), 2025.

[2] **Variance-Dependent Regret Bounds for Non-stationary Linear Bandits,**

Zhiyong Wang, Jize Xie, Yi Chen, John C.S. Lui, Dongruo Zhou,

Adaptive Learning in Complex Environments TTIC Workshop, 2024.

ICML 2024 Workshop: Foundations of Reinforcement Learning and Control – Connections and Perspectives.

Presented at the 25th International Symposium on Mathematical Programming (ISMP), 2024.

Accepted in the 28th International Conference on Artificial Intelligence and Statistics (AISTATS), 2025.

[3] **Online Learning and Detecting Corrupted Users for Conversational Recommendation Systems,**

Xiangxiang Dai*, Zhiyong Wang*#, Jize Xie, Tong Yu, John C.S. Lui,

Accepted in the IEEE Transactions on Knowledge and Data Engineering (TKDE), 2024.

[4] **Conversational Recommendation with Online Learning and Clustering on Misspecified Users,**

Xiangxiang Dai*, Zhiyong Wang*#, Jize Xie, Xutong Liu,

John C.S. Lui,
Accepted in the IEEE Transactions on Knowledge and Data Engineering (TKDE), 2024.

[5] **Combinatorial Multivariant Multi-Armed Bandits with Applications to Episodic Reinforcement Learning and Beyond,**
Xutong Liu, Siwei Wang, Jinhang Zuo, Han Zhong, Xuchuang Wang, Zhiyong Wang, Shuai Li, Mohammad Hajiesmaili, John C.S. Lui, Wei Chen,
Accepted in the Forty-first International Conference on Machine Learning (ICML), 2024.

[6] **Quantifying the Merits of Network-Assist Online Learning in Optimizing Network Protocols,**
Xiangxiang Dai*, Zhiyong Wang*, Jiancheng Ye, John C.S. Lui,
Accepted in the IEEE/ACM International Symposium on Quality of Service (IWQoS), 2024.

[7] **Online Optimal Service Caching for Multi-Access Edge Computing: A Constrained Multi-Armed Bandit Optimization Approach,**
Weibo Chu, Xiaoyan Zhang, Xinming Jia, John C.S. Lui, Zhiyong Wang,
Accepted in the Computer Networks, 2024.

[8] **Federated Contextual Cascading Bandits with Asynchronous Communication and Heterogeneous Users,**
Hantao Yang, Xutong Liu, Zhiyong Wang, Hong Xie, John

C.S. Lui, Defu Lian, Enhong Chen,
Accepted in the AAAI Conference on Artificial Intelligence (AAAI), 2024.

[9] **Learning Context-Aware Probabilistic Maximum Coverage Bandits: A Variance-Adaptive Approach,**
Xutong Liu, Jinhang Zuo, Junkai Wang, Zhiyong Wang, Yuedong Xu, John C.S. Lui,
IEEE International Conference on Computer Communications (INFOCOM), 2024.

[10] **Online Clustering of Bandits with Misspecified User Models,**
Zhiyong Wang, Jize Xie, Xutong Liu, Shuai Li, John C.S. Lui,
Thirty-seventh Conference on Neural Information Processing Systems (NeurIPS), 2023.

[11] **Online Corrupted User Detection and Regret Minimization,**
Zhiyong Wang, Jize Xie, Xutong Liu, Shuai Li, John C.S. Lui,
Thirty-seventh Conference on Neural Information Processing Systems (NeurIPS), 2023.

[12] **Adversarial Attacks on Online Learning to Rank with Click Feedback,**
Jinhang Zuo, Zhiyao Zhang, Zhiyong Wang, Shuai Li, Mohammad Hajiesmaili, Adam Wierman,
Thirty-seventh Conference on Neural Information Processing Systems (NeurIPS), 2023.

[13] **Efficient Explorative Key-term Selection Strategies**

for Conversational Contextual Bandits,
Zhiyong Wang, Jize Xie, Xutong Liu, Shuai Li, John C.S. Lui,
Thirty-seventh AAAI Conference on Artificial Intelligence (AAAI),
2023.

Chapter 1

Introduction

Online learning methods—such as reinforcement learning (RL) and multi-armed bandits (MAB)—have achieved remarkable progress across a broad range of applications. These include surpassing human-level performance in complex games like Atari and Go, driving breakthroughs in robotics, powering large-scale recommendation systems, and facilitating the fine-tuning of large language models (LLMs).

Nevertheless, despite these advances, many state-of-the-art algorithms are designed under idealized assumptions, which can be fragile in the face of model misspecifications or adversarial perturbations. Such assumptions—often involving prior knowledge of the model class or environment stationarity—rarely hold in practice, particularly in dynamic systems influenced by unknown or even malicious factors. This disconnect between theoretical assumptions and practical realities calls for the development of algorithms that are both robust and adaptive to unforeseen conditions.

Moreover, worst-case theoretical guarantees, while valuable

for providing general reliability, often overlook the performance advantages that can be gained from exploiting instance-specific structure. Such conservatism may result in inefficient learning in practice. A further key challenge is generalization—the ability to extend what is learned in the training phase to unseen environments or tasks. This ability is essential for deploying learning systems in real-world scenarios that are dynamic, partially observable, or fundamentally different from the training conditions.

Motivated by these critical issues, this thesis aims to address them by advancing the field toward:

more efficient, robust, instance-adaptive, and generalizable online learning.

To this end, we focus on developing theoretical foundations and algorithms for both reinforcement learning and multi-armed bandits that embody these four properties. Below, we outline the major research problems tackled in this thesis.

1.1 Model-based RL as a Minimalist Approach to Horizon-Free and Second-Order Bounds

Model-based reinforcement learning (MBRL) typically involves two steps: first, learning a model of the environment's transition dynamics using collected data; second, performing planning or policy optimization within the learned model. This paradigm is attractive due to its simplicity and has been successfully applied to a wide array of real-world domains such as control and robotics (e.g., [16, 17, 18, 19, 20, 21, 22]).

The simplicity of MBRL also lends itself to theoretical analysis. Prior work has studied its performance in both online RL [23] and offline RL [24] settings. For instance, [25] showed that in the classic linear quadratic regulator (LQR) setting, the basic model-fitting and planning scheme enjoys strong performance guarantees. Similarly, [26] demonstrated that when optimism is incorporated, MBRL achieves solid sample complexity bounds for online RL with rich function classes. In the offline case, [24] showed that pessimism-augmented MBRL can provide robust guarantees, while [27] further established its effectiveness in hybrid settings involving both online and offline data, even without explicit optimism or pessimism mechanisms.

Rather than proposing new MBRL algorithms, this thesis shows that the standard MBRL approach—using Maximum Likelihood Estimation (MLE) for model fitting, combined with optimistic or pessimistic planning (depending on whether the setting is online or offline)—already achieves strong theoretical results. Specifically, under the conditions where the trajectory-level reward is normalized and transitions are time-homogeneous, these algorithms can yield nearly horizon-free and instance-dependent regret and sample complexity bounds even in the presence of general, non-linear function approximation.

Nearly horizon-free bounds imply that the regret or sample complexity does not scale polynomially with the time horizon H, suggesting that long-term planning is not necessarily the bottleneck for statistical efficiency. For instance-dependent analysis, we focus on second-order bounds, where regret scales with the variance of policy returns and directly implies first-order bounds

as a special case. This leads to significantly smaller regret when the environment is nearly deterministic or when the optimal policy has low return variance. In the case of deterministic transitions (which the algorithm does not need to know in advance), we demonstrate that these algorithms can converge faster than what worst-case analysis would suggest.

Simple and standard MLE-based MBRL algorithms are sufficient for achieving nearly horizon-free and second-order bounds in online and offline RL with function approximation.

1.2 Provable Zero-Shot Generalization in Offline Reinforcement Learning

Offline RL has become an increasingly vital framework as it enables learning policies from fixed datasets without requiring direct interaction with the environment. However, in practical deployments, the training dataset often comes from environments that differ from the ones where the policy will ultimately be applied. This leads to the need for zero-shot generalization (ZSG), where an agent is trained on a finite number of environments sampled from a distribution and then evaluated on previously unseen environments—without access to additional interactions. This problem has been explored in the online RL literature [28, 29, 30, 31, 32, 33], but remains under-theorized in the offline setting.

Although recent empirical studies [3, 34, 35] have tackled this problem by proposing various ZSG-capable methods, most suffer from strong limitations. Some methods work only when environ-

ment differences are restricted to observations [35], while others reduce to imitation learning setups [34], limiting their generality. On the theoretical front, multi-task offline RL approaches [36, 37] leverage shared representations but rely on access to downstream interactions, thus deviating from the offline ZSG formulation.

This motivates the central question:

Can we design provable offline RL algorithms that support zero-shot generalization?

To this end, we develop novel algorithmic frameworks for offline RL that deliver provable ZSG guarantees and significantly outperform prior methods in large-scale experiments.

1.3 Online Clustering of Bandits with Misspecified User Models

Stochastic multi-armed bandits (MAB) are foundational models for sequential decision-making under uncertainty. In each round, a learning agent selects an action and observes a corresponding reward, with the goal of maximizing cumulative rewards. They are widely adopted in applications such as recommendation systems and network optimization [38, 39, 7, 40].

To handle complex settings, contextual linear bandits incorporate side information, modeling expected rewards as linear functions of observed features. These methods enable personalization in large-scale systems [41, 42, 43, 44, 45], but do not exploit similarities among users. Clustering of bandits (CB) addresses this by

adaptively grouping users and sharing information across clusters [46].

However, prior CB methods assume perfectly linear reward models and identical preferences within clusters. This fails to reflect real-world variation caused by noise or user heterogeneity [47, 48]. To overcome this, we introduce clustering of bandits with misspecified user models (CBMUM), where users in the same cluster share a linear reward component but have individual deviations that better capture diverse behaviors.

1.4 Online Corrupted User Detection and Regret Minimization

In online recommendation platforms, user data arrives sequentially, and some users may behave adversarially—through click fraud, fake reviews, or coordinated disruptions. These corrupted signals can degrade the system' s performance by misleading preference estimations [49, 50, 51, 52, 53].

Prior bandit algorithms with corruption tolerance are limited to single-user settings [49, 53, 54], and offline user detection approaches [55, 56, 57] cannot operate dynamically in streaming scenarios.

We propose a novel learning framework called Learning and Online Corrupted Users Detection (LOCUD), which simultaneously performs preference learning, cluster inference, and online anomaly detection under potential adversarial corruption. This setting models latent user clusters and assumes only a minority of users are corrupted. The algorithm aims to maximize reward and

detect corrupted users on the fly—despite dynamic and partially adversarial behavior.

1.5 Online Clustering of Dueling Bandits

In many applications like recommendation and prompt tuning for LLMs, it is more realistic to obtain relative feedback (i.e., preferences) instead of absolute scores. Dueling bandits formalize this feedback mode by asking users to compare two options. Classical dueling bandit algorithms, however, do not incorporate user collaboration.

We introduce the first clustering of dueling bandits framework, which enables adaptive user grouping in preference-feedback environments. This approach combines pairwise comparison modeling with collaborative structure, expanding the applicability of contextual dueling bandits [58, 59, 60].

1.6 Efficient Explorative Key-term Selection Strategies for Conversational Contextual Bandits

Conversational recommender systems (CRS) improve learning efficiency by eliciting user preferences through occasional interactions involving explicit feedback on key terms [61, 62]. Existing conversational bandit methods treat feedback from different levels independently and lack effective strategies to select informative key terms.

We propose ConLinUCB, a unified framework that jointly integrates key-term and arm-level feedback into a single estimation process. Within this framework, we design two explorative strategies: ConLinUCB-BS, which samples from a barycentric spanner of key terms, and ConLinUCB-MCR, which selects key terms based on confidence radius to maximize exploration. These methods significantly improve the speed and quality of recommendation.

1.7 Variance-Dependent Regret Bounds for Non-stationary Linear Bandits

In non-stationary linear bandits, the expected reward functions change over time. Most existing approaches focus on bounding regret in terms of total variation in reward means (e.g., B_K), but ignore the influence of reward variance.

We propose new algorithms that exploit both mean drift and variance information to produce regret bounds that scale more favorably in heteroscedastic environments. This advancement is motivated by real-world applications like hyperparameter tuning in physical systems, where the noise profile depends on the evaluation point. Our methods demonstrate that variance-awareness can yield sharper bounds and better adaptivity in non-stationary settings.

Chapter 2

Literature Review

In this chapter, we summarize previous researches that are related to this thesis and differentiate our results from theirs.

2.1 Model-based RL

Learning transition models with function approximation and planning with the learned model is a standard approach in RL and control. In the control literature, certainty-equivalence control learns a model from some data and plans using the learned model, which is simple but effective for controlling systems such as Linear Quadratic Regulators (LQRs) [25]. In RL, such a simple model-based framework has been widely used in theory with rich function approximation, for online RL [23, 63, 64, 65, 66, 26, 67], offline RL [24], RL with representation learning [68, 69], and hybrid RL using both online and offline data for model fitting [27]. Our work builds on the maximum-likelihood estimation (MLE) approach, a standard method for estimating transition models in model-based RL.

2.2 Horizon-free and Instance-dependent bounds

Most existing works on horizon-free RL typically focus on tabular settings or linear settings. For instance, [70] firstly studied horizon-free RL for tabular MDPs and proposed an algorithm that depends on horizon logarithmically. Several follow-up work studied horizon-free RL for tabular MDP with better sample complexity [71], offline RL [72], stochastic shortest path [73] and RL with linear function approximation [74, 75, 76, 77, 78, 79, 80]. Note that all these works have logarithmic dependence on the horizon H. For the tabular setting, recent work further improved the regret or sample complexity to be completely independent of the horizon (i.e., removing the logarithmic dependence on the horizon) [81, 82] with a worse dependence on the cardinality of state and action spaces $|\mathcal{S}|$ and $|\mathcal{A}|$. To compare with, we show that simple MBRL algorithms are already enough to achieve completely horizon-free (i.e., no log dependence) sample complexity for offline RL when the transition model class is finite, and we provide a simpler approach to achieve the nearly horizon-free results for tabular MDPs, compared with [71]. A recent work [83] also studied the horizon-free and instance-dependent online RL in the function approximation setting with small Eluder dimensions. They estimated the variances to conduct variance-weighted regression. To compare, in our online RL part, we use the simple and standard MLE-based MBRL approach and analysis to get similar guarantees. A more recent work also studied horizon-free behavior cloning [84], which is different from our settings.

2.3 Offline RL

Offline reinforcement learning (RL) [85, 86, 87, 88] addresses the challenge of learning a policy from a pre-collected dataset without direct online interactions with the environment. A central issue in offline RL is the inadequate dataset coverage, stemming from a lack of exploration [88, 89]. A common strategy to address this issue is the application of the pessimism principle, which penalizes the estimated value of under-covered state-action pairs. Numerous studies have integrated pessimism into various single-environment offline RL methodologies. This includes model-based approaches [90, 24, 91, 92, 93, 69, 94], model-free techniques [95, 96, 97, 98, 99], and policy-based strategies [100, 101, 102, 103]. [104] has observed that with sufficient offline data diversity and coverage, the need for pessimism to mitigate extrapolation errors and distribution shift might be reduced. To the best of our knowledge, we are the first to theoretically study the generalization ability of offline RL in the contextual MDP setting.

2.4 Generalization in online RL

There are extensive empirical studies on training online RL agents that can generalize to new transition and reward functions [28, 29, 30, 31, 32, 33, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122]. They use techniques including implicit regularization [118], data augmentation [120, 121], uncertainty-driven exploration [122], successor feature [123],

etc. These works focus mostly on the online RL setting and do not provide theoretical guarantees, thus differing a lot from ours. Moreover, [123] has studied zero-shot generalization in offline RL, but to unseen reward functions rather than unseen environments.

There are also some recent works aimed at understanding online RL generalization from a theoretical perspective. [124] examined a specific class of reparameterizable RL problems and derived generalization bounds using Rademacher complexity and the PAC-Bayes bound. [125] established lower bounds and introduced efficient algorithms that ensure a near-optimal policy for deterministic MDPs. A recent work [126] studied how much pretraining can improve online RL test performance under different generalization settings. To the best of our knowledge, no previous work exists on theoretical understanding of the zero-shot generalization of offline RL.

Our paper is also related to recent works studying multi-task learning in reinforcement learning (RL) [127, 128, 129, 130, 131, 36, 37, 132, 133], which focus on transferring the knowledge learned from upstream tasks to downstream ones. Additionally, these works typically assume that all tasks share similar transition dynamics or common representations while we do not. Meanwhile, they typically require the agent to interact with the downstream tasks, which does not fall into the ZSG regime.

2.5 Online Clustering of Bandits (CB)

The paper [46] first formulates the CB problem and proposes a graph-based algorithm. The work [134] further considers lever-

aging the collaborative effects on items to guide the clustering of users. The work [135] considers the CB problem in the cascading bandits setting with random prefix feedback. The paper [136] also considers users with different arrival frequencies. A recent work [137] proposes the setting of clustering of federated bandits, considering both privacy protection and communication requirements. However, all these works assume that the reward model for each user follows a perfectly linear model, which is unrealistic in many real-world applications. To the best of our knowledge, this paper is the first work to consider user model misspecifications in the CB problem.

2.6 Misspecified Linear Bandits (MLB)

The work [48] first proposes the misspecified linear bandits (MLB) problem, shows the vulnerability of linear bandit algorithms under deviations, and designs an algorithm RLB that is only robust to non-sparse deviations. The work [138] proposes two algorithms to handle general deviations, which are modifications of the phased elimination algorithm [139] and LinUCB [43]. Some recent works [140, 141] use model selection methods to deal with unknown exact maximum model misspecification level. Note that the work [141] has an additional assumption on the access to an online regression oracle, and the paper [140] still needs to know an upper bound of the unknown exact maximum model deviation level. None of them consider the CB setting with multiple users, thus differing from ours.

2.7 Bandits with Adversarial Corruption

The work [49] first studies stochastic bandits with adversarial corruption, where the rewards are corrupted with the sum of corruption magnitudes in all rounds constrained by the *corruption level* C. They propose a robust elimination-based algorithm. The paper [53] proposes an improved algorithm with a tighter regret bound.

The paper [54] first studies stochastic linear bandits with adversarial corruptions. To tackle the contextual linear bandit setting where the arm set changes over time, the work [142] proposes a variant of the OFUL [43] that achieves a sub-linear regret. A recent work [51] proposes the CW-OFUL algorithm that achieves a nearly optimal regret bound. All these works focus on designing robust bandit algorithms for a single user; none consider how to robustly learn and leverage the implicit relations among potentially corrupted users for more efficient learning. Moreover, none of them consider how to online detect corrupted users in the multiple-user case.

2.8 Dueling Bandits and Neural Bandits

Dueling bandits has been receiving growing attention over the years since its introduction [143, 144, 58] due to the prevelance of preference or relative feedback in real-world applications. Many earlier works on dueling bandits have focused on MAB problems with a finte number of arms [145, 146, 147, 148, 149, 150, 151, 152, 153, 154]. More recently, contextual dueing bandits, which

model the reward function using a parametric function of the features of the arms, have attracted considerable attention [155, 156, 157, 158, 159, 60].

To apply MABs to complicated real-world applications with non-linear reward functions, neural bandits have been proposed which use a neural network to model the reward function [160, 161]. Recently, we have witnessed a significant growing interest in further improving the theoretical and empirical performance of neural bandits and applying it to solve real-world problems [162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177]. In particular, the work of [178] has adopted a neural network as a meta-learner for adapting to users in different clusters within the framework of clustering of bandits, and the work of [60] has combined neural bandits with dueling bandits.

2.9 Conversational Contextual Bandits

Contextual linear bandit is an online sequential decision-making problem where at each time step, the agent has to choose an action and receives a corresponding reward whose expected value is an unknown linear function of the action [41, 42, 43, 179]. The objective is to collect as much reward as possible in T rounds.

Traditional linear bandits need extensive exploration to capture the user preferences in recommender systems. To speed up online recommendations, the idea of conversational contextual bandits was first proposed in [62], where conversational feedback on key-terms is leveraged to assist the user preference elicitation. In that work, they propose the ConUCB algorithm with a theo-

retical regret bound of $O(d\sqrt{T}\log T)$. Some follow-up works try to improve the performance of ConUCB with the help of additional information, such as self-generated key-terms [180], relative feedback [181], and knowledge graph [182]. Unlike these works, we adopt the same problem settings as ConUCB and improve the underlying mechanisms without relying on additional information. Yet one can use the principles of efficient information incorporation and explorative conversations proposed in this work to enhance these works when additional information is available, which is left as an interesting future work.

2.10 Non-stationary (Linear) Bandits

There have been a series of works about non-stationary bandits [183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195, 196, 197, 198, 199, 200, 201].

In non-stationary linear bandits, the unknown feature vector $\boldsymbol{\theta}_k$ can be dynamically and adversarially adjusted, with the total change upper bounded by the *total variation budget* B_K over K rounds, *i.e.*, $\sum_{k=1}^{K-1} \|\boldsymbol{\theta}_{k+1} - \boldsymbol{\theta}_k\|_2 \leq B_K$. To tackle this problem, some works proposed forgetting strategies such as sliding window, restart, and weighted regression [187, 188, 192]. [193] also introduced the randomized exploration with weighting strategy. The regret upper bounds in these works are all of $\tilde{O}(B_K^{\frac{1}{4}}K^{\frac{3}{4}})$. A recent work by [194] proposed the MASTER-OFUL algorithm based on a black-box approach, which can achieve a regret bound of $\tilde{O}(B_K^{\frac{1}{3}}K^{\frac{2}{3}})$ in the case where the arm set is fixed over K rounds. To the best of our knowledge, none of the existing works con-

sider how to utilize the variance information to improve the regret bound in the case with time-dependent variances. The only exception of utilizing the variance information in the non-stationary bandit setting is [186], which proposed the Rerun-UCB-V algorithm for the non-stationary MAB setting with a regret dependent on the action set size $|\mathcal{A}|$. To compare with, the regret upper bounds of our algorithms are independent of the action set size, thus our algorithms are more efficient for the case where the number of actions is large.

2.11 Linear Bandits with Heteroscedastic Noises

Some recent works study the heteroscedastic linear bandit problem, where the noise distribution is assumed to vary over time. [202] first proposed the linear bandit model with heteroscedastic noise. In this model, the noise at round $k \in [K]$ is assumed to be σ_k-sub-Gaussian. Some follow-up works relaxed the σ_k-sub-Gaussian assumption by assuming the noise at the k-th round to be of variance σ_k^2 [203, 75, 74, 76, 204, 80]. Specifically, [203] and [76] considered the case where σ_k is observed by the learner after the k-th round. [75] and [74] proposed statistically efficient but computationally inefficient algorithms for the unknown-variance case. A recent work by [80] proposed an algorithm that achieves both statistical and computational efficiency in the unknown-variance setting. [204] also considered a specific heteroscedastic linear bandit problem where the linear model is sparse.

Chapter 3

Model-based RL as a Minimalist Approach to Horizon-Free and Second-Order Bounds

Learning a transition model via Maximum Likelihood Estimation (MLE) followed by planning inside the learned model is perhaps the most standard and simplest Model-based Reinforcement Learning (RL) framework. In this work, we show that such a simple Model-based RL scheme, when equipped with optimistic and pessimistic planning procedures, achieves strong regret and sample complexity bounds in online and offline RL settings. Particularly, we demonstrate that under the conditions where the trajectory-wise reward is normalized between zero and one and the transition is time-homogenous, it achieves nearly horizon-free and second-order bounds. This chapter is based on our publication [1].

3.1 Introduction

The framework of model-based Reinforcement Learning (RL) often consists of two steps: fitting a transition model using data and then performing planning inside the learned model. Such a simple framework turns out to be powerful and has been used extensively in practice on applications such as robotics and control (e.g., [16, 17, 18, 19, 20, 21, 22]).

The simplicity of model-based RL also attracts researchers to analyze its performance in settings such as online RL [23] and offline RL [24]. [25] showed that this simple scheme — fitting model via data followed by optimal planning inside the model, has a strong performance guarantee under the classic linear quadratic regulator (LQR) control problems. [26] showed that this simple MBRL framework when equipped with optimism in the face of the uncertainty principle, can achieve strong sample complexity bounds for a wide range of online RL problems with rich function approximation for the models. For offline settings where the model can only be learned from a static offline dataset, [24] showed that MBRL equipped with the pessimism principle can again achieve robust performance guarantees for a large family of MDPs. [27] showed that in the hybrid RL setting where one has access to both online and offline data, this simple MBRL framework again achieves favorable performance guarantees without any optimism/pessimism algorithm design.

In this work, we do not create new MBRL algorithms, instead, we show that the extremely simple and standard MBRL algorithm – fitting models using Maximum Likelihood Estima-

tion (MLE), followed by optimistic/pessimistic planning (depending on whether operating in online RL or offline RL mode), can already achieve surprising theoretical guarantees. Particularly, we show that under the conditions that trajectory-wise reward is normalized between zero and one, and the transition is time-homogenous, they can achieve nearly horizon-free and instance-dependent regret and sample complexity bounds, in both online and offline RL with non-linear function approximation. Nearly horizon-free bounds mean that the regret or sample complexity bounds have no explicit polynomial dependence on the horizon H. The motivation for studying horizon-free RL is to see if RL problems are harder than bandits due to the longer horizon planning in RL. *Our result here indicates that, even under non-linear function approximation, long-horizon planning is not the bottleneck of achieving statistical efficiency in RL.* For instance-dependent bounds, we focus on second-order bounds. A second-order regret bound scales with respect to the variances of the returns of policies and also directly implies a first-order regret bound which scales with the expected reward of the optimal policy. Thus our instance-dependent bounds can be small under situations such as nearly-deterministic systems or the optimal policy having a small value. When specializing to the case of deterministic ground truth transitions (but the algorithm does not need to know this a priori), we show that these simple MBRL algorithms demonstrate a faster convergence rate than the worst-case rates. The key message of our work is

Simple and standard MLE-based MBRL algorithms are sufficient for achieving nearly horizon-free and second-order bounds in

online and offline RL with function approximation.

We provide a fairly standard analysis to support the above claim. Our analysis follows the standard frameworks of optimism/pessimism in the face of uncertainty. For online RL. we use ℓ_1 Eluder dimension [66, 205], a condition that uses both the MDP structure and the function class, to capture the structural complexity of exploration. For offline RL, we use the similar concentrability coefficient in [206] to capture the coverage condition of the offline data. The key technique we leverage is the *triangular discrimination* – a divergence that is equivalent to the squared Hellinger distance up to some universal constants. Triangular discrimination was used in contextual bandit and model-free RL for achieving first-order and second-order instance-dependent bounds [207, 208, 205]. Here we show that it also plays an important role in achieving horizon-free bounds. Our contributions can be summarized as follows.

[1] Our results extend the scope of the prior work on horizon-free RL which only applies to tabular MDPs or MDPs with linear functions. Given a finite model class $\mathcal{P}$ (which could be exponentially large), we show that in online RL, the agent achieves an $O\left(\sqrt{(\sum_k \mathrm{VaR}_{\pi^k}) \cdot d_{\mathrm{RL}} \log(KH|\mathcal{P}|/\delta)} + d_{\mathrm{RL}} \log(KH|\mathcal{P}|/\delta)\right)$ regret, where K is the number of episodes, d_{RL} is the ℓ_1 Eluder dimension, VaR_{π^k} is the variance of the total reward of policy π^k learned in episode k and $\delta \in (0,1)$ denotes the failure probability. Similarly, for offline RL, the agent achieves an $O\left(\sqrt{C^{\pi^*}\mathrm{VaR}_{\pi^*} \log(|\mathcal{P}|/\delta)/K} + C^{\pi^*} \log(|\mathcal{P}|/\delta)/K\right)$ performance gap in finding a comparator policy π^*, where C^{π^*} is the single policy concentrability coefficient over π^*, K denotes the number of

offline trajectories, VaR_{π^*} is the variance of the total reward of π^*. For offline RL with finite $\mathcal{P}$, our result is *completely horizon-free*, not even with $\log H$ dependence.

[2] When specializing to MDPs with deterministic ground truth transition (but rewards, and models in the model class could still be stochastic), we show that the same simple MBRL algorithms can adapt to the deterministic environment and achieve a better statistical complexity. For online RL, the regret becomes $O(d_{\mathrm{RL}} \log(KH|\mathcal{P}|/\delta))$, which only depends on the number of episodes K poly-logarithmically. For offline RL, the performance gap to a comparator policy π^* becomes $O\left(C^{\pi^*} \log(|\mathcal{P}|/\delta)/K\right)$, which is tighter than the worst-case $O(1/\sqrt{K})$ rate. All our results can be extended to continuous model class $\mathcal{P}$ using bracket number as the complexity measure.

Overall, our work identifies the *minimalist* algorithms and analysis for nearly horizon-free and instance-dependent (first & second-order) online & offline RL.

3.2 Preliminaries

Markov Decision Processes. We consider finite horizon time homogenous MDP $\mathcal{M} = \{\mathcal{S}, \mathcal{A}, H, P^\star, r, s_0\}$ where $\mathcal{S}, \mathcal{A}$ are the state and action space (could be large or even continuous), $H \in \mathbb{N}^+$ is the horizon, $P^\star : \mathcal{S} \times \mathcal{A} \mapsto \Delta(\mathcal{S})$ is the ground truth transition, $r : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}$ is the reward signal which we assume is known to the learner, and s_0 is the fixed initial state.[1] Note that the

[1]For simplicity, we assume initial state s_0 is fixed and known. Our analysis can be easily extended to a setting where the initial state is sampled from an unknown fixed distribution.

transition $P^\star$ here is time-homogenous. For notational easiness, we denote $[K-1] = \{0, 1, \ldots, K-1\}$.

We denote π as a deterministic non-stationary policy $\pi = \{\pi_0, \ldots, \pi_{H-1}\}$ where $\pi_h : \mathcal{S} \mapsto \mathcal{A}$ maps from a state to an action. Let Π denote the set of all such policies. $V_h^\pi(s)$ represents the expected total reward of policy π starting at $s_h = s$, and $Q_h^\pi(s,a)$ is the expected total reward of the process of executing a at s at time step h followed by executing π to the end. The optimal policy $\pi^\star$ is defined as $\pi^\star = \operatorname{argmax}_\pi V_0^\pi(s_0)$. For notation simplicity, we denote $V^\pi := V_0^\pi(s_0)$. We will denote $d_h^\pi(s,a)$ as the state-action distribution induced by policy π at time step h. We sometimes will overload notation and denote $d_h^\pi(s)$ as the corresponding state distribution at h. Sampling $s \sim d_h^\pi$ means executing π starting from s_0 to h and returning the state at time step h.

Since we use the model-based approach for learning, we define a general model class $\mathcal{P} \subset \mathcal{S} \times \mathcal{A} \mapsto \Delta(\mathcal{S})$. Given a transition P, we denote $V_{h;P}^\pi$ and $Q_{h;P}^\pi$ as the value and Q functions of policy π under the model P. Given a function $f : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}$, we denote the $(Pf)(s,a) := \mathbb{E}_{s' \sim P(s,a)} f(s')$. We then denote the *variance induced by one-step transition P and function f* as $(\mathbb{V}_P f)(s,a) := \left(Pf^2\right)(s,a) - \left(Pf(s,a)\right)^2$ which is equal to $\mathbb{E}_{s' \sim P(s,a)} f^2(s') - \left(\mathbb{E}_{s' \sim P(s,a)} f(s')\right)^2$.

Assumptions. We make the realizability assumption that $P^\star \in \mathcal{P}$. We assume that the rewards are normalized such that $r(\tau) \in [0,1]$ for any trajectory $\tau := \{s_0, a_0, \ldots, s_{H-1}, a_{H-1}\}$ where $r(\tau)$ is short for $\sum_{h=0}^{H-1} r(s_h, a_h)$. Note that this setting is more general

than assuming each one-step reward is bounded, i.e., $r(s_h, a_h) \in [0, 1/H]$, and allows to represent the sparse reward setting. Without loss of generalizability, we assume $V^{\pi}_{h;P}(s) \in [0, 1]$, for all $\pi \in \Pi, h \in [0, H], P \in \mathcal{P}, s \in \mathcal{S}$[2].

Online RL. For the online RL setting, we focus on the episodic setting where the learner can interact with the environment for K episodes. At episode k, the learner proposes a policy π^k (based on the past interaction history), executes π^k starting from s_0 to time step $H-1$. We measure the performance of the online learning via *regret*: $\sum_{k=0}^{K-1} \left(V^{\pi^\star} - V^{\pi^k}\right)$. To achieve meaningful regret bounds, we often need additional structural assumptions on the MDP and the model class $\mathcal{P}$. We use a ℓ_1 Eluder dimension [66] as the structural condition due to its ability to capture nonlinear function approximators (formal definition will be given in Section 3.3).

Offline RL. For the offline RL setting, we assume that we have a pre-collected offline dataset $\mathcal{D} = \{\tau^i\}_{i=1}^K$ which contains K trajectories. For each trajectory, we allow it to potentially be generated by an adversary, i.e., at step h in trajectory k, (i.e., s_h^k), the adversary can select a_h^k based on all history (the past $k-1$ trajectories and the steps before h within trajectory k) with a fixed strategy, with the only condition that the state transitions follow the underlying transition dynamics, i.e., $s_{h+1}^i \sim P^\star(s_h^i, a_h^i)$. We emphasize that $\mathcal{D}$ is not necessarily generated by some offline

[2] $r(\tau) \in [0, 1]$ implies $V^{\pi}_{h;P^\star}(s) \in [0, 1]$. If we do not assume $V^{\pi}_{h;P}(s) \in [0, 1]$ for all $P \in \mathcal{P}$, we can simply add a filtering step in the algorithm to only choose π,P with $V^{\pi}_{h;P}(s_0) \in [0, 1]$ to get the same guarantees.

trajectory distribution. Given $\mathcal{D}$, we can split the data into HK many state-action-next state (s, a, s') tuples which we can use to learn the transition. To succeed in offline learning, we typically require the offline dataset to have good coverage over some high-quality comparator policy π^* (formal definition of coverage will be given in Section 3.4). Our goal here is to learn a policy $\widehat{\pi}$ that is as good as π^*, and we are interested in the *performance gap* between $\hat{\pi}$ and π^*, i.e., $V^{\pi^*} - V^{\hat{\pi}}$.

Horizon-free and Second-order Bounds. Our goal is to achieve regret bounds (online RL) or performance gaps (offline RL) that are (nearly) horizon-free, i.e., logarithmical dependence on H. In addition to the horizon-free guarantee, we also want our bounds to scale with respect to the variance of the policies. Denote VaR_π as the variance of trajectory reward, i.e., $\mathrm{VaR}_\pi := \mathbb{E}_{\tau\sim\pi}(r(\tau) - \mathbb{E}_{\tau\sim\pi} r(\tau))^2$. Second-order bounds in offline RL scales with VaR_{π^*} – the variance of the comparator policy. Second-order regret bound in online setting scales with respect to $\sqrt{\sum_k \mathrm{VaR}_{\pi^k}}$ instead of $\sqrt{K}$. Note that in the worst case, $\sqrt{\sum_k \mathrm{VaR}_{\pi^k}}$ scales in the order of $\sqrt{K}$, but can be much smaller in benign cases such as nearly deterministic MDPs. We also note that second-order regret bound immediately implies first-order regret bound in the reward maximization setting, which scales in the order $\sqrt{KV^{\pi^\star}}$ instead of just $\sqrt{K}$. The first order regret bound $\sqrt{KV^{\pi^\star}}$ is never worse than $\sqrt{K}$ since $V^{\pi^\star} \le 1$. Thus, by achieving a second-order regret bound, our algorithm immediately achieves a first-order regret bound.

Additional notations. Given two distributions $p \in \Delta(\mathcal{X})$ and

$q \in \Delta(\mathcal{X})$, we denote the triangle discrimination $D_{\triangle}(p \parallel q) = \sum_{x\in\mathcal{X}} \frac{(p(x)-q(x))^2}{p(x)+q(x)}$, and squared Hellinger distance $\mathbb{H}^2(p \parallel q) = \frac{1}{2}\sum_{x\in\mathcal{X}} \left(\sqrt{q(x)} - \sqrt{p(x)}\right)^2$ (we replace sum via integral when $\mathcal{X}$ is continuous and p and q are pdfs). Note that $D_{\triangle}$ and $\mathbb{H}^2$ are equivalent up to universal constants. We will frequently use the following key lemma in [205] to control the difference between means of two distributions.

Lemma 3.2.1 (Lemma 4.3 in [205])**.** *For two distributions $f \in \Delta([0,1])$ and $g \in \Delta([0,1])$:*

$$|\mathbb{E}_{x\sim f}[x] - \mathbb{E}_{x\sim g}[x]| \le 4\sqrt{\mathrm{VaR}_f \cdot D_{\triangle}(f \parallel g)} + 5D_{\triangle}(f \parallel g). \quad (3.1)$$

where $\mathrm{VaR}_f := \mathbb{E}_{x\sim f}(x - \mathbb{E}_{x\sim f}[x])^2$ denotes the variance of the distribution f.

The lemma plays a key role in achieving second-order bounds [205]. The intuition is the means of the two distributions can be closer if one of the distributions has a small variance. A more naive way of bounding the difference in means is $|\mathbb{E}_{x\sim f}[x] - \mathbb{E}_{x\sim g}[x]| \le (\max_{x\in\mathcal{X}} |x|)\|f - g\|_1 \lesssim (\max_{x\in\mathcal{X}} |x|)\mathbb{H}(f \parallel g) \lesssim (\max_{x\in\mathcal{X}} |x|)\sqrt{D_{\triangle}(f \parallel g)}$. Such an approach would have to pay the maximum range $\max_{x\in\mathcal{X}} |x|$ and thus can not leverage the variance VaR_f. In the next sections, we show this lemma plays an important role in achieving horizon-free and second-order bounds.

3.3 Online Setting

In this section, we study the online setting. We present the optimistic model-based RL algorithm (O-MBRL) in Algorithm 1.

Algorithm 1 Optimistic Model-based RL (O-MBRL)

1: **Input:** model class $\mathcal{P}$, confidence parameter $\delta \in (0,1)$, threshold β.
2: Initialize π^0, initialize dataset $\mathcal{D} = \emptyset$.
3: **for** $k = 0 \to K-1$ **do**
4: Collect a trajectory $\tau = \{s_0, a_0, \cdots, s_{H-1}, a_{H-1}\}$ from π^k, split it into tuples of $\{s, a, s'\}$ and add to $\mathcal{D}$.
5: Construct a version space $\widehat{\mathcal{P}}^k$:

$$\widehat{\mathcal{P}}^k = \left\{ P \in \mathcal{P} : \sum_{s,a,s' \in \mathcal{D}} \log P(s_i'|s_i, a_i) \geq \max_{\tilde{P} \in \mathcal{P}} \sum_{s,a,s' \in \mathcal{D}} \log \tilde{P}(s_i'|s_i, a_i) - \beta \right\}.$$

6: Set $(\pi^k, \widehat{P}^k) \leftarrow \text{argmax}_{\pi \in \Pi, P \in \widehat{\mathcal{P}}^k} V_{0;P}^{\pi}(s_0)$.
7: **end for**

The algorithm starts from scratch, and iteratively maintains a version space $\widehat{\mathcal{P}}^k$ of the model class using the historical data collected so far. Again the version space is designed such that for all $k \in [0, K-1]$, we have $P^\star \in \widehat{\mathcal{P}}_k$ with high probability. The policy π^k in this case is computed via the optimism principle, i.e., it selects π^k and $\widehat{P}^k$ such that $V_{\widehat{P}^k}^{\pi^k} \geq V^{\pi^\star}$.

Note that the algorithm design in Algorithm 1 is not new and in fact is quite standard in the model-based RL literature. For instance, [23] presented a similar style of algorithm except that they use a min-max GAN style objective for learning models. [65] used MLE oracle with optimism planning for Partially observable systems such as Predictive State Representations (PSRs), and [26] used them for both partially and fully observable systems. However, their analyses do not give horizon-free and instance-dependent bounds. We show that under the structural condition that captures nonlinear function class with small eluder dimen-

sions, Algorithm 1 achieves horizon-free and second-order bounds. Besides, since second-order regret bound implies first-order bound [205], our result immediately implies a first-order bound as well.

We first introduce the ℓ_p Eluder dimension as follows.

Definition 3.1 (ℓ_p Eluder Dimension)**.** *$DE_p(\Psi, \mathcal{X}, \epsilon)$ is the eluder dimension for $\mathcal{X}$ with function class Ψ, when the longest ϵ-independent sequence $x^1, \ldots, x^L \subseteq \mathcal{X}$ enjoys the length less than $DE_p(\Psi, \mathcal{X}, \epsilon)$, i.e., there exists $g \in \Psi$ such that for all $t \in [L]$, $\sum_{l=1}^{t-1} |g(x^l)|^p \leq \epsilon^p$ and $|g(x^t)| > \epsilon$.*

We work with the ℓ_1 Eluder dimension $DE_1(\Psi, \mathcal{S} \times \mathcal{A}, \epsilon)$ with the function class Ψ specified as:

$$\Psi = \{(s,a) \mapsto \mathbb{H}^2(P^\star(s,a) \parallel P(s,a)) : P \in \mathcal{P}\}.$$

Remark 1. *The ℓ_1 Eluder dimension has been used in previous works such as [66]. We have the following corollary to demonstrate that the ℓ_1 dimension generalizes the original ℓ_2 dimension of [209], it can capture tabular, linear, and generalized linear models.*

Lemma 3.3.1 (Proposition 19 in [66])**.** *For any $\Psi, \mathcal{X}$, $\epsilon > 0$,* $\mathrm{DE}_1(\Psi, \mathcal{X}, \epsilon) \leq \mathrm{DE}_2(\Psi, \mathcal{X}, \epsilon)$.

We are ready to present our main theorem for the online RL setting.

Theorem 3.3.2 (Main theorem for online setting)**.** *For any $\delta \in (0,1)$, let $\beta = 4\log\left(\frac{K|\mathcal{P}|}{\delta}\right)$, with probability at least $1-\delta$, Algo-*

rithm 1 achieves the following regret bound:

$$\sum_{k=0}^{K-1}(V^{\pi^\star}-V^{\pi^k}) \le O\Big(\sqrt{\sum_{k=0}^{K-1}\mathrm{VaR}_{\pi^k}\cdot DE_1(\Psi,\mathcal{S}\times\mathcal{A},1/KH)\cdot\log(KH\,|\mathcal{P}|\,/\delta)\log(KH)} + DE_1(\Psi,\mathcal{S}\times\mathcal{A},1/KH)\cdot\log(KH\,|\mathcal{P}|\,/\delta)\log(KH)\Big). \tag{3.2}$$

The above theorem indicates the standard and simple O-MBRL algorithm is already enough to achieve horizon-free and second-order regret bounds: our bound does not have explicit polynomial dependences on horizon H, the leading term scales with $\sqrt{\sum_k \mathrm{VaR}_{\pi^k}}$ instead of the typical $\sqrt{K}$.

We have the following result about the first-order regret bound.

Corollary 3.1 (Horizon-free and First-order regret bound)**.** *Let $\beta = 4\log\left(\frac{K|\mathcal{P}|}{\delta}\right)$, with probability at least $1-\delta$, Algorithm 1 achieves the following regret bound:*

$$\sum_{k=0}^{K-1}V^{\pi^\star}-V^{\pi^k} \le O\Big(\sqrt{KV^{\pi^\star}\cdot DE_1(\Psi,\mathcal{S}\times\mathcal{A},1/KH)\cdot\log(KH\,|\mathcal{P}|\,/\delta)\log(KH)} + DE_1(\Psi,\mathcal{S}\times\mathcal{A},1/KH)\cdot\log(KH\,|\mathcal{P}|\,/\delta)\log(KH)\Big).$$

Proof. Note that $\mathrm{VaR}_\pi \le V^\pi \le V^{\pi^\star}$ where the first inequality is because the trajectory-wise reward is bounded in $[0,1]$. Therefore, combining with Theorem 3.3.2, we directly obtain the first-order result. □

Note that the above bound scales with respect to $\sqrt{KV^{\pi^\star}}$ instead of just $\sqrt{K}$. Since $V^{\pi^\star}\le 1$, this bound improves the worst-case regret bound when the optimal policy has total reward less than one.[3]

[3]Typically a first-order regret bound makes more sense in the cost minimization setting instead of reward maximization setting. We believe that our results are transferable to the cost-minimization setting.

Faster rates for deterministic transitions. When the underlying MDP has deterministic transitions, we can achieve a smaller regret bound that only depends on the number of episodes logarithmically.

Corollary 3.2 ($\log K$ regret bound with deterministic transitions)**.** *When the transition dynamics of the MDP are deterministic, setting* $\beta = 4 \log\left(\frac{K|\mathcal{P}|}{\delta}\right)$*, w.p. at least* $1-\delta$*, Algorithm 1 achieves:*

$$\sum_{k=0}^{K-1} V^{\pi^\star} - V^{\pi^k} \leq O\left(DE_1(\Psi, \mathcal{S}\times\mathcal{A}, 1/KH) \cdot \log(KH\,|\mathcal{P}|\,/\delta)\log(KH)\right).$$

Extension to infinite class $\mathcal{P}$. For infinite model class $\mathcal{P}$, we have a similar result. First, we define the bracketing number of an infinite model class as follows.

Definition 3.2 (Bracketing Number [210])**.** *Let* $\mathcal{G}$ *be a set of functions mapping* $\mathcal{X} \to \mathbb{R}$*. Given two functions* l, u *such that* $l(x) \leq u(x)$ *for all* $x \in \mathcal{X}$*, the bracket* $[l,u]$ *is the set of functions* $g \in \mathcal{G}$ *such that* $l(x) \leq g(x) \leq u(x)$ *for all* $x \in \mathcal{X}$*. We call* $[l,u]$ *an* ϵ*-bracket if* $\|u-l\| \leq \epsilon$*. Then, the* ϵ*-bracketing number of* $\mathcal{G}$ *with respect to* $\|\cdot\|$*, denoted by* $\mathcal{N}_{[]}(\epsilon, \mathcal{G}, \|\cdot\|)$ *is the minimum number of* ϵ*-brackets needed to cover* $\mathcal{G}$*.*

We use the bracketing number of $\mathcal{P}$ to denote the complexity of the model class, similar to $|\mathcal{P}|$ in the finite class case. Next, we propose a corollary to characterize the regret with an infinite model class.

Corollary 3.3 (Regret bound for Algorithm 1 with infinite model class $\mathcal{P}$)**.** *When* $\mathcal{P}$ *is infinite, let* $\beta = 7\log(K\mathcal{N}_{[]}((KH|\mathcal{S}|)^{-1}, \mathcal{P}, \|\cdot$

$\|_\infty)/\delta)$, *with probability at least* $1-\delta$, *Algorithm 1 achieves the following regret bound:*

$$\sum_{k=0}^{K-1} V^{\pi^\star} - V^{\pi^k} \le O\Bigg(DE_1(\Psi, \mathcal{S}\times\mathcal{A}, \frac{1}{KH}) \log(\frac{KH\mathcal{N}_{[]}((KH|\mathcal{S}|)^{-1}, \mathcal{P}, \|\cdot\|_\infty)}{\delta}) \log(KH) + \sqrt{\sum_{k=0}^{K-1} \mathrm{VaR}_{\pi^k} \cdot DE_1(\Psi, \mathcal{S}\times\mathcal{A}, \frac{1}{KH}) \log(\frac{KH\mathcal{N}_{[]}((KH|\mathcal{S}|)^{-1}, \mathcal{P}, \|\cdot\|_\infty)}{\delta}) \log(KH)} \Bigg),$$

where $\mathcal{N}_{[]}((KH|\mathcal{S}|)^{-1}, \mathcal{P}, \|\cdot\|_\infty)$ *is the bracketing number defined in Definition 3.2.*

A specific example of the infinite model class is the tabular MDP, where $\mathcal{P}$ is the collection of all the conditional distributions over $\mathcal{S}\times\mathcal{A}\to\Delta(\mathcal{S})$. By Corollary 3.3, we also have a new regret bound for MBRL under the tabular MDP setting, which is nearly horizon-free and second-order.

Example 1 (Tabular MDPs)**.** *When specializing to tabular MDPs, use the fact that tabular MDP has* ℓ_2 *Eluder dimension being at most* $|\mathcal{S}||\mathcal{A}|$ *(Section D.1 in [209]),* ℓ_1 *dimension is upper bounded by* ℓ_2 *dimension (Lemma 3.3.1), and use the standard* ϵ*-net argument to show that* $\mathcal{N}_{[]}(\epsilon, \mathcal{P}, \|\cdot\|_\infty)$ *is upper-bounded by* $(c/\epsilon)^{|\mathcal{S}|^2|\mathcal{A}|}$ *(e.g., see [24]), we can show that Algorithm 1 achieves the following regret bound for tabular MDP: with probability at least* $1-\delta$,

$$\sum_k V^{\pi^\star} - V^{\pi^k} \le O\Big(|\mathcal{S}|^{1.5}|\mathcal{A}| \sqrt{\sum_k \mathrm{VaR}_{\pi^k} \cdot \log(\frac{KH|\mathcal{S}|}{\delta}) \log(KH)} + |\mathcal{S}|^3|\mathcal{A}|^2 \log(\frac{KH|\mathcal{S}|}{\delta}) \log(KH) \Big).$$

In summary, we have shown that a simple MLE-based MBRL algorithm is enough to achieve nearly horizon-free and second-order regret bounds under non-linear function approximation.

3.3.1 Proof Sketch of Theorem 3.3.2

Now we are ready to provide a proof sketch of Theorem 3.3.2 with the full proof deferred to Appendix A.1.5. For ease of presentation, we use d_{RL} to denote $\mathrm{DE}_1(\Psi, \mathcal{S}\times\mathcal{A}, 1/KH)$, and ignore some log terms.

Overall, our analysis follows the general framework of optimism in the face of uncertainty, but with (1) careful analysis in leveraging the MLE generalization bound and (2) more refined proof in the training-to-testing distribution transfer via Eluder dimension.

By standard MLE analysis, we can show w.p. $1-\delta$, for all $k \in [K-1]$, we have $P^\star \in \widehat{\mathcal{P}}^k$, and

$$\sum_{i=0}^{k-1}\sum_{h=0}^{H-1} \mathbb{H}^2(P^\star(s_h^i, a_h^i)||\widehat{P}^k(s_h^i, a_h^i)) \le O(\log(K\left|\mathcal{P}\right|/\delta)). \quad (3.3)$$

From here, trivially applying training-to-testing distribution transfer via the Eluder dimension as previous works (e.g., [205]) would cause poly-dependence on H. With new techniques detailed in Appendix A.1.2, which is one of our technical contributions and may be of independent interest, we can get: there exists a set $\mathcal{K} \subseteq [K-1]$ such that $|\mathcal{K}| \le O(d_{\mathsf{RL}}\log(K|\mathcal{P}|/\delta))$, and

$$\sum_{k\in[K-1]\setminus\mathcal{K}}\sum_{h} \mathbb{H}^2\Big(P^\star(s_h^k, a_h^k) \parallel \widehat{P}^k\big(s_h^k, a_h^k\big)\Big)$$
$$\le O(d_{\mathsf{RL}}\cdot\log(K\left|\mathcal{P}\right|/\delta)\log(KH)). \quad (3.4)$$

Recall that $(\pi^k, \widehat{P}^k) \leftarrow \operatorname{argmax}_{\pi\in\Pi, P\in\widehat{\mathcal{P}}^k} V^{\pi}_{0;P}(s_0)$, with the above realization guarantee $P^\star \in \widehat{\mathcal{P}}^k$, we can get the following optimism guarantee: $V^\star_{0;P^\star} \le \max_{\pi\in\Pi, P\in\widehat{\mathcal{P}}^k} V^{\pi}_{0;P} = V^{\pi^k}_{0;\widehat{P}^k}$.

At this stage, one straight-forward way to proceed is to use the standard simulation lemma (Lemma A.1.5):

$$\begin{aligned}
&\sum_{k=0}^{K-1} V^{\pi^k}_{0;\widehat{P}^k} - V^{\pi^k}_{0;P^\star} \\
&\le \sum_{k=0}^{K-1}\sum_{h=0}^{H-1} \mathbb{E}_{s,a\sim d_h^{\pi^k}} \left[\left|\mathbb{E}_{s'\sim P^\star(s,a)} V^{\pi^k}_{h+1;\widehat{P}^k}(s') - \mathbb{E}_{s'\sim \widehat{P}^k(s,a)} V^{\pi^k}_{h+1;\widehat{P}^k}(s')\right|\right].
\end{aligned} \tag{3.5}$$

However, from here, if we naively bound each term on the RHS via $\mathbb{E}_{s,a\sim d_h^{\pi^k}} \|P^\star(s,a) - \widehat{P}(s,a)\|_1$, which is what previous works such as [24] did exactly, we would end up paying a linear horizon dependence H due to the summation over H on the RHS the above expression. Given the mean-to-variance lemma (Lemma 3.2.1), we may consider using it to bound the difference between two means $\mathbb{E}_{s'\sim P^\star(s,a)} V^{\pi^k}_{h+1;\widehat{P}^k}(s') - \mathbb{E}_{s'\sim \widehat{P}^k(s,a)} V^{\pi^k}_{h+1;\widehat{P}^k}(s')$. This still can not work if we start from here, because we would eventually get $\sum_k \sum_h \mathbb{E}_{s,a\sim d_h^{\pi^k}} [\mathbb{H}^2(P^\star(s,a)||\widehat{P}^k(s,a))]$ terms, which can not be further upper bounded easily with the MLE generalization guarantee.

To achieve horizon-free and second-order bounds, we need a novel and more careful analysis.

First, we carefully decompose and upper bound the regret in $\tilde{\mathcal{K}} := [K-1] \setminus \mathcal{K}$ w.h.p. as follows using Bernstain's inequality (for regret in $\mathcal{K}$ we simply upper bound it by $|\mathcal{K}|$)

$$\sum_{k\in\tilde{\mathcal{K}}} \left(V^{\pi^k}_{0;\widehat{P}^k}(s_h^k) - \sum_{h=0}^{H-1} r(s_h^k, a_h^k)\right) + \sum_{k\in\tilde{\mathcal{K}}} \left(\sum_{h=0}^{H-1} r(s_h^k, a_h^k) - V^{\pi^k}_{0;P^\star}\right) \lesssim \sqrt{\sum_{k\in\tilde{\mathcal{K}}}\sum_h (\mathbb{V}_{P^\star} V^{\pi^k}_{h+1;\widehat{P}^k})(s_h^k, a_h^k)}$$

$$+\sum_{k\in\tilde{\mathcal{K}}}\sum_{h}\left|\mathbb{E}_{s'\sim\widehat{P}^k(s_h^k,a_h^k)}V^{\pi^k}_{h+1;\widehat{P}^k}(s')-\mathbb{E}_{s'\sim P^*(s_h^k,a_h^k)}V^{\pi^k}_{h+1;\widehat{P}^k}(s')\right|+\sqrt{\sum_k \mathrm{VaR}_{\pi^k}\log(1/\delta)}\,. \tag{3.6}$$

Then, we bound the difference of two means $\mathbb{E}_{s'\sim\widehat{P}^k(s_h^k,a_h^k)}V^{\pi^k}_{h+1;\widehat{P}^k}(s')-\mathbb{E}_{s'\sim P^*(s_h^k,a_h^k)}V^{\pi^k}_{h+1;\widehat{P}^k}(s')$ using variances and the triangle discrimination (see Lemma 3.2.1 for more details), together with the fact that $D_\triangle\le 4\mathbb{H}^2$, and information processing inequality on the squared Hellinger distance, we have

$$\begin{aligned}
&|\mathbb{E}_{s'\sim\widehat{P}^k(s_h^k,a_h^k)}V^{\pi^k}_{h+1;\widehat{P}^k}(s')-\mathbb{E}_{s'\sim P^*(s_h^k,a_h^k)}V^{\pi^k}_{h+1;\widehat{P}^k}(s')|\\
&\quad\le O\Big(\sqrt{(\mathbb{V}_{P^\star}V^{\pi^k}_{h+1;\widehat{P}^k})(s_h^k,a_h^k)D_\triangle\Big(V^{\pi^k}_{h+1;\widehat{P}^k}\big(s'\sim P^\star(s_h^k,a_h^k)\big)\,\|\,V^{\pi^k}_{h+1;\widehat{P}^k}(s'\sim\widehat{P}^k\big(s_h^k,a_h^k\big)\big)\Big)}\\
&\qquad\quad+D_\triangle\Big(V^{\pi^k}_{h+1;\widehat{P}^k}\big(s'\sim P^\star(s_h^k,a_h^k)\big)\,\|\,V^{\pi^k}_{h+1;\widehat{P}^k}(s'\sim\widehat{P}^k\big(s_h^k,a_h^k\big)\big)\Big)\Big)\\
&\le O\Big(\sqrt{(\mathbb{V}_{P^\star}V^{\pi^k}_{h+1;\widehat{P}^k})(s_h^k,a_h^k)\mathbb{H}^2\Big(P^\star(s_h^k,a_h^k)\,\|\,\widehat{P}^k\big(s_h^k,a_h^k\big)\Big)}+\mathbb{H}^2\Big(P^\star(s_h^k,a_h^k)\,\|\,\widehat{P}^k\big(s_h^k,a_h^k\big)\Big)\Big)
\end{aligned}$$

where we denote $V^{\pi^*}_{h+1;\widehat{P}}(s'\sim P^\star(s,a))$ as the distribution of the random variable $V^{\pi^*}_{h+1;\widehat{P}}(s')$ with $s'\sim P^\star(s,a)$. This is the key lemma used by [205] to show distributional RL can achieve second-order bounds. We show that this is also crucial for achieving a horizon-free bound.

Then, summing up over k,h, with Cauchy-Schwartz and the MLE generalization bound via Eluder dimension in Equation 3.4, we have

$$\begin{aligned}
&\sum_{k\in\tilde{\mathcal{K}}}\sum_{h}\left|\mathbb{E}_{s'\sim\widehat{P}^k(s_h^k,a_h^k)}V^{\pi^k}_{h+1;\widehat{P}^k}(s')-\mathbb{E}_{s'\sim P^*(s_h^k,a_h^k)}V^{\pi^k}_{h+1;\widehat{P}^k}(s')\right|\le O\Big(\sum_{k\in\tilde{\mathcal{K}}}\sum_{h}\mathbb{H}^2\Big(P^\star(s_h^k,a_h^k)\,\|\,\widehat{P}^k\big(s_h^k,a_h^k\big)\Big)\\
&+\sqrt{\sum_{k\in\tilde{\mathcal{K}}}\sum_{h}\big(\mathbb{V}_{P^\star}V^{\pi^k}_{h+1;\widehat{P}^k}\big)(s_h^k,a_h^k)\sum_{k\in\tilde{\mathcal{K}}}\sum_{h}\mathbb{H}^2\Big(P^\star(s_h^k,a_h^k)\,\|\,\widehat{P}^k\big(s_h^k,a_h^k\big)\Big)}\Big)\\
&\le O\Big(\sqrt{\sum_{k\in\tilde{\mathcal{K}}}\sum_{h}\big(\mathbb{V}_{P^\star}V^{\pi^k}_{h+1;\widehat{P}^k}\big)(s_h^k,a_h^k)d_{\mathsf{RL}}\log(K\,|\mathcal{P}|\,/\delta)\log(KH)}+d_{\mathsf{RL}}\log(K\,|\mathcal{P}|\,/\delta)\log(KH)\Big)\,.
\end{aligned} \tag{3.7}$$

Note that we have $\big(\mathbb{V}_{P^\star}V^{\pi^k}_{h+1;\widehat{P}^k}\big)(s_h^k,a_h^k)$ depending on $\widehat{P}^k$. To get a second-order bound, we need to convert it to the variance

under ground truth transition $P^\star$, and we want to do it without incurring any H dependence. *This is another key difference from [205].*

We aim to replace $\big(\mathbb{V}_{P^\star} V^{\pi^k}_{h+1;\widehat{P}^k}\big)(s^k_h, a^k_h)$ by $\big(\mathbb{V}_{P^\star} V^{\pi^k}_{h+1}\big)(s^k_h, a^k_h)$ which is the variance under $P^\star$ (recall that V^π is the value function of π under $P^\star$), and we want to control the difference $\big(\mathbb{V}_{P^\star}\left(V^{\pi^k}_{h+1;\widehat{P}^k} - V^{\pi^k}_{h+1}\right)\big)(s^k_h, a^k_h)$. To do so, we need to bound the 2^m moment of the difference $V^{\pi^k}_{h+1;\widehat{P}^k} - V^{\pi^k}_{h+1}$ following the strategy in [71, 76, 80]. Let us define the following terms:

$$A := \sum_{k\in\tilde{\mathcal{K}}}\sum_h \left[(\mathbb{V}_{P^\star} V^{\pi^k}_{h+1;\widehat{P}^k})(s^k_h, a^k_h)\right], C_m := \sum_{k\in\tilde{\mathcal{K}}}\sum_h \left[(\mathbb{V}_{P^\star}(V^{\pi^k}_{h+1;\widehat{P}^k} - V^{\pi^k}_{h+1})^{2^m})(s^k_h, a^k_h)\right],$$

$$B := \sum_{k\in\tilde{\mathcal{K}}}\sum_h \left[(\mathbb{V}_{P^\star} V^{\pi^k}_{h+1})(s^k_h, a^k_h)\right], G := \sqrt{A\cdot d_{\mathsf{RL}}\log(\frac{K\,|\mathcal{P}|}{\delta})\log(KH)} + d_{\mathsf{RL}}\log(\frac{K\,|\mathcal{P}|}{\delta})\log(KH)\,.$$

With the fact $\mathbb{V}_{P^\star}(a+b) \le 2\mathbb{V}_{P^\star}(a) + 2\mathbb{V}_{P^\star}(b)$ we have $A \le 2B + 2C_0$. For C_m, we prove that w.h.p. it has the recursive form $C_m \lesssim 2^m G + \sqrt{\log(1/\delta) C_{m+1}} + \log(1/\delta)$, during which process we also leverage the above Equation 3.7 and some careful analysis (detailed in Appendix A.1.5). Then, with the recursion lemma (Lemma A.1.9), we can get $C_0 \lesssim G$, which further gives us

$$\begin{aligned} A &\lesssim B + d_{\mathsf{RL}}\log(\frac{K\,|\mathcal{P}|}{\delta})\log(KH) + \sqrt{A\cdot d_{\mathsf{RL}}\log(\frac{K\,|\mathcal{P}|}{\delta})\log(KH)} \\ &\le O\big(B + d_{\mathsf{RL}}\log(\frac{K\,|\mathcal{P}|}{\delta})\log(KH)\big)\,, \end{aligned}$$

where in the last step we use the fact $x \le 2a + b^2$ if $x \le a + b\sqrt{x}$. Finally, we note that $B \le O(\sum_k \mathrm{VaR}_{\pi^k} + \log(1/\delta))$ w.h.p.. Plugging the upper bound of A back into Equation 3.7 and then to Equation 3.6, we conclude the proof.

Algorithm 2 ([24]) Constrained Pessimistic Policy Optimization with Likelihood-Ratio based constraints (CPPO-LR)

1: **Input:** dataset $\mathcal{D} = \{s, a, s'\}$, model class $\mathcal{P}$, policy class Π, confidence parameter $\delta \in (0, 1)$, threshold β.

2: Calculate the confidence set based on the offline dataset:

$$\widehat{\mathcal{P}} = \left\{ P \in \mathcal{P} : \sum_{i=1}^{n} \log P(s_i'|s_i, a_i) \geq \max_{\tilde{P} \in \mathcal{P}} \sum_{i=1}^{n} \log \tilde{P}(s_i'|s_i, a_i) - \beta \right\}.$$

3: **Output:** $\hat{\pi} \leftarrow \text{argmax}_{\pi \in \Pi} \min_{P \in \widehat{\mathcal{P}}} V_{0;P}^{\pi}(s_0)$.

3.4 Offline Setting

For the offline setting, we directly analyze the Constrained Pessimism Policy Optimization (CPPO-LR) algorithm (Algorithm 2) proposed by [24]. We first explain the algorithm and then present its performance gap guarantee in finding the comparator policy π^*.

Algorithm 2 splits the offline trajectory data that contains K trajectories into a dataset of (s, a, s') tuples (note that in total we have $n := KH$ many tuples) which is used to perform maximum likelihood estimation $\max_{\tilde{P} \in \mathcal{P}} \sum_{i=1}^{n} \log \tilde{P}(s_i'|s_i, a_i)$. It then builds a version space $\widehat{\mathcal{P}}$ which contains models $P \in \mathcal{P}$ whose log data likelihood is not below by too much than that of the MLE estimator. The threshold for the version space is constructed so that with high probability, $P^\star \in \widehat{\mathcal{P}}$. Once we build a version space, we perform pessimistic planning to compute $\widehat{\pi}$.

We first define the single policy coverage condition as follows.

Definition 3.3 (Single policy coverage)**.** *Given any comparator policy π^*, denote the data-dependent single policy concentrability*

coefficient $C_{\mathcal{D}}^{\pi^*}$ *as follows:*

$$C_{\mathcal{D}}^{\pi^*} := \max_{h, P \in \mathcal{P}} \frac{\mathbb{E}_{s,a \sim d_h^{\pi^*}} \mathbb{H}^2 \left(P(s,a) \parallel P^\star(s,a) \right)}{1/K \sum_{k=1}^K \mathbb{H}^2 \left(P(s_h^k, a_h^k) \parallel P^\star(s_h^k, a_h^k) \right)}.$$

We assume w.p. at least $1-\delta$ *over the randomness of the generation of* $\mathcal{D}$*, we have* $C_{\mathcal{D}}^{\pi^*} \leq C^{\pi^*}$*.*

The existence of C^{π^*} is certainly an assumption. We now give an example in the tabular MDP where we show that if the data is generated from some fixed behavior policy π^b which has non-trivial probability of visiting every state-action pair, then we can show the existence of C^{π^*}.

Example 2 (Tabular MDP with good behavior policy coverage)**.** *If the* K *trajectories are collected i.i.d. with a fixed behavior policy* π^b*, and* $d_h^{\pi^b}(s,a) \geq \rho_{\min}, \forall s, a, h$ *(similar to [72]), then we have: if* K *is large enough, i.e.,* $K \geq 2\log(|\mathcal{S}||\mathcal{A}|H)/\rho_{\min}^2$*, w.p. at least* $1-\delta$*,* $C_{\mathcal{D}}^{\pi^*} \leq 2/\rho_{\min}$*.*

Our coverage definition (Definition 3.3) shares similar spirits as the one in [206]. It reflects how well the state-action samples in the offline dataset $\mathcal{D}$ cover the state-action pairs induced by the comparator policy $\pi^\star$. It is different from the coverage definition in [24] in which the denominator is $\mathbb{E}_{s,a \sim d_h^{\pi^b}} \mathbb{H}^2 \left(P(s,a) \parallel P^\star(s,a) \right)$ where π^b is the fixed behavior policy used to collect $\mathcal{D}$. This definition does not apply in our setting since $\mathcal{D}$ is not necessarily generated by some underlying fixed behavior policy. On the other hand, our horizon-free result does not hold in the setting of [24] where $\mathcal{D}$ is collected with a fixed behavior policy π^b with the concentrability coefficient defined in their way. We leave the derivation of horizon-free results in the setting from [24] as a future work.

Now we are ready to present the main theorem of Algorithm 2, which provides a tighter performance gap than that by [24].

Theorem 3.4.1 (Performance gap of Algorithm 2)**.** *For any $\delta \in (0,1)$, let $\beta = 4\log(|\mathcal{P}|/\delta)$, w.p. at least $1-\delta$, Algorithm 2 learns a policy $\widehat{\pi}$ that enjoys the following performance gap with respect to any comparator policy π^*:*

$$V^{\pi^*} - V^{\widehat{\pi}} \leq O\left(\sqrt{C^{\pi^*}\mathrm{VaR}_{\pi^*}\log(|\mathcal{P}|/\delta)/K} + C^{\pi^*}\log(|\mathcal{P}|/\delta)/K\right) .$$

Comparing to the theorem (Theorem 2) of CPPO-LR from [24], our bound has two improvements. First, our bound is horizon-free (not even any $\log(H)$ dependence), while the bound in [24] has $\mathrm{poly}(H)$ dependence. Second, our bound scales with $\mathrm{VaR}_{\pi^*} \in [0,1]$, which can be small when $\mathrm{VaR}_{\pi^*} \ll 1$. For deterministic system and policy π^*, we have $\mathrm{VaR}_{\pi^*} = 0$ which means the sample complexity now scales at a faster rate C^{π^*}/K. The proof is in Appendix A.1.9.

We show that the same algorithm can achieve $1/K$ rate when $P^\star$ is deterministic (but rewards could be random, and the algorithm does not need to know the condition that $P^\star$ is deterministic).

Corollary 3.4 (C^{π^*}/K performance gap of Algorithm 2 with deterministic transitions)**.** *When the ground truth transition $P^\star$ of the MDP is deterministic, for any $\delta \in (0,1)$, let $\beta = 4\log(|\mathcal{P}|/\delta)$, w.p. at least $1-\delta$, Algorithm 2 learns a policy $\widehat{\pi}$ that enjoys the following performance gap with respect to any comparator policy π^*:*

$$V^{\pi^*} - V^{\widehat{\pi}} \leq O\left(C^{\pi^*}\log(|\mathcal{P}|/\delta)/K\right) .$$

For infinite model class $\mathcal{P}$, we have a similar result in the following corollary.

Corollary 3.5 (Performance gap of Algorithm 2 with infinite model class $\mathcal{P}$)**.** *When the model class $\mathcal{P}$ is infinite, for any $\delta \in (0,1)$, let $\beta = 7\log(\mathcal{N}_{[]}((KH|\mathcal{S}|)^{-1}, \mathcal{P}, \|\cdot\|_\infty)/\delta)$, w.p. at least $1-\delta$, Algorithm 2 learns a policy $\widehat{\pi}$ that enjoys the following PAC bound w.r.t. any comparator policy π^*:*

$$V^{\pi^*} - V^{\widehat{\pi}} \le O\left(\sqrt{\frac{C^{\pi^*}\mathrm{VaR}_{\pi^*}\log(\mathcal{N}_{[]}((KH|\mathcal{S}|)^{-1}, \mathcal{P}, \|\cdot\|_\infty)/\delta)}{K}} + \frac{C^{\pi^*}\log(\mathcal{N}_{[]}((KH|\mathcal{S}|)^{-1}, \mathcal{P}, \|\cdot\|_\infty)/\delta)}{K}\right),$$

where $\mathcal{N}_{[]}((KH|\mathcal{S}|)^{-1}, \mathcal{P}, \|\cdot\|_\infty)$ is the bracketing number defined in Definition 3.2.

Our next example gives the explicit performance gap bound for tabular MDPs.

Example 3 (Tabular MDPs)**.** *For tabular MDPs, we have $\mathcal{N}_{[]}(\epsilon, \mathcal{P}, \|\cdot\|_\infty)$ upper-bounded by $(c/\epsilon)^{|\mathcal{S}|^2|\mathcal{A}|}$ (e.g., see [24]). Then with probability at least $1-\delta$, let $\beta = 7\log(\mathcal{N}_{[]}((KH|\mathcal{S}|)^{-1}, \mathcal{P}, \|\cdot\|_\infty)/\delta)$, Algorithm 2 learns a policy $\widehat{\pi}$ satisfying the following performance gap with respect to any comparator policy π^*:*

$$V^{\pi^*} - V^{\widehat{\pi}} \le O\Bigg(|\mathcal{S}|\sqrt{|\mathcal{A}|C^{\pi^*}\mathrm{VaR}_{\pi^*}\log(KH|\mathcal{S}|/\delta)/K} + |\mathcal{S}|^2|\mathcal{A}|C^{\pi^*}\log(KH|\mathcal{S}|/\delta)/K\Bigg), \tag{3.8}$$

The closest result to us is from [72], which analyzes the MBRL for tabular MDPs and obtains a performance gap $\tilde{O}(\sqrt{\frac{1}{Kd_m}} + \frac{|\mathcal{S}|}{Kd_m})$, where d_m is the minimum visiting probability for the behavior policy to visit each state and action. Note that their result is not instance-dependent, which makes their gap only $\tilde{O}(1/\sqrt{K})$ even

when the environment is deterministic and π^* is deterministic. In a sharp contrast, our analysis shows a better $\tilde{O}(1/K)$ gap under the deterministic environment. Our result would still have the $\log H$ dependence, and we leave getting rid of this logarithmic dependence on the horizon H as an open problem.

Chapter 4

Provable Zero-Shot Generalization in Offline Reinforcement Learning

In this chapter, we study offline reinforcement learning (RL) with zero-shot generalization property (ZSG), where the agent has access to an offline dataset including experiences from different environments, and the goal of the agent is to train a policy over the training environments which performs well on test environments without further interaction. Existing work showed that classical offline RL fails to generalize to new, unseen environments. We propose pessimistic empirical risk minimization (PERM) and pessimistic proximal policy optimization (PPPO), which leverage pessimistic policy evaluation to guide policy learning and enhance generalization. We show that both PERM and PPPO are capable of finding a near-optimal policy with ZSG. Our result serves as a first step in understanding the foundation of the generalization phenomenon in offline reinforcement learning. This chapter is based on our publication [2].

4.1 Introduction

Offline reinforcement learning (RL) has become increasingly significant in modern RL because it eliminates the need for direct interaction between the agent and the environment; instead, it relies solely on learning from an offline training dataset. However, in practical applications, the offline training dataset often originates from a different environment than the one of interest. This discrepancy necessitates evaluating RL agents in a generalization setting, where the training involves a finite number of environments drawn from a specific distribution, and the testing is conducted on a distinct set of environments from the same or different distribution. This scenario is commonly referred to as the zero-shot generalization (ZSG) challenge which has been studied in online RL[28, 29, 30, 31, 32, 33], as the agent receives no training data from the environments it is tested on.

A number of recent empirical studies [3, 34, 35] have recognized this challenge and introduced various offline RL methodologies that are capable of ZSG. Notwithstanding the lack of theoretical backing, these methods are somewhat restrictive; for instance, some are only effective for environments that vary solely in observations[35], while others are confined to the realm of imitation learning[34], thus limiting their applicability to a comprehensive framework of offline RL with ZSG capabilities. Concurrently, theoretical advancements [36, 37] in this domain have explored multi-task offline RL by focusing on representation learning. These approaches endeavor to derive a low-rank representation of states and actions, which inherently requires additional

interactions with the downstream tasks to effectively formulate policies based on these representations. Therefore, we raise a natural question:

Can we design provable offline RL with zero-shot generalization ability?

We propose novel offline RL frameworks that achieve ZSG to address this question affirmatively. Our contributions are listed as follows.

- We first analyze when existing offline RL approaches fail to generalize without further algorithm modifications. Specifically, we prove that if the offline dataset does not contain context information, then it is impossible for vanilla RL that equips a Markovian policy to achieve a ZSG property. We show that the offline dataset from a contextual Markov Decision Process (MDP) is not distinguishable from a vanilla MDP which is the average of contextual Markov Decision Process over all contexts. Such an analysis verifies the necessity of new RL methods with ZSG property.

- We propose two meta-algorithms called pessimistic empirical risk minimization (PERM) and pessimistic proximal policy optimization (PPPO) that enable ZSG for offline RL [91]. In detail, both of our algorithms take a pessimistic policy evaluation (PPE) oracle as its component and output policies based on offline datasets from multiple environments. Our result shows that the sub-optimalities of the output policies are bounded by both the supervised learning error, which is controlled by

Table 4.1: Summary of our algorithms and their suboptimality gaps, where $\mathcal{A}$ is the action space, H is the length of episode, n is the number of environments in the offline dataset. Note that in the multi-environment setting, π^* is the near-optimal policy w.r.t. expectation (defined in Section 8.2). $\mathcal{N}$ is the covering number of the policy space Π w.r.t. distance $\mathrm{d}(\pi^1, \pi^2) = \max_{s\in\mathcal{S}, h\in[H]} \|\pi_h^1(\cdot|s) - \pi_h^2(\cdot|s)\|_1$. The uncertainty quantifier $\Gamma_{i,h}$ are tailored with the oracle return in the corresponding algorithms (details are in Section 4.4).

Algorithm	**Suboptimality Gap**
PERM (our Algo.4)	$\sqrt{\log(\mathcal{N})/n} + n^{-1}\sum_{i=1}^{n}\sum_{h=1}^{H} \mathbb{E}_{i,\pi^*}\big[\Gamma_{i,h}(s_h, a_h) \,\big|\, s_1 = x_1\big]$
PPPO (our Algo.5)	$\sqrt{\log\lvert\mathcal{A}\rvert\, H^2/n} + n^{-1}\sum_{i=1}^{n}\sum_{h=1}^{H} \mathbb{E}_{i,\pi^*}\big[\Gamma_{i,h}(s_h, a_h) \,\big|\, s_1 = x_1\big]$

the number of different environments, and the reinforcement learning error, which is controlled by the coverage of the offline dataset to the optimal policy. Please refer to Table 4.1 for a summary of our results. To the best of our knowledge, our proposed algorithms are the first offline RL methods that provably enjoy the ZSG property.

Notation We use lower case letters to denote scalars, and use lower and upper case bold face letters to denote vectors and matrices respectively. We denote by $[n]$ the set $\{1, \ldots, n\}$. For a vector $\mathbf{x} \in \mathbb{R}^d$ and a positive semi-definite matrix $\mathbf{\Sigma} \in \mathbb{R}^{d\times d}$, we denote by $\|\mathbf{x}\|_2$ the vector's Euclidean norm and define $\|\mathbf{x}\|_{\mathbf{\Sigma}} = \sqrt{\mathbf{x}^\top \mathbf{\Sigma}\mathbf{x}}$. For two positive sequences $\{a_n\}$ and $\{b_n\}$ with $n = 1, 2, \ldots$, we write $a_n = O(b_n)$ if there exists an absolute constant $C > 0$ such that $a_n \le Cb_n$ holds for all $n \ge 1$ and write $a_n = \Omega(b_n)$ if there exists an absolute constant $C > 0$ such that $a_n \ge Cb_n$ holds for all $n \ge 1$. We use $\tilde{O}(\cdot)$ to further hide the polylogarithmic factors. We use $(x_i)_{i=1}^n$ to denote sequence $(x_1, ..., x_n)$, and we use $\{x_i\}_{i=1}^n$ to denote the set $\{x_1, ..., x_n\}$. We use $\mathrm{KL}(p\|q)$ to denote the KL

distance between distributions p and q, defined as $\int p \log(p/q)$. We use $\mathbb{E}[x], \mathbb{V}[x]$ to denote expectation and variance of a random variable x.

4.2 Preliminaries

Contextual MDP We study *contextual episodic MDPs*, where each MDP $\mathcal{M}_c$ is associated with a context $c \in C$ belongs to the context space C. Furthermore, $\mathcal{M}_c = \{M_{c,h}\}_{h=1}^H$ consists of H different individual MDPs, where each individual MDP $M_{c,h} := (\mathcal{S}, \mathcal{A}, P_{c,h}(s'|s,a), r_{c,h}(s,a))$. Here $\mathcal{S}$ denotes the state space, $\mathcal{A}$ denotes the action space, $P_{c,h}$ denotes the transition function and $r_{c,h}$ denotes the reward function at stage h. We assume the starting state for each $\mathcal{M}_c$ is the same state x_1. In this work, we interchangeablely use "environment" or MDP to denote the MDP $\mathcal{M}_c$ with different contexts.

Policy and value function We denote the policy π_h at stage h as a mapping $\mathcal{S} \to \Delta(\mathcal{A})$, which maps the current state to a distribution over the action space. We use $\pi = \{\pi_h\}_{h=1}^H$ to denote their collection. Then for any episodic MDP $\mathcal{M}$, we define the value function for some policy π as

$$V^\pi_{\mathcal{M},h}(x) := \mathbb{E}[r_h + ... + r_H | s_h = x, a_{h'} \sim \pi_{h'}, r_{h'} \sim r_{h'}(s_{h'}, a_{h'}), s_{h'+1} \sim P_{h'}(\cdot|s_{h'}, a_{h'}),\ h' \geq h],$$
$$Q^\pi_{M,h}(x,a) := \mathbb{E}[r_h + ... + r_H | s_h = x, a_h = a, r_h \sim r_h(s_h, a_h), s_{h'} \sim P_{h'-1}(\cdot|s_{h'-1}, a_{h'-1}), a_{h'} \sim \pi_{h'},$$
$$r_{h'} \sim r_{h'}(s_{h'}, a_{h'}),\ h' \geq h+1].$$

For any individual MDP M with reward r and transition dynamic P, we denote its Bellman operator $[\mathbb{B}_M f](x,a)$ as $[\mathbb{B}_M f](s,a) := \mathbb{E}[r_h(s,a) + f(s')|s' \sim P(\cdot|s,a)]$. Then we have the well-known Bellman equation

$$V^\pi_{\mathcal{M},h}(x) = \langle Q^\pi_{\mathcal{M},h}(x,\cdot), \pi_h(\cdot|x)\rangle_{\mathcal{A}},\ Q^\pi_{\mathcal{M},h}(x,a) = [\mathbb{B}_{M_h} V^\pi_{\mathcal{M},h+1}](x,a).$$

For simplicity, we use $V^{\pi}_{c,h}, Q^{\pi}_{c,h}, \mathbb{B}_{c,h}$ to denote $V^{\pi}_{\mathcal{M}_c,h}, Q^{\pi}_{\mathcal{M}_c,h}, \mathbb{B}_{M_{c,h}}$. We also use $\mathbb{P}_c$ to denote $\mathbb{P}_{\mathcal{M}_c}$, the joint distribution of any potential objects under the $\mathcal{M}_c$ episodic MDP. We would like to find the near-optimal policy π^* w.r.t. expectation, i.e., $\pi^* := \operatorname{argmax}_{\pi\in\Pi} \mathbb{E}_{c\sim C} V^{\pi}_{c,1}(x_c)$, where Π is the set of collection of Markovian policies, and with a little abuse of notation, we use $\mathbb{E}_{c\sim C}$ to denote the expectation taken w.r.t. the i.i.d. sampling of context c from the context space. Then our goal is to develop the *generalizable RL* with small *zero-shot generalization gap (ZSG gap)*, defined as follows:

$$\mathrm{SubOpt}(\pi) := \mathbb{E}_{c\sim C}\big[V^{\pi^*}_{c,1}(x_1)\big] - \mathbb{E}_{c\sim C}\big[V^{\pi}_{c,1}(x_1)\big].$$

Remark 2. *We briefly compare generalizable RL with several related settings. Robust RL [211] aims to find the best policy for the worst-case environment, whereas generalizable RL seeks a policy that performs well in the average-case environment. Meta-RL [212] enables few-shot adaptation to new environments, either through policy updates [213] or via history-dependent policies [214]. In contrast, generalizable RL primarily focuses on the zero-shot setting. In the general POMDP framework [215], agents need to maintain history-dependent policies to implicitly infer environment information, while generalizable RL aims to discover a single state-dependent policy that generalizes well across all environments.*

Remark 3. *[126] showed that in online RL, for a certain family of contextual MDPs, it is inherently impossible to determine an optimal policy for each individual MDP. Given that offline RL poses greater challenges than its online counterpart, this impossibility extends to finding optimal policies for each MDP in a zero-shot offline RL setting as well, which justifies our optimiza-*

tion objective on the ZSG gap. Moreover, [126] showed that the few-shot RL is able to find the optimal policy for individual MDPs. Clearly, such a setting is stronger than ours, and the additional interactions are often hard to be satisfied in real-world practice. We leave the study of such a setting for future work.

Offline RL data collection process The data collection process is as follows. An experimenter i.i.d. samples number n of contextual episodic MDP M_i from the context set (*e.g.*, $i \sim C$). For each episodic MDP M_i, the experimenter collects dataset $\mathcal{D}_i := \{(x_{i,h}^\tau, a_{i,h}^\tau, r_{i,h}^\tau)_{h=1}^H\}_{\tau=1}^K$ which includes K trajectories. Note that the action $a_{i,h}^\tau$ selected by the experimenter can be arbitrary, and it does not need to follow a specific behavior policy [91]. We assume that $\mathcal{D}_i$ is compliant with the episodic MDP $\mathcal{M}_i$, which is defined as follows.

Definition 4.1 ([91])**.** *For $\mathcal{D}_i := \{(x_{i,h}^\tau, a_{i,h}^\tau, r_{i,h}^\tau)_{h=1}^H\}_{\tau=1}^K$, let $\mathbb{P}_{\mathcal{D}_i}$ be the joint distribution of the data collecting process. We say $\mathcal{D}_i$ is compliant with episodic MDP $\mathcal{M}_i$ if for any $x' \in \mathcal{S}, r', \tau \in [K], h \in [H]$, we have*

$$\begin{aligned}&\mathbb{P}_{\mathcal{D}_i}(r_{i,h}^\tau = r', x_{i,h+1}^\tau = x' | \{(x_{i,h}^j, a_{i,h}^j)\}_{j=1}^\tau, \{(r_{i,h}^j, x_{i,h+1}^j)\}_{j=1}^{\tau-1}) \\ &\quad = \mathbb{P}_i(r_{i,h}(s_h, a_h) = r', s_{h+1} = x' | s_h = x_h^\tau, a_h = a_h^\tau).\end{aligned}$$

In general, we claim $\mathcal{D}_i$ is compliant with $\mathcal{M}_i$ when the conditional distribution of any tuple of reward and next state in $\mathcal{D}_i$ follows the conditional distribution determined by MDP $\mathcal{M}_i$.

4.3 Offline RL without context indicator information

In this section, we show that directly applying existing offline RL algorithms over datasets from multiple environments *without* maintaining their identity information cannot yield a sufficient ZSG property, which is aligned with the existing observation of the poor generalization performance of offline RL [3].

In detail, given contextual MDPs $\mathcal{M}_1, ..., \mathcal{M}_n$ and their corresponding offline datasets $\mathcal{D}_1, ..., \mathcal{D}_n$, we assume the agent only has the access to the offline dataset $\bar{\mathcal{D}} = \cup_{i=1}^{n} \mathcal{D}_i$, where $\bar{\mathcal{D}} = \{(x^{\tau}_{c_\tau,h}, a^{\tau}_{c_\tau,h}, r^{\tau}_{c_\tau,h})_{h=1}^{H}\}_{\tau=1}^{K}$. Here $c_\tau \in C$ is the context information of trajectory τ, which is *unknown* to the agent. To explain why offline RL without knowing context information performs worse, we have the following proposition suggesting the offline dataset from multiple MDPs is not distinguishable from an "average MDP" if the offline dataset does not contain context information.

Proposition 4.1. *$\bar{\mathcal{D}}$ is compliant with* average MDP $\bar{\mathcal{M}} := \{\bar{M}_h\}_{h=1}^{H}$, $\bar{M}_h := \big(\mathcal{S}, \mathcal{A}, H, \bar{P}_h, \bar{r}_h\big)$,

$$\bar{P}_h(x'|x,a) := \mathbb{E}_{c\sim C} \frac{P_{c,h}(x'|x,a)\mu_{c,h}(x,a)}{\mathbb{E}_{c\sim C}\mu_{c,h}(x,a)},$$
$$\mathbb{P}(\bar{r}_h = r|x,a) := \mathbb{E}_{c\sim C} \frac{\mathbb{P}(\bar{r}_{c,h} = r|x,a)\mu_{c,h}(x,a)}{\mathbb{E}_{c\sim C}\mu_{c,h}(x,a)},$$

where $\mu_{c,h}(\cdot,\cdot)$ is the data collection distribution of (s,a) at stage h in dataset $\mathcal{D}_c$.

Proof. See Appendix A.2.1.1. □

Proposition 4.1 suggests that if no context information is revealed, then the merged offline dataset $\bar{\mathcal{D}}$ is equivalent to a dataset collected from the average MDP $\bar{\mathcal{M}}$. Therefore, for any offline RL which outputs a Markovian policy, it converges to the optimal policy $\bar{\pi}^*$ of the average MDP $\bar{\mathcal{M}}$.

In general, $\bar{\pi}^*$ can be very different from π^* when the transition probability functions of each environment are different. For example, consider the 2-context cMDP problem shown in Figure 4.1, each context consists of one state and three possible actions. The offline dataset distributions μ are marked on the arrows that both of the distributions are following near-optimal policy. By Proposition 4.1, in average MDP $\bar{\mathcal{M}}$ the reward of the middle action is deterministically 0, while both upper and lower actions are deterministically 1. As a result, the optimal policy $\bar{\pi}^*$ will only have positive probabilities toward upper and lower actions. This leads to $\mathbb{E}_{c\sim C}[V_{c,1}^{\bar{\pi}^*}(x_1)] = 0$, though we can see that π^* is deterministically choosing the middle action and $\mathbb{E}_{c\sim C}[V_{c,1}^{\pi^*}(x_1)] = 0.5$. This theoretically illustrates that the generalization ability of offline RL algorithms without leveraging context information is weak. In sharp contrast, imitation learning such as behavior cloning (BC) converges to the teacher policy that is independent of the specific MDP. Therefore, offline RL methods such as CQL [95] might enjoy worse generalization performance compared with BC, which aligns with the observation made by [3].

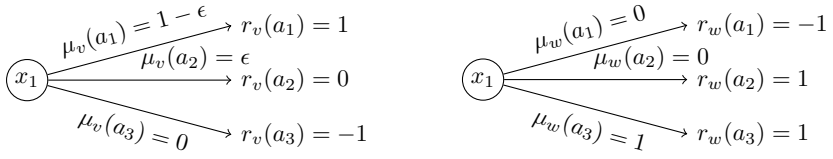


Figure 4.1: Two Contextual MDPs with the same compliant average MDPs. The discrete contextual space is defined as $C = \{v, w\}$ and both MDPs satisfies $\mathcal{S} = \{x_1\}, \mathcal{A} = \{a_1, a_2, a_3\}, H = 1$. The data collection distributions μ and rewards r for each action of each context are specified in the graph.

4.4 Provable offline RL with zero-shot generalization

In this section, we propose offline RL with small ZSG gaps. We show that two popular offline RL approaches, *model-based RL* and *policy optimization-based RL*, can output RL agent with ZSG ability, with a pessimism-style modification that encourages the agent to follow the offline dataset pattern.

4.4.1 Pessimistic policy evaluation

We consider a meta-algorithm to evaluate any policy π given an offline dataset, which serves as a key component in our proposed offline RL with ZSG. To begin with, we consider a general individual MDP and an oracle $\mathbb{O}$, which returns us an empirical Bellman operator and an uncertainty quantifier, defined as follows.

Definition 4.2 ([91])**.** *For any individual MDP M, a dataset $\mathcal{D} \subseteq \mathcal{S} \times \mathcal{A} \times \mathcal{S} \times [0, 1]$ that is compliant with M, a test function $V_{\mathcal{D}} \subseteq [0, H]^{\mathcal{S}}$ and a confidence level ξ, we have an oracle $\mathbb{O}(\mathcal{D}, V_{\mathcal{D}}, \xi)$ that returns $(\hat{\mathbb{B}}V_{\mathcal{D}}(\cdot, \cdot), \Gamma(\cdot, \cdot))$, a tuple of Empirical Bellman operator*

Algorithm 3 Pessimistic Policy Evaluation (PPE)

Require: Offline dataset $\{\mathcal{D}_{i,h}\}_{h=1}^{H}$, policy $\pi = (\pi_h)_{h=1}^{H}$, confidence probability $\delta \in (0,1)$.

1: Initialize $\hat{V}_{i,H+1}^{\pi}(\cdot) \leftarrow 0,\ \forall i \in [n]$.
2: **for** step $h = H, H-1, \ldots, 1$ **do**
3: Let $(\hat{\mathbb{B}}_{i,h}\hat{V}_{i,h+1}^{\pi})(\cdot,\cdot), \Gamma_{i,h}(\cdot,\cdot) \leftarrow \mathbb{O}(\mathcal{D}_{i,h}, \hat{V}_{i,h+1}^{\pi}, \delta)$
4: Set $\hat{Q}_{i,h}^{\pi}(\cdot,\cdot) \leftarrow \min\{H-h+1, (\hat{\mathbb{B}}_{i,h}\hat{V}_{i,h+1}^{\pi})(\cdot,\cdot) - \Gamma_{i,h}(\cdot,\cdot)\}^{+}$
5: Set $\hat{V}_{i,h}^{\pi}(\cdot) \leftarrow \langle \hat{Q}_{i,h}^{\pi}(\cdot,\cdot), \pi_h(\cdot|\cdot)\rangle_{\mathcal{A}}$
6: **end for**
7: **return** $\hat{V}_{i,1}^{\pi}(\cdot), \ldots, \hat{V}_{i,H}^{\pi}(\cdot), \hat{Q}_{i,1}^{\pi}(\cdot,\cdot), \ldots, \hat{Q}_{i,H}^{\pi}(\cdot,\cdot)$.

and uncertainty quantifier, satisfying

$$\mathbb{P}_{\mathcal{D}}\Big(\big|(\hat{\mathbb{B}}V_{\mathcal{D}})(x,a) - (\mathbb{B}_M V_{\mathcal{D}})(x,a)\big| \le \Gamma(x,a) \text{ for all } (x,a) \in \mathcal{S}\times\mathcal{A}\Big) \ge 1-\xi.$$

Remark 4. *Here we adapt a test function $V_{\mathcal{D}}$ that can depend on the dataset $\mathcal{D}$ itself. Therefore, Γ is a function that depends on both the dataset and the test function class. We do not specify the test function class in this definition, and we will discuss its specific realization in Section 4.5.*

Remark 5. *For general non-linear MDPs, one may employ the bootstrapping technique to estimate uncertainty, in line with the bootstrapped DQN approach developed by [216]. We note that when the bootstrapping method is straightforward to implement, the assumption of having access to an uncertainty quantifier is reasonable.*

Based on the oracle $\mathbb{O}$, we propose our pessimistic policy evaluation (PPE) algorithm as Algorithm 3. In general, PPE takes a given policy π as its input, and its goal is to evaluate the V value and Q value $\{(V_{i,h}^{\pi}, Q_{i,h}^{\pi})\}_{h=1}^{H}$ of π on MDP $\mathcal{M}_i$. Since the agent is not allowed to interact with $\mathcal{M}_i$, PPE evaluates the value based

on the offline dataset $\{\mathcal{D}_{i,h}\}_{h=1}^{H}$. At each stage h, PPE utilizes the oracle $\mathbb{O}$ and obtains the empirical Bellman operator based on $\mathcal{D}_{i,h}$ as well as its uncertainty quantifier, with high probability. Then PPE applies the *pessimism principle* to build the estimation of the Q function based on the empirical Bellman operator and the uncertainty quantifier. Such a principle has been widely studied and used in offline policy optimization, such as pessimistic value iteration (PEVI) [91]. To compare with, we use the pessimism principle in the policy evaluation problem.

Remark 6. *In our framework, pessimism can indeed facilitate generalization, rather than hinder it. Specifically, we employ pessimism to construct reliable Q functions for each environment individually. This approach supports broader generalization by maintaining multiple Q-networks separately. By doing so, we ensure that each Q function is robust within its specific environment, while the collective set of Q functions enables the system to generalize across different environments.*

4.4.2 Model-based approach: pessimistic empirical risk minimization

Given PPE, we propose algorithms that have the ZSG ability. We first propose a pessimistic empirical risk minimization (PERM) method which is model-based and conceptually simple. The algorithm details are in Algorithm 4. In detail, for each dataset $\mathcal{D}_i$ drawn from i-th environments, PERM builds a model using PPE to evaluate the policy π under the environment $\mathcal{M}_i$. Then PERM outputs a policy $\pi^{\text{PERM}} \in \Pi$ that maximizes the

Algorithm 4 Pessimistic Empirical Risk Minimization (PERM)

Require: Offline dataset $\mathcal{D} = \{\mathcal{D}_i\}_{i=1}^n, \mathcal{D}_i := \{(x_{i,h}^\tau, a_{i,h}^\tau, r_{i,h}^\tau)_{h=1}^H\}_{\tau=1}^K$, policy class Π, confidence probability $\delta \in (0,1)$, a pessimistic offline policy evaluation algorithm **Evaluation** as a subroutine.

1: Set $\mathcal{D}_{i,h} = \{(x_{i,h}^\tau, a_{i,h}^\tau, r_{i,h}^\tau, x_{i,h+1}^\tau)\}_{\tau=1}^K$

2: $\pi^{\mathrm{PERM}} = \mathrm{argmax}_{\pi\in\Pi} \frac{1}{n}\sum_{i=1}^n \hat{V}_{i,1}^\pi(x_1)$,
where $[\hat{V}_{i,1}^\pi(\cdot), \cdot, \ldots, \cdot] = \textbf{Evaluation}\Big(\{\mathcal{D}_{i,h}\}_{h=1}^H, \pi, \delta/(3nH\mathcal{N}_{(Hn)^{-1}}^\Pi)\Big)$

3: **return** π^{PERM}.

average pessimistic value, i.e., $1/n\sum_{i=1}^n \hat{V}_{i,1}^\pi(x_1)$. Our approach is inspired by the classical empirical risk minimization approach adopted in supervised learning, and the Optimistic Model-based ERM proposed in [126] for online RL. Our setting is more challenging than the previous ones due to the RL setting and the offline setting, where the interaction between the agent and the environment is completely disallowed. Therefore, unlike [126], which adopted an optimism-style estimation to the policy value, we adopt a pessimism-style estimation to fight the distribution shift issue in the offline setting.

Next we propose a theoretical analysis of PERM. Denote $\mathcal{N}_\epsilon^\Pi$ as the ϵ-covering number of the policy space Π w.r.t. distance $\mathrm{d}(\pi^1, \pi^2) = \max_{s\in\mathcal{S}, h\in[H]} \|\pi_h^1(\cdot|s) - \pi_h^2(\cdot|s)\|_1$. Then we have the following theorem to provide an upper bound of the suboptimality gap of the output policy π^{PERM}.

Theorem 4.4.1. *Set the Evaluation subroutine in Algorithm 4 as PPE (Algo.3). Let $\Gamma_{i,h}$ be the uncertainty quantifier returned by $\mathbb{O}$ through the PERM. Then w.p. at least $1-\delta$, the output π^{PERM}*

of Algorithm 4 satisfies

$$\text{SubOpt}(\pi^{\text{PERM}}) \leq \underbrace{7\sqrt{\frac{2\log(6\mathcal{N}^{\Pi}_{(Hn)^{-1}}/\delta)}{n}}}_{I_1\text{:Supervised learning (SL) error}} + \underbrace{\frac{2}{n}\sum_{i=1}^{n}\sum_{h=1}^{H}\mathbb{E}i, \pi^{*}\Gamma_{i,h}(s_h, a_h)|s_1 = x_1}_{I_2\text{:Reinforcement learning (RL) error}}, \tag{4.1}$$

where $\mathbb{E}_{i,\pi^*}$ is w.r.t. the trajectory induced by π^* with the transition $\mathcal{P}_i$ in the underlying MDP $\mathcal{M}_i$.

Proof. See Appendix A.2.2.1. □

Remark 7. *The covering number $\mathcal{N}^{\Pi}_{(Hn)^{-1}}$ depends on the policy class Π. Without any specific assumptions, the policy class Π that consists of all the policies $\pi = \{\pi_h\}_{h=1}^{H}, \pi_h : \mathcal{S} \mapsto \Delta(\mathcal{A})$ and the log ϵ-covering number $\log \mathcal{N}^{\Pi}_{\epsilon} = O(|\mathcal{A}||\mathcal{S}|H\log(1 + |\mathcal{A}|/\epsilon))$.*

Remark 8. *The SL error can be easily improved to a distribution-dependent bound $\log \mathcal{N} \cdot Var/\sqrt{n}$, where $\mathcal{N}$ is the covering number term denoted in I_1, $Var = \max_{\pi} \mathbb{V}_{c\sim C} V^{\pi}_{c,1}(x_1)$ is the variance of the context distribution, by using a Bernstein-type concentration inequality in our proof. Therefore, for the singleton environment case where $|C| = 1$, our suboptimality gap reduces to the one of PEVI in [91].*

Remark 9. *In real-world settings, as the number of sampled contexts n may be very large, it is unrealistic to manage n models simultaneously in the implementation of PERM algorithm, thus we provide the suboptimality bound in line with Theorem 4.4.1 when the offline dataset is merged into m contexts such that $m < n$. See Theorem A.2.7 in Appendix A.2.3.*

Theorem 4.4.1 shows that the ZSG gap of PERM is bounded by two terms I_1 and I_2. I_1, which we call *supervised learning*

Algorithm 5 Pessimistic Proximal Policy Optimzation (PPPO)

Require: Offline dataset $\mathcal{D} = \{\mathcal{D}_i\}_{i=1}^n, \mathcal{D}_i := \{(x_{i,h}^\tau, a_{i,h}^\tau, r_{i,h}^\tau)_{h=1}^H\}_{\tau=1}^K$, confidence probability $\delta \in (0,1)$, a pessimistic offline policy evaluation algorithm **Evaluation** as a subroutine.

1: Set $\mathcal{D}_{i,h} = \{(x_{i,h}^{\tau\cdot H+h}, a_{i,h}^{\tau\cdot H+h}, r_{i,h}^{\tau\cdot H+h}, x_{i,h+1}^{\tau\cdot H+h})\}_{\tau=0}^{\lfloor K/H\rfloor-1}$
2: Set $\pi_{0,h}(\cdot|\cdot)$ as uniform distribution over $\mathcal{A}$ and $\hat{Q}_{0,h}^{\pi_0}(\cdot,\cdot)$ as zero functions.
3: **for** $i = 1, 2, \cdots, n$ **do**
4: Set $\pi_{i,h}(\cdot|\cdot) \propto \pi_{i-1,h}(\cdot|\cdot) \cdot \exp(\alpha \cdot \hat{Q}_{i-1,h}^{\pi_{i-1}}(\cdot,\cdot))$
5: Set $[\cdot, \ldots, \cdot, \hat{Q}_{i,1}^{\pi_i}(\cdot,\cdot), \ldots, \hat{Q}_{i,H}^{\pi_i}(\cdot,\cdot)] =$ **Evaluation**$(\{\mathcal{D}_{i,h}\}_{h=1}^H, \pi_i, \delta/(nH))$
6: **end for**
7: **return** $\pi^{\mathrm{PPPO}} = \mathrm{random}(\pi_1, ..., \pi_n)$

error, depends on the number of environments n in the offline dataset $\mathcal{D}$ and the covering number of the function (policy) class, which is similar to the generalization error in supervised learning. I_2, which we call it *reinforcement learning error*, is decided by the optimal policy π^* that achieves the best zero-shot generalization performance and the uncertainty quantifier $\Gamma_{i,h}$. In general, I_2 is the "intrinsic uncertainty" denoted by [91] over n MDPs, which characterizes how well each dataset $\mathcal{D}_i$ covers the optimal policy π^*.

4.4.3 Model-free approach: pessimistic proximal policy optimization

PERM in Algorithm 4 works as a general model-based algorithm framework to enable ZSG for any pessimistic policy evaluation oracle. However, note that in order to implement PERM, one needs to maintain n different models or critic functions simultaneously in order to evaluate $\sum_{i=1}^n \hat{V}_{i,1}^\pi(x_1)$ for any candidate policy π. Note that existing online RL [110] achieves ZSG by

a model-free approach, which only maintains n policies rather than models/critic functions. Therefore, one natural question is whether we can design a *model-free* offline RL algorithm also with access only to policies.

We propose the pessimistic proximal policy optimization (PPPO) in Algorithm 5 to address this issue. Our algorithm is inspired by the optimistic PPO [217] originally proposed for online RL. PPPO also adapts PPE as its subroutine to evaluate any given policy pessimistically. Unlike PERM, PPPO only maintains n policies $\pi_1, ..., \pi_n$, each of them is associated with an MDP $\mathcal{M}_n$ from the offline dataset. In detail, PPPO assigns an order for MDPs in the offline dataset and names them $\mathcal{M}_1, ..., \mathcal{M}_n$. For i-th MDP $\mathcal{M}_i$, PPPO selects the i-th policy π_i as the solution of the proximal policy optimization starting from π_{i-1}, which is

$$\pi_i \leftarrow \operatorname*{argmax}_{\pi} V^{\pi}_{i-1,1}(x_1) - \alpha^{-1}\mathbb{E}_{i-1,\pi_{i-1}}[\mathrm{KL}(\pi\|\pi_{i-1})|s_1 = x_1], \tag{4.2}$$

where α is the step size parameter. Since $V^{\pi}_{i-1,1}(x_1)$ is not achievable, we use a linear approximation $L_{i-1}(\pi)$ to replace $V^{\pi}_{i-1,1}(x_1)$, where

$$L_{i-1}(\pi) = V^{\pi_{i-1}}_{i-1,1}(x_1) + \mathbb{E}_{i-1,\pi_{i-1}}\left[\sum_{h=1}^{H}\langle \hat{Q}^{\pi_{i-1}}_{i-1,h}(x_h,\cdot), \pi_h(\cdot|x_h) - \pi_{i-1,h}(\cdot|x_h)\rangle \middle| s_1 = x_1\right], \tag{4.3}$$

where $\hat{Q}^{\pi_{i-1}}_{i-1,h} \approx Q^{\pi_{i-1}}_{i-1,h}$ are the Q values evaluated on the offline dataset for $\mathcal{M}_{i-1}$. (4.2) and (4.3) give us a close-form solution of π in Line 4 in Algorithm 5. Such a routine corresponds to one iteration of PPO [218]. Finally, PPPO outputs π^{PPPO} as a random selection from $\pi_1, ..., \pi_n$.

Remark 10. *In Algorithm 5, we adopt a data-splitting trick [91] to build $\mathcal{D}_{i,h}$, where we only utilize each trajectory once for one data tuple at some stage h. It is only used to avoid the statistical dependency of $\hat{V}_{i,h+1}^{\pi_i}(\cdot)$ and $x_{i,h+1}^{\tau}$ for the purpose of theoretical analysis.*

The following theorem bounds the suboptimality of PPPO.

Theorem 4.4.2. *Set the Evaluation subroutine in Algorithm 5 as Algorithm 3. Let $\Gamma_{i,h}$ be the uncertainty quantifier returned by $\mathbb{O}$ through the PPPO. Selecting $\alpha = 1/\sqrt{H^2 n}$. Then selecting $\delta = 1/8$, w.p. at least $2/3$, we have*

$$SubOpt(\pi^{PPPO}) \leq 10\Big(\underbrace{\sqrt{\frac{\log|\mathcal{A}|H^2}{n}}}_{I_1:SL\ error} + \underbrace{\frac{1}{n}\sum_{i=1}^{n}\sum_{h=1}^{H}\mathbb{E}i, \pi^*\Gamma_{i,h}(s_h,a_h)|s_1 = x_1}_{I_2:RL\ error} \Big).$$

where $\mathbb{E}_{i,\pi^*}$ is w.r.t. the trajectory induced by π^* with the transition $\mathcal{P}_i$ in the underlying MDP $\mathcal{M}_i$.

Proof. See Appendix A.2.2.2. □

Remark 11. *As in Remark 9, we also provide the suboptimality bound in line with Theorem 4.4.2 when the offline dataset is merged into m contexts such that $m < n$. See Theorem A.2.8 in Appendix A.2.3.*

Theorem 4.4.2 shows that the suboptimality gap of PPPO can also be bounded by the SL error I_1 and RL error I_2. Interestingly, I_1 in Theorem 4.4.2 for PPPO only depends on the cardinality of the action space $|\mathcal{A}|$, which is different from the covering number term in I_1 for PERM. Such a difference is due to the fact that PPPO outputs the final policy π^{PPPO} as a random selection from n existing policies, while PERM outputs one policy π^{PERM}.

Whether these two guarantees can be unified into one remains an open question.

4.5 Provable generalization for offline linear MDPs

In this section, we instantiate our Algo.4 and Algo.5 for general MDPs on specific MDP classes. We consider the linear MDPs defined as follows.

Assumption 4.1 ([219, 220])**.** *We assume $\forall i \in C, \mathcal{M}_i$ is a linear MDP with a known feature map $\phi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$ if there exist d unknown measures $\mu_{i,h} = (\mu_{i,h}^{(1)}, \ldots, \mu_{i,h}^{(d)})$ over $\mathcal{S}$ and an unknown vector $\theta_{i,h} \in \mathbb{R}^d$ such that*

$$\begin{aligned} &P_{i,h}(x' \,|\, x, a) = \langle \phi(x,a), \mu_{i,h}(x') \rangle, \\ &\mathbb{E}\big[r_{i,h}(s_h, a_h) \,\big|\, s_h = x, a_h = a\big] = \langle \phi(x,a), \theta_{i,h} \rangle \end{aligned} \tag{4.4}$$

for all $(x, a, x') \in \mathcal{S} \times \mathcal{A} \times \mathcal{S}$ at every step $h \in [H]$. We assume $\|\phi(x,a)\| \le 1$ for all $(x,a) \in \mathcal{S} \times \mathcal{A}$ and $\max\{\|\mu_{i,h}(\mathcal{S})\|, \|\theta_{i,h}\|\} \le \sqrt{d}$ at each step $h \in [H]$, and we define $\|\mu_{i,h}(\mathcal{S})\| = \int_{\mathcal{S}} \|\mu_{i,h}(x)\| \, \mathrm{d}x$.

We first specialize the general PPE algorithm (Algo.3) to obtain the PPE algorithm tailored for linear MDPs (Algo.6). This specialization is achieved by constructing $\hat{\mathbb{B}}_{i,h}\hat{V}^{\pi}_{i,h+1}$, $\Gamma_{i,h}$, and $\hat{V}^{\pi}_{i,h}$ based on the dataset $\mathcal{D}_i$. We denote the set of trajectory indexes in $\mathcal{D}_{i,h}$ as $\mathcal{B}_{i,h}$. Algo.6 subsequently functions as the policy evaluation subroutine in Algo.4 and Algo.5 for linear MDPs. In detail,

we construct $\hat{\mathbb{B}}_{i,h}\hat{V}_{i,h+1}$ (which is the estimation of $\mathbb{B}_{i,h}\hat{V}_{i,h+1}$) as $(\hat{\mathbb{B}}_{i,h}\hat{V}_{i,h+1})(x,a) = \phi(x,a)^\top \hat{w}_{i,h}$, where

$$\hat{w}_{i,h} = \text{argmin}_{w\in\mathbb{R}^d} \sum_{\tau\in\mathcal{B}_{i,h}} \big(r_{i,h}^\tau + \hat{V}_{i,h+1}(x_{i,h}^{-,\tau}) - \phi(x_{i,h}^\tau, a_{i,h}^\tau)^\top w\big)^2 + \lambda \cdot \|w\|_2^2 \tag{4.5}$$

with $\lambda > 0$ being the regularization parameter. The closed-form solution to (4.5) is in Line 4 in Algorithm 6. Besides, we construct the uncertainty quantifier $\Gamma_{i,h}$ based on $\mathcal{D}_i$ as

$$\Gamma_{i,h}(x,a) = \beta(\delta) \cdot \|\phi(x,a)\|_{\Lambda_{i,h}^{-1}}, \Lambda_{i,h} = \sum_{\tau\in\mathcal{B}_{i,h}} \phi(x_{i,h}^\tau, a_{i,h}^\tau)\phi(x_{i,h}^\tau, a_{i,h}^\tau)^\top + \lambda \cdot I,$$

with $\beta(\delta) > 0$ being the scaling parameter.

Algorithm 6 Pessimistic Policy Evaluation (PPE): Linear MDP

Require: Offline dataset $\{\mathcal{D}_{i,h}\}_{h=1}^H, \mathcal{D}_{i,h} = \{(x_{i,h}^\tau, a_{i,h}^\tau, r_{i,h}^\tau, x_{i,h}^{-,\tau})\}_{\tau\in\mathcal{B}_{i,h}}$, policy π, confidence probability $\delta \in (0,1)$.

1: Initialize $\hat{V}_{i,H+1}^\pi(\cdot) \leftarrow 0, \ \forall i \in [n]$.
2: **for** step $h = H, H-1, \ldots, 1$ **do**
3: Set $\Lambda_{i,h} \leftarrow \sum_{\tau\in\mathcal{B}_{i,h}} \phi(x_{i,h}^\tau, a_{i,h}^\tau)\phi(x_{i,h}^\tau, a_{i,h}^\tau)^\top + \lambda \cdot I$.
4: Set $\hat{w}_{i,h} \leftarrow \Lambda_{i,h}^{-1}(\sum_{\tau\in\mathcal{B}_{i,h}} \phi(x_{i,h}^\tau, a_{i,h}^\tau) \cdot (r_{i,h}^\tau + \hat{V}_{i,h+1}^\pi(x_{i,h}^{-,\tau})))$.
5: Set $\Gamma_{i,h}(\cdot,\cdot) \leftarrow \beta(\delta) \cdot (\phi(\cdot,\cdot)^\top \Lambda_{i,h}^{-1} \phi(\cdot,\cdot))^{1/2}$.
6: Set $\hat{Q}_{i,h}^\pi(\cdot,\cdot) \leftarrow \min\{\phi(\cdot,\cdot)^\top \hat{w}_{i,h} - \Gamma_{i,h}(\cdot,\cdot), H-h+1\}^+$.
7: Set $\hat{V}_{i,h}^\pi(\cdot) \leftarrow \langle \hat{Q}_{i,h}^\pi(\cdot,\cdot), \pi_h(\cdot|\cdot)\rangle_{\mathcal{A}}$
8: **end for**
9: **return** $\hat{V}_{i,1}^\pi(\cdot), \ldots, \hat{V}_{i,H}^\pi(\cdot), \hat{Q}_{i,1}^\pi(\cdot,\cdot), \ldots, \hat{Q}_{i,H}^\pi(\cdot,\cdot)$.

The following theorem shows the suboptimality gaps for Algo.4 (utilizing subroutine Algo.6) and Algo.5 (also with subroutine Algo.6).

Theorem 4.5.1. *Under Assumption 4.1, in Algorithm 6, we set* $\lambda = 1, \quad \beta(\delta) = c \cdot dH\sqrt{\log(2dHK/\delta)}$, *where* $c > 0$ *is a positive*

constant. Then, we have:
(i) for the output policy π^{PERM} *of Algo.4 with subroutine Algo.6, w.p. at least* $1-\delta$*, the suboptimality gap satisfies*

$$\begin{aligned} SubOpt(\pi^{PERM}) \leq 7\sqrt{\frac{7\log(6\mathcal{N}^{\Pi}_{(Hn)^{-1}}/\delta)}{n}} \\ + \frac{2\beta\big(\frac{\delta}{3nH\mathcal{N}^{\Pi}_{(Hn)^{-1}}}\big)}{n} \cdot \sum_{i=1}^{n}\sum_{h=1}^{H} \mathbb{E}_{i,\pi^*}\Big[\|\phi(s_h,a_h)\|_{\tilde{\Lambda}^{-1}_{i,h}} \,\Big|\, s_1 = x_1\Big], \end{aligned} \tag{4.6}$$

(ii) for the output policy π^{PPPO} *of Algo.5 with subroutine Algo.6, setting* $\delta = 1/8$*, then with probability at least* $2/3$*, the suboptimality gap satisfies*

$$\mathrm{SubOpt}(\pi^{\mathrm{PPPO}}) \leq 10\bigg(\sqrt{\frac{\log|\mathcal{A}|H^2}{n}} + \frac{\beta\big(\frac{1}{4nH}\big)}{n} \cdot \sum_{i=1}^{n}\sum_{h=1}^{H} \mathbb{E}_{i,\pi^*}\Big[\|\phi(s_h,a_h)\|_{\bar{\Lambda}^{-1}_{i,h}} \,\Big|\, s_1 = x_1\Big]\bigg), \tag{4.7}$$

where $\mathbb{E}_{i,\pi^*}$ *is with respect to the trajectory induced by* π^* *with the transition* $\mathcal{P}_i$ *in the underlying MDP* $\mathcal{M}_i$ *given the fixed matrix* $\tilde{\Lambda}_{i,h}$ *or* $\bar{\Lambda}_{i,h}$*.*

$\|\phi(s_h,a_h)\|_{\Lambda^{-1}_{i,h}}$ indicates how well the state-action pair (s_h,a_h) is covered by the dataset $\mathcal{D}_i$. $\sum_{i=1}^{n}\sum_{h=1}^{H}\mathbb{E}_{i,\pi^*}\Big[\|\phi(s_h,a_h)\|_{\Lambda^{-1}_{i,h}} \,\Big|\, s_1 = x_1\Big]$ in the suboptimality gap in Theorem 4.5.1 is small if for each context $i \in [n]$, the dataset $\mathcal{D}_i$ well covers the trajectory induced by the optimal policy π^* on the corresponding MDP $\mathcal{M}_i$.

Well-explored behavior policy Next we consider a case where the dataset $\mathcal{D}$ consists of i.i.d. trajectories collecting from different environments. Suppose $\mathcal{D}$ consists of n independent datasets

$\mathcal{D}_1, \ldots, \mathcal{D}_n$, and for each environment i, $\mathcal{D}_i$ consists of K trajectories $\mathcal{D}_i = \{(x_{i,h}^\tau, a_{i,h}^\tau, r_{i,h}^\tau)_{h=1}^H\}_{\tau=1}^K$ independently and identically induced by a fixed behavior policy $\bar{\pi}_i$ in the linear MDP $\mathcal{M}_i$. We have the following assumption on well-explored policy:

Definition 4.3 ([221, 91])**.** *For an behavior policy $\bar{\pi}$ and an episodic linear MDP $\mathcal{M}$ with feature map ϕ, we say $\bar{\pi}$ well-explores $\mathcal{M}$ with constant c if there exists an absolute positive constant $c > 0$ such that*

$$\forall h \in [H], \lambda_{\min}(\Sigma_h) \geq c/d, \text{where } \Sigma_h = \mathbb{E}_{\bar{\pi},\mathcal{M}}\left[\phi(s_h, a_h)\phi(s_h, a_h)^\top\right].$$

A well-explored policy guarantees that the obtained trajectories is "uniform" enough to represent any policy and value function. The following corollary shows that with the above assumption, the suboptimality gaps of Algo.4 (with subroutine Algo.6) and Algo.5 (with subroutine Algo.6) decay to 0 when n and K are large enough.

Corollary 4.1. *Suppose that for each $i \in [n]$, $\mathcal{D}_i$ is generated by behavior policy $\bar{\pi}_i$ which well-explores MDP $\mathcal{M}_i$ with constant $c_i \geq c_{min}$. In Algo.6, we set $\lambda = 1, \beta(\delta) = c' \cdot dH\sqrt{\log(4dHK/\delta)}$ where $c' > 0$ is a positive constant. Suppose we have $K \geq 40d/c_{min}\log(4dnH/\delta)$ and set $C_n^* := 1/n \cdot \sum_{i=1}^n c_i^{-1/2}$. Then we have:*
(i) for the output π^{PERM} of Algo.4 with subroutine Algo.6, w.p. at least $1-\delta$, the suboptimality gap satisfies

$$SubOpt(\pi^{PERM}) \leq 7\sqrt{\frac{2\log(6\mathcal{N}^{\Pi}_{(Hn)^{-1}}/\delta)}{n}}$$
$$+ 2\sqrt{2}c' \cdot d^{3/2}H^2K^{-1/2}\sqrt{\log(12dHnK\mathcal{N}^{\Pi}_{(Hn)^{-1}}/\delta)} \cdot C^*_n\,, \quad (4.8)$$

(ii) for the output policy π^{PPPO} of Algo.5 with subroutine Algo.6, setting $\delta = 1/8$, then with probability at least $2/3$, the suboptimality gap satisfies

$$\mathrm{SubOpt}(\pi^{\mathrm{PPPO}}) \leq 10\bigg(\sqrt{\frac{\log|\mathcal{A}|H^2}{n}} + 2\sqrt{2}c' \cdot d^{3/2}H^{2.5}K^{-1/2}\sqrt{\log(16dHnK)} \cdot C^*_n\bigg). \quad (4.9)$$

Remark 12. *The mixed coverage parameter $C^*_n = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{\sqrt{c_i}}$ is small if for any $i \in [n]$, c_i is large,* i.e., *the minimum eigenvalue of $\Sigma_{i,h} = \mathbb{E}_{\bar{\pi}_i,\mathcal{M}_i}\left[\phi(s_h,a_h)\phi(s_h,a_h)^\top\right]$ is large. Note that $\lambda_{min}(\Sigma_{i,h})$ indicates how well the behavior policy $\bar{\pi}_i$ explores the state-action pairs on MDP $\mathcal{M}_i$; this shows that if for each environment $i \in [n]$, the behavior policy explores $\mathcal{M}_i$ well, the suboptimality gap will be small.*

Remark 13. *Under the same conditions of Corollary 4.1:*
(i) If $n \geq \frac{392\log(6\mathcal{N}^{\Pi}_{(Hn)^{-1}}/\delta)}{\epsilon^2}$
and $K \geq \max\{\frac{40d}{c_{min}}\log(\frac{4dnH}{\delta}), \frac{32c'^2d^3H^4\log(12dHnK\mathcal{N}^{\Pi}_{(Hn)^{-1}}/\delta)C^{*2}_n}{\epsilon^2}\}$, then w.p. at least $1-\delta$, $SubOpt(\pi^{PERM}) \leq \epsilon$.
(ii) If $n \geq \frac{400H^2\log(|\mathcal{A}|)}{\epsilon^2}$
and $K \geq \max\{\frac{40d}{c_{min}}\log(16dnH), \frac{32c'^2d^3H^5\log(16dHnK)C^{*2}_n}{\epsilon^2}\}$, then w.p. at least $2/3$, $SubOpt(\pi^{PPPO}) \leq \epsilon$.

Corollary 4.1 suggests that both of our proposed algorithms enjoy the $O(n^{-1/2} + K^{-1/2} \cdot C^*_n)$ convergence rate to the optimal policy π^* given a well-exploration data collection assumption, where

C_n^* is a mixed coverage parameter over n environments defined in Corollary 4.1.

Chapter 5

Online Clustering of Bandits with Misspecified User Models

The contextual linear bandit is an important online learning problem where given arm features, a learning agent selects an arm at each round to maximize the cumulative rewards in the long run. A line of works, called the clustering of bandits (CB), utilize the collaborative effect over user preferences and have shown significant improvements over classic linear bandit algorithms. However, existing CB algorithms require well-specified linear user models and can fail when this critical assumption does not hold. Whether robust CB algorithms can be designed for more practical scenarios with misspecified user models remains an open problem. In this paper, we are the first to present the important problem of clustering of bandits with misspecified user models (CBMUM), where the expected rewards in user models can be perturbed away from perfect linear models. We devise two robust CB algorithms, RCLUMB and RSCLUMB (representing the learned clustering structure with dynamic graph and sets, respectively), that can accommodate the inaccurate user preference estimations and er-

roneous clustering caused by model misspecifications. We prove regret upper bounds of $O(\epsilon_* T\sqrt{md\log T} + d\sqrt{mT}\log T)$ for our algorithms under milder assumptions than previous CB works (notably, we move past a restrictive technical assumption on the distribution of the arms), which match the lower bound asymptotically in T up to logarithmic factors, and also match the state-of-the-art results in several degenerate cases. The techniques in proving the regret caused by misclustering users are quite general and may be of independent interest. Experiments on both synthetic and real-world data show our outperformance over previous algorithms. This chapter is based on our publication [4].

5.1 Introduction

Stochastic multi-armed bandit (MAB) [222, 223, 139] is an online sequential decision-making problem, where the learning agent selects an action and receives a corresponding reward at each round, so as to maximize the cumulative reward in the long run. MAB algorithms have been widely applied in recommendation systems and computer networks to handle the exploration and exploitation trade-off [38, 39, 7, 40].

To deal with large-scale applications, the contextual linear bandits [41, 42, 43, 44, 45] have been studied, where the expected reward of each arm is assumed to be perfectly linear in their features. Leveraging the contextual side information about the user and arms, linear bandits can provide more personalized recommendations [224]. Classical linear bandit approaches, however, ignore the often useful tool of collaborative filtering. To

utilize the relationships among users, the problem of clustering of bandits (CB) has been proposed [46]. Specifically, CB algorithms adaptively partition users into clusters and utilize the collaborative effect of users to enhance learning performance.

Although existing CB algorithms have shown great success in improving recommendation qualities, there exist two major limitations. First, all previous works on CB [46, 135, 136, 225] assume that for each user, the expected rewards follow a *perfectly linear* model with respect to the user preference vector and arms' feature vectors. In many real-world scenarios, due to feature noises or uncertainty [47], the reward may not necessarily conform to a perfectly linear function, or even deviates a lot from linearity [48]. Second, previous CB works assume that for users within the same cluster, their preferences are exactly the same. Due to the heterogeneity in users' personalities and interests, similar users may not have identical preferences, invalidating this strong assumption.

To address these issues, we propose a novel problem of clustering of bandits with misspecified user models (CBMUM). In CBMUM, the expected reward model of each user does not follow a perfectly linear function but with possible additive deviations. We assume users in the same underlying cluster share a common preference vector, meaning they have the same linear part in reward models, but the deviation parts are allowed to be different, better reflecting the varieties of user personalities.

The relaxation of perfect linearity and the reward homogeneity within the same cluster bring many challenges to the CBMUM problem. In CBMUM, we not only need to handle the uncer-

tainty from the *unknown* user preference vectors, but also have to tackle the additional uncertainty from model misspecifications. Due to such uncertainties, it becomes highly challenging to design a robust algorithm that can cluster the users appropriately and utilize the clustered information judiciously. On the one hand, the algorithm needs to be more tolerant in the face of misspecifications so that more similar users can be clustered together to utilize the collaborative effect. On the other hand, it has to be more selective to rule out the possibility of *misclustering* users with large preference gaps.

5.1.1 Our Contributions

This paper makes the following four contributions.

New Model Formulation. We are the first to formulate the clustering of bandits with misspecified user models (CBMUM) problem, which is more practical by removing the perfect linearity assumption in previous CB works.

Novel Algorithm Designs. We design two novel algorithms, RCLUMB and RSCLUMB, which robustly learn the clustering structure and utilize this collaborative information for faster user preference elicitation. Specifically, RCLUMB keeps updating a dynamic graph over all users, where users connected directly by edges are supposed to be in the same cluster. RCLUMB adaptively removes edges and recommends items based on historical interactions. RSCLUMB represents the clustering structure with sets, which are dynamicly merged and split during the learning process. Due to the page limit, we only illustrate the RCLUMB

algorithm in the main paper. We leave the exposition, illustration, and regret analysis of the RSCLUMB algorithm in Appendix A.3.11.

To overcome the challenges brought by model misspecifications, we do the following key steps in the RCLUMB algorithm. (i) To ensure that with high probability, similar users will not be partitioned apart, we design a more tolerant edge deletion rule by taking model misspecifications into consideration. (ii) Due to inaccurate user preference estimations caused by model misspecifications, trivially following previous CB works [46, 135, 137] to directly use connected components in the maintained graph as clusters would *miscluster* users with big preference gaps, causing a large regret. To be discriminative in cluster assignments, we filter users directly linked with the current user in the graph to form the cluster used in this round. With these careful designs of (i) and (ii), we can guarantee that with high probability, information of all similar users can be leveraged, and only users with close enough preferences might be *misclustered*, which will only mildly impair the learning accuracy. Additionally: (iii) we design an enlarged confidence radius to incorporate both the exploration bonus and the additional uncertainty from misspecifications when recommending arms. The design of RSCLUMB follows similar ideas, which we leave in the Appendix A.3.11 due to page limit.

Theoretical Analysis with Milder Assumptions. We prove regret upper bounds for our algorithms of $O(\epsilon_* T\sqrt{md\log T} + d\sqrt{mT}\log T)$ in CBMUM under much milder and practical assumptions (in arm generation distribution) than previous CB works, which match the state-of-the-art results in degenerate cases.

Our proof is quite different from the typical proof flow of previous CB works (details in Appendix A.3.3). One key challenge is to bound the regret caused by *misclustering* users with close but not the same preference vectors and use the inaccurate cluster-based information to recommend arms. To handle the challenge, we prove a key lemma (Lemma 5.4.4) to bound this part of regret. We defer its details in Section 5.4 and Appendix A.3.7. The techniques and results for bounding this part are quite general and may be of independent interest. We also give a regret lower bound of $\Omega(\epsilon_* T\sqrt{d})$ for CBMUM, showing that our upper bounds are asymptotically tight with respect to T up to logarithmic factors. We leave proving a tighter lower bound for CBMUM as an open problem.

Good Experimental Performance. Extensive experiments on both synthetic and real-world data show the advantages of our proposed algorithms over the existing algorithms.

5.2 Problem Setup

This section formulates the problem of "clustering of bandits with misspecified user models" (CBMUM). We use boldface **lowercase** and boldface **CAPITALIZED** letters for vectors and matrices. We use $|\mathcal{A}|$ to denote the number of elements in $\mathcal{A}$, $[m]$ to denote $\{1, \ldots, m\}$, and $\|\boldsymbol{x}\|_{\boldsymbol{M}} = \sqrt{\boldsymbol{x}^\top \boldsymbol{M} \boldsymbol{x}}$ to denote the matrix norm of vector $\boldsymbol{x}$ regarding the positive semi-definite (PSD) matrix $\boldsymbol{M}$.

In CBMUM, there are u users denoted by $\mathcal{U} = \{1, 2, \ldots, u\}$. Each user $i \in \mathcal{U}$ is associated with an *unknown* preference vector $\boldsymbol{\theta}_i \in \mathbb{R}^d$, with $\|\boldsymbol{\theta}_i\|_2 \leq 1$. We assume there is an *unknown* underlying clustering structure over users representing the simi-

larity of their behaviors. Specifically, $\mathcal{U}$ can be partitioned into a small number m (i.e., $m \ll u$) clusters, $V_1, V_2, \dots V_m$, where $\cup_{j\in[m]} V_j = \mathcal{U}$, and $V_j \cap V_{j'} = \emptyset$, for $j \neq j'$. We call these clusters *ground-truth clusters* and use $\mathcal{V} = \{V_1, V_2, \dots, V_m\}$ to denote the set of these clusters. Users in the same *ground-truth cluster* share the same preference vector, while users from different *ground-truth clusters* have different preference vectors. Let $\boldsymbol{\theta}^j$ denote the common preference vector for V_j and $j(i) \in [m]$ denote the index of the *ground-truth cluster* that user i belongs to. For any $\ell \in \mathcal{U}$, if $\ell \in V_{j(i)}$, then $\boldsymbol{\theta}_\ell = \boldsymbol{\theta}_i = \boldsymbol{\theta}^{j(i)}$.

At each round $t \in [T]$, a user $i_t \in \mathcal{U}$ comes to be served. The learning agent receives a finite arm set $\mathcal{A}_t \subseteq \mathcal{A}$ to choose from (with $|\mathcal{A}_t| \leq C, \forall t$), where each arm $a \in \mathcal{A}$ is associated with a feature vector $\boldsymbol{x}_a \in \mathbb{R}^d$, and $\|\boldsymbol{x}_a\|_2 \leq 1$. The agent assigns an appropriate cluster $\overline{V}_t$ for user i_t and recommends an item $a_t \in \mathcal{A}_t$ based on the aggregated historical information gathered from cluster $\overline{V}_t$. After receiving the recommended item a_t, user i_t gives a random reward $r_t \in [0, 1]$ to the agent. To better model real-world scenarios, we assume that the reward r_t follows a misspecified linear function of the item feature vector $\boldsymbol{x}_{a_t}$ and the *unknown* user preference vector $\boldsymbol{\theta}_{i_t}$. Formally,

$$r_t = \boldsymbol{x}_{a_t}^\top \boldsymbol{\theta}_{i_t} + \boldsymbol{\epsilon}_{a_t}^{i_t,t} + \eta_t \,, \tag{5.1}$$

where $\boldsymbol{\epsilon}^{i_t,t} = [\boldsymbol{\epsilon}_1^{i_t,t}, \boldsymbol{\epsilon}_2^{i_t,t}, \dots, \boldsymbol{\epsilon}_{|\mathcal{A}_t|}^{i_t,t}]^\top \in \mathbb{R}^{|\mathcal{A}_t|}$ denotes the *unknown* deviation in the expected rewards of arms in $\mathcal{A}_t$ from linearity for user i_t at t, and η_t is the 1-sub-Gaussian noise. We allow the deviation vectors for users in the same *ground-truth cluster* to be different.

We assume the clusters, users, items, and model misspecifications satisfy the following assumptions.

Assumption 5.1 (Gap between different clusters)**.** *The gap between any two preference vectors for different ground-truth clusters is at least an unknown positive constant* γ

$$\left\|\boldsymbol{\theta}^{j}-\boldsymbol{\theta}^{j'}\right\|_2 \geq \gamma > 0\,, \forall j, j' \in [m]\,, j \neq j'\,.$$

Assumption 5.2 (Uniform arrival of users)**.** *At each round* t*, a user* i_t *comes uniformly at random from* $\mathcal{U}$ *with probability* $1/u$*, independent of the past rounds.*

Assumption 5.3 (Item regularity)**.** *At each time step* t*, the feature vector* $\boldsymbol{x}_a$ *of each arm* $a \in \mathcal{A}_t$ *is drawn independently from a fixed but unknown distribution* ρ *over* $\{\boldsymbol{x} \in \mathbb{R}^d : \|\boldsymbol{x}\|_2 \leq 1\}$*, where* $\mathbb{E}_{\boldsymbol{x}\sim\rho}[\boldsymbol{x}\boldsymbol{x}^\top]$ *is full rank with minimal eigenvalue* $\lambda_x > 0$*. Additionally, at any time* t*, for any fixed unit vector* $\boldsymbol{\theta} \in \mathbb{R}^d$*,* $(\boldsymbol{\theta}^\top \boldsymbol{x})^2$ *has sub-Gaussian tail with variance upper bounded by* σ^2*.*

Assumption 5.4 (Bounded misspecification level)**.** *We assume that there is a pre-specified maximum misspecification level parameter* ϵ_* *such that* $\left\|\boldsymbol{\epsilon}^{i,t}\right\|_\infty \leq \epsilon_*$*,* $\forall i \in \mathcal{U}, t \in [T]$*.*

Remark 1. All these assumptions basically follow previous works on CB [46, 226, 135, 227, 137] and MLB [138]. Note that Assumption 9.4 is less stringent and more practical than previous CB works which also put restrictions on the variance upper bound σ^2. For Assumption 9.3, our results can easily generalize to the case where the user arrival follows any distributions with minimum arrival probability greater than p_{min}. For Assumption 9.1, note that ϵ_* can be an upper bound on the maximum misspecification level, not the exact maximum itself. In real-world applications, the deviations are usually small [48], and we can set a relatively

big ϵ_* as an upper bound. For more discussions please refer to Appendix A.3.2

Let $a_t^* \in \arg\max_{a\in\mathcal{A}_t} \boldsymbol{x}_a^\top \boldsymbol{\theta}_{i_t} + \boldsymbol{\epsilon}_a^{i_t,t}$ denote an optimal arm which gives the highest expected reward at t. The goal of the agent is to minimize the expected cumulative regret

$$R(T) = \mathbb{E}[\textstyle\sum_{t=1}^{T}(\boldsymbol{x}_{a_t^*}^\top \boldsymbol{\theta}_{i_t} + \boldsymbol{\epsilon}_{a_t^*}^{i_t,t} - \boldsymbol{x}_{a_t}^\top \boldsymbol{\theta}_{i_t} - \boldsymbol{\epsilon}_{a_t}^{i_t,t})]\,. \tag{5.2}$$

5.3 Algorithm

This section introduces our algorithm called "Robust CLUstering of Misspecified Bandits" (RCLUMB) (Algo.7). RCLUMB is a graph-based algorithm. The ideas and techniques of RCLUMB can be easily generalized to set-based algorithms. To illustrate this generalizability, we also design a set-based algorithm RSCLUMB. We leave the exposition and analysis of RSCLUMB in Appendix A.3.11.

For ease of interpretation, we define the coefficient

$$\zeta \triangleq 2\epsilon_* \sqrt{\frac{2}{\tilde{\lambda}_x}}\,, \tag{5.3}$$

where $\tilde{\lambda}_x \triangleq \int_0^{\lambda_x} (1 - e^{-\frac{(\lambda_x - x)^2}{2\sigma^2}})^C dx$. ζ is theoretically the minimum gap between two users' preference vectors that an algorithm can distinguish with high probability, as supported by Eq.(A.129) in the proof of Lemma A.3.2 in Appendix A.3.8. Note that the algorithm does not require knowledge of ζ. We also make the following definition for illustration.

Definition 5.1 (ζ-close users and ζ-good clusters)**.** *Two users $i, i' \in \mathcal{U}$ are ζ-close if $\|\boldsymbol{\theta}_i - \boldsymbol{\theta}_{i'}\|_2 \leq \zeta$. Cluster $\overline{V}$ is a ζ-good*

Algorithm 7 Robust Clustering of Misspecified Bandits Algorithm (RCLUMB)

1: **Input:** Deletion parameter $\alpha_1, \alpha_2 > 0$, $f(T) = \sqrt{\frac{1+\ln(1+T)}{1+T}}$, $\lambda, \beta, \epsilon_* > 0$.
2: **Initialization:** $\boldsymbol{M}_{i,0} = 0_{d\times d}, \boldsymbol{b}_{i,0} = 0_{d\times 1}, T_{i,0} = 0$, $\forall i \in \mathcal{U}$; a complete Graph $G_0 = (\mathcal{U}, E_0)$ over $\mathcal{U}$.
3: **for all** $t = 1, 2, \ldots, T$ **do**
4: Receive the index of the current user $i_t \in \mathcal{U}$, and the current feasible arm set $\mathcal{A}_t$;
5: Filter user i_t and users $i \in \mathcal{U}$ that are *directly* connected with user i_t via edge $(i, i_t) \in E_{t-1}$, to form the cluster $\overline{V}_t$;
6: Compute the estimated statistics for cluster $\overline{V}_t$

$$\overline{\boldsymbol{M}}_{\overline{V}_t,t-1} = \lambda \boldsymbol{I} + \textstyle\sum_{i\in\overline{V}_t} \boldsymbol{M}_{i,t-1}\,, \overline{\boldsymbol{b}}_{\overline{V}_t,t-1} = \sum_{i\in\overline{V}_t} \boldsymbol{b}_{i,t-1}\,, \hat{\boldsymbol{\theta}}_{\overline{V}_t,t-1} = \overline{\boldsymbol{M}}^{-1}_{\overline{V}_t,t-1}\overline{\boldsymbol{b}}_{\overline{V}_t,t-1};$$

7: Recommend an arm a_t with the largest UCB index (Eq.(6.3)), and receive the reward $r_t \in [0, 1]$;
8: Update the statistics for user i_t $\boldsymbol{M}_{i_t,t} = \boldsymbol{M}_{i_t,t-1} + \boldsymbol{x}_{a_t}\boldsymbol{x}_{a_t}^\top\,, \boldsymbol{b}_{i_t,t} = \boldsymbol{b}_{i_t,t-1} + r_t\boldsymbol{x}_{a_t}\,, T_{i_t,t} = T_{i_t,t-1} + 1\,, \hat{\boldsymbol{\theta}}_{i_t,t} = (\lambda\boldsymbol{I} + \boldsymbol{M}_{i_t,t})^{-1}\boldsymbol{b}_{i_t,t}$;
9: Keep the statistics of other users unchanged $\boldsymbol{M}_{\ell,t} = \boldsymbol{M}_{\ell,t-1}, \boldsymbol{b}_{\ell,t} = \boldsymbol{b}_{\ell,t-1}, T_{\ell,t} = T_{\ell,t-1}, \hat{\boldsymbol{\theta}}_{\ell,t} = \hat{\boldsymbol{\theta}}_{\ell,t-1}$, for all $\ell \in \mathcal{U}, \ell \neq i_t$;
10: Delete the edge $(i_t, \ell) \in E_{t-1}$, if

$$\left\|\hat{\boldsymbol{\theta}}_{i_t,t} - \hat{\boldsymbol{\theta}}_{\ell,t}\right\|_2 \geq \alpha_1\left(f(T_{i_t,t}) + f(T_{\ell,t})\right) + \alpha_2\epsilon_*\,,$$

and get an updated graph $G_t = (\mathcal{U}, E_t)$;
11: **end for**

cluster at time t, if $\forall\, i \in \overline{V}$, user i and the coming user i_t are ζ-close.

We also say that two *ground-truth clusters* are "ζ-close" if their preference vectors' gap is less than ζ.

Now we introduce the process and intuitions of RCLUMB (Algo.7). The algorithm maintains an undirected user graph $G_t = (\mathcal{U}, E_t)$, where users are connected with edges if they are

inferred to be in the same cluster. We denote the connected component in G_{t-1} containing user i_t at round t as $\tilde{V}_t$.

Cluster Detection. G_0 is initialized to be a complete graph, and will be updated adaptively based on the interactive information. At round t, user $i_t \in \mathcal{U}$ comes to be served with a feasible arm set $\mathcal{A}_t$ (Line 4).

Due to model misspecifications, it is impossible to cluster users with exactly the same preference vector $\boldsymbol{\theta}$, but similar users whose preference vectors are within the distance of ζ. According to the proof of Lemma A.3.2, after a sufficient time, with high probability, any pair of users directly connected by an edge in E_{t-1} are ζ-close. However, if we trivially follow previous CB works [46, 135, 137] to directly use the connected component $\tilde{V}_t$ as the inferred cluster for user i_t at round t, it will cause a large regret. The reason is that in the worst case, the preference vector $\boldsymbol{\theta}$ of the user in $\tilde{V}_t$ who is h-hop away from user i_t could deviate by $h\zeta$ from $\boldsymbol{\theta}_{i_t}$, where h can be as large as $|\tilde{V}_t|$. Based on this reasoning, our key point is to select the cluster $\overline{V}_t$ as the users at most 1-hop away from i_t in the graph. In other words, after some interactions, $\overline{V}_t$ forms a ζ-good cluster with high probability; thus, RCLUMB can avoid using misleading information from dissimilar users for recommendations.

Cluster-based Recommendation. After finding the appropriate cluster $\overline{V}_t$ for i_t, the agent estimates the common user preference vector based on the historical information associated with cluster $\overline{V}_t$ by

$$\hat{\boldsymbol{\theta}}_{\overline{V}_t,t-1} = \underset{\boldsymbol{\theta}\in\mathbb{R}^d}{\arg\min} \sum_{\substack{s\in[t-1]\\ i_s\in\overline{V}_t}} (r_s - \boldsymbol{x}_{a_s}^\top \boldsymbol{\theta})^2 + \lambda \left\|\boldsymbol{\theta}\right\|_2^2 , \tag{5.4}$$

where $\lambda > 0$ is a regularization coefficient. Its closed-form solution is $\hat{\boldsymbol{\theta}}_{\overline{V}_t,t-1} = \overline{\boldsymbol{M}}_{\overline{V}_t,t-1}^{-1} \overline{\boldsymbol{b}}_{\overline{V}_t,t-1}$,
where $\overline{\boldsymbol{M}}_{\overline{V}_t,t-1} = \lambda \boldsymbol{I} + \sum_{\substack{s\in[t-1]\\ i_s\in\overline{V}_t}} \boldsymbol{x}_{a_s} \boldsymbol{x}_{a_s}^\top$, $\overline{\boldsymbol{b}}_{\overline{V}_t,t-1} = \sum_{\substack{s\in[t-1]\\ i_s\in\overline{V}_t}} r_{a_s} \boldsymbol{x}_{a_s}$.

Based on this estimation, in Line 7, the agent recommends an arm using the UCB strategy

$$a_t = \underset{a\in\mathcal{A}_t}{\operatorname{argmax}} \min\{1, \underbrace{\boldsymbol{x}_a^\top \hat{\boldsymbol{\theta}}_{\overline{V}_t,t-1}}_{\hat{R}_{a,t}} + \underbrace{\beta \left\|\boldsymbol{x}_a\right\|_{\overline{\boldsymbol{M}}_{\overline{V}_t,t-1}^{-1}} + \epsilon_* \sum_{\substack{s\in[t-1]\\ i_s\in\overline{V}_t}} \left| \boldsymbol{x}_a^\top \overline{\boldsymbol{M}}_{\overline{V}_t,t-1}^{-1} \boldsymbol{x}_{a_s} \right|}_{C_{a,t}}\} , \tag{5.5}$$

where $\beta = \sqrt{\lambda} + \sqrt{2\log(\frac{1}{\delta}) + d\log(1+\frac{T}{\lambda d})}$, $\hat{R}_{a,t}$ denotes the estimated reward of arm a at t, $C_{a,t}$ denotes the confidence radius of arm a at round t.

Due to deviations from linearity, the estimation $\hat{R}_{a,t}$ computed by a linear function is no longer accurate. To handle the estimation uncertainty of model misspecifications, we design an enlarged confidence radius $C_{a,t}$. The first term of $C_{a,t}$ in Eq.(6.3) captures the uncertainty of online learning for the linear part, and the second term related to ϵ_* reflects the additional uncertainty from deviations from linearity. The design of $C_{a,t}$ theoretically relies on Lemma 6.4.2 which will be given in Section 5.4.

Update User Statistics. Based the feedback r_t, in Line 8 and 9, the agent updates the statistics for user i_t. Specifically, the agent estimates the preference vector $\boldsymbol{\theta}_{i_t}$ by

$$\hat{\boldsymbol{\theta}}_{i_t,t} = \underset{\boldsymbol{\theta}\in\mathbb{R}^d}{\arg\min} \sum_{\substack{s\in[t]\\ i_s=i_t}} (r_s - \boldsymbol{x}_{a_s}^\top \boldsymbol{\theta})^2 + \lambda \left\|\boldsymbol{\theta}\right\|_2^2 , \tag{5.6}$$

with solution $\hat{\boldsymbol{\theta}}_{i_t,t} = (\lambda \boldsymbol{I} + \boldsymbol{M}_{i_t,t})^{-1} \boldsymbol{b}_{i_t,t}$,
where $\boldsymbol{M}_{i_t,t} = \sum_{\substack{s \in [t] \\ i_s = i_t}} \boldsymbol{x}_{a_s} \boldsymbol{x}_{a_s}^\top$, $\boldsymbol{b}_{i_t,t} = \sum_{\substack{s \in [t] \\ i_s = i_t}} r_{a_s} \boldsymbol{x}_{a_s}$.

Update the Graph G_t. Finally, in Line 10, the agent verifies whether the similarities between user i_t and other users are still true based on the updated estimation $\hat{\boldsymbol{\theta}}_{i_t,t}$. For every user $\ell \in \mathcal{U}$ connected with user i_t via edge $(i_t, \ell) \in E_{t-1}$, if the gap between her estimated preference vector $\hat{\boldsymbol{\theta}}_{\ell,t}$ and $\hat{\boldsymbol{\theta}}_{i_t,t}$ is larger than a threshold supported by Lemma A.3.2, the agent will delete the edge (i_t, ℓ) to split them apart. The threshold in Line 10 is carefully designed, taking both estimation uncertainty in a linear model and deviations from linearity into consideration. As shown in the proof of Lemma A.3.2 (in Appendix A.3.8), using this threshold, with high probability, edges between users in the same *ground-truth clusters* will not be deleted, and edges between users that are not ζ-close will always be deleted. Together with the filtering step in Line 5, with high probability, the algorithm will leverage all the collaborative information of similar users and avoid misusing the information of dissimilar users. The updated graph G_t will be used in the next round.

5.4 Theoretical Analysis

In this section, we theoretically analyze the performance of the RCLUMB algorithm by giving an upper bound of the expected regret defined in Eq.(6.1). Due to the space limitation, we only show the main result (Theorem 6.4.3), key lemmas, and a sketched proof for Theorem 6.4.3. Detailed proofs, other technical lemmas, and the regret analysis of the RSLUMB algorithm can be

found in the Appendix.

To state our main result, we first give two definitions as follows. The first definition is about the minimum separable gap constant γ_1 of a CBMUM problem instance.

Definition 5.2 (Minimum separable gap γ_1)**.** *The minimum separable gap constant γ_1 of a CBMUM problem instance is the minimum gap over the gaps among users that are greater than ζ (Eq. (5.3))*

$$\gamma_1 = \min\{\|\boldsymbol{\theta}_i - \boldsymbol{\theta}_\ell\|_2 : \|\boldsymbol{\theta}_i - \boldsymbol{\theta}_\ell\|_2 > \zeta, \forall i, \ell \in \mathcal{U}\}, \textit{with} \min \emptyset = \infty.$$

Remark 2. In CBMUM, the role of $\gamma_1 - \zeta$ is similar to that of γ (given in Assumption 9.2) in the previous CB problem with perfectly linear models, quantifying the hardness of performing clustering on the problem instance. Intuitively, users are easier to cluster if γ_1 is larger, and the deduction of ζ shows the additional difficulty due to model diviations. If there are no misspecifications, i.e., $\zeta = 2\epsilon_* \sqrt{\frac{2}{\lambda_x}} = 0$, then $\gamma_1 = \gamma$, recovering the minimum separable gap between clusters in the classic CB problem [46, 135] without model misspecifications.

The second definition is about the number of "hard-to-cluster users" $\tilde{u}$.

Definition 5.3 (Number of "hard-to-cluster users" $\tilde{u}$)**.** *The number of "hard-to-cluster users" $\tilde{u}$ is the number of users in the ground-truth clusters which are ζ-close to some other ground-truth clusters*

$$\tilde{u} = \sum_{j \in [m]} |V_j| \times \mathbb{I}\{\exists j' \in [m], j' \neq j : \left\|\boldsymbol{\theta}^{j'} - \boldsymbol{\theta}^{j}\right\|_2 \leq \zeta\},$$

where $\mathbb{I}\{\cdot\}$ *denotes the indicator function of the argument,* $|V_j|$ *denotes the number of users in* V_j*.*

Remark 3. $\tilde{u}$ captures the number of users who belong to different *ground-truth clusters* but their gaps are less than ζ. These users may be merged into one cluster by mistake and cause certain regret.

The following theorem gives an upper bound on the expected regret achieved by RCLUMB.

Theorem 5.4.1 (Main result on regret bound)**.** *Suppose that the assumptions in Section 5.2 are satisfied. Then the expected regret of the RCLUMB algorithm for* T *rounds satisfies*

$$R(T) \leq O\left(u\left(\frac{d}{\tilde{\lambda}_x(\gamma_1-\zeta)^2}+\frac{1}{\tilde{\lambda}_x^2}\right)\log T+\frac{\tilde{u}}{u}\frac{\epsilon_*\sqrt{d}T}{\tilde{\lambda}_x^{1.5}}+\epsilon_* T\sqrt{md\log T}+d\sqrt{mT}\log T\right) \tag{5.7}$$

$$\leq O(\epsilon_* T\sqrt{md\log T}+d\sqrt{mT}\log T)\,, \tag{5.8}$$

where γ_1 *is defined in Definition 5.2, and* $\tilde{u}$ *is defined in Definition 5.3).*

Discussion and Comparison. The bound in Eq.(5.7) has four terms. The first term is the time needed to gather enough information to assign appropriate clusters for users. The second term is the regret caused by *misclustering* ζ-close but not precisely similar users together, which is unavoidable with model misspecifications. The third term is from the preference estimation errors caused by model deviations. The last term is the usual term in CB with perfectly linear models [46, 135, 136].

Let us discuss how the parameters affect this regret bound.

• If $\gamma_1-\zeta$ is large, the gaps between clusters that are not "ζ-close"

are much greater than the minimum gap ζ for the algorithm to distinguish, the first term in Eq.(5.7) will be small as it is easy to identify their dissimilarities. The role of $\gamma_1 - \zeta$ in CBMUM is similar to that of γ in the previous CB.

• If $\tilde{u}$ is small, indicating that few *ground-truth clusters* are "ζ-close", RCLUMB will hardly *miscluster* different *ground-truth clusters* together thus the second term in Eq.(5.7) will be small.

• If the deviation level ϵ_* is small, the user models are close to linearity and the misspecifications will not affect the estimations much, then both the second and third term in Eq.(5.7) will be small.

The following theorem gives a regret lower bound of the CBMUM problem.

Theorem 5.4.2 (Regret lower bound for CBMUM)**.** *There exists a problem instance for the CBMUM problem such that for any algorithm* $R(T) \geq \Omega(\epsilon_* T\sqrt{d})$ *.*

The proof can be found in Appendix A.3.6. The upper bounds in Theorem 6.4.3 asymptotically match this lower bound with respect to T up to logarithmic factors (and a constant factor of $\sqrt{m}$ where m is typically small in real-applications), showing the tightness of our theoretical results. Additionally, we conjecture the gap for the m factor is due to the strong assumption that cluster structures are known to prove this lower bound, and whether there exists a tighter lower bound is left for future work.

We then compare our results with two degenerate cases. First, when $m = 1$ (indicating $\tilde{u} = 0$), our setting degenerates to the MLB problem where all users share the same preference vector.

In this case, our regret bound is $O(\epsilon_* T\sqrt{d\log T} + d\sqrt{T}\log T)$, exactly matching the current best bound of MLB [138]. Second, when $\epsilon_* = 0$, our setting reduces to the CB problem with perfectly linear user models and our bounds become $O(d\sqrt{mT}\log T)$, also perfectly match the existing best bound of the CB problem [135, 136]. The above discussions and comparisons show the tightness of our regret bounds. Additionally, we also provide detailed discussions on why trivially combining existing works on CB and MLB would not get any non-vacuous regret upper bound in Appendix A.3.4.

We define the following "good partition" for ease of interpretation.

Definition 5.4 (Good partition)**.** *RCLUMB does a "good partition" at t, if the cluster $\overline{V}_t$ assigned to i_t is a ζ-good cluster, and it contains all the users in the same ground-truth cluster as i_t, i.e.,*

$$\|\boldsymbol{\theta}_{i_t} - \boldsymbol{\theta}_\ell\|_2 \leq \zeta, \forall \ell \in \overline{V}_t\,, \text{and} V_{j(i_t)} \subseteq \overline{V}_t\,. \tag{5.9}$$

Note that when the algorithm does a "good partition" at t, $\overline{V}_t$ will contain all the users in the same *ground-truth cluster* as i_t and may only contain some other ζ-close users with respect to i_t, which means the gathered information associated with $\overline{V}_t$ can be used to infer user i_t's preference with high accuracy. Also, it is obvious that under a "good partition", if $\overline{V}_t \in \mathcal{V}$, then $\overline{V}_t = V_{j(i_t)}$ by definition.

Next, we give a sketched proof for Theorem 6.4.3.

Proof. **[Sketch for Theorem 6.4.3]** The proof mainly contains two parts. First, we prove there is a sufficient time T_0 for

RCLUMB to get a "good partition" with high probability. Second, we prove the regret upper bound for RCLUMB after maintaining a "good partition". The most challenging part is to bound the regret caused by *misclustering* ζ-close users after getting a "good partition".

1. Sufficient time to maintain a "good partition". With the item regularity (Assumption 9.4), we can prove after some T_0 (defined in Lemma A.3.2 in Appendix A.3.8), RCLUMB will always have a "good partition". After $t \geq O\big(u\left(\frac{d}{\tilde{\lambda}_x(\gamma_1-\zeta)^2}+\frac{1}{\tilde{\lambda}_x^2}\right)\log T\big)$, for any user $i \in \mathcal{U}$, the gap between the estimated $\hat{\boldsymbol{\theta}}_{i,t}$ and the ground-truth $\boldsymbol{\theta}^{j(i)}$ is less than $\frac{\gamma_1}{4}$ with high probability. With this, we can get: for any two users i and ℓ, if their gap is greater than ζ, it will trigger the deletion of the edge (i,ℓ) (Line 10 of Algo.7) with high probability; on the other hand, when the deletion condition of the edge (i,ℓ) is satisfied, then $\|\boldsymbol{\theta}^{j(i)}-\boldsymbol{\theta}^{j(\ell)}\|_2 > 0$, which means user i and ℓ belong to different *ground-truth clusters* by Assumption 9.2 with high probability. Therefore, we can get that with high probability, all those users in the same *ground-truth cluster* as i_t will be directly connected with i_t, and users directly connected with i_t must be ζ-close to i_t. By filtering users directly linked with i_t as the cluster $\overline{V}_t$ (Algo.7 Line 5) and the definition of "good partition", we can ensure that RCLUMB will keep a "good partition" afterward with high probability.

2. Bounding the regret after getting a "good partition". After T_0, with the "good partition", we can prove the following lemma that gives a bound of the difference between $\hat{\boldsymbol{\theta}}_{\overline{V}_t,t-1}$ and ground-truth $\boldsymbol{\theta}_{i_t}$ in direction of action vector $\boldsymbol{x}_a$, and supports

the design of the confidence radius $C_{a,t}$ in Eq.(6.3).

Lemma 5.4.3. *With probability at least* $1-5\delta$ *for some* $\delta \in (0, \frac{1}{5})$, $\forall t \geq T_0$

$$\left|\boldsymbol{x}_a^\top(\boldsymbol{\theta}_{i_t} - \hat{\boldsymbol{\theta}}_{\overline{V}_t,t-1})\right| \leq \frac{\epsilon_*\sqrt{2d}}{\tilde{\lambda}_x^{\frac{3}{2}}}\mathbb{I}\{\overline{V}_t \notin \mathcal{V}\} + \epsilon_* \sum_{\substack{s\in[t-1]\\ i_s\in\overline{V}_t}} \left|\boldsymbol{x}_a^\top \overline{\boldsymbol{M}}^{-1}_{\overline{V}_t,t-1}\boldsymbol{x}_{a_s}\right| + \beta \left\|\boldsymbol{x}_a\right\|_{\overline{\boldsymbol{M}}^{-1}_{\overline{V}_t,t-1}} .$$

To prove this lemma, we consider the following two situations.

(i) Assigning a perfect cluster for i_t. In this case, $\overline{V}_t \in \mathcal{V}$, meaning the cluster assigned for user i_t is the same as her *ground-truth cluster*, i.e., $\overline{V}_t = V_{j(i_t)}$. Therefore, we have that $\forall \ell \in \overline{V}_t, \boldsymbol{\theta}_\ell = \boldsymbol{\theta}_{i_t}$. With careful analysis, we can bound $\left|\boldsymbol{x}_a^\top(\boldsymbol{\theta}_{i_t} - \hat{\boldsymbol{\theta}}_{\overline{V}_t,t-1})\right|$ by $C_{a,t}$ (defined in Eq.(6.3)).

(ii) Bounding the term of *misclustering* i_t's ζ-close users. In this case, $\overline{V}_t \notin \mathcal{V}$, meaning the algorithm *misclusters* user i_t, i.e., $\overline{V}_t \neq V_{j(i_t)}$. Thus, we do not have $\forall \ell \in \overline{V}_t, \boldsymbol{\theta}_\ell = \boldsymbol{\theta}_{i_t}$ anymore, but we have all the users in $\overline{V}_t$ are ζ-close to i_t (by "good partition"), i.e., $\left\|\boldsymbol{\theta}_{i_s} - \boldsymbol{\theta}_{i_t}\right\|_2 \leq \zeta\,, \forall \ell \in \overline{V}_t$. Then an additional term can be caused by using the information of i_t's ζ-close users in $\overline{V}_t$ lying in different *ground-truth clusters* from i_t to estimate $\boldsymbol{\theta}_{i_t}$. It is highly challenging to bound this part.

We will get an extra term $\left|\boldsymbol{x}_a^\top \overline{\boldsymbol{M}}^{-1}_{\overline{V}_t,t-1} \sum_{\substack{s\in[t-1]\\ i_s\in\overline{V}_t}} \boldsymbol{x}_{a_s}\boldsymbol{x}_{a_s}^\top(\boldsymbol{\theta}_{i_s} - \boldsymbol{\theta}_{i_t})\right|$ when bounding the regret in this case, where $\left\|\boldsymbol{\theta}_\ell - \boldsymbol{\theta}_{i_t}\right\|_2 \leq \zeta\,, \forall \ell \in \overline{V}_t$. It is an easy-to-be-made mistake to directly drag $\left\|\boldsymbol{\theta}_{i_s} - \boldsymbol{\theta}_{i_t}\right\|_2$ out to bound it by $\left\|\boldsymbol{x}_a^\top \overline{\boldsymbol{M}}^{-1}_{\overline{V}_t,t-1} \sum_{\substack{s\in[t-1]\\ i_s\in\overline{V}_t}} \boldsymbol{x}_{a_s}\boldsymbol{x}_{a_s}^\top\right\|_2 \times \zeta$. With subtle analysis, we propose the following lemma to bound the above term.

Lemma 5.4.4 (Bound of error caused by *misclustering*)**.** $\forall t \geq T_0$, *if the current partition by RCLUMB is a "good partition", and* $\overline{V}_t \notin \mathcal{V}$, *then for all* $\boldsymbol{x}_a \in \mathbb{R}^d, \left\|\boldsymbol{x}_a\right\|_2 \leq 1$, *with probability at least*

$1-\delta$:

$$\left|\boldsymbol{x}_a^\top \overline{\boldsymbol{M}}_{\overline{V}_t,t-1}^{-1} \sum_{\substack{s\in[t-1]\\ i_s\in\overline{V}_t}} \boldsymbol{x}_{a_s}\boldsymbol{x}_{a_s}^\top(\boldsymbol{\theta}_{i_s}-\boldsymbol{\theta}_{i_t})\right| \leq \frac{\epsilon_*\sqrt{2d}}{\tilde{\lambda}_x^{\frac{3}{2}}}\,.$$

This lemma is quite general. Please see Appendix A.3.7 for details about its proof.

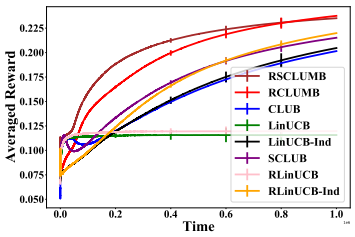
The expected occurrences of $\{\overline{V}_t \notin \mathcal{V}\}$ is bounded by $\frac{\tilde{u}}{u}T$ with Assumption 9.3, Definition 5.3 and 5.4. The result follows by bounding the expected sum of the bounds for the instantaneous regret using Lemma 6.4.2 with delicate analysis due to the time-varying clustering structure kept by RCLUMB. □

5.5 Experiments

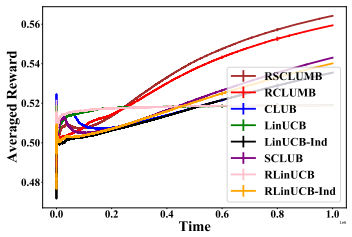
This section compares RCLUMB and RSCLUMB with CLUB [46], SCLUB [136], LinUCB with a single estimated vector for all users, LinUCB-Ind with separate estimated vectors for each user, and two modifications of LinUCB in [138] which we name as RLinUCB and RLinUCB-Ind. We use averaged reward as the evaluation metric, where the average is taken over ten independent trials.

5.5.1 Synthetic Experiments

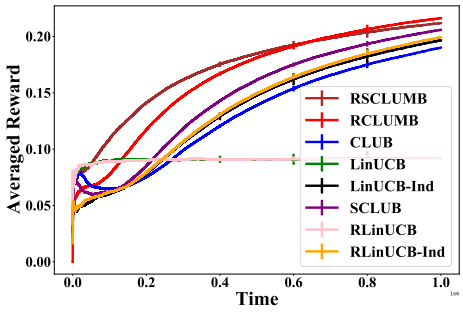
We consider a setting with $u = 1,000$ users, $m = 10$ clusters and $T = 10^6$ rounds. The preference and feature vectors are in $d = 50$ dimension with each entry drawn from a standard Gaussian distribution, and are normalized to vectors with $\|.\|_2 = 1$ [136]. We fix an arm set with $|\mathcal{A}| = 1000$ items, at each round



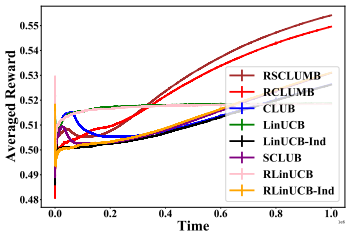
(a) Synthetic



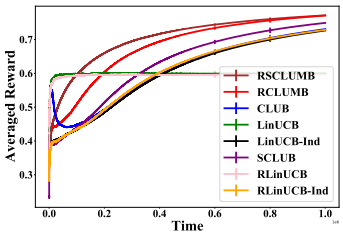
(b) Yelp Case 1



(c) Yelp Case 2



(d) Movielens Case 1



(e) Movielens Case 2

Figure 5.1: The figures compare RCLUMB and RSCLUMB with the baselines. (a) shows the result on synthetic data, (b) and (c) show the results on Yelp dataset, (d) and (e) show the results on Movielens dataset. All experiments are under the setting of $u = 1,000$ users, $m = 10$ clusters, and $d = 50$. All results are averaged under 10 random trials. The error bars are standard deviations divided by $\sqrt{10}$.

t, 20 items are randomly selected to form a set $\mathcal{A}_t$ for the user to choose from. We construct a matrix $\boldsymbol{\epsilon} \in \mathbb{R}^{1,000\times 1,000}$ in which each element $\boldsymbol{\epsilon}(i,j)$ is drawn uniformly from the range $(-0.2, 0.2)$ to represent the deviation. At t, for user i_t and the item a_t, $\boldsymbol{\epsilon}(i_t, a_t)$ will be added to the feedback as the deviation, which corresponds to the $\boldsymbol{\epsilon}_{a_t}^{i_t,t}$ defined in Eq.(5.1).

The result is provided in Figure 5.1(a), showing that our algorithms have clear advantages: RCLUMB improves over CLUB by 21.9%, LinUCB by 194.8%, LinUCB-Ind by 20.1%, SCLUB by 12.0%, RLinUCB by 185.2% and RLinUCB-Ind by 10.6%. The performance difference between RCLUMB and RSCLUMB is very small as expected. RLinUCB performs better than LinUCB; RLinUCB-Ind performs better than LinUCB-Ind and CLUB, showing that the modification of the recommendation policy is effective. The set-based RSCLUMB and SCLUB can separate clusters quicker and have advantages in the early period, but eventually RCLUMB catches up with RSCLUMB, and SCLUB is surpassed by RLinUCB-Ind because it does not consider misspecifications. RCLUMB and RSCLUMB perform better than RLinUCB-Ind, which shows the advantage of the clustering. So it can be concluded that both the modification for misspecification and the clustering structure are critical to improving the algorithm's performance. We also have done some ablation experiments on different scales of ϵ^* in Appendix A.3.16 , and we can notice that under different ϵ^* , our algorithms always outperform the baselines, and some baselines will perform worse as ϵ^* increases.

5.5.2 Experiments on Real-world Datasets

We conduct experiments on the Yelp data and the $20m$ MovieLens data [228]. For both data, we have two cases due to the different methods for generating feedback. For case 1, we extract 1,000 items with most ratings and 1,000 users who rate most; then we construct a binary matrix $\boldsymbol{H}^{1,000\times1,000}$ based on the user rating [180, 229]: if the user rating is greater than 3, the feedback is 1; otherwise, the feedback is 0. Then we use this binary matrix to generate the preference and feature vectors by singular-value decomposition (SVD) [136, 135, 180]. Similar to the synthetic experiment, we construct a matrix $\boldsymbol{\epsilon} \in \mathbb{R}^{1,000\times1,000}$ in which each element is drawn uniformly from the range $(-0.2, 0.2)$. For case 2, we extract 1,100 users who rate most and 1000 items with most ratings. We construct a binary feedback matrix $\boldsymbol{H}^{1,100\times1,000}$ based on the same rule as case 1. Then we select the first 100 rows $\boldsymbol{H}_1^{100\times1,000}$ to generate the feature vectors by SVD. The remaining 1,000 rows $\boldsymbol{F}^{1,000\times1,000}$ is used as the feedback matrix, meaning user i receives $\boldsymbol{F}(i, j)$ as feedback while choosing item j. In both cases, at time t, we randomly select 20 items for the algorithms to choose from. In case 1, the feedback is computed by the preference and feature vector with misspecification, in case 2, the feedback is from the feedback matrix.

The results on Yelp are shown in Fig 5.1(b) and Fig 5.1(c). In case 1, RCLUMB improves CLUB by 45.1%, SCLUB by 53.4%, LinUCB-One by 170.1% , LinUCB-Ind by 46.2%, RLinUCB by 171.0% and RLinUCB-Ind by 21.5%. In case 2, RCLUMB improves over CLUB by 13.9%, SCLUB by 5.1%, LinUCB-One by

135.6% , LinUCB-Ind by 10.1%, RLinUCB by 138.6% and RLinUCB by 8.5%. It is notable that our modeling assumption 9.1 is violated in case 2 since the misspecification range is unknown. We set $\epsilon_* = 0.2$ following our synthetic dataset and it can still perform better than other algorithms. When the misspecification level is known as in case 1, our algorithms' improvement is significantly enlarged, e.g., RCLUMB improves over SCLUB from 5.1% to 53.4%.

The results on Movielens are shown in Fig 5.1(d) and 5.1(e). In case 1, RCLUMB improves CLUB by 58.8%, SCLUB by 92.1%, LinUCB-One by 107.7%, LinUCB-Ind by 61.5 %, RLinUCB by 109.5%, and RLinUCB-Ind by 21.3%. In case 2, RCLUMB improves over CLUB by 5.5%, SCLUB by 2.9%, LinUCB-One by 28.5%, LinUCB-Ind by 6.1%, RLinUCB by 29.3% and RLinUCB-Ind by 5.8%. The results are consistent with the Yelp data, confirming our superior performance.

Chapter 6

Online Corrupted User Detection and Regret Minimization

In real-world online web systems, multiple users usually arrive sequentially into the system. For applications like click fraud and fake reviews, some users can maliciously perform corrupted (disrupted) behaviors to trick the system. Therefore, it is crucial to design efficient online learning algorithms to robustly learn from potentially corrupted user behaviors and accurately identify the corrupted users in an online manner. Existing works propose bandit algorithms robust to adversarial corruption. However, these algorithms are designed for a single user, and cannot leverage the implicit social relations among multiple users for more efficient learning. Moreover, none of them consider how to detect corrupted users online in the multiple-user scenario. In this paper, we present an important online learning problem named LOCUD to learn and utilize unknown user relations from disrupted behaviors to speed up learning, and identify the corrupted users

in an online setting. To robustly learn and utilize the unknown relations among potentially corrupted users, we propose a novel bandit algorithm RCLUB-WCU. To detect the corrupted users, we devise a novel online detection algorithm OCCUD based on RCLUB-WCU's inferred user relations. We prove a regret upper bound for RCLUB-WCU, which asymptotically matches the lower bound with respect to T up to logarithmic factors, and matches the state-of-the-art results in degenerate cases. We also give a theoretical guarantee for the detection accuracy of OCCUD. With extensive experiments, our methods achieve superior performance over previous bandit algorithms and high corrupted user detection accuracy. This chapter is based on our publication [5].

6.1 Introduction

In real-world online recommender systems, data from many users arrive in a streaming fashion [42, 38, 230, 231, 7, 44, 232]. There may exist some corrupted (malicious) users, whose behaviors (*e.g.*, click, rating) can be adversarially corrupted (disrupted) over time to fool the system [49, 50, 51, 52, 53]. These corrupted behaviors could disrupt the user preference estimations of the algorithm. As a result, the system would easily be misled and make sub-optimal recommendations [233, 234, 231, 15], which would hurt the user experience. Therefore, it is essential to design efficient online learning algorithms to robustly learn from potentially disrupted behaviors and detect corrupted users in an online manner.

There exist some works on bandits with adversarial corruption

[49, 53, 54, 142, 51, 45]. However, they have the following limitations. First, existing algorithms are initially designed for robust online preference learning of a single user. In real-world scenarios with multiple users, they cannot robustly infer and utilize the implicit user relations for more efficient learning. Second, none of them consider how to identify corrupted users online in the multiple-user scenario. Though there also exist some works on corrupted user detection [55, 56, 57, 235, 236], they all focus on detection with *known* user information in an offline setting, thus can not be applied to do online detection from bandit feedback.

To address these limitations, we propose a novel bandit problem "*Learning and Online Corrupted Users Detection from bandit feedback*" (LOCUD). To model and utilize the relations among users, we assume there is an *unknown* clustering structure over users, where users with similar preferences lie in the same cluster [46, 135, 237]. The agent can infer the clustering structure to leverage the information of similar users for better recommendations. Among these users, there exists a small fraction of corrupted users. They can occasionally perform corrupted behaviors to fool the agent [51, 49, 50, 53] while mimicking the behaviors of normal users most of the time to make themselves hard to discover. The agent not only needs to learn the *unknown* user preferences and relations robustly from potentially disrupted feedback, balance the exploration-exploitation trade-off to maximize the cumulative reward, but also needs to detect the corrupted users online from bandit feedback.

The LOCUD problem is very challenging. First, the corrupted behaviors would cause inaccurate user preference estimations,

which could lead to erroneous user relation inference and sub-optimal recommendations. Second, it is nontrivial to detect corrupted users online since their behaviors are dynamic over time (sometimes regular while sometimes corrupted), whereas, in the offline setting, corrupted users' information can be fully represented by static embeddings and the existing approaches [238, 239] can typically do binary classifications offline, which are not adaptive over time.

We propose a novel learning framework composed of two algorithms to address these challenges.

RCLUB-WCU. To robustly estimate user preferences, learn the unknown relations from potentially corrupted behaviors, and perform high-quality recommendations, we propose a novel bandit algorithm "*Robust CLUstering of Bandits With Corrupted Users*" (RCLUB-WCU), which maintains a dynamic graph over users to represent the learned clustering structure, where users linked by edges are inferred to be in the same cluster. RCLUB-WCU adaptively deletes edges and recommends arms based on aggregated interactive information in clusters. We do the following to ensure robust clustering structure learning. (i) To relieve the estimation inaccuracy caused by disrupted behaviors, we use weighted ridge regressions for robust user preference estimations. Specifically, we use the inverse of the confidence radius to weigh each sample. If the confidence radius associated with user i_t and arm a_t is large at t, the learner is quite uncertain about the estimation of i_t's preference on a_t, indicating the sample at t is likely to be corrupted. Therefore, we use the inverse of the confidence radius to assign minor importance to the possibly disrupted samples

when doing estimations. (ii) We design a robust edge deletion rule to divide the clusters by considering the potential effect of corruptions, which, together with (i), can ensure that after some interactions, users in the same connected component of the graph are in the same underlying cluster with high probability.

OCCUD. To detect corrupted users online, based on the learned clustering structure of RCLUB-WCU, we devise a novel algorithm named "*Online Cluster-based Corrupted User Detection*" (OCCUD). At each round, we compare each user's non-robustly estimated preference vector (by ridge regression) and the robust estimation (by weighted regression) of the user's inferred cluster. If the gap exceeds a carefully-designed threshold, we detect this user as corrupted. The intuitions are as follows. With misleading behaviors, the non-robust preference estimations of corrupted users would be far from ground truths. On the other hand, with the accurate clustering of RCLUB-WCU, the robust estimations of users' inferred clusters should be close to ground truths. Therefore, for corrupted users, their non-robust estimates should be far from the robust estimates of their inferred clusters.

We summarize our contributions as follows.

• We present a novel online learning problem LOCUD, where the agent needs to (i) robustly learn and leverage the unknown user relations to improve online recommendation qualities under the disruption of corrupted user behaviors; (ii) detect the corrupted users online from bandit feedback.

• We propose a novel online learning framework composed of two algorithms, RCLUB-WCU and OCCUD, to tackle the challenging LOCUD problem. RCLUB-WCU robustly learns and utilizes

the unknown social relations among potentially corrupted users to efficiently minimize regret. Based on RCLUB-WCU' s inferred user relations, OCCUD accurately detects corrupted users online.
• We prove a regret upper bound for RCLUB-WCU, which matches the lower bound asymptotically in T up to logarithmic factors and matches the state-of-the-art results in several degenerate cases. We also give a theoretical performance guarantee for the online detection algorithm OCCUD.
• Experiments on both synthetic and real-world data clearly show the advantages of our methods.

6.2 Problem Setup

This section formulates the problem of "*Learning and Online Corrupted Users Detection from bandit feedback*" (LOCUD) (illustrated in Fig.6.1). We denote $\|\boldsymbol{x}\|_{\boldsymbol{M}} = \sqrt{\boldsymbol{x}^\top \boldsymbol{M} \boldsymbol{x}}$, $[m] = \{1, \ldots, m\}$, number of elements in set $\mathcal{A}$ as $|\mathcal{A}|$.

In LOCUD, there are u users, which we denote by set $\mathcal{U} = \{1, 2, \ldots, u\}$. Some of them are corrupted users, denoted by set $\tilde{\mathcal{U}} \subseteq \mathcal{U}$. These corrupted users, on the one hand, try to mimic normal users to make themselves hard to detect; on the other hand, they can occasionally perform corrupted behaviors to fool the agent into making sub-optimal decisions. Each user $i \in \mathcal{U}$, no matter a normal one or corrupted one, is associated with a (possibly mimicked for corrupted users) preference feature vector $\boldsymbol{\theta}_i \in \mathbb{R}^d$ that is *unknown* and $\|\boldsymbol{\theta}_i\|_2 \leq 1$. There is an underlying clustering structure among all the users representing the similarity of their preferences, but it is *unknown* to the agent and needs

to be learned via interactions. Specifically, the set of users $\mathcal{U}$ can be partitioned into m ($m \ll u$) clusters, $V_1, V_2, \dots V_m$, where $\cup_{j\in[m]} V_j = \mathcal{U}$, and $V_j \cap V_{j'} = \emptyset$, for $j \neq j'$. Users in the same cluster have the same preference feature vector, while users in different clusters have different preference vectors. We use $\boldsymbol{\theta}^j$ to denote the common preference vector shared by users in the j-th cluster V_j, and use $j(i)$ to denote the index of cluster user i belongs to (*i.e.*, $i \in V_{j(i)}$). For any two users $k, i \in \mathcal{U}$, if $k \in V_{j(i)}$, then $\boldsymbol{\theta}_k = \boldsymbol{\theta}^{j(i)} = \boldsymbol{\theta}_i$; otherwise $\boldsymbol{\theta}_k \neq \boldsymbol{\theta}_i$. We assume the arm set $\mathcal{A} \subseteq \mathbb{R}^d$ is finite. Each arm $a \in \mathcal{A}$ is associated with a feature vector $\boldsymbol{x}_a \in \mathbb{R}^d$ with $\|\boldsymbol{x}_a\|_2 \leq 1$.

The learning process of the agent is as follows. At each round $t \in [T]$, a user $i_t \in \mathcal{U}$ comes to be served, and the learning agent receives a set of arms $\mathcal{A}_t \subseteq \mathcal{A}$ to choose from. The agent infers the cluster V_t that user i_t belongs to based on the interaction history, and recommends an arm $a_t \in \mathcal{A}_t$ according to the aggregated information gathered in the cluster V_t. After receiving the recommended arm a_t, a normal user i_t will give a random reward with expectation $\boldsymbol{x}_{a_t}^\top \boldsymbol{\theta}_{i_t}$ to the agent.

To model the behaviors of corrupted users, following [49, 53, 142, 51], we assume that they can occasionally corrupt the rewards to mislead the agent into recommending sub-optimal arms. Specifically, at each round t, if the current served user is a corrupted user (i.e., $i_t \in \tilde{\mathcal{U}}$), the user can corrupt the reward by c_t. In summary, we model the reward received by the agent at round t as

$$r_t = \boldsymbol{x}_{a_t}^\top \boldsymbol{\theta}_{i_t} + \eta_t + c_t \,,$$

where $c_t = 0$ if i_t is a normal user, (*i.e.*, $i_t \notin \tilde{\mathcal{U}}$), and η_t is 1-sub-Gaussian random noise.

As the number of corrupted users is usually small, and they only corrupt the rewards occasionally with small magnitudes to make themselves hard to detect, we assume the sum of corruption magnitudes in all rounds is upper bounded by the *corruption level* C, *i.e.*, $\sum_{t=1}^{T} |c_t| \leq C$ [49, 53, 142, 51].

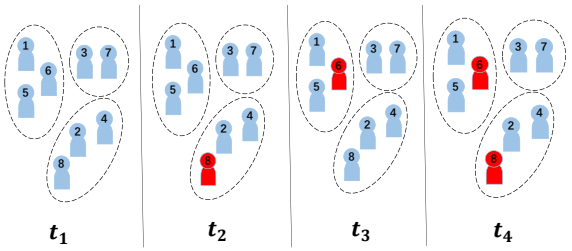


Figure 6.1: Illustration of LOCUD. The *unknown* user relations are represented by dotted circles, *e.g.*, user 3, 7 have similar preferences and thus can be in the same user segment (*i.e.*, cluster). Users 6 and 8 are corrupted users with dynamic behaviors over time (*e.g.*, for user 8, the behaviors are normal at t_1 and t_3 (blue), but are adversarially corrupted at t_2 and t_4 (red)[49, 51]), making them hard to be detected online. The agent needs to learn user relations to utilize information among similar users to speed up learning, and detect corrupted users 6, 8 online from bandit feedback.

We assume the clusters, users, and items satisfy the following assumptions. Note that all these assumptions basically follow the settings from classical works on clustering of bandits [46, 135, 137, 225].

Assumption 6.1 (Gap between different clusters)**.** *The gap between any two preference vectors for different clusters is at least an unknown positive constant* γ

$$\left\| \boldsymbol{\theta}^{j} - \boldsymbol{\theta}^{j'} \right\|_2 \geq \gamma > 0 \,, \forall j, j' \in [m] \,, j \neq j' \,.$$

Assumption 6.2 (Uniform arrival of users)**.** *At each round* t*, a user* i_t *comes uniformly at random from* $\mathcal{U}$ *with probability* $1/u$*,*

independent of the past rounds.

Assumption 6.3 (Item regularity)**.** *At each round t, the feature vector $\boldsymbol{x}_a$ of each arm $a \in \mathcal{A}_t$ is drawn independently from a fixed unknown distribution ρ over $\{\boldsymbol{x} \in \mathbb{R}^d : \|\boldsymbol{x}\|_2 \leq 1\}$, where $\mathbb{E}_{\boldsymbol{x}\sim\rho}[\boldsymbol{x}\boldsymbol{x}^\top]$'s minimal eigenvalue $\lambda_x > 0$. At $\forall t$, for any fixed unit vector $\boldsymbol{z} \in \mathbb{R}^d$, $(\boldsymbol{\theta}^\top \boldsymbol{z})^2$ has sub-Gaussian tail with variance no greater than σ^2.*

Let $a_t^* \in \arg\max_{a\in\mathcal{A}_t} \boldsymbol{x}_a^\top \boldsymbol{\theta}_{i_t}$ denote an optimal arm with the highest expected reward at round t. One objective of the learning agent is to minimize the expected cumulative regret

$$R(T) = \mathbb{E}[\textstyle\sum_{t=1}^{T}(\boldsymbol{x}_{a_t^*}^\top \boldsymbol{\theta}_{i_t} - \boldsymbol{x}_{a_t}^\top \boldsymbol{\theta}_{i_t})]\,. \tag{6.1}$$

Another objective is to detect corrupted users online accurately. Specifically, at round t, the agent will give a set of users $\tilde{\mathcal{U}}_t$ as the detected corrupted users, and we want $\tilde{\mathcal{U}}_t$ to be as close to the ground-truth set of corrupted users $\tilde{\mathcal{U}}$ as possible.

6.3 Algorithms

This section introduces our algorithms RCLUB-WCU (Algo.8) and OCCUD (Algo.9). RCLUB-WCU robustly learns the unknown user clustering structure and preferences from corrupted feedback, and leverages the cluster-based information to accelerate learning. Based on the clustering structure learned by RCLUB-WCU, OCCUD can accurately detect corrupted users online.

Algorithm 8 RCLUB-WCU

Input: Regularization parameter λ, confidence radius parameter β, threshold parameter α, edge deletion parameter α_1, $f(T) = \sqrt{(1+\ln(1+T))/(1+T)}$.

Initialization: $\boldsymbol{M}_{i,0} = \mathbf{0}_{d\times d}, \boldsymbol{b}_{i,0} = \mathbf{0}_{d\times 1}$, $\tilde{\boldsymbol{M}}_{i,0} = \mathbf{0}_{d\times d}, \tilde{\boldsymbol{b}}_{i,0} = \mathbf{0}_{d\times 1}, T_{i,0} = 0$, $\forall i \in \mathcal{U}$;

A complete graph $G_0 = (\mathcal{U}, E_0)$ over $\mathcal{U}$.

for all $t = 1, 2, \ldots, T$ **do**

Receive the index of the current served user $i_t \in \mathcal{U}$, get the feasible arm set at this round $\mathcal{A}_t$.

Determine the connected components V_t in the current maintained graph $G_{t-1} = (\mathcal{U}, E_{t-1})$ such that $i_t \in V_t$.

Calculate the robustly estimated statistics for the cluster V_t:
$\boldsymbol{M}_{V_t,t-1} = \lambda \boldsymbol{I} + \sum_{i\in V_t} \boldsymbol{M}_{i,t-1}$, $\boldsymbol{b}_{V_t,t-1} = \sum_{i\in V_t} \boldsymbol{b}_{i,t-1}$, $\hat{\boldsymbol{\theta}}_{V_t,t-1} = \boldsymbol{M}_{V_t,t-1}^{-1} \boldsymbol{b}_{V_t,t-1}$;

Select an arm a_t with largest UCB index in Eq.(6.3) and receive the corresponding reward r_t;

Update the statistics for robust estimation of user i_t:
$\boldsymbol{M}_{i_t,t} = \boldsymbol{M}_{i_t,t-1} + w_{i_t,t-1} \boldsymbol{x}_{a_t} \boldsymbol{x}_{a_t}^\top$, $\boldsymbol{b}_{i_t,t} = \boldsymbol{b}_{i_t,t-1} + w_{i_t,t-1} r_t \boldsymbol{x}_{a_t}$, $T_{i_t,t} = T_{i_t,t-1} + 1$,
$\boldsymbol{M}'_{i_t,t} = \lambda \boldsymbol{I} + \boldsymbol{M}_{i_t,t}$, $\hat{\boldsymbol{\theta}}_{i_t,t} = \boldsymbol{M}'^{-1}_{i_t,t} \boldsymbol{b}_{i_t,t}$, $w_{i_t,t} = \min\{1, \alpha / \|\boldsymbol{x}_{a_t}\|_{\boldsymbol{M}'^{-1}_{i_t,t}}\}$;

Keep robust estimation statistics of other users unchanged:
$\boldsymbol{M}_{\ell,t} = \boldsymbol{M}_{\ell,t-1}, \boldsymbol{b}_{\ell,t} = \boldsymbol{b}_{\ell,t-1}, T_{\ell,t} = T_{\ell,t-1}$, $\hat{\boldsymbol{\theta}}_{\ell,t} = \hat{\boldsymbol{\theta}}_{\ell,t-1}$, for all $\ell \in \mathcal{U}, \ell \neq i_t$;

Delete the edge $(i_t, \ell) \in E_{t-1}$, if

$$\left\| \hat{\boldsymbol{\theta}}_{i_t,t} - \hat{\boldsymbol{\theta}}_{\ell,t} \right\|_2 \geq \alpha_1 \big(f(T_{i_t,t}) + f(T_{\ell,t}) + \alpha C \big) ,$$

and get an updated graph $G_t = (\mathcal{U}, E_t)$;

Use the OCCUD Algorithm (Algo.9) to detect the corrupted users.

end for

6.3.1 RCLUB-WCU

The corrupted behaviors may cause inaccurate preference estimations, leading to erroneous relation inference and sub-optimal decisions. In this case, how to learn and utilize unknown user

relations to accelerate learning becomes non-trivial. Motivated by this, we design RCLUB-WCU as follows.

Assign the inferred cluster V_t for user i_t. RCLUB-WCU maintains a dynamic undirected graph $G_t = (\mathcal{U}, E_t)$ over users, which is initialized to be a complete graph (Algo.8 Line 2). Users with similar learned preferences will be connected with edges in E_t. The connected components in the graph represent the inferred clusters by the algorithm. At round t, user i_t comes to be served with a feasible arm set $\mathcal{A}_t$ for the agent to choose from (Line 4). In Line 5, RCLUB-WCU detects the connected component V_t in the graph containing user i_t to be the current inferred cluster for i_t.

Robust preference estimation of cluster V_t. After determining the cluster V_t, RCLUB-WCU estimates the common preferences for users in V_t using the historical feedback of all users in V_t and recommends an arm accordingly. The corrupted behaviors could cause inaccurate preference estimates, which can easily mislead the agent. To address this, inspired by [240, 51], we use weighted ridge regression to make corruption-robust estimations. Specifically, RCLUB-WCU robustly estimates the common preference vector of cluster V_t by solving the following weighted ridge regression

$$\hat{\boldsymbol{\theta}}_{V_t,t-1} = \underset{\boldsymbol{\theta}\in\mathbb{R}^d}{\arg\min} \sum_{\substack{s\in[t-1]\\ i_s\in V_t}} w_{i_s,s}(r_s - \boldsymbol{x}_{a_s}^\top \boldsymbol{\theta})^2 + \lambda \|\boldsymbol{\theta}\|_2^2 \,, \tag{6.2}$$

where $\lambda > 0$ is a regularization coefficient. Its closed-form solution is $\hat{\boldsymbol{\theta}}_{V_t,t-1} = \boldsymbol{M}_{V_t,t-1}^{-1}\boldsymbol{b}_{V_t,t-1}$, where $\boldsymbol{M}_{V_t,t-1} = \lambda\boldsymbol{I} + \sum_{\substack{s\in[t-1]\\ i_s\in V_t}} w_{i_s,s}\boldsymbol{x}_{a_s}\boldsymbol{x}_{a_s}^\top$, $\boldsymbol{b}_{V_t,t-1} = \sum_{\substack{s\in[t-1]\\ i_s\in V_t}} w_{i_s,s} r_{a_s}\boldsymbol{x}_{a_s}$.

We set the weight of sample for user i_s in V_t at round s as

$w_{i_s,s} = \min\{1, \alpha/\left\|\boldsymbol{x}_{a_s}\right\|_{M'^{-1}_{i_s,s}}\}$, where α is a coefficient to be determined later. The intuitions of designing these weights are as follows. The term $\left\|\boldsymbol{x}_{a_s}\right\|_{M'^{-1}_{i_s,s}}$ is the confidence radius of arm a_s for user i_s at s, reflecting how confident the algorithm is about the estimation of i_s's preference on a_s at s. If $\left\|\boldsymbol{x}_{a_s}\right\|_{M'^{-1}_{i_s,s}}$ is large, it means the agent is uncertain of user i_s's preference on a_s, indicating this sample is probably corrupted. Therefore, we use the inverse of confidence radius to assign a small weight to this round's sample if it is potentially corrupted. In this way, uncertain information for users in cluster V_t is assigned with less importance when estimating the V_t's preference vector, which could help relieve the estimation inaccuracy caused by corruption. For technical details, please refer to Section 6.4.1 and Appendix.

Recommend a_t with estimated preference of cluster V_t. Based on the corruption-robust preference estimation $\hat{\boldsymbol{\theta}}_{V_t,t-1}$ of cluster V_t, in Line 7, the agent recommends an arm using the upper confidence bound (UCB) strategy to balance exploration and exploitation

$$a_t = \operatorname*{argmax}_{a\in\mathcal{A}_t} \boldsymbol{x}_a^\top \hat{\boldsymbol{\theta}}_{V_t,t-1} + \beta \left\|\boldsymbol{x}_a\right\|_{\boldsymbol{M}^{-1}_{V_t,t-1}} \triangleq \hat{R}_{a,t} + C_{a,t}\,, \tag{6.3}$$

where $\beta = \sqrt{\lambda} + \sqrt{2\log(\frac{1}{\delta}) + d\log(1+\frac{T}{\lambda d})} + \alpha C$ is the confidence radius parameter, $\hat{R}_{a,t}$ denotes the estimated reward of arm a at t, $C_{a,t}$ denotes the confidence radius of arm a at t. The design of $C_{a,t}$ theoretically relies on Lemma 6.4.2 that will be given in Section 6.4.

Update the robust estimation of user i_t. After receiving r_t, the algorithm updates the estimation statistics of user i_t, while

keeping the statistics of others unchanged (Line 8 and Line 9). Specifically, RCLUB-WCU estimates the preference vector of user i_t by solving a weighted ridge regression

$$\hat{\boldsymbol{\theta}}_{i_t,t} = \arg\min_{\boldsymbol{\theta}\in\mathbb{R}^d} \sum\nolimits_{\substack{s\in[t]\\ i_s=i_t}} w_{i_s,s}(r_s - \boldsymbol{x}_{a_s}^\top \boldsymbol{\theta})^2 + \lambda \left\|\boldsymbol{\theta}\right\|_2^2 \tag{6.4}$$

with closed-form solution $\hat{\boldsymbol{\theta}}_{i_t,t} = (\lambda \boldsymbol{I} + \boldsymbol{M}_{i_t,t})^{-1}\boldsymbol{b}_{i_t,t}$, where $\boldsymbol{M}_{i_t,t} = \sum_{\substack{s\in[t]\\ i_s=i_t}} w_{i_s,s}\boldsymbol{x}_{a_s}\boldsymbol{x}_{a_s}^\top$, $\boldsymbol{b}_{i_t,t} = \sum_{\substack{s\in[t]\\ i_s=i_t}} w_{i_s,s} r_{a_s}\boldsymbol{x}_{a_s}$, and we design the weights in the same way by the same reasoning. **Update the**

Algorithm 9 OCCUD (At round t, used in Line 11 in Algo.8)

1: Initialize $\tilde{\mathcal{U}}_t = \emptyset$; input probability parameter δ.
2: Update the statistics for non-robust estimation of user i_t
$\tilde{\boldsymbol{M}}_{i_t,t} = \tilde{\boldsymbol{M}}_{i_t,t-1} + \boldsymbol{x}_{a_t}\boldsymbol{x}_{a_t}^\top$, $\tilde{\boldsymbol{b}}_{i_t,t} = \tilde{\boldsymbol{b}}_{i_t,t-1} + r_t\boldsymbol{x}_{a_t}$, $\tilde{\boldsymbol{\theta}}_{i_t,t} = (\lambda\boldsymbol{I} + \tilde{\boldsymbol{M}}_{i_t,t})^{-1}\tilde{\boldsymbol{b}}_{i_t,t}$,
3: Keep non-robust estimation statistics of other users unchanged
$\tilde{\boldsymbol{M}}_{\ell,t} = \tilde{\boldsymbol{M}}_{\ell,t-1}, \tilde{\boldsymbol{b}}_{\ell,t} = \tilde{\boldsymbol{b}}_{\ell,t-1}, \tilde{\boldsymbol{\theta}}_{\ell,t} = \tilde{\boldsymbol{\theta}}_{\ell,t-1}$, for all $\ell \in \mathcal{U}, \ell \neq i_t$.
4: **for all** connected component $V_{j,t} \in G_t$ **do**
5: Calculate the robust estimation statistics for the cluster $V_{j,t}$:
$\boldsymbol{M}_{V_{j,t},t} = \lambda\boldsymbol{I} + \sum_{\ell\in V_{j,t}} \boldsymbol{M}_{\ell,t}$, $T_{V_{j,t},t} = \sum_{\ell\in V_{j,t}} T_{\ell,t}$,
$\boldsymbol{b}_{V_{j,t},t} = \sum_{\ell\in V_{j,t}} \boldsymbol{b}_{\ell,t}$, $\hat{\boldsymbol{\theta}}_{V_{j,t},t} = \boldsymbol{M}_{V_{j,t},t}^{-1}\boldsymbol{b}_{V_{j,t},t}$;
6: **for all** user $i \in V_{j,t}$ **do**
7: Detect user i to be a corrupted user and add user i to the set $\tilde{\mathcal{U}}_t$ if the following holds:

$$\left\|\tilde{\boldsymbol{\theta}}_{i,t} - \hat{\boldsymbol{\theta}}_{V_{i,t},t}\right\|_2 > \frac{\sqrt{d\log(1+\frac{T_{i,t}}{\lambda d}) + 2\log(\frac{1}{\delta})}\sqrt{\lambda}}{\sqrt{\lambda_{\min}(\tilde{\boldsymbol{M}}_{i,t}) + \lambda}} + \frac{\sqrt{d\log(1+\frac{T_{V_{i,t},t}}{\lambda d}) + 2\log(\frac{1}{\delta})} + \sqrt{\lambda} + \alpha C}{\sqrt{\lambda_{\min}(\boldsymbol{M}_{V_{i,t},t})}}, \tag{6.5}$$

where $\lambda_{\min}(\cdot)$ denotes the minimum eigenvalue of the matrix argument.
8: **end for**
9: **end for**

dynamic graph. Finally, with the updated statistics of user i_t, RCLUB-WCU checks whether the inferred i_t's preference similarities with other users are still true, and updates the graph accordingly. Precisely, if gap between the updated estimation $\hat{\boldsymbol{\theta}}_{i_t,t}$ of i_t and the estimation $\hat{\boldsymbol{\theta}}_{\ell,t}$ of user ℓ exceeds a threshold in Line 10, RCLUB-WCU will delete the edge (i_t, ℓ) in G_{t-1} to split them apart. The threshold is carefully designed to handle the estimation uncertainty from both stochastic noises and potential corruptions. The updated graph $G_t = (\mathcal{U}, E_t)$ will be used in the next round.

6.3.2 OCCUD

Based on the inferred clustering structure of RCLUB-WCU, we devise a novel online detection algorithm OCCUD (Algo.9). The design ideas and process of OCCUD are as follows.

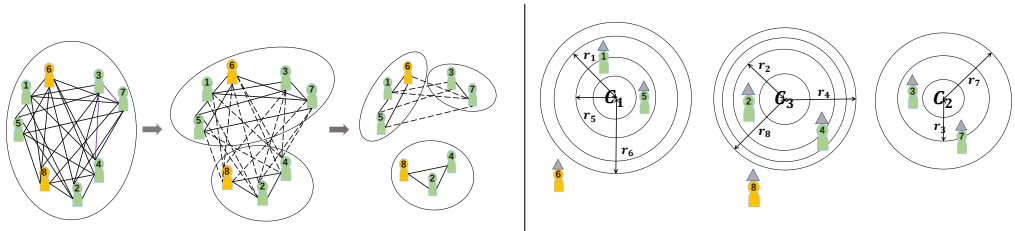
Besides the robust preference estimations (with weighted regression) of users and clusters kept by RCLUB-WCU, OCCUD also maintains the non-robust estimations for each user by online ridge regression without weights (Line 2 and Line 3). Specifically, at round t, OCCUD updates the non-robust estimation of user i_t by solving the following online ridge regression:

$$\tilde{\boldsymbol{\theta}}_{i_t,t} = \underset{\boldsymbol{\theta}\in\mathbb{R}^d}{\arg\min} \sum\nolimits_{\substack{s\in[t]\\ i_s=i_t}} (r_s - \boldsymbol{x}_{a_s}^\top \boldsymbol{\theta})^2 + \lambda \left\|\boldsymbol{\theta}\right\|_2^2 , \tag{6.6}$$

with solution $\tilde{\boldsymbol{\theta}}_{i_t,t} = (\lambda \boldsymbol{I} + \tilde{\boldsymbol{M}}_{i_t,t})^{-1} \tilde{\boldsymbol{b}}_{i_t,t}$, where $\tilde{\boldsymbol{M}}_{i_t,t} = \sum_{\substack{s\in[t]\\ i_s=i_t}} \boldsymbol{x}_{a_s} \boldsymbol{x}_{a_s}^\top$, $\tilde{\boldsymbol{b}}_{i_t,t} = \sum_{\substack{s\in[t]\\ i_s=i_t}} r_{a_s} \boldsymbol{x}_{a_s}$.

With the robust and non-robust preference estimations, OCCUD does the following to detect corrupted users based on the

clustering structure inferred by RCLUB-WCU. First, OCCUD finds the connected components in the graph kept by RCLUB-WCU, which represent the inferred clusters. Then, for each inferred cluster $V_{j,t} \in G_t$: (1) OCCUD computes its robustly estimated preferences vector $\hat{\boldsymbol{\theta}}_{V_{i,t},t}$ (Line 5). (2) For each user i whose inferred cluster is $V_{j,t}$ ($i.e.,i \in V_{j,t}$), OCCUD computes the gap between user i's non-robustly estimated preference vector $\tilde{\boldsymbol{\theta}}_{i,t}$ and the robust estimation $\hat{\boldsymbol{\theta}}_{V_{i,t},t}$ for user i's inferred cluster $V_{j,t}$. If the gap exceeds a carefully-designed threshold, OCCUD will detect user i as corrupted and add i to the detected corrupted user set $\tilde{\mathcal{U}}_t$ (Line 7).



(a) RCLUB-WCU (b) OCCUD

Figure 6.2: Algorithm illustrations. Users 6 and 8 are corrupted users (orange), and the others are normal (green). (a) illustrates RCLUB-WCU, which starts with a complete user graph, and adaptively deletes edges between users (dashed lines) with dissimilar robustly learned preferences. The corrupted behaviors of users 6 and 8 may cause inaccurate preference estimations, leading to erroneous relation inference. In this case, how to delete edges correctly is non-trivial, and RCLUB-WCU addresses this challenge (detailed in Section 6.3.1). (b) illustrates OCCUD at some round t, where person icons with triangle hats represent the non-robust user preference estimations. The gap between the non-robust estimation of user 6 and the robust estimation of user 6's inferred cluster (circle C_1) exceeds the threshold r_6 at this round (Line 7 in Algo.9), so OCCUD detects user 6 to be corrupted.

The intuitions of OCCUD are as follows. On the one hand,

after some interactions, RCLUB-WCU will infer the user clustering structure accurately. Thus, at round t, the robust estimation $\hat{\boldsymbol{\theta}}_{V_{i,t},t}$ for user i's inferred cluster should be pretty close to user i's ground-truth preference vector $\boldsymbol{\theta}_i$. On the other hand, since the feedback of normal users are always regular, at round t, if user i is a normal user, the non-robust estimation $\tilde{\boldsymbol{\theta}}_{i,t}$ should also be close to the ground-truth $\boldsymbol{\theta}_i$. However, the non-robust estimation of a corrupted user should be quite far from the ground truth due to corruptions. Based on this reasoning, OCCUD compares each user's non-robust estimation and the robust estimation of the user's inferred cluster to detect the corrupted users. For technical details, please refer to Section 6.4.2 and Appendix. Simple illustrations of our proposed algorithms can be found in Fig.6.2.

6.4 Theoretical Analysis

In this section, we theoretically analyze the performances of our proposed algorithms, RCLUB-WCU and OCCUD. Due to the page limit, we put the proofs in the Appendix.

6.4.1 Regret Analysis of RCLUB-WCU

This section gives an upper bound of the expected regret (defined in Eq.(6.1)) for RCLUB-WCU.

The following lemma provides a sufficient time $T_0(\delta)$, after which RCLUB-WCU can cluster all the users correctly with high probability.

Lemma 6.4.1. *With probability at least* $1-3\delta$*, RCLUB-WCU will cluster all the users correctly after*

$$T_0(\delta) \triangleq 16u\log(\frac{u}{\delta}) + 4u\max\{\frac{288d}{\gamma^2\alpha\sqrt{\lambda}\tilde{\lambda}_x}\log(\frac{u}{\delta}), \frac{16}{\tilde{\lambda}_x^2}\log(\frac{8d}{\tilde{\lambda}_x^2\delta}), \frac{72\sqrt{\lambda}}{\alpha\gamma^2\tilde{\lambda}_x}, \frac{72\alpha C^2}{\gamma^2\sqrt{\lambda}\tilde{\lambda}_x}\}$$

for some $\delta \in (0, \frac{1}{3})$*, where* $\tilde{\lambda}_x \triangleq \int_0^{\lambda_x}(1-e^{-\frac{(\lambda_x-x)^2}{2\sigma^2}})^K dx$*,* $|\mathcal{A}_t| \leq K, \forall t \in [T]$*.*

After $T_0(\delta)$, the following lemma gives a bound of the gap between $\hat{\boldsymbol{\theta}}_{V_t,t-1}$ and the ground-truth $\boldsymbol{\theta}_{i_t}$ in direction of action vector $\boldsymbol{x}_a$ for RCLUB-WCU, which supports the design in Eq.(6.3).

Lemma 6.4.2. *With probability at least* $1-4\delta$ *for some* $\delta \in (0, \frac{1}{4})$*,* $\forall t \geq T_0(\delta)$*, we have:*

$$\left|\boldsymbol{x}_a^{\mathrm{T}}(\hat{\boldsymbol{\theta}}_{V_t,t-1} - \boldsymbol{\theta}_{i_t})\right| \leq \beta \left\|\boldsymbol{x_a}\right\|_{\boldsymbol{M}_{V_t,t-1}^{-1}} \triangleq C_{a,t}\,.$$

With Lemma A.3.2 and 6.4.2, we prove the following theorem on the regret upper bound of RCLUB-WCU.

Theorem 6.4.3 (Regret Upper Bound of RCLUB-WCU). *With the assumptions in Section 6.2, and picking* $\alpha = \frac{\sqrt{d}+\sqrt{\lambda}}{C}$*, the expected regret of the RCLUB-WCU algorithm for* T *rounds satisfies*

$$R(T) \leq O\big((\frac{C\sqrt{d}}{\gamma^2\tilde{\lambda}_x} + \frac{1}{\tilde{\lambda}_x^2})u\log(T)\big) + O\big(d\sqrt{mT}\log(T)\big) + O\big(mCd\log^{1.5}(T)\big)\,. \quad (6.7)$$

Discussion and Comparison. The regret bound in Eq.(6.7) has three terms. The first term is the time needed to get enough information for accurate robust estimations such that RCLUB-WCU could cluster all users correctly afterward with high probability. This term is related to the *corruption level* C, which is inevitable since, if there are more corrupted user feedback, it will

be harder for the algorithm to learn the clustering structure correctly. The last two terms correspond to the regret after T_0 with the correct clustering. Specifically, the second term is caused by stochastic noises when leveraging the aggregated information within clusters to make recommendations; the third term associated with the *corruption level* C is the regret caused by the disruption of corrupted behaviors.

When the *corruption level* C is *unknown*, we can use its estimated upper bound $\hat{C} \triangleq \sqrt{T}$ to replace C in the algorithm. In this way, if $C \leq \hat{C}$, the bound will be replacing C with $\hat{C}$ in Eq.(6.7); when $C > \sqrt{T}$, $R(T) = O(T)$, which is already optimal for a large class of bandit algorithms [51].

The following theorem gives a regret lower bound of the LOCUD problem.

Theorem 6.4.4 (Regret lower bound for LOCUD)**.** *There exists a problem instance for the LOCUD problem such that for any algorithm*

$$R(T) \geq \Omega(d\sqrt{mT} + dC)\,.$$

Its proof and discussions can be found in Appendix A.4.4. The upper bound in Theorem 6.4.3 asymptotically matches this lower bound in T up to logarithmic factors, showing our regret bound is nearly optimal.

We then compare our regret upper bound with several degenerated cases. First, when $C = 0$, *i.e.*, all users are normal, our setting degenerates to the classic CB problem [46]. In this case the bound in Theorem 6.4.3 becomes $O(1/\tilde{\lambda}_x^2 \cdot u \log(T)) + O(d\sqrt{mT}\log(T))$, perfectly matching the state-of-the-art results

in CB [46, 135, 237]. Second, when $m = 1$ and $u = 1$, *i.e.*, there is only one user, our setting degenerates to linear bandits with adversarial corruptions [54, 51], and the bound in Theorem 6.4.3 becomes $O(d\sqrt{T}\log(T)) + O(Cd\log^{1.5}(T))$, it also perfectly matches the nearly optimal result in [51]. The above comparisons also show the tightness of the regret bound of RCLUB-WCU.

6.4.2 Theoretical Performance Guarantee for OCCUD

The following theorem gives a performance guarantee of the online detection algorithm OCCUD.

Theorem 6.4.5 (**Theoretical Guarantee for OCCUD**). *With assumptions in Section 6.2, at $\forall t \geq T_0(\delta)$, for any detected corrupted user $i \in \tilde{\mathcal{U}}_t$, with probability at least $1 - 5\delta$, i is indeed a corrupted user.*

This theorem guarantees that after RCLUB-WCU learns the clustering structure accurately, with high probability, the corrupted users detected by OCCUD are indeed corrupted, showing the high detection accuracy of OCCUD. The proof of Theorem 6.4.5 can be found in Appendix A.4.4.

6.5 Experiments

This section shows experimental results on synthetic and real data to evaluate RCLUB-WCU's recommendation quality and OCCUD's detection accuracy. We compare RCLUB-WCU to LinUCB [43] with a single non-robust estimated vector for all users, LinUCB-Ind with separate non-robust estimated vectors for each

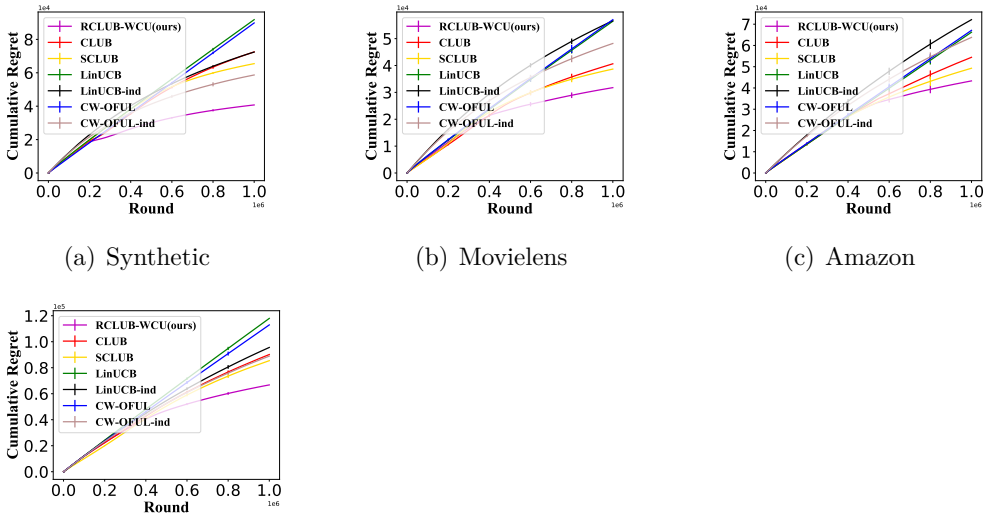
user, CW-OFUL [51] with a single robust estimated vector for all users, CW-OFUL-Ind with separate robust estimated vectors for each user, CLUB[46], and SCLUB[237]. More description of these baselines are in Appendix A.4.6. To show that the design of OCCUD is non-trivial, we develop a straightforward detection algorithm GCUD, which leverages the same cluster structure as OCCUD but detects corrupted users by selecting users with highest $\left\|\hat{\boldsymbol{\theta}}_{i,t} - \hat{\boldsymbol{\theta}}_{V_{i,t},t-1}\right\|_2$ in each inferred cluster. GCUD selects users according to the underlying percentage of corrupted users, which is unrealistic in practice, but OCCUD still performs better in this unfair condition.

Remark. The offline detection methods [57, 56, 238, 239] need to know all the user information in advance to derive the user embedding for classification, so they cannot be directly applied in online detection with bandit feedback thus cannot be directly compared to OCCUD. However, we observe the AUC achieved by OCCUD on Amazon and Yelp (in Tab.6.1) is similar to recent offline methods [238, 239]. Additionally, OCCUD has rigorous theoretical performance guarantee (Section 6.4.2).

6.5.1 Experiments on Synthetic Dataset

We use $u = 1,000$ users and $m = 10$ clusters, where each cluster contains 100 users. We randomly select 100 users as the corrupted users. The preference and arm (item) vectors are drawn in $d-1$ ($d = 50$) dimensions with each entry a standard Gaussian variable and then normalized, added one more dimension with constant 1, and divided by $\sqrt{2}$ [237]. We fix an arm set with $|\mathcal{A}| = 1000$ items,

at each round, 20 items are randomly selected to form a set $\mathcal{A}_t$ to choose from. Following [240, 241], in the first k rounds, we always flip the reward of corrupted users by setting $r_t = -\boldsymbol{x}_{a_t}^{\mathrm{T}} \boldsymbol{\theta}_{i_t,t} + \eta_t$. And we leave the remaining $T - k$ rounds intact. Here we set $T = 1,000,000$ and $k = 20,000$.



(a) Synthetic (b) Movielens (c) Amazon

(d) Yelp

Figure 6.3: Recommendation results on the synthetic and real-world datasets

Fig.6.3(a) shows the recommendation results. RCLUB-WCU outperforms all baselines and achieves a sub-linear regret. LinUCB and CW-OFUL perform worst as they ignore the preference differences among users. CW-OFUL-Ind outperforms LinUCB-Ind because it considers the corruption, but worse than RCLUB-WCU since it does not consider leveraging user relations to speed up learning.

The detection results are shown in Tab.6.1. We test the AUC of OCCUD and GCUD in every $200,000$ rounds. OCCUD's performance improves over time with more interactions, while GCUD's performance is much worse as it detects corrupted users

only relying on the robust estimations. OCCUD finally achieves an AUC of 0.855, indicating it can identify most of the corrupted users.

6.5.2 Experiments on Real-world Datasets

We use three real-world data Movielens [228], Amazon[242], and Yelp [243]. The Movielens data does not have the corrupted users' labels, so following [244], we manually select the corrupted users. On Amazon data, following [57], we label the users with more than 80% helpful votes as normal users, and label users with less than 20% helpful votes as corrupted users. The Yelp data contains users and their comments on restaurants with true labels of the normal users and corrupted users.

We select 1,000 users and 1,000 items for Movielens; 1,400 users and 800 items for Amazon; 2,000 users and 2,000 items for Yelp. The ratios of corrupted users on these data are 10%, 3.5%, and 30.9%, respectively. We generate the preference and item vectors following [180, 237]. We first construct the binary feedback matrix through the users' ratings: if the rating is greater than 3, the feedback is 1; otherwise, the feedback is 0. Then we use SVD to decompose the extracted binary feedback matrix $R_{u\times m} = \boldsymbol{\theta} S X^{\mathrm{T}}$, where $\boldsymbol{\theta} = (\boldsymbol{\theta}_i), i \in [u]$ and $X = (\boldsymbol{x}_j), j \in [m]$, and select $d = 50$ dimensions. We have 10 clusters on Movielens and Amazon, and 20 clusters on Yelp. We use the same corruption mechanism as the synthetic data with $T = 1,000,000$ and $k = 20,000$. We conduct more experiments in different environments to show our algorithms' robustness in Appendix.A.4.7.

The recommendation results are shown in Fig.6.3(b)-(d). RCLUB-WCU outperforms all baselines. On the Amazon dataset, the percentage of corrupted users is lowest, RCLUB-WCU's advantages over baselines decrease because of the weakened corruption. The corrupted user detection results are provided in Tab.6.1. OCCUD's performance improves over time and is much better than GCUD. On the Movielens dataset, OCCUD achieves an AUC of 0.85; on the Amazon dataset, OCCUD achieves an AUC of 0.84; and on the Yelp dataset, OCCUD achieves an AUC of 0.628. According to recent works on offline settings [238, 239], our results are relatively high.

Dataset	Alg \ Time	0.2M	0.4M	0.6M	0.8M	1M
Synthetic	OCCUD	0.599	0.651	0.777	0.812	**0.855**
	GCUD	0.477	0.478	0.483	0.484	0.502
Movielens	OCCUD	0.65	0.750	0.785	0.83	**0.85**
	GCUD	0.450	0.474	0.485	0.489	0.492
Amazon	OCCUD	0.639	0.735	0.761	0.802	**0.840**
	GCUD	0.480	0.480	0.486	0.500	0.518
Yelp	OCCUD	0.452	0.489	0.502	0.578	**0.628**
	GCUD	0.473	0.481	0.496	0.500	0.510

Table 6.1: Detection results on synthetic and real datasets

Chapter 7

Efficient Explorative Key-term Selection Strategies for Conversational Contextual Bandits

Conversational contextual bandits elicit user preferences by occasionally querying for explicit feedback on key-terms to accelerate learning. However, there are aspects of existing approaches which limit their performance. First, information gained from key-term-level conversations and arm-level recommendations is not appropriately incorporated to speed up learning. Second, it is important to ask explorative key-terms to quickly elicit the user's potential interests in various domains to accelerate the convergence of user preference estimation, which has never been considered in existing works. To tackle these issues, we first propose "ConLinUCB", a general framework for conversational bandits with better information incorporation, combining arm-level and key-term-level feedback to estimate user preference in one

step at each time. Based on this framework, we further design two bandit algorithms with explorative key-term selection strategies, ConLinUCB-BS and ConLinUCB-MCR. We prove tighter regret upper bounds of our proposed algorithms. Particularly, ConLinUCB-BS achieves a regret bound of $O(d\sqrt{T \log T})$, better than the previous result $O(d\sqrt{T} \log T)$. Extensive experiments on synthetic and real-world data show significant advantages of our algorithms in learning accuracy (up to 54% improvement) and computational efficiency (up to 72% improvement), compared to the classic ConUCB algorithm, showing the potential benefit to recommender systems. This chapter is based on our publication [7].

7.1 Introduction

Nowadays, recommender systems are widely used in various areas. The learning speed for traditional online recommender systems is usually slow since extensive exploration is needed to discover user preferences. To accelerate the learning process and provide more personalized recommendations, the conversational recommender system (CRS) has been proposed [61, 245, 246, 247, 248, 249]. In a CRS, a learning agent occasionally asks for the user's explicit feedback on some "key-terms", and leverages this additional conversational information to better elicit the user's preferences [62, 181].

Despite the recent success of CRS, there are crucial limitations in using conversational contextual bandit approaches to design recommender systems. These limitations include: (a) The

information gained from key-term-level conversations and arm-level recommendations is not incorporated properly to speed up learning, as the user preferences are essentially assumed to be the same in these two stages but are estimated separately [62, 181, 180]; (b) Queries using traditional key-terms were restrictive and not explorative enough. Specifically, we say a key-term is "explorative" if it is under-explored so far and the system is uncertain about the user's preferences in its associated items. Asking for the user's feedback on explorative key-terms can efficiently elicit her potential interests in various domains (e.g., sports, science), which means we can quickly estimate the user preference vector in all directions of the feature space, thus accelerating the learning speed. Therefore, it is crucial to design explorative key-term selection strategies, which existing works have not considered.

Motivated by the above considerations, we propose to design conversational bandit algorithms that (i) estimate the user's preferences utilizing both arm-level and key-term-level interactions simultaneously to properly incorporate the information gained from both two levels and (ii) use effective strategies to choose explorative key-terms when conducting conversations for quick user preference inference.

To better utilize the interactive feedback from both recommendations and conversations, we propose ConLinUCB, a general framework for conversational bandits with possible flexible key-term selection strategies. ConLinUCB estimates the user preference vector by solving *one single optimization problem* that minimizes the mean squared error of both arm-level estimated rewards and key-term-level estimated feedback simultaneously, instead of

separately estimating at different levels as in previous works. In this manner, the information gathered from these two levels can be better combined to guide the learning.

Based on this ConLinUCB framework, we design two new algorithms with explorative key-term selection strategies, ConLinUCB-BS and ConLinUCB-MCR.

- ConLinUCB-BS makes use of a barycentric spanner containing linearly independent vectors, which can be an efficient exploration basis in bandit problems [250]. Whenever a conversation is allowed, ConLinUCB-BS selects an explorative key-term uniformly at random from a precomputed barycentric spanner $\mathcal{B}$ of the given key-term set $\mathcal{K}$.

- ConLinUCB-MCR applies in a more general setting when the key-term set can be time-varying, and it can leverage interactive histories to choose explorative key-terms adaptively. Note that in the bandit setting, we often use confidence radius to adaptively evaluate whether an arm has been sufficiently explored, and the confidence radius of an arm will shrink whenever it is selected [139]. This implies that an explorative key-term should have a large confidence radius. Based on this reasoning, ConLinUCB-MCR selects the most explorative key-terms with maximal confidence radius when conducting conversations.

Equipped with explorative conversations, our algorithms can quickly elicit user preferences for better recommendations. For example, if the key-term *sports* is explorative at round t, indicating that so far the agent is not sure whether the user favors

items associated with *sports* (e.g., basketball, volleyball), it will ask for the user's feedback on *sports* directly and conduct recommendations accordingly. In this manner, the agent can quickly find suitable items for the user. We prove the regret upper bounds of our algorithms, which are better than the classic ConUCB algorithm.

In summary, our paper makes the following contributions:

- We propose a new and general framework for conversational contextual bandits, ConLinUCB, which can efficiently incorporate the interactive information gained from both recommendations and conversations.
- Based on ConLinUCB, we design two new algorithms with explorative key-term selection strategies, ConLinUCB-BS and ConUCB-MCR, which can accelerate the convergence of user preference estimation.
- We prove that our algorithms achieve tight regret upper bounds. Particularly, ConLinUCB-BS achieves a bound of $O(d\sqrt{T \log T})$, better than the previous $O(d\sqrt{T} \log T)$ in the conversational bandits literature.
- Experiments on both synthetic and real-world data validate the advantages of our algorithms in both learning accuracy (up to 54% improvement) and computational efficiency (up to 72% improvement)[1].

[1]Codes are available at https://github.com/ZhiyongWangWzy/ConLinUCB.

7.2 Problem Settings

This section states the problem setting of conversational contextual bandits. Suppose there is a finite set $\mathcal{A}$ of arms. Each arm $a \in \mathcal{A}$ represents an item to be recommended and is associated with a feature vector $\boldsymbol{x}_a \in \mathbb{R}^d$. Without loss of generality, the feature vectors are assumed to be normalized, i.e., $\|\boldsymbol{x}_a\|_2 = 1$, $\forall a \in \mathcal{A}$. The agent interacts with a user in $T \in \mathbb{N}_+$ rounds, whose preference of items is represented by an *unknown* vector $\boldsymbol{\theta}^* \in \mathbb{R}^d$, $\|\boldsymbol{\theta}^*\|_2 \leq 1$.

At each round $t = 1, 2, ..., T$, a subset of arms $\mathcal{A}_t \subseteq \mathcal{A}$ are available to the agent to choose from. Based on historical interactions, the agent selects an arm $a_t \in \mathcal{A}_t$, and receives a corresponding reward $r_{a_t,t} \in [0, 1]$. The reward is assumed to be a linear function of the contextual vectors

$$r_{a_t,t} = \boldsymbol{x}_{a_t}^\top \boldsymbol{\theta}^* + \epsilon_t \,, \tag{7.1}$$

where ϵ_t is 1-sub-Gaussian random noise with zero mean.

Let $a_t^* \in \arg\max_{a \in \mathcal{A}_t} \boldsymbol{x}_a^\top \boldsymbol{\theta}^*$ denote an optimal arm with the largest expected reward at t. The learning objective is to minimize the cumulative regret

$$R(T) = \sum_{t=1}^{T} \boldsymbol{x}_{a_t^*}^\top \boldsymbol{\theta}^* - \sum_{t=1}^{T} \boldsymbol{x}_{a_t}^\top \boldsymbol{\theta}^*. \tag{7.2}$$

The agent can also occasionally query the user's feedback on some conversational key-terms to help elicit user preferences. In particular, a "key-term" is a keyword or topic related to a subset of arms. For example, the key-term *sports* is related to the arms like basketball, football, swimming, etc.

Suppose there is a finite set $\mathcal{K}$ of key-terms. The relationship between arms and key-terms is given by a weighted bipartite graph $(\mathcal{A}, \mathcal{K}, \boldsymbol{W})$, where $\boldsymbol{W} \triangleq [w_{a,k}]_{a\in\mathcal{A},k\in\mathcal{K}}$ represents the relationship between arms and key-terms, i.e., a key-term $k \in \mathcal{K}$ is associated to an arm $a \in \mathcal{A}$ with weight $w_{a,k} \geq 0$. We assume that each key-term k has positive weights with some related arms (i.e., $\sum_{a\in\mathcal{A}} w_{a,k} > 0$, $\forall k \in \mathcal{K}$), and the weights associated with each arm sum up to 1, i.e., $\sum_{k\in\mathcal{K}} w_{a,k} = 1$, $a \in \mathcal{A}$. The feature vector of a key-term k is given by $\tilde{\boldsymbol{x}}_k = \sum_{a\in\mathcal{A}} \frac{w_{a,k}}{\sum_{a'\in\mathcal{A}} w_{a',k}} \boldsymbol{x}_a$. The key-term-level feedback on the key-term k at t is defined as

$$\tilde{r}_{k,t} = \tilde{\boldsymbol{x}}_k^\top \boldsymbol{\theta}^* + \tilde{\epsilon}_t \,, \tag{7.3}$$

where $\tilde{\epsilon}_t$ is assumed to be 1-sub-Gaussian random noise. One thing to stress is that in the previous works [62, 180, 181, 182], the *unknown* user preference vector $\boldsymbol{\theta}^*$ is essentially assumed to be the same at both the arm level and the key-term level.

To avoid affecting the user experience, the agent should not conduct conversations too frequently. Following [62], we define a function $b : \mathbb{N}_+ \to \mathbb{R}_+$, where $b(t)$ is increasing in t, to control the conversation frequency of the agent. At each round t, if $b(t) - b(t-1) > 0$, the agent is allowed to conduct $q(t) = \lfloor b(t) - b(t-1) \rfloor$ conversations by asking for user's feedback on $q(t)$ key-terms. Using this modeling arrangement, the agent will have $b(t)$ conversational interactions with the user up to round t.

7.3 Algorithms and Theoretical Analysis

This section first introduces ConLinUCB, a framework for conversational bandits with better information incorporation, which is general for "any" key-term selection strategies. Based on ConLinUCB, we further propose two bandit algorithms, ConLinUCB-BS and ConLinUCB-MCR, with explorative key-term selection strategies.

To simplify the exposition, we merge the ConLinUCB framework, ConLinUCB-BS and ConLinUCB-MCR in Algorithm 10. We also theoretically give regret bounds of our proposed algorithms.

7.3.1 General ConLinUCB Algorithm Framework

In conversational bandits, it is common that the *unknown* preference vector $\boldsymbol{\theta}^*$ is essentially assumed to be the same at both arm level and key-term level [62, 181, 180]. However, all existing works treat $\boldsymbol{\theta}^*$ differently at these two levels. Specifically, they take two different steps to estimate user preference vectors at the arm level and key-term level, and use a discounting parameter $\lambda \in (0, 1)$ to balance learning from these two levels' interactions. In this manner, the contributions of the arm-level and key-term-level information to the convergence of estimation are discounted by λ and $1 - \lambda$, respectively. Therefore, such discounting will cause waste of observations, indicating that information at these two levels can not be fully leveraged to accelerate the learning process.

To handle the above issues, we propose a general framework

called ConLinUCB, for conversational contextual bandits. In this new framework, in order to fully leverage interactive information from two levels, we simultaneously estimate the user preference vector by solving *one single optimization problem* that minimizes the mean squared error of both arm-level estimated rewards and key-term-level estimated feedback. Specifically, in ConLinUCB, at round t, the user preference vector is estimated by solving the following linear regression

$$\begin{aligned}\boldsymbol{\theta}_t = \arg\min_{\boldsymbol{\theta}\in\mathbb{R}^d} \sum_{\tau=1}^{t-1}(\boldsymbol{x}_{a_\tau}^\top\boldsymbol{\theta} - r_{a_\tau,\tau})^2 + \sum_{\tau=1}^{t}\sum_{k\in\mathcal{K}_\tau}(\tilde{\boldsymbol{x}}_k^\top\boldsymbol{\theta} - \tilde{r}_{k,\tau})^2 \\ + \beta\left\|\boldsymbol{\theta}\right\|_2^2 ,\end{aligned} \tag{7.4}$$

where $\mathcal{K}_\tau$ denotes the set of key-terms asked at round τ, and the coefficient $\beta > 0$ controls regularization. The closed-form solution of this optimization problem is

$$\boldsymbol{\theta}_t = \boldsymbol{M}_t^{-1}\boldsymbol{b}_t , \tag{7.5}$$

where

$$\begin{aligned}\boldsymbol{M}_t &= \sum_{\tau=1}^{t-1}\boldsymbol{x}_{a_\tau}\boldsymbol{x}_{a_\tau}^\top + \sum_{\tau=1}^{t}\sum_{k\in\mathcal{K}_\tau}\tilde{\boldsymbol{x}}_k\tilde{\boldsymbol{x}}_k^\top + \beta\boldsymbol{I} , \\ \boldsymbol{b}_t &= \sum_{\tau=1}^{t-1}\boldsymbol{x}_{a_\tau}r_{a_\tau,\tau} + \sum_{\tau=1}^{t}\sum_{k\in\mathcal{K}_\tau}\tilde{\boldsymbol{x}}_k\tilde{r}_{k,\tau}.\end{aligned} \tag{7.6}$$

To balance exploration and exploitation, ConLinUCB selects arms using the upper confidence bound (UCB) strategy

$$a_t = \arg\max_{a\in\mathcal{A}_t} \underbrace{\boldsymbol{x}_a^\top\boldsymbol{\theta}_t}_{\hat{R}_{a,t}} + \underbrace{\alpha_t\left\|\boldsymbol{x}_a\right\|_{\boldsymbol{M}_t^{-1}}}_{C_{a,t}} , \tag{7.7}$$

where $\|\boldsymbol{x}\|_{\boldsymbol{M}} = \sqrt{\boldsymbol{x}^\top \boldsymbol{M}\boldsymbol{x}}$, $\hat{R}_{a,t}$ and $C_{a,t}$ denote the estimated reward and confidence radius of arm a at round t, and

$$\alpha_t = \sqrt{2\log(\frac{1}{\delta}) + d\log(1 + \frac{t + b(t)}{\beta d})} + \sqrt{\beta}\,, \tag{7.8}$$

which comes from the following Lemma 7.3.1.

The ConLinUCB algorithm framework is shown in Alg. 10. The key-term-level interactions take place in line 3-14. At round t, the agent first determines whether conversations are allowed using $b(t)$. When conducting conversations, the agent asks for the user's feedback on $q(t)$ key-terms and uses the feedback to update the parameters. Line 15-20 summarise the arm-level interactions. Based on historical interactions, the agent calculates the estimated $\boldsymbol{\theta}^*$, selects an arm with the largest UCB index, receives the corresponding reward, and updates the parameters accordingly. ConLinUCB only maintains one set of covariance matrix $\boldsymbol{M}_t$ and regressand vector $\boldsymbol{b}_t$, containing the feedback from both arm-level and key-term-level interactions. By doing so, ConLinUCB better leverages the feedback information than ConUCB. Note that ConLinUCB is a general framework with the specified key-term selection strategy $\boldsymbol{\pi}$ to be determined.

7.3.2 ConLinUCB with key-terms from Barycentric Spanner (ConLinUCB-BS)

Based on the ConLinUCB framework, we propose the ConLinUCB-BS algorithm with an explorative key-term selection strategy. Specifically, ConLinUCB-BS selects key-terms from the barycentric spanner $\mathcal{B}$ of the key-term set $\mathcal{K}$, which is an efficient ex-

ploration basis in online learning [250], to conduct explorative conversations. Below is the formal definition of the barycentric spanner for the key-term set $\mathcal{K}$.

Definition 7.1 (Barycentric Spanner of $\mathcal{K}$)**.** *$\mathcal{B} = \{k_1, k_2, ..., k_d\} \subseteq \mathcal{K}$ is a barycentric spanner for $\mathcal{K}$ if for any $k \in \mathcal{K}$, there exists a set of coefficients $\boldsymbol{c} \in [-1, 1]^d$, such that $\tilde{\boldsymbol{x}}_k = \sum_{i=1}^{d} \boldsymbol{c}_i \tilde{\boldsymbol{x}}_{k_i}$.*

We assume that the key-term set $\mathcal{K}$ is finite and $\{\tilde{\boldsymbol{x}}_k\}_{k\in\mathcal{K}}$ span $\mathbb{R}^d$, thus the existence of a barycentric spanner $\mathcal{B}$ of $\mathcal{K}$ is guaranteed [251].

Corresponding vectors in the barycentric spanner are linearly independent. By choosing key-terms from the barycentric spanner, we can quickly explore the *unknown* user preference vector $\boldsymbol{\theta}^*$ in various directions. Based on this reasoning, whenever a conversation is allowed, ConLinUCB-BS selects a key-term

$$k \sim \text{unif}(\mathcal{B}), \tag{7.9}$$

which means sampling a key-term k uniformly at random from the barycentric spanner $\mathcal{B}$ of $\mathcal{K}$. ConLinUCB-BS is completed using the above strategy as $\boldsymbol{\pi}$ in the ConLinUCB framework (Alg. 10). As shown in the following Lemma 7.3.1 and Lemma 7.3.2, in ConLinUCB-BS, the statistical estimation uncertainty shrinks quickly. Additionally, since the barycentric spanner $\mathcal{B}$ of the key-term set $\mathcal{K}$ can be precomputed offline, ConLinUCB-BS is computationally efficient, which is vital for real-time recommendations.

7.3.3 ConLinUCB with key-terms having Max Confidence Radius (ConLinUCB-MCR)

We can further improve ConLinUCB-BS in the following aspects. First, ConLinUCB-BS does not apply in a more general setting where the key-term set $\mathcal{K}$ varies over time since it needs a precomputed barycentric spanner $\mathcal{B}$ of $\mathcal{K}$. Second, as the selection of key-terms is independent of past observations, ConLinUCB-BS does not fully leverage the historical information. For example, suppose the agent is already certain about whether the user favors *sports* based on previous interactions. In that case, it does not need to ask for the user's feedback on the key-term *sports* anymore. To address these issues, we propose the ConLinUCB-MCR algorithm that (i) is applicable when the key-term set $\mathcal{K}$ varies with t and (ii) can adaptively conduct explorative conversations based on historical interactions.

In multi-armed bandits, confidence radius is used to capture whether an arm has been well explored in the interactive history, and it will shrink whenever the arm is selected. Motivated by this, if a key-term has a large confidence radius, it means the system has not sufficiently explored the user's preferences in its related items, indicating that this key-term is explorative. Based on this reasoning, ConLinUCB-MCR selects key-terms with maximal confidence radius to conduct explorative conversations apdaptively. Specifically, when a conversation is allowed at t, ConLinUCB-MCR chooses a key-term as follow

$$k \in \arg\max_{k \in \mathcal{K}_t} \alpha_t \left\| \tilde{\boldsymbol{x}}_k \right\|_{\boldsymbol{M}_t^{-1}} , \tag{7.10}$$

where α_t is defined in Eq. (7.8) and $\mathcal{K}_t \subseteq \mathcal{K}$ denotes the possibly time-varying key-terms set available at round t. ConLinUCB-MCR is completed using the above strategy (Eq. (7.10)) as $\boldsymbol{\pi}$ in ConLinUCB (Alg. 10).

7.3.4 Theoretical Analysis

We give upper bounds of the regret for our algorithms. As a convention, the conversation frequency satisfies $b(t) \leq t$, so we assume $b(t) = b \cdot t$, $b \in (0, 1)$. We leave the proofs of Lemma 7.3.1-7.3.2 and Theorem 7.3.3-7.3.4 to the Appendix due to the space limitation.

The following lemma shows a high probability upper bound of the difference between $\boldsymbol{\theta}_t$ and $\boldsymbol{\theta}^*$ in the direction of the action vector $\boldsymbol{x}_a$ for algorithms based on ConLinUCB.

Lemma 7.3.1. *At $\forall t$, for any $a \in \mathcal{A}$, with probability at least $1 - \delta$ for some $\delta \in (0, 1)$*

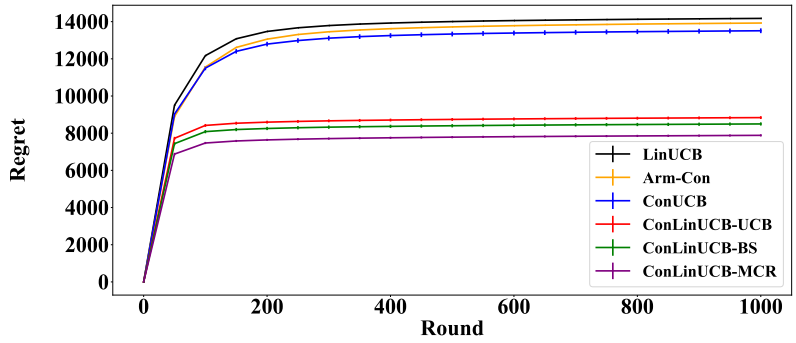
$$\left|\boldsymbol{x}_a^\top (\boldsymbol{\theta}_t - \boldsymbol{\theta}^*)\right| \leq \alpha_t \left\|\boldsymbol{x}_a\right\|_{\boldsymbol{M}_t^{-1}} = C_{a,t},$$

where $\alpha_t = \sqrt{2\log(\frac{1}{\delta}) + d\log(1 + \frac{t+b(t)}{\beta d})} + \sqrt{\beta}$.

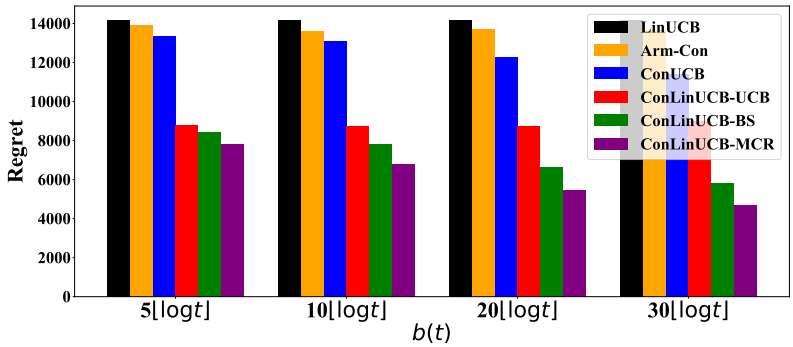
For a barycentric spanner $\mathcal{B}$ of the key-term set $\mathcal{K}$, let

$$\lambda_{\mathcal{B}} \coloneqq \lambda_{\min}(\boldsymbol{E}_{k\sim\text{unif}(\mathcal{B})}[\tilde{\boldsymbol{x}}_k \tilde{\boldsymbol{x}}_k^\top]) > 0\,, \tag{7.11}$$

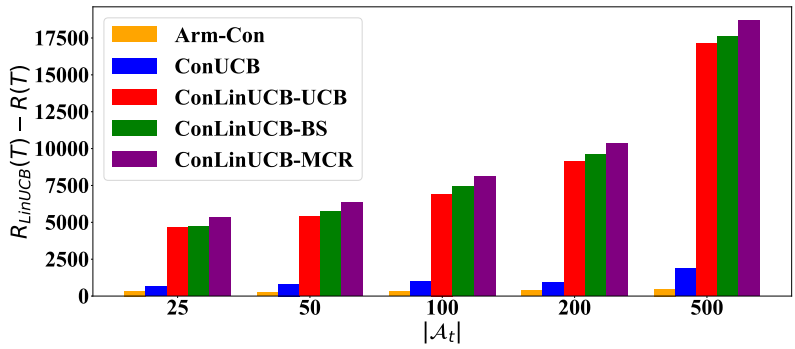
where $\lambda_{\min}(\cdot)$ denotes the minimum eigenvalue of the augment. We can get the following Lemma that gives a high probability upper bound of $\left\|\boldsymbol{x}_a\right\|_{\boldsymbol{M}_t^{-1}}$ for ConLinUCB-BS.



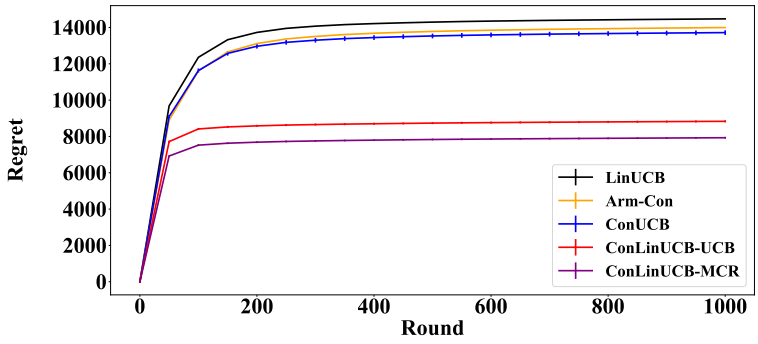
(a) Cumulative regret for fixed key-term set $\mathcal{K}$



(b) Impact of b(t)



(c) Impact of the poolsize $|\mathcal{A}_t|$



(d) Regret for time-varying key-term sets with $|\mathcal{K}_t| = 300$

Figure 7.1: Experimental results on synthetic dataset

Lemma 7.3.2. *For ConLinUCB-BS, $\forall a \in \mathcal{A}$, at $\forall t \geq t_0 = \frac{256}{b\lambda_{\mathcal{B}}^2}\log(\frac{128d}{\lambda_{\mathcal{B}}^2\delta})$, with probability at least $1-\delta$ for $\delta \in (0, \frac{1}{8}]$*

$$\|\boldsymbol{x}_a\|_{\boldsymbol{M}_t^{-1}} \leq \sqrt{\frac{2}{\lambda_{\mathcal{B}} bt}}\,.$$

The following theorem gives a high probability regret upper bound of our ConLinUCB-BS.

Theorem 7.3.3. *With probability at least $1-\delta$ for some $\delta \in (0, \frac{1}{4}]$, the regret $R(T)$ of ConLinUCB-BS satisfies*

$$R(T) \leq 4\sqrt{\frac{2}{b\lambda_{\mathcal{B}}}}\sqrt{T}\left(\sqrt{2\log(\frac{2}{\delta}) + d\log(1+\frac{(1+b)T}{\beta d})} + \sqrt{\beta}\right) + \frac{256}{b\lambda_{\mathcal{B}}^2}\log(\frac{256d}{\lambda_{\mathcal{B}}^2\delta}) + 1\,.$$

Recall that the regret upper bound of ConUCB [62] is

$$R(T) \leq 2\sqrt{2Td\log(1+\frac{\lambda(T+1)}{(1-\lambda)d})}\left(\sqrt{\frac{1-\lambda}{\lambda}} + \sqrt{\frac{1-\lambda}{\lambda\beta}}\sqrt{2\log(\frac{2}{\delta}) + d\log(1+\frac{bT}{\beta d})} + \sqrt{2\log(\frac{2}{\delta}) + d\log(1+\frac{\lambda T}{(1-\lambda)d})}\right),$$

which is of $O(d\sqrt{T}\log T)$. The regret bound of ConLinUCB-BS given in Theorem 7.3.3 is of $O(d\sqrt{T\log T})$ (as $\lambda_{\mathcal{B}}$ is of order $O(\frac{1}{d})$), better than ConUCB by reducing a multiplicative $\sqrt{\log T}$ term.

Next, the following theorem gives a high-probability regret upper bound of ConLinUCB-MCR.

Theorem 7.3.4. *With probability at least* $1-\delta$ *for some* $\delta \in (0,1)$*, the regret* $R(T)$ *of ConLinUCB-MCR satisfies*

$$R(T) \leq 2\sqrt{2Td\log(1+\frac{T+1}{\beta d})} \times \left(\sqrt{\beta}+\sqrt{2\log(\frac{1}{\delta})+d\log(1+\frac{(b+1)T}{\beta d})}\right).$$

Note that the regret upper bound of ConLinUCB-MCR is smaller than ConUCB by reducing some additive terms.

7.4 Experiments on Synthetic Dataset

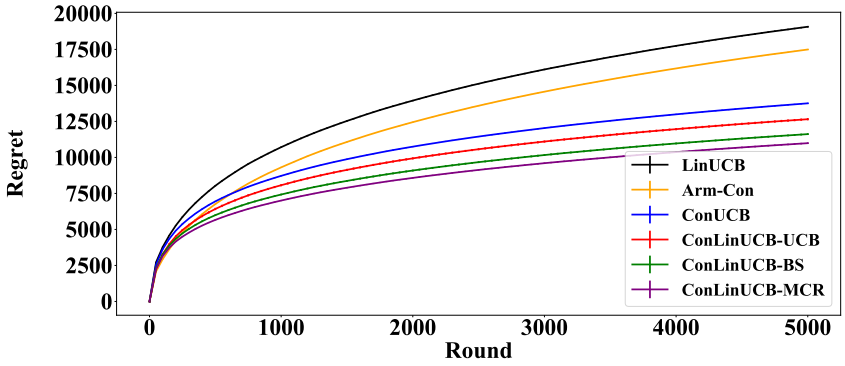
In this section, we show the experimental results on synthetic data. To obtain the offline-precomputed barycentric spanner $\mathcal{B}$, we use the method proposed in [251].

7.4.1 Experimental Settings

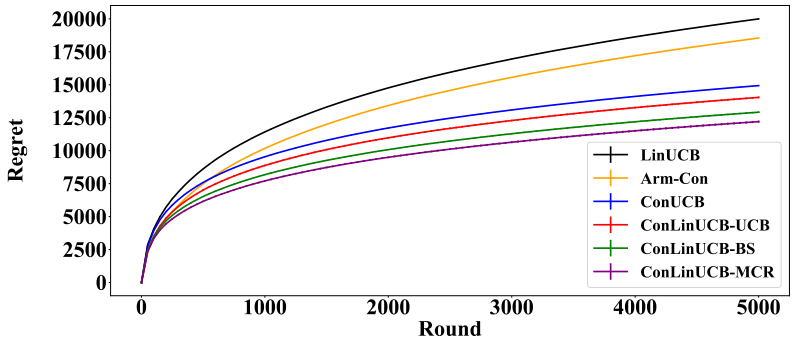
7.4.1.1 Generation of the synthetic dataset.

We create a set of arms $\mathcal{A}$ with $|\mathcal{A}| = 5,000$ arms, and a set of key-terms $\mathcal{K}$ with $|\mathcal{K}| = 500$. We set the dimension of the feature space to be $d = 50$ and the number of users $N_u = 200$.

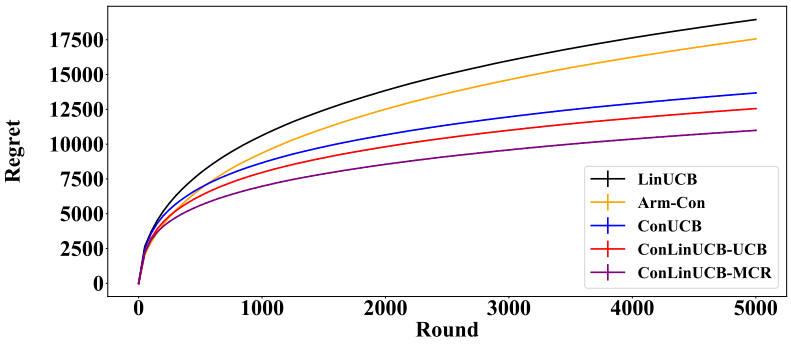
For each user preference vector $\boldsymbol{\theta}_u^*$ and each arm feature vector $\boldsymbol{x}_a$, each entry is generated by independently drawing from the standard normal distribution $\mathcal{N}(0,1)$, and all these vectors are normalized such that $\|\boldsymbol{\theta}_u^*\|_2 = 1$, $\|\boldsymbol{x}_a\|_2 = 1$. The weight matrix $\boldsymbol{W} \triangleq [w_{a,k}]$ is generated as follows: First, for each key-term k, we select an integer $n_k \in [1,10]$ uniformly at random, then randomly



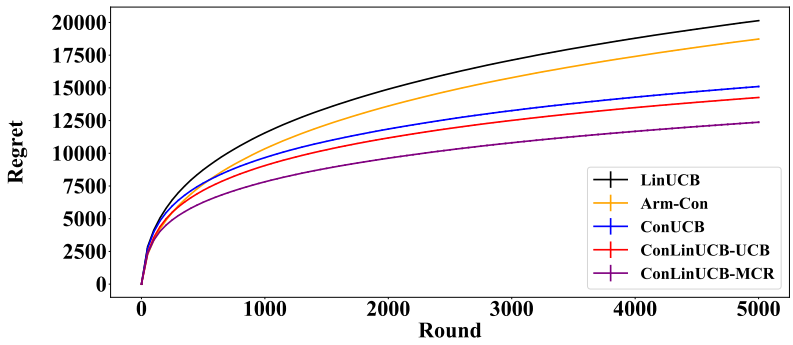
(a) Regret for fixed key-term set $\mathcal{K}$ on Last.FM



(b) Regret for fixed key-term set $\mathcal{K}$ on Movielens



(c) Regret for time-varying $\mathcal{K}_t$ on Last.FM



(d) Regret for time-varying $\mathcal{K}_t$ on Movielens

Figure 7.2: Experimental results on real-word datasets

select a subset of n_k arms $\mathcal{A}_k$ to be the related arms for key-term k; second, for each arm a, if it is related to a set of n_a key-terms $\mathcal{K}_a$, we assign equal weights $w_{a,k} = \frac{1}{n_a}$, $\forall k \in \mathcal{K}_a$. Following [62], the feature vector for each key-term k is computed using $\tilde{\boldsymbol{x}}_k = \sum_{a \in \mathcal{A}} \frac{w_{a,k}}{\sum_{a' \in \mathcal{A}} w_{a',k}} \boldsymbol{x}_a$. The arm-level rewards and key-term-level feedback are generated following Eq. (7.1) and Eq. (7.3).

7.4.1.2 Baselines.

We compare our algorithms with the following baselines:

- LinUCB [41]: A state-of-the-art contextual linear bandit algorithm that selects arms only based on the arm-level feedback without using conversational feedback.
- Arm-Con [61]: A conversational bandit algorithm that conducts conversations on arms without considering key-terms, and uses LinUCB for arm selection.
- ConUCB [62]: The core conversational bandit algorithm that selects a key-term to minimize some estimation error whenever a conversation is allowed.
- ConLinUCB-UCB: An algorithm using a LinUCB-alike method as the key-term selection strategy in our proposed ConLinUCB framework, i.e., choose key-term $k \in \arg\max_{k \in \mathcal{K}_t} \tilde{\boldsymbol{x}}_k^\top \boldsymbol{\theta}_t + \alpha_t \|\tilde{\boldsymbol{x}}_k\|_{\boldsymbol{M}_t^{-1}}$ at round t.

7.4.2 Evaluation Results

This section first shows the results when the key-term set $\mathcal{K}$ is fixed. In this case, we evaluate the regret $R(T)$ for all algo-

Alogrithm	Total time	Total time for selecting arms	Total time for selecting key-terms
ConUCB	11,297	5,217	6,080
ConLinUCB-UCB	5,738	3,060	2,678
ConLinUCB-MCR	4,821	3,030	1,791
ConLinUCB-BS	3,127	3,120	6

Table 7.1: Total runninng time (in seconds) of algorithms on Movielens with $T = 5,000$.

rithms, and we study the impact of the conversation frequency function $b(t)$ and the number of arms $|\mathcal{A}_t|$ available at each round t. When $\mathcal{K}$ varies with time, ConLinUCB-BS does not apply, and we compare the regret of other algorithms. Following [62], we set $T = 1,000$, $b(t) = 5\lfloor \log(t+1) \rfloor$ and $|\mathcal{A}_t| = 50$, unless otherwise stated.

7.4.2.1 Cumulative regret

We run the experiments 10 times and calculate the average regret of all the users for each algorithm. We include $\pm std$ as the error bar, where *std* stands for the *standard deviation*. The results are given in Figure 7.1 (a). First, all other algorithms outperform LinUCB, showing the advantage of conversations. Further, with our proposed ConLinUCB framework, even if we use ConLinUCB-UCB with a simple LinUCB-alike key-term selection strategy, the performance is already better than ConUCB (34.91% improvement), showing more efficient information incorporation. With explorative conversations, ConLinUCB-BS and ConLinUCB-MCR achieve much lower regrets (37.00% and 43.10% improvement over ConUCB respectively), indicating better learning accuracy. ConLinUCB-MCR further leverages historical information to conduct explorative conversations adaptively,

thus achieving the lowest regret.

7.4.2.2 Impact of conversation frequency function $b(t)$

A larger $b(t)$ means the agent can conduct more conversations. We set $b(t) = f_q \cdot \lfloor \log t \rfloor$ and vary f_q to change the conversation frequencies, i.e., $f_q \in \{5, 10, 20, 30\}$. The results are shown in Figure 7.1 (b). With larger $b(t)$, our algorithms have less regret, showing the power of conversations. In all cases, ConLinUCB-BS and ConLinUCB-MCR have lower regrets than ConUCB, and ConLinUCB-MCR performs the best.

7.4.2.3 Impact of $|\mathcal{A}_t|$

We vary $|\mathcal{A}_t|$ to be 25, 50, 100, 200, 500. To clearly show the advantage of our algorithms, we evaluate the difference in regrets between LinUCB and other algorithms, i.e., $R_{\text{LinUCB}}(T) - R(T)$, representing the improved accuracy of the conversational bandit algorithms as compared with LinUCB. Note that the larger $|\mathcal{A}_t|$ is, the harder it is for the algorithm to identify the best arm. Results in Figure 7.1 (c) show that as $|\mathcal{A}_t|$ increases, the advantages of ConLinUCB-BS and ConLinUCB-MCR become more significant. Particularly, when $|\mathcal{A}_t|$=25, ConLinUCB-BS and ConLinUCB-MCR achieve 34.99% and 40.21% improvement over ConUCB respectively; when $|\mathcal{A}_t|$=500, ConLinUCB-BS and ConLinUCB-MCR achieve 50.36% and 53.77% improvement over ConUCB, respectively. In real applications, the size of arm set $|\mathcal{A}_t|$ is usually very large. Therefore, our proposed algorithms are expected to significantly outperform ConUCB in practice.

7.4.2.4 Cumulative regret for time-varying $\mathcal{K}$

This section studies the case when only a subset of key-terms $\mathcal{K}_t \subseteq \mathcal{K}$ are available to the agent at each round t, where ConLinUCB-BS is not applicable as mentioned before. The number of key-terms available at each time t is set to be $|\mathcal{K}_t| = 300$. At round t, 300 key-terms are chosen uniformly at random from $\mathcal{K}$ to form $\mathcal{K}_t$. We evaluate the regret of all algorithms except ConLinUCB-BS. The results are shown in Figure 7.1 (d). We can observe that ConLinUCB-MCR outperforms all baselines and achieves 43.02% improvement over ConUCB.

7.5 Experiments on Real-world Datasets

This section shows the experimental results on two real-world datasets, Last.FM and Movielens. The baselines, generations of arm-level rewards and key-term-level feedback, and the computation method of the barycentric spanner are the same as in the last section. Following the experiments on real data of [62], we set $T = 5,000$, $b(t) = 5\lfloor \log(t+1) \rfloor$ and $|\mathcal{A}_t| = 50$, unless otherwise stated.

7.5.1 Experiment Settings

7.5.1.1 Last.FM and Movielens datasets [252]

Last.FM is a dataset for music artist recommendations containing 186,479 interaction records between 1,892 users and 17,632 artists. Movielens is a dataset for movie recommendation containing 47,957 interaction records between 2,113 users and 10,197

movies.

7.5.1.2 Generation of the data

The data is generated following [136, 62, 180]. We treat each music artist and each movie as an arm. For both datasets, we extract $|\mathcal{A}| = 2,000$ arms with the most assigned tags by users and $N_u = 500$ users who have assigned the most tags. For each arm, we keep at most 20 tags that are related to the most arms, and consider them as the associated key-terms of the arm. All the kept key-terms associated with the arms form the key-term set $\mathcal{K}$. The number of key-terms for Last.FM is $|\mathcal{K}| = 2,726$ and that for Movielens is $5,585$. The weights of all key-terms related to the same arm are set to be equal. Based on the interactive recordings, the user feedback is constructed as follows: if the user has assigned tags to the item, the feedback is 1, otherwise the feedback is 0. To generate the feature vectors of users and arms, following [136], we construct a feedback matrix $\boldsymbol{F} \in \mathbb{R}^{N_u \times N}$ based on the above user feedback, and decompose it using the singular-value decomposition (SVD): $\boldsymbol{F} = \boldsymbol{\Theta} \boldsymbol{S} \boldsymbol{X}^\top$, where $\boldsymbol{\Theta} = (\boldsymbol{\theta}_{\boldsymbol{u}}^*)$, $u \in [N_u]$ and $\boldsymbol{X} = (\boldsymbol{x}_a)$, $a \in [N]$. We select $d = 50$ dimensions with highest singular values in $\boldsymbol{S}$. Following [62], feature vectors of key-terms are calculated using $\tilde{\boldsymbol{x}}_k = \sum_{a \in \mathcal{A}} \frac{w_{a,k}}{\sum_{a' \in \mathcal{A}} w_{a',k}} \boldsymbol{x}_a$. The arm-level rewards and key-term-level feedback are then generated following Eq. (7.1) and Eq. (7.3).

7.5.2 Evaluation Results

This section first shows the results on both datasets in two cases: $\mathcal{K}$ is fixed and $\mathcal{K}$ is varying with time t. We also compare the running time of all algorithms on the Movielens dataset, since it has more key-terms than Last.FM.

7.5.2.1 Cumulative regret

We run the experiments 10 times and calculate the average regret of all the users over $T = 5,000$ rounds on the fixed generated datasets. The randomness of experiments comes from the randomly chosen $\mathcal{A}_t$ (also $\mathcal{K}_t$ in the varying key-term set case) and the randomness in the ConLinUCB-BS algorithm. We also include $\pm std$ as the error bar. For the time-varying key-term sets case, we set $|\mathcal{K}_t| = 1,000$ and randomly select $|\mathcal{K}_t|$ key-terms from $\mathcal{K}$ to form $\mathcal{K}_t$ at round t. Results on Last.FM and Movielens for fixed key-term set are shown in Figure 7.2 (a) and Figure 7.2 (b). On both datasets, the regrets of ConLinUCB-BS and ConLinUCB-MCR are much smaller than ConUCB (13.28% and 17.12% improvement on Last.FM, 13.08% and 16.93% improvement on Movielens, respectively) and even the simple ConLinUCB-UCB based on our ConLinUCB framework outperforms ConUCB. Results on Last.FM and Movielens for varying key-term sets are given in Figure 7.2 (c) and Figure 7.2 (d). ConLinUCB-MCR performs much better than ConUCB on both datasets (19.66% and 17.85% improvement on Last.FM and Movielens respectively).

7.5.2.2 Running time

We evaluate the running time of all the conversational bandit algorithms on the representative Movielens dataset to compare their computational efficiency. For clarity, we report the total running time for selecting arms and key-terms. We set $T = 5,000$ and the results are summarized in Table 7.1. It is clear that our algorithms cost much less time in both key-term selection and arm selection than ConUCB. Specifically, the improvements of total running time over ConUCB are 72.32% for ConLinUCB-BS and 57.32% for ConLinUCB-MCR. The main reason is that our algorithms estimate the *unknown* user preference vector in one single step, whereas ConUCB does it in two separate steps as mentioned before. For ConLinUCB-BS, the time costed in the key-term selection is almost negligible, since it just randomly chooses a key-term from the precomputed barycentric spanner whenever a conversation is allowed.

Algorithm 10 General ConLinUCB framework

Input: graph$(\mathcal{A}, \mathcal{K}, \boldsymbol{W})$, conversation frequency function $b(t)$, key-term selection strategy $\boldsymbol{\pi}$.

Initialization: $\boldsymbol{M}_0 = \beta \boldsymbol{I}$, $\boldsymbol{b}_0 = \mathbf{0}$.
for *t = 1 to T* **do**
  if $b(t) - b(t-1) > 0$ **then**
    $q(t) = \lfloor b(t) - b(t-1) \rfloor$;
    while $q(t) > 0$ **do**
      Select a key-term $k \in \mathcal{K}$ using the specified key-term selection strategy $\boldsymbol{\pi}$ (e.g., Eq. (7.9) for ConLinUCB-BS and Eq. (7.10) for ConLinUCB-MCR), and query the user's preference over it;
      Receive the user's feedback $\tilde{r}_{k,t}$;
      $\boldsymbol{M}_t = \boldsymbol{M}_{t-1} + \tilde{\boldsymbol{x}}_k \tilde{\boldsymbol{x}}_k^\top$;
      $\boldsymbol{b}_t = \boldsymbol{b}_{t-1} + \tilde{\boldsymbol{x}}_k \tilde{r}_{k,t}$;
      $q(t) -= 1$;
    end
  else
    $\boldsymbol{M}_t = \boldsymbol{M}_{t-1}$, $\boldsymbol{b}_t = \boldsymbol{b}_{t-1}$;
  end
  $\boldsymbol{\theta}_t = \boldsymbol{M}_t^{-1} \boldsymbol{b}_t$;
  Select $a_t = \arg\max_{a \in \mathcal{A}_t} \boldsymbol{x}_a^\top \boldsymbol{\theta}_t + \alpha_t \left\| \boldsymbol{x}_a \right\|_{\boldsymbol{M}_t^{-1}}$;
  Ask the user's preference on arm a_t and receive the reward $r_{a_t,t}$;
  $\boldsymbol{M}_t = \boldsymbol{M}_{t-1} + \boldsymbol{x}_{a_t} \boldsymbol{x}_{a_t}^\top$;
  $\boldsymbol{b}_t = \boldsymbol{b}_{t-1} + \boldsymbol{x}_{a_t} r_{a_t,t}$;
end

Chapter 8

Variance-Dependent Regret Bounds for Non-stationary Linear Bandits

We investigate the non-stationary stochastic linear bandit problem where the reward distribution evolves each round. Existing algorithms characterize the non-stationarity by the *total variation budget* B_K, which is the summation of the change of the consecutive feature vectors of the linear bandits over K rounds. However, such a quantity only measures the non-stationarity with respect to the expectation of the reward distribution, which makes existing algorithms sub-optimal under the general non-stationary distribution setting. In this work, we propose algorithms that utilize the *variance* of the reward distribution as well as the B_K, and show that they can achieve tighter regret upper bounds. Specifically, we introduce two novel algorithms: Restarted WeightedOFUL$^+$ and Restarted SAVE$^+$. These algorithms address cases where the variance information of the rewards is known and unknown, respectively. Notably, when the total variance V_K is much smaller

than K, our algorithms outperform previous state-of-the-art results on non-stationary stochastic linear bandits under different settings. Experimental evaluations further validate the superior performance of our proposed algorithms over existing works. This chapter is based on our publication [10].

8.1 Introduction

In this work, we study non-stationary stochastic bandits, which is a generalization of the classical stationary stochastic bandits, where the reward distribution is non-stationary. The intuition about the non-stationary setting comes from real-world applications such as dynamic pricing and ads allocation, where the environment changes rapidly and deviates significantly from stationarity [183, 253]. Most of the existing works in stochastic bandits consider a stationary setting where the goal of the agent is to minimize the *static regret*, *i.e.*, the summation of suboptimality gaps between the agent's selected arm and the fixed, time-independent best arm that maximizes the expectation of the reward distribution. In contrast, for the non-stationary setting, the emphasis shifts to minimizing the *dynamic regret*, which represents the gap between the cumulative reward of selecting the time-dependent optimal arm at each time and that of the learner. As we can always treat a stationary bandit instance as a special case of the non-stationary bandit instance, designing algorithms that work well under the non-stationary setting is significantly more challenging.

There have been a series of works aiming to minimize the *dy-*

namic regret for non-stationary stochastic bandits, such as Multi-Armed Bandits (MAB) [183, 184, 185, 186], linear bandits [253, 187, 192, 194, 254], general function approximation [255, 191, 195], and the even more challenging reinforcement learning (RL) setting [256, 257, 258, 259, 194]. In this work, we mainly consider the linear bandit setting, where each arm is a contextual vector, and the expected reward of each arm is assumed to be the linear product of the arm with an unknown feature vector. Most existing *dynamic regret* results for non-stationary linear bandits depend on both the *non-stationarity measurement* and the number of interaction rounds. Specifically, assume K is the total number of rounds, and for each $k \in [K]$, $\mathbf{x}$ is one of the arms, $\boldsymbol{\theta}_k$ and $\boldsymbol{\theta}_{k+1}$ are the feature vectors at k and $k+1$ rounds, satisfying $\|\mathbf{x}\|_2 \leq 1$. Then, the non-stationarity measurement is often defined as the summation of the changes in the mean of the reward distribution, which is

$$B_K := \sum_{k=1}^{K} \max_{\mathbf{x} \in \mathbb{R}^d} |\langle \mathbf{x}, \boldsymbol{\theta}_k - \boldsymbol{\theta}_{k+1} \rangle| = \sum_{k=1}^{K} \|\boldsymbol{\theta}_k - \boldsymbol{\theta}_{k+1}\|_2 \,. \tag{8.1}$$

Existing works for non-stationary linear bandits [188, 260, 261, 257, 253, 192] achieved a regret upper bound of $\tilde{O}(d^{7/8} B_K^{\frac{1}{4}} K^{\frac{3}{4}})$, where d is the problem dimension. A recent work by [194] proposed a black-box reduction method that can achieve a regret upper bound of $\tilde{O}(d B_K^{\frac{1}{3}} K^{\frac{2}{3}})$ in the setting with a fixed arm set across all rounds. Such regret bounds clearly demonstrate that regret grows as long as the non-stationarity grows, which is aligned with intuition.

Although existing works clearly demonstrate the relationship

between the B_K and the regret, we claim that it is not sufficient for us to fully characterize the non-stationary level of the reward distributions. Consider applications such as hyperparameter tuning in physical systems, the noise distribution may highly depend on the evaluation point since the measurement noise often largely varies with the chosen parameter settings [202]. For linear bandits, such examples suggest that the non-stationarity not only consists of the change of the mean of the distribution, but also the variance of the distribution. However, none of the previous works on non-stationary linear bandits considered how to leverage the variance information to improve regret bounds in the above heteroscedastic noise setting. Therefore, an open question arises:

Can we design even better algorithms for non-stationary linear bandits by considering its variance information?

In this paper, we answer this question affirmatively. We assume that at the k-th round, the reward distribution of an arm $\mathbf{x}$ satisfies $r_k \sim \langle \boldsymbol{\theta}_k, \mathbf{x} \rangle + \epsilon_k$, where ϵ_k is a zero-mean noise variable with variance σ_k^2. Our contributions are:

- We establish the first variance-dependent regret lower bound for non-stationary linear bandits. This result captures the interplay between non-stationarity and variance, which is not addressed in existing literature for non-stationary linear bandits.

- For the case where the reward variance σ_k^2 at round k can be observed and the *total variation budget* B_K is known, we propose the Restarted-WeightedOFUL$^+$ algorithm, which uses

variance-based weighted linear regression to deal with heteroscedastic noises [203, 76] and a restarted scheme to forget some historical data to hedge against the non-stationarity. We prove that the regret upper bound of Restarted-WeightedOFUL$^+$ is $\tilde{O}(d^{7/8}(B_K V_K)^{1/4}\sqrt{K} + d^{5/6}B_K^{1/3}K^{2/3})$. Our regret surpasses the best result for non-stationary linear bandits $\tilde{O}(dB_K^{1/3}K^{2/3})$ [194] when the total variance $V_K = \tilde{O}(1)$ is small, which indicates that additional variance information benefits non-stationary linear bandit algorithms.

- For the case where the reward variance σ_k^2 is unknown but the total variance V_K and variation budget B_K are known, we propose the Restarted-SAVE$^+$ algorithm. It maintains a multi-layer weighted linear regression structure with carefully-designed weight within each layer to handle the unknown variances [80]. We prove that Restarted-SAVE$^+$ can achieve a regret upper bound of $\tilde{O}(d^{\frac{4}{5}}V_K^{\frac{2}{5}}B_K^{\frac{1}{5}}K^{\frac{2}{5}} + d^{\frac{2}{3}}B_K^{\frac{1}{3}}K^{\frac{2}{3}})$. Specifically, when $V_K = \tilde{O}(1)$, our regret is also better than the existing best result $\tilde{O}(dB_K^{1/3}K^{2/3})$ [194], which again verifies the effect of the variance information.

- Lastly, we propose Restarted-SAVE$^+$-BOB for the case where both the reward variance σ_k^2 and B_K are unknown. Restarted-SAVE$^+$-BOB equips a *bandit-over-bandit* (BOB) framework to handle the unknown B_K [187], and also maintains a multi-layer structure as Restarted-SAVE$^+$. We show that Restarted-SAVE$^+$-BOB achieves a regret upper bound of $\tilde{O}(d^{\frac{4}{5}}V_K^{\frac{2}{5}}B_K^{\frac{1}{5}}K^{\frac{2}{5}} + d^{\frac{2}{3}}B_K^{\frac{1}{3}}K^{\frac{2}{3}} + d^{\frac{1}{5}}K^{\frac{7}{10}})$, and it behaves the same as Restarted-SAVE$^+$ when $V_K = \tilde{O}(1)$ and $B_K = \Omega(d^{-14}K^{1/10})$.

- We also conduct experimental evaluations to validate the outperformance of our proposed algorithms over existing works.

Notation We use lower case letters to denote scalars, and use lower and upper case bold face letters to denote vectors and matrices respectively. We denote by $[n]$ the set $\{1, \ldots, n\}$. For a vector $\mathbf{x} \in \mathbb{R}^d$ and a positive semi-definite matrix $\mathbf{\Sigma} \in \mathbb{R}^{d \times d}$, we denote by $\|\mathbf{x}\|_2$ the vector's Euclidean norm and define $\|\mathbf{x}\|_{\mathbf{\Sigma}} = \sqrt{\mathbf{x}^\top \mathbf{\Sigma} \mathbf{x}}$. For two positive sequences $\{a_n\}$ and $\{b_n\}$ with $n = 1, 2, \ldots$, we write $a_n = O(b_n)$ if there exists an absolute constant $C > 0$ such that $a_n \leq C b_n$ holds for all $n \geq 1$ and write $a_n = \Omega(b_n)$ if there exists an absolute constant $C > 0$ such that $a_n \geq C b_n$ holds for all $n \geq 1$. We use $\tilde{O}(\cdot)$ to further hide the polylogarithmic factors.

8.2 Problem Setting

We consider a heteroscedastic variant of the classic non-stationary linear contextual bandit problem. Let K be the total number of rounds. At each round $k \in [K]$, the learner interacts with the environment as follows: (1) the environment generates an arbitrary arm set $\mathcal{D}_k \subseteq \mathbb{R}^d$ where each element represents a feasible arm for the learner to choose, and also generates an *unknown* feature vector $\boldsymbol{\theta}_k$; (2) the leaner observes $\mathcal{D}_k$ and selects $\mathbf{a}_k \in \mathcal{D}_k$; (3) the environment generates the stochastic noise ϵ_k and reveals the stochastic reward $r_k = \langle \boldsymbol{\theta}_k, \mathbf{a}_k \rangle + \epsilon_k$ to the leaner. We assume that for all $k \geq 1$ and all $\mathbf{a} \in \mathcal{D}_k$, $\langle \mathbf{a}, \boldsymbol{\theta}_k \rangle \in [-1, 1]$, $\|\boldsymbol{\theta}_k\|_2 \leq B$, $\|\mathbf{a}\|_2 \leq A$.

Following [203, 80], we assume the following condition on the

Model	Algorithm	Regret	Variance -Dependent	Varying Arm Set	Require B_K
Linear Bandit	SW-UCB [253]	$\tilde{O}(d^{\frac{7}{8}}B_K^{\frac{1}{4}}K^{\frac{3}{4}})$	No	Yes	Yes
	BOB [253]	$\tilde{O}\Big(d^{\frac{7}{8}}B_K^{\frac{1}{4}}K^{\frac{3}{4}}\Big)$	No	Yes	No
	RestartUCB [192]	$\tilde{O}\Big(d^{\frac{7}{8}}B_K^{\frac{1}{4}}K^{\frac{3}{4}}\Big)$	No	Yes	Yes
	RestartUCB-BOB [192]	$\tilde{O}\Big(d^{\frac{7}{8}}B_K^{\frac{1}{4}}K^{\frac{3}{4}}\Big)$	No	Yes	No
	LB-WeightUCB [254]	$\tilde{O}\Big(d^{\frac{3}{4}}B_K^{\frac{1}{4}}K^{\frac{3}{4}}\Big)$	No	Yes	Yes
	MASTER + OFUL [194]	$\tilde{O}\Big(dB_K^{\frac{1}{3}}K^{\frac{2}{3}}\Big)$	No	No	No
	Restarted-WeightedOFUL$^+$ **(Ours)**	$\tilde{O}\Big(d^{\frac{7}{8}}(B_KV_K)^{\frac{1}{4}}K^{\frac{1}{2}} + d^{\frac{5}{6}}B_K^{\frac{1}{3}}K^{\frac{2}{3}}\Big)$	Yes	Yes	Yes
	Restarted SAVE$^+$ **(Ours)**	$\tilde{O}\Big(d^{\frac{4}{5}}V_K^{\frac{2}{5}}B_K^{\frac{1}{5}}K^{\frac{2}{5}} + d^{\frac{2}{3}}B_K^{\frac{1}{3}}K^{\frac{2}{3}}\Big)$	Yes	Yes	Yes
	Restarted SAVE$^+$-BOB **(Ours)**	$\tilde{O}\Big(d^{\frac{4}{5}}V_K^{\frac{2}{5}}B_K^{\frac{1}{5}}K^{\frac{2}{5}} + d^{\frac{2}{3}}B_K^{\frac{1}{3}}K^{\frac{2}{3}} + d^{\frac{1}{5}}K^{\frac{7}{10}}\Big)$	Yes	Yes	No
	Lower Bound **(Ours)**	$\tilde{\Omega}\Big(d^{2/3}B_K^{1/3}V_K^{1/3}K^{1/3} \wedge V_K + \sqrt{B_KK}\Big)$	Yes	Yes	-
MAB	Rerun-UCB-V [186]	$\tilde{O}\Big(\lvert\mathcal{A}\rvert^{\frac{2}{3}}B_K^{\frac{1}{3}}V_K^{\frac{1}{3}}K^{\frac{1}{3}} + \lvert\mathcal{A}\rvert^{\frac{1}{2}}B_K^{\frac{1}{2}}K^{\frac{1}{2}}\Big)$	Yes	No	Yes
	Lower Bound [186]	$\tilde{\Omega}\Big(B_K^{\frac{1}{3}}V_K^{\frac{1}{3}}K^{\frac{1}{3}} + B_K^{\frac{1}{2}}K^{\frac{1}{2}}\Big)$	Yes	No	-

Table 8.1: Comparison of non-stationary bandits in terms of regret guarantee. K is the total rounds, d is the problem dimension for linear bandits, B_K is the *total variation budget* defined in Section 8.2 (for the MAB setting, $B_K = \sum_{k=1}^{K}\|\mu_k - \mu_{k+1}\|_\infty$, where μ_k is the mean of the reward distribution at round k), V_K is the *total variance* defined in Section 8.2, $|\mathcal{A}|$ is the number of arms for MAB.

random noise ϵ_k at each round k:

$$\mathbb{P}\left(|\epsilon_k| \leq R\right) = 1, \quad \mathbb{E}[\epsilon_k|\mathbf{a}_{1:k}, \epsilon_{1:k-1}] = 0,$$
$$\mathbb{E}[\epsilon_k^2|\mathbf{a}_{1:k}, \epsilon_{1:k-1}] \leq \sigma_k^2. \tag{8.2}$$

Following [253, 187, 188, 192], we assume the summation of ℓ_2 differences of consecutive $\boldsymbol{\theta}_k$'s is upper bounded by the *total vari-*

ation budget B_K, *i.e.*, $\sum_{k=1}^{K-1} \|\boldsymbol{\theta}_{k+1} - \boldsymbol{\theta}_k\|_2 \leq B_K$, where the $\boldsymbol{\theta}_k$'s can be adversarially chosen by an oblivious adversary. We also assume that the *total variance* is upper bounded by V_K, which is $\sum_{k=1}^{K} \sigma_k^2 \leq V_K$. The goal of the agent is to minimize the *dynamic regret* defined as follows: $\text{Regret}(K) = \sum_{k\in[K]} \big(\langle \mathbf{a}_k^*, \boldsymbol{\theta}_k\rangle - \langle \mathbf{a}_k, \boldsymbol{\theta}_k\rangle\big)$, where $\mathbf{a}_k^* = \text{argmax}_{\mathbf{a}\in\mathcal{D}_k}\langle \mathbf{a}, \boldsymbol{\theta}_k\rangle$ is the optimal arm at round k with the highest expected reward.

8.3 Lower Bound

In this section, we establish a novel variance-dependent regret lower bound for non-stationary linear bandits, which reveals new insights into the problem structure.

Theorem 8.3.1. *Given $K > 0$. For any bandit algorithm there exists $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_K$ satisfying the problem setting denoted in Section 8.2, such that*

$$\begin{aligned} &\textit{Regret}(K) \\ &\geq \Omega(\min\{d^{2/3}B_K^{1/3}V_K^{1/3}K^{1/3}, V_K\} + \sqrt{B_K K}). \end{aligned}$$

Proof. See Appendix A.6.3. □

Remark 14. *Note that [187] proposed a lower bound of $\Omega(d^{2/3}B_K^{1/3}K^{2/3})$ for general non-stationary linear bandits. However, their result applies only to cases without the variance restriction V_K, making it inapplicable to our setting.*

Theorem 8.3.1 represents the first variance-dependent regret lower bound specifically tailored for non-stationary linear bandits.

Algorithm 11 Restarted-WeightedOFUL$^+$

Require: Regularization parameter $\lambda > 0$; B, an upper bound on the ℓ_2-norm of $\boldsymbol{\theta}_k$ for all $k \in [K]$; confidence radius $\hat{\beta}_k$, variance parameters α, γ; restart window size w.

$\hat{\boldsymbol{\Sigma}}_1 \leftarrow \lambda \mathbf{I}$, $\hat{\mathbf{b}}_1 \leftarrow \mathbf{0}$, $\hat{\boldsymbol{\theta}}_1 \leftarrow \mathbf{0}$, $\hat{\beta}_1 = \sqrt{\lambda} B$

for $k = 1, \ldots, K$ **do**

  if $k \% w == 0$ **then**

    $\hat{\boldsymbol{\Sigma}}_k \leftarrow \lambda \mathbf{I}$, $\hat{\mathbf{b}}_k \leftarrow \mathbf{0}$, $\hat{\boldsymbol{\theta}}_k \leftarrow \mathbf{0}$, $\hat{\beta}_k = \sqrt{\lambda} B$

  end if

  Observe $\mathcal{D}_k$ and choose $\mathbf{a}_k \leftarrow \operatorname{argmax}_{\mathbf{a} \in \mathcal{D}_k} \langle \mathbf{a}, \boldsymbol{\theta}_k \rangle + \hat{\beta}_k \|\mathbf{a}_k\|_{\hat{\boldsymbol{\Sigma}}_k^{-1}}$

  Observe (r_k, σ_k), set $\bar{\sigma}_k$ as

$$\bar{\sigma}_k \leftarrow \max\{\sigma_k, \alpha, \gamma \|\mathbf{a}_k\|^{1/2}_{\hat{\boldsymbol{\Sigma}}_k^{-1}}\} \tag{8.3}$$

  $\hat{\boldsymbol{\Sigma}}_{k+1} \leftarrow \hat{\boldsymbol{\Sigma}}_k + \mathbf{a}_k \mathbf{a}_k^\top / \bar{\sigma}_k^2$, $\hat{\mathbf{b}}_{k+1} \leftarrow \hat{\mathbf{b}}_k + r_k \mathbf{a}_k / \bar{\sigma}_k^2$, $\hat{\boldsymbol{\theta}}_{k+1} \leftarrow \hat{\boldsymbol{\Sigma}}_{k+1}^{-1} \hat{\mathbf{b}}_{k+1}$

end for

The bound highlights the inherent complexity of balancing variance and non-stationarity, offering a foundation for future work aimed at designing algorithms with matching upper bounds. Notably, our result improves the existing variance-dependent lower bound $\Omega(B_K^{1/3} V_K^{1/3} K^{1/3} + B_K^{1/2} K^{1/2})$ [186] by a factor of $d^{2/3}$ for the linear bandits setting.

8.4 Non-stationary Linear Contextual Bandit with Known Variance

In this section, we introduce our Algorithm 11 under the setting where the variance σ_k^2 at k-th iteration is known to the agent in prior. We start from WeightedOFUL$^+$ [76], an *weighted ridge*

regression-based algorithm for heteroscedastic linear bandits under the stationary reward assumption. For our non-stationary linear bandit setting where $\boldsymbol{\theta}_k$ is changing over the round k, WeightedOFUL$^+$ aims to build an $\hat{\boldsymbol{\theta}}_k$ which estimates the feature vector $\boldsymbol{\theta}_k$ by using the solution to the following regression problem:

$$\hat{\boldsymbol{\theta}}_k \leftarrow \arg\min_{\boldsymbol{\theta}} \sum_{t=1}^{k-1} \bar{\sigma}_t^{-2}(\langle \boldsymbol{\theta}, \mathbf{a}_t \rangle - r_t)^2 + \lambda \|\boldsymbol{\theta}\|_2^2, \tag{8.4}$$

where the weight is defined as in (8.3). After obtaining $\hat{\boldsymbol{\theta}}_k$, WeightedOFUL$^+$ chooses arm $\mathbf{a}_k$ by maximizing the upper confidence bound (UCB) of $\langle \mathbf{a}, \hat{\boldsymbol{\theta}} \rangle$, with an exploration bonus $\hat{\beta}_k \|\mathbf{a}_k\|_{\hat{\boldsymbol{\Sigma}}_k^{-1}}$, where $\hat{\boldsymbol{\Sigma}}_k$ is the covariance matrix over $\mathbf{a}_k$. The weight $\bar{\sigma}_k^2$ is introduced to balance the different past examples based on their reward variance σ_k^2, and such a strategy has been proved as a state-of-the-art algorithm for the stationary heteroscedastic linear bandits [76]. However, the non-stationary nature of our setting prevents us from directly using $\hat{\boldsymbol{\theta}}_k$ defined in (8.4) as an estimate to $\boldsymbol{\theta}$. Therefore, inspired by the *restarting* strategy which has been adopted by previous algorithms for non-stationary linear bandits [192], we propose Restarted-WeightedOFUL$^+$, which periodically restarts itself and runs WeightedOFUL$^+$ as its submodule. The restart window size is set as w, which is used to balance the nonstationarity and the total regret and will be fine-tuned in the next steps. Combined with the restart window size w, we set $\{\hat{\beta}_k\}_{k\geq 1}$ to

$$\hat{\beta}_k = 12\sqrt{d\log(1+\frac{(k\%w)A^2}{\alpha^2 d\lambda})\log(32(\log(\frac{\gamma^2}{\alpha}+1)\frac{(k\%w)^2}{\delta})}$$

$$+ 30\log(32(\log(\frac{\gamma^2}{\alpha})+1)\frac{(k\%w)^2}{\delta})\frac{R}{\gamma^2} + \sqrt{\lambda}B. \tag{8.5}$$

We now propose the theoretical guarantee for Algorithm 8. The following key lemma shows how nonstationarity affects our estimation of the reward of each arm.

Lemma 8.4.1. *Let $0 < \delta < 1$. Then with probability at least $1-\delta$, for any action $\mathbf{a} \in \mathbb{R}^d$, we have*

$$|\mathbf{a}^\top(\hat{\boldsymbol{\theta}}_k - \boldsymbol{\theta}_k)| \leq \underbrace{\frac{A^2}{\alpha}\sqrt{\frac{dw}{\lambda}} \sum_{t=w\cdot\lfloor k/w\rfloor+1}^{k-1} \|\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t+1}\|_2}_{\textit{Drifting term}} + \underbrace{\hat{\beta}_k \|\mathbf{a}\|_{\hat{\boldsymbol{\Sigma}}_k^{-1}}}_{\textit{Stochastic term}}.$$

Proof. See Appendix A.6.4 for the full proof. □

Here we provide a proof sketch of Lemma 8.4.1 to show the technical challenge we need to overcome. Without loss of generality, we prove the lemma for $k \in [1, w]$. We have

$$\begin{aligned}|\mathbf{a}^\top(\hat{\boldsymbol{\theta}}_k - \boldsymbol{\theta}_k)| \leq & \left|\mathbf{a}^\top \hat{\boldsymbol{\Sigma}}_k^{-1} \sum_{t=1}^{k-1} \frac{\mathbf{a}_t \mathbf{a}_t^\top}{\bar{\sigma}_t^2}(\boldsymbol{\theta}_t - \boldsymbol{\theta}_k)\right| \\ & + \|\mathbf{a}\|_{\hat{\boldsymbol{\Sigma}}_k^{-1}} \| \sum_{t=1}^{k-1} \frac{\mathbf{a}_t \epsilon_t}{\bar{\sigma}_t^2} \|_{\hat{\boldsymbol{\Sigma}}_k^{-1}} + \sqrt{\lambda} B \|\mathbf{a}\|_{\hat{\boldsymbol{\Sigma}}_k^{-1}},\end{aligned} \tag{8.6}$$

For the first term, it gets involved by the nonstationarity of $\boldsymbol{\theta}_k$. By rearranging the summation orders and several calculation steps, we have

$$\left|\mathbf{a}^\top \hat{\mathbf{\Sigma}}_k^{-1} \sum_{t=1}^{k} \frac{\mathbf{a}_t \mathbf{a}_t^\top}{\bar{\sigma}_t^2} (\boldsymbol{\theta}_t - \boldsymbol{\theta}_k)\right| \leq \sum_{t=1}^{k-1} |\mathbf{a}^\top \hat{\mathbf{\Sigma}}_k^{-1} \frac{\mathbf{a}_t}{\bar{\sigma}_t}| \cdot \|\frac{\mathbf{a}_t}{\bar{\sigma}_t}\|_2$$
$$\cdot \|\sum_{s=t}^{k-1} (\boldsymbol{\theta}_s - \boldsymbol{\theta}_{s+1})\|_2 \leq \frac{A^2}{\alpha} \sqrt{\frac{dw}{\lambda}} \sum_{s=1}^{k-1} \|\boldsymbol{\theta}_s - \boldsymbol{\theta}_{s+1}\|_2 \ ,$$

We would like to highlight the subtleties in both our algorithm design and analysis to get the desired improvement. First, from here, we can see the necessity of introducing α in the design of $\bar{\sigma}_k$ in Eq.(8.3), which makes it possible to upper bound $\bar{\sigma}_k^{-1}$ and get a tunable α in the drifting term, which can subsequently be used to optimize the regret bound. Second, we show that it is essential to split the term $\bar{\sigma}_t^{-2}$ as how we did. Only by doing that can we bound the $\sum_{t=1}^{s} \frac{\mathbf{a}_t}{\bar{\sigma}_t}^\top \hat{\mathbf{\Sigma}}_k^{-1} \frac{\mathbf{a}_t}{\bar{\sigma}_t}$ term by d with the elliptical potential lemma. Otherwise, we can get a $1/\alpha^2$ term rather than the A/α term, which will hurt the final regret bound. For the second term in Eq.(8.6), a vanilla way to control it is adopting a self-normalized concentration inequality from [43]. However, it can not utilize variance information, but just the magnitude of the noise, which fails to get a tight bound with the variance information. Inspired by [76, 203, 80], we adapt a variance-adaptive concentration inequality in Theorem A.6.3 to get a tighter bound. Similar arguments also hold for the proof of Theorem 8.5.1 for the unknown variance case. We refer to Appendix A.6.4 for the full proof. Lemma 8.4.1 suggests that under the non-stationary setting, the difference between the true expected reward and our estimated reward will be upper bounded by two separate terms. The first drifting term characterizes the

error caused by the non-stationary environment, and the second stochastic term characterizes the error caused by the estimation of the stochastic environment. Note thata similar bound has also been discovered in [257]. We want to emphasize that our bound differs from existing ones in 1) an additional variance parameter α in the drifting term, and 2) a weighted convariance matrix $\hat{\mathbf{\Sigma}}$ rather than a vanilla convariance matrix.

Next we present our main theorem.

Theorem 8.4.2. *Let $0 < \delta < 1$. By treating A, λ, B, R as constants and setting $\gamma^2 = R/\sqrt{d}$, with probability at least $1-\delta$, the regret of Restarted-WeightedOFUL$^+$ is bounded by*

$$\begin{aligned} \textit{Regret}(K) = \tilde{O}(B_K w^{3/2} d^{1/2} \alpha^{-1} + dK\alpha/\sqrt{w} \\ + d\sqrt{KV_K/w} + dK/w). \end{aligned} \tag{8.7}$$

Proof. See Appendix A.6.5. □

Remark 15. *For the stationary linear bandit case where $B_K = 0$, we can set the restart window size $w = K$ and the variance parameter $\alpha = 1/\sqrt{K}$, then we obtain an $\tilde{O}(d\sqrt{V_K} + d)$ regret for Algorithm 8, which is identical to the one in [76].*

Next, we aim to select parameters α and w in order to optimize (8.7).

Corollary 8.1. *Assume that $B_K, V_K \in [\Omega(1), O(K)]$. Then by selecting*

$$\begin{aligned} w &= d^{1/4}\sqrt{V_K/B_K}, & dV_K^6 \geq K^4 B_K^2, \\ w &= d^{1/6}(K/B_K)^{1/3} & \textit{otherwise.} \end{aligned}$$

and $\alpha = d^{-1/4}B_K^{1/2}wK^{-1/2}$*, the regret is in the order*

$$\textit{Regret}(K) = \tilde{O}(d^{7/8}(B_K V_K)^{1/4}\sqrt{K} + d^{5/6}B_K^{1/3}K^{2/3}). \tag{8.8}$$

Remark 16. *We compare the regret of Algo.8 in Corollary 8.1 with previous results in the special cases below.*

- *In the worst case where* $V_K = O(K)$*, our result becomes* $\tilde{O}(d^{7/8}B_K^{1/4}K^{3/4})$*, matching the state-of-the-art results for restarting and sliding window strategies [253, 192].*
- *In the case where the* total variance *is small,* i.e., $V_K = \tilde{O}(1)$*, assuming that* $K^4 > d$*, our result becomes* $\tilde{O}(d^{5/6}B_K^{1/3}K^{2/3})$*, better than all the previous results [253, 192, 254, 194].*

Remark 17. *[186] has studied non-stationary MAB with dynamic variance. With the knowledge of* V_K *and* B_K*, [186] proposed a restart-based Rerun-UCB-V algorithm with a* $\tilde{O}(|\mathcal{A}|^{\frac{2}{3}}B_K^{\frac{1}{3}}V_K^{\frac{1}{3}}K^{\frac{1}{3}} + |\mathcal{A}|^{\frac{1}{2}}B_K^{\frac{1}{2}}K^{\frac{1}{2}})$ *regret, where* $\mathcal{A}$ *is the action set. Reduced to the MAB setting, our Restarted-WeightedOFUL*$^+$ *achieves an* $\tilde{O}(|\mathcal{A}|^{7/8}(B_K V_K)^{1/4}\sqrt{K} + |\mathcal{A}|^{5/6}B_K^{1/3}K^{2/3})$ *regret, which is worse than [186]. We claim that this is due to the generality of the linear bandits, which brings us a looser bound to the drifting term in Lemma 8.4.1. When restricting to the MAB setting, our drifting term enjoys a tighter bound, which could further tighten our final regret. To develop an algorithm achieving the same regret as [186] is beyond the scope of this work.*

Remark 18. *[186] has established a lower bound* $\tilde{\Omega}(B_K^{\frac{1}{3}}V_K^{\frac{1}{3}}K^{\frac{1}{3}} + B_K^{\frac{1}{2}}K^{\frac{1}{2}})$ *for MAB with total variance* V_K *and total variation budget*

B_K. There still exist gaps between our regret and their lower bound regarding the dependence of K, V_K, B_K, and we leave to fix the gaps as future work.

8.5 Non-stationary Linear Contextual Bandit with Unknown Variance and Total Variation Budget

By Theorem 8.4.2, we know that Algorithm 8 is able to utilize the total variance V_K and obtain a better regret result compared with existing algorithms which do not utilize V_K. However, the success of Algorithm 8 depends on the knowledge of the per-round variance σ_k, and it also depends on a good selection of restart window size w, whose optimal selection depends on both V_K and B_K. In this section, we aim to relax these two requirements with still better regret results.

8.5.1 Unknown Per-round Variance, Known V_K and B_K

We first aim to relax the requirement that each σ_k^2 is known to the agent at the beginning of k-th round. We follow the SAVE algorithm [80] which introduces a multi-layer structure [42, 262] to deal with unknown σ_k^2. In detail, SAVE maintains multiple estimates to the current feature vector θ_k, which we denote them as $\hat{\boldsymbol{\theta}}_{k,1}, ..., \hat{\boldsymbol{\theta}}_{k,L}$ in line 2. Each $\hat{\boldsymbol{\theta}}_{k,\ell}$ is calculated based on a subset $\hat{\Psi}_{k,\ell} \subseteq [k-1]$ of samples $\{(\mathbf{a}_t, r_t)\}$. The rule that whether to add the current k to some $\hat{\Psi}_{k,\ell}$ is based on the uncertainty of $\mathbf{a}_k$ with the sample set $\{(\mathbf{a}_t, r_t)\}_{t \in \hat{\Psi}_{k,\ell}}$. As long as $\mathbf{a}_k$ is too uncertain

Algorithm 12 Restarted SAVE$^+$

Require: $\alpha > 0$; the upper bound on the ℓ_2-norm of $\mathbf{a}$ in $\mathcal{D}_k (k \geq 1)$, i.e., A; the upper bound on the ℓ_2-norm of $\boldsymbol{\theta}_k$ $(k \geq 1)$, i.e., B; restart window size w.

1: Initialize $L \leftarrow \lceil \log_2(1/\alpha) \rceil$.
2: Initialize the estimators for all layers: $\hat{\boldsymbol{\Sigma}}_{1,\ell} \leftarrow 2^{-2\ell} \cdot \mathbf{I}$, $\hat{\mathbf{b}}_{1,\ell} \leftarrow \mathbf{0}$, $\hat{\boldsymbol{\theta}}_{1,\ell} \leftarrow \mathbf{0}$, $\hat{\beta}_{1,\ell} \leftarrow 2^{-\ell+1}$, $\hat{\Psi}_{1,\ell} \leftarrow \emptyset$ for all $\ell \in [L]$.
3: **for** $k = 1, \ldots, K$ **do**
4: **if** $k\%w == 0$ **then**
5: Set $\hat{\boldsymbol{\Sigma}}_{k,\ell} \leftarrow 2^{-2\ell} \cdot \mathbf{I}$, $\hat{\mathbf{b}}_{k,\ell} \leftarrow \mathbf{0}$, $\hat{\boldsymbol{\theta}}_{k,\ell} \leftarrow \mathbf{0}$, $\hat{\beta}_{1,\ell} \leftarrow 2^{-\ell+1}$, $\hat{\Psi}_{k,\ell} \leftarrow \emptyset$ for all $\ell \in [L]$.
6: **end if**
7: Observe $\mathcal{D}_k$, choose $\mathbf{a}_k \leftarrow \operatorname{argmax}_{\mathbf{a} \in \mathcal{D}_k} \min_{\ell \in [L]} \langle \mathbf{a}, \hat{\boldsymbol{\theta}}_{k,\ell} \rangle + \hat{\beta}_{k,\ell} \|\mathbf{a}\|_{\hat{\boldsymbol{\Sigma}}_{k,\ell}^{-1}}$ and observe r_k.
8: Set $\ell_k \leftarrow L + 1$
9: Let $\mathcal{L}_k \leftarrow \{\ell \in [L] : \|\mathbf{a}_k\|_{\hat{\boldsymbol{\Sigma}}_{k,\ell}^{-1}} \geq 2^{-\ell}\}$, set $\ell_k \leftarrow \min(\mathcal{L}_k)$ if $\mathcal{L}_k \neq \emptyset$
10: $\hat{\Psi}_{k,\ell_k} \leftarrow \hat{\Psi}_{k,\ell_k} \cup \{k\}$
11: **if** $\mathcal{L}_k \neq \emptyset$ **then**
12: Set $w_k \leftarrow \frac{2^{-\ell_k}}{\|\mathbf{a}_k\|_{\hat{\boldsymbol{\Sigma}}_{k,\ell_k}^{-1}}}$ and update

$$\hat{\boldsymbol{\Sigma}}_{k+1,\ell_k} \leftarrow \hat{\boldsymbol{\Sigma}}_{k,\ell_k} + w_k^2 \mathbf{a}_k \mathbf{a}_k^\top, \hat{\mathbf{b}}_{k+1,\ell} \leftarrow \hat{\mathbf{b}}_{k,\ell_k} + w_k^2 \cdot r_k \mathbf{a}_k, \hat{\boldsymbol{\theta}}_{k+1,\ell_k} \leftarrow \hat{\boldsymbol{\Sigma}}_{k+1,\ell_k}^{-1} \hat{\mathbf{b}}_{k+1,\ell_k}.$$

13: Compute the adaptive confidence radius $\hat{\beta}_{k+1,l}$for the next round according to (8.9).
14: **end if**
15: For $\ell \neq \ell_k$ let $\hat{\boldsymbol{\Sigma}}_{k+1,\ell} \leftarrow \hat{\boldsymbol{\Sigma}}_{k,\ell}, \hat{\mathbf{b}}_{k+1,\ell} \leftarrow \hat{\mathbf{b}}_{k,\ell}, \hat{\boldsymbol{\theta}}_{k+1,\ell} \leftarrow \hat{\boldsymbol{\theta}}_{k,\ell}, \hat{\beta}_{k+1,\ell} \leftarrow \hat{\beta}_{k,\ell}$.
16: **end for**

w.r.t. some level ℓ_k (line 9), we add k to $\hat{\Psi}_{k,\ell}$ and update the estimate $\hat{\boldsymbol{\theta}}_{k,\ell_k}$ accordingly (line 12). Each $\hat{\boldsymbol{\theta}}_{k,\ell_k}$ is calculated as the solution of a weighted regression problem, where the weight w_k is selected as the inverse of the uncertainty of the arm $\mathbf{a}_k$ w.r.t.

the samples in the ℓ-th layer. Maintaining L different $\hat{\boldsymbol{\theta}}_{k,\ell}, \ell \in [L]$, Algorithm 12 then calculates L number of UCB for each arm $\mathbf{a}$ w.r.t. L different $\hat{\boldsymbol{\theta}}_{k,\ell}$, and selects the arm which maximizes the minimization of L UCBs (line 7). It has been shown in [80] that such a multilayer structure is able to utilize the V_K information without knowing the per-round variance σ_k^2. Similar to Algorithm 8, in order to deal with the nonstationarity issue, we introduce a restarting scheme that Algorithm 12 restarts itself by a restart window size w (line 5).

Next we show the theoretical guarantee of Algorithm 12. We call the restart time rounds *grids* and denote them by $g_1, g_2, \dots g_{\lceil \frac{K}{w} \rceil - 1}$, where $g_i \% w = 0$ for all $i \in [\lceil \frac{K}{w} \rceil - 1]$. Let i_k be the grid index of time round k, *i.e.*, $g_{i_k} \leq k < g_{i_k+1}$. We denote $\hat{\Psi}_{k,\ell} := \{t : t \in [g_{i_k}, k-1], \ell_t = \ell\}$. We define the confidence radius $\hat{\beta}_{k,\ell}$ at round k and layer ℓ as

$$\hat{\beta}_{k,\ell} := 16 \cdot 2^{-\ell} \sqrt{\left(8\hat{\mathrm{Var}}_{k,\ell} + 6R^2 \log(\frac{4(w+1)^2 L}{\delta}) + 2^{-2\ell+4}\right)} \times \sqrt{\log(\frac{4w^2 L}{\delta})} + 6 \cdot 2^{-\ell} R \log(\frac{4w^2 L}{\delta}) + 2^{-\ell} B, \tag{8.9}$$

where we set $\hat{\mathrm{Var}}_{k,\ell}$ as $\sum_{i \in \hat{\Psi}_{k,\ell}} w_i^2 \big(r_i - \langle \hat{\boldsymbol{\theta}}_{k,\ell}, \mathbf{a}_i \rangle\big)^2$, if $2^\ell \geq 64\sqrt{\log\left(\frac{4(w+1)^2 L}{\delta}\right)}$, or $R^2 \left|\hat{\Psi}_{k,\ell}\right|$ for the remaining cases.

Note that our selection of the confidence radius $\hat{\beta}_{k,\ell}$ only depends on $\hat{\mathrm{Var}}_{k,\ell}$, which serves as an estimate of the total variance of samples at ℓ-th layer without knowing σ_k^2.

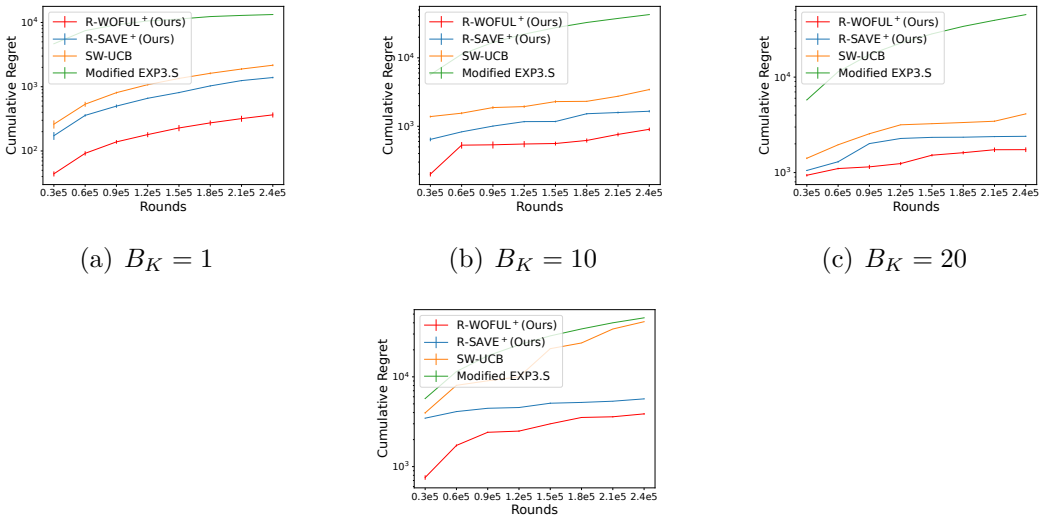
We build the theoretical guarantee of Algorithm 12 as follows.

Theorem 8.5.1. *Let* $0 < \delta < 1$. *Define* $\{\beta_{k,\ell}\}_{k\geq 1,\ell\in[L]}$ *as in* (8.9), *regarding* A, R *as constants, we have*

$$\begin{aligned} \textit{Regret}(K) = \tilde{O}(\sqrt{d}w^{1.5}B_K/\alpha + \alpha^2(K + \sqrt{wKV_K}) \\ + d\sqrt{KV_K/w} + dK/w). \end{aligned}$$

Proof. See Appendix A.6.6 for the full proof. □

Remark 19. *Like Remark 15, we consider the case where* $B_K = 0$. *We set* $w = K$ *and* $\alpha^2 = 1/K\sqrt{V_K}$, *then we obtain a regret* $\tilde{O}(d\sqrt{V_K} + d)$, *which matches the regret of the SAVE algorithm in [80].*



(a) $B_K = 1$ (b) $B_K = 10$ (c) $B_K = 20$

(d) $B_K = K^{1/3}$

Figure 8.1: The regret of Restarted-WeightedOFUL$^+$, Restarted SAVE$^+$, SW-UCB and Modified EXP3.S under different total rounds.

Corollary 8.2. *Assume that* $B_K, V_K \in [\Omega(1), O(K)]$, *then by selecting*

$$\begin{aligned} w &= d^{1/3}(K/B_K)^{1/3}, & K^2 &\geq V_K^3 d/B_K, \\ w &= d^{2/5}(KV_K)^{1/5}/B_K^{2/5} & &\textit{otherwise.} \end{aligned}$$

and $\alpha = d^{1/6}\sqrt{w}B_K^{1/3}/(K^{1/3} + (V_K K w)^{1/6})$*, we have*

$$Regret(K) = \tilde{O}(d^{4/5}V_K^{2/5}B_K^{1/5}K^{2/5} + d^{2/3}B_K^{1/3}K^{2/3}).$$

Remark 20. *We discuss the regret of Algo.12 in Corollary 8.2 in the following special cases. In the case where the* total variance *is small,* i.e., $V_K = \tilde{O}(1)$*, assuming that* $K^2 > d$*, our result becomes* $\tilde{O}(d^{2/3}B_K^{1/3}K^{2/3})$*, better than all the previous results [253, 192, 254, 194]. In the worst case where* $V_K = O(K)$*, our result becomes* $\tilde{O}(d^{4/5}B_K^{1/5}K^{4/5})$*.*

Unknown Per-round Variance, Unknown V_K and B_K In Corollary 8.2, we need to know the *total variance* V_K and *total variation budget* B_K to select the optimal w and α. To deal with the more general case where V_K and B_K are unknown, we can employ the *Bandits-over-Bandits* (BOB) mechanism ([187, 254, 192]). We name the Restarted SAVE$^+$ algorithm with BOB mechanism as "Restarted SAVE$^+$-BOB". Due to the space limit, we put the algorithm design, descriptions, and theoretical analysis of Restarted SAVE$^+$-BOB (Algo.18) in Appendix A.6.1.

8.6 Experiments

To validate the effectiveness of our methods, we conduct a series of experiments on the synthetic data.

Problem Setting and Baselines Following the experimental set up in [187], we consider the 2-armed bandits setting, where

the action set $\mathcal{D}_k = \{(1,0),(0,1)\}$, and

$$\boldsymbol{\theta}_k = \begin{pmatrix} 0.5 + \frac{3}{10}\sin(5B_K\pi k/K) \\ 0.5 + \frac{3}{10}\sin(\pi + 5B_K\pi k/K) \end{pmatrix}.$$

It is easy to see that the total variation budget can be bounded as B_K. At each round k, the ϵ_k satisfies the following distribution:

$$\epsilon_k \sim \text{Bernoulli}(0.5/k) - 0.5/k.$$

We can verify that under such a distribution for ϵ_k, the variance of the reward distribution at k-th round is $(1-0.5/k)\cdot 0.5/k$, and the total variance $V_K \sim \log K$.

We compare the proposed Restarted-WeightedOFUL$^+$ and Restarted SAVE$^+$ with SW-UCB [187] and Modified EXP3.S [263]. We leave the detailed setup for the baselines in Appendix A.6.2.

Result We plot the results in Figure.8.1, where all the empirical results are averaged over ten independent trials and the error bar is the standard error divided by $\sqrt{10}$. The results are consistent with our theoretical findings. It is evident that our algorithms significantly outperform both SW-UCB and Modified EXP3.S. Among our proposed algorithms,
Restarted-WeightedOFUL$^+$ achieves the best performance. This can be attributed to the fact that it knows the variance and can make more informed decisions. Although Restarted SAVE$^+$ performed slightly worse than Restarted-WeightedOFUL$^+$, it still outperforms the baseline algorithms, particularly when $B_K = K^{1/3}$. These results highlight the superiority of our methods.

Chapter 9

Online Clustering of Dueling Bandits

The contextual multi-armed bandit (MAB) is a widely used framework for problems requiring sequential decision-making under uncertainty, such as recommendation systems. In applications involving a large number of users, the performance of contextual MAB can be significantly improved by facilitating collaboration among multiple users. This has been achieved by the clustering of bandits (CB) methods, which adaptively group the users into different clusters and achieve collaboration by allowing the users in the same cluster to share data. However, classical CB algorithms typically rely on numerical reward feedback, which may not be practical in certain real-world applications. For instance, in recommendation systems, it is more realistic and reliable to solicit *preference feedback* between pairs of recommended items rather than absolute rewards. To address this limitation, we introduce the first "clustering of dueling bandit algorithms" to enable collaborative decision-making based on preference feedback. We propose two novel algorithms: (1) Clustering of Linear Dueling Ban-

dits (COLDB) which models the user reward functions as linear functions of the context vectors, and (2) Clustering of Neural Dueling Bandits (CONDB) which uses a neural network to model complex, non-linear user reward functions. Both algorithms are supported by rigorous theoretical analyses, demonstrating that user collaboration leads to improved regret bounds. Extensive empirical evaluations on synthetic and real-world datasets further validate the effectiveness of our methods, establishing their potential in real-world applications involving multiple users with preference-based feedback.

9.1 Introduction

The contextual multi-armed bandit (MAB) is a widely used method in real-world applications requiring sequential decision-making under uncertainty, such as recommendation systems, computer networks, among others [41]. In a contextual MAB problem, a user faces a set of K arms (i.e., context vectors) in every round, selects one of these K arms, and then observes a corresponding numerical reward [139]. In order to select the arms to maximize the cumulative reward (or equivalently minimize the cumulative regret), we often need to consider the trade-off between the *exploration* of the arms whose unknown rewards are associated with large uncertainty and *exploitation* of the available observations collected so far. To carefully handle this trade-off, we often model the reward function using a surrogate model, such as a linear model [42] or a neural network [160].

Some important applications of contextual MAB, such as rec-

ommendation systems, often involve a large number (e.g., in the scale of millions) of users, which opens up the possibility of further improving the performance of contextual MAB via user collaboration. To this end, the method of *online Clustering of Bandits* (CB) has been proposed, which adaptively partitions the users into a number of clusters and leverages the collaborative effect of the users in the same cluster to achieve improved performance [46, 4, 136].

Classical CB algorithms usually require an absolute real-valued numerical reward as feedback for each arm [4]. However, in some crucial applications of contextual MAB, it is often more realistic and reliable to request the users for *preference feedback*. For example, in recommendation systems, it is often preferable to recommend a pair of items to a user and then ask the user for relative feedback (i.e., which item is preferred) [58]. As another example, contextual MAB has been successfully adopted to optimize the input prompt for large language models (LLMs), which is often referred to as *prompt optimization* [59, 264]. In this application, instead of requesting an LLM user for a numerical score as feedback, it is more practical to show the user a pair of LLM responses generated by two candidate prompts and ask the user which response is preferred [59, 60].

A classical and principled approach to account for preference feedback in contextual MAB is the framework of contextual *dueling bandit* [155, 157, 156, 159]. In every round of contextual dueling bandits, a pair of arms are selected, after which a binary observation is collected reflecting which arm is preferred. However, classical dueling bandit algorithms are not able to leverage

the collaboration of multiple users, which leaves significant untapped potential to further improve the performance in these applications involving preference feedback. In this work, we bring together the merits of both approaches, and hence introduce the first *clustering of dueling bandit* algorithms, enabling multi-user collaboration in scenarios involving preference feedback.

We firstly proposed our *Clustering Of Linear Dueling Bandits* (COLDB) algorithm (Sec. 9.3.1), which assumes that the latent reward function of each user is a linear function of the context vectors (i.e., the arm features). In addition, to handle challenging real-world scenarios with complicated non-linear reward functions, we extend our COLDB algorithm to use a *neural network to model the reward function*, hence introducing our *Clustering Of Neural Dueling Bandits* (CONDB) algorithm (Sec. 9.3.2). Both algorithms adopt a graph to represent the estimated clustering structure of all users, and adaptively update the graph to iteratively refine the estimate. After receiving a user in every round, our both algorithms firstly assign the user to its estimated cluster, and then leverage the data from all users in the estimated cluster to learn a linear model (COLDB) or a neural network (CONDB), which is then used to select a pair of arms for the user to query for preference feedback. After that, we update the reward function estimate for the user based on the newly observed feedback, and then update the graph to remove its connection with users who are estimated to belong to a different cluster.

We conduct rigorous theoretical analysis for both our COLDB and CONDB algorithms, and our theoretical results demonstrate that the regret upper bounds of both algorithms are sub-linear

and that a larger degree of user collaboration (i.e., when a larger number of users belong to the same cluster on average) leads to theoretically guaranteed improvement (Sec. 9.4). In addition, we also perform both synthetic and real-world experiments to demonstrate the practical advantage of our algorithms and the benefit of user collaboration in contextual MAB problems with preference feedback (Sec. 9.5).

9.2 Problem Setting

This section formulates the problem of *clustering of dueling bandits*. In the following, we use boldface lowercase letters for vectors and boldface uppercase letters for matrices. The number of elements in a set $\mathcal{A}$ is denoted as $|\mathcal{A}|$, while $[m]$ refers to the index set $\{1, 2, \ldots, m\}$, and $\|\boldsymbol{x}\|_{\boldsymbol{M}} = \sqrt{\boldsymbol{x}^\top \boldsymbol{M} \boldsymbol{x}}$ represents the matrix norm of vector $\boldsymbol{x}$ with respect to the positive semi-definite (PSD) matrix $\boldsymbol{M}$.

Clustering Structure. Consider a scenario with u users, indexed by $\mathcal{U} = \{1, 2, \ldots, u\}$, where each user $i \in \mathcal{U}$ is associated with a unknown reward function $f_i : \mathbb{R}^{d'} \to \mathbb{R}$ which maps an arm $\boldsymbol{x} \in \mathcal{X} \subset \mathbb{R}^{d'}$ to its corresponding reward value $f_i(\boldsymbol{x})$. We assume that there exists an underlying, yet unknown, clustering structure over the users reflecting their behavior similarities. Specifically, the set of users $\mathcal{U}$ is partitioned into m clusters $C_1, C_2, \ldots, C_m$, where $m \ll u$, and the clusters are mutually disjoint: $\cup_{j \in [m]} C_j = \mathcal{U}$ and $C_j \cap C_{j'} = \emptyset$ for $j \neq j'$. These clusters are referred to as *ground-truth clusters*, and the set of clusters is denoted by $\mathcal{C} = \{C_1, C_2, \ldots, C_m\}$. Let f^j denote the common reward function

of all users in cluster j and let $j(i) \in [m]$ be the index of the cluster to which user i belongs. If two users i and l belong to the same cluster, they have the same reward function. That is, for any $\ell \in \mathcal{U}$, if $\ell \in C_{j(i)}$, then $f_\ell = f_i = f^{j(i)}$. Meanwhile, users from different clusters have distinct reward functions.

Modeling Preference Feedback. At each time step $t \in [T]$, a user $i_t \in \mathcal{U}$ is served. The learning agent observes a set of context vectors (i.e., arms) $\mathcal{X}_t \subseteq \mathcal{X} \subset \mathbb{R}^{d'}$, where $|\mathcal{X}_t| = K \leq C$ for all t. Each arm $\boldsymbol{x} \in \mathcal{X}_t$ is a feature vector in $\mathbb{R}^{d'}$ with $\|\boldsymbol{x}\|_2 \leq 1$. The agent assigns the cluster $\overline{C}_t$ to user i_t and recommends two arms $\boldsymbol{x}_{t,1}, \boldsymbol{x}_{t,2} \in \mathcal{X}_t$ based on the aggregated historical data from cluster $\overline{C}_t$. After receiving the recommended pair of arms, the user provides a binary preference feedback $y_t \in \{0, 1\}$, in which $y_t = 1$ if $\boldsymbol{x}_{t,1}$ is preferred over $\boldsymbol{x}_{t,2}$ and $y_t = 0$ otherwise. We model the binary preference feedback following the widely used Bradley-Terry-Luce (BTL) model [265, 266]. Specifically, the BTL model assumes that for user i_t, the probability that the first arm $\boldsymbol{x}_{t,1}$ is preferred over the second arm $\boldsymbol{x}_{t,2}$ is given by

$$\mathbb{P}_t(\boldsymbol{x}_{t,1} \succ \boldsymbol{x}_{t,2}) = \mu(f_{i_t}(\boldsymbol{x}_{t,1}) - f_{i_t}(\boldsymbol{x}_{t,2})),$$

where $\mu : \mathbb{R} \to [0, 1]$ is the logistic function: $\mu(z) = \frac{1}{1+e^{-z}}$. In other words, the binary feedback y_t is sampled from the Bernoulli distribution with the probability $\mathbb{P}_t(\boldsymbol{x}_{t,1} \succ \boldsymbol{x}_{t,2})$.

We make the following assumption about the preference model:

Assumption 9.1 (Standard Dueling Bandits Assumptions)**.** *1.* $|\mu(f(\boldsymbol{x})) - \mu(g(\boldsymbol{x}))| \leq L_\mu |f(\boldsymbol{x}) - g(\boldsymbol{x})|, \forall x \in \mathcal{X}$ *, for any functions*

$f, g : \mathbb{R}^{d'} \to \mathbb{R}$.

2. $\min_{\boldsymbol{x} \in \mathcal{X}} \nabla \mu(f(\boldsymbol{x})) \geq \kappa_\mu > 0$.

Assumption 9.1 is the standard assumption in the analysis of linear bandits and dueling bandits [267, 157], and when μ is the logistic function, $L_\mu = 1/4$. The regret incurred by the learning agent is defined as:

$$R_T = \sum_{t=1}^{T} r_t = \sum_{t=1}^{T} \left(2 f_{i_t}(\boldsymbol{x}_t^*) - f_{i_t}(\boldsymbol{x}_{t,1}) - f_{i_t}(\boldsymbol{x}_{t,2})\right),$$

where $\boldsymbol{x}_t^* = \arg\max_{\boldsymbol{x} \in \mathcal{X}_t} f_{i_t}(\boldsymbol{x})$ represents the optimal arm at round t. This is a commonly adopted notion of regret in the analysis of dueling bandits [157, 156].

9.2.1 Clustering of Linear Dueling Bandits

For the linear setting, we assume that each reward function f_i is linear in a fixed feature space $\phi(\cdot)$, such that $f_i(\boldsymbol{x}) = \boldsymbol{\theta}_i^\top \phi(\boldsymbol{x}), \forall \boldsymbol{x} \in \mathcal{X}$. The feature mapping $\phi : \mathbb{R}^{d'} \to \mathbb{R}^d$ is a fixed mapping with $\|\phi(\boldsymbol{x})\|_2 \leq 1$ for all $\boldsymbol{x} \in \mathcal{X}$. In the special case of classical linear dueling bandits, we have that $\phi(\boldsymbol{x}) = \boldsymbol{x}$, i.e., $\phi(\cdot)$ is the identity mapping. The use of $\phi(\boldsymbol{x})$ enables us to potentially model non-linear reward functions given an appropriate feature mapping.

In this case, the reward function of every user i is represented by its corresponding *preference vector* $\boldsymbol{\theta}_i$, and all users in the same cluster share the same preference vector while users from different clusters have distinct preference vectors. Denote $\boldsymbol{\theta}^j$ as the common preference vector of users in cluster C_j, and let $j(i) \in [m]$ be the index of the cluster to which user i belongs. Therefore, for any $\ell \in \mathcal{U}$, if $\ell \in C_{j(i)}$, then $\boldsymbol{\theta}_\ell = \boldsymbol{\theta}_i = \boldsymbol{\theta}^{j(i)}$.

The following assumptions are made regarding the clustering structure, users, and items:

Assumption 9.2 (Cluster Separation)**.** *The preference vectors of users from different clusters are at least separated by a constant gap $\gamma > 0$, i.e.,*

$$\left\|\boldsymbol{\theta}^{j} - \boldsymbol{\theta}^{j'}\right\|_2 \geq \gamma \quad \textit{for all} \quad j \neq j' \in [m].$$

Assumption 9.3 (Uniform User Arrival)**.** *At each time step t, the user i_t is selected uniformly at random from $\mathcal{U}$, with probability $1/u$, independent of previous rounds.*

Assumption 9.4 (Item regularity)**.** *At each time step t, the feature vector $\phi(\boldsymbol{x})$ of each arm $\boldsymbol{x} \in \mathcal{X}_t$ is drawn independently from a fixed but unknown distribution ρ over $\{\phi(\boldsymbol{x}) \in \mathbb{R}^d : \|\phi(\boldsymbol{x})\|_2 \leq 1\}$, where $\mathbb{E}_{\boldsymbol{x}\sim\rho}[\phi(\boldsymbol{x})\phi(\boldsymbol{x})^\top]$ is full rank with minimal eigenvalue $\lambda_x > 0$. Additionally, at any time t, for any fixed unit vector $\boldsymbol{\theta} \in \mathbb{R}^d$, $(\boldsymbol{\theta}^\top \phi(\boldsymbol{x}))^2$ has sub-Gaussian tail with variance upper bounded by σ^2.*

Remark 1. All these assumptions above follow the previous works on clustering of bandits [46, 226, 135, 227, 137, 4, 5]. For Assumption 9.3, our results can easily generalize to the case where the user arrival follows any distribution with minimum arrival probability $\geq p_{min}$.

9.2.2 Clustering of Neural Dueling Bandits

Here we allow the reward functions f_i's to be non-linear functions. To estimate the unknown reward functions f_i's, we use

fully connected neural networks (NNs) with ReLU activations, and denote the depth and width (of every layer) of the NN by $L \geq 2$ and m_{NN}, respectively [160, 161]. Let $h(\boldsymbol{x}; \theta)$ represent the output of an NN with parameters $\boldsymbol{\theta}$ and input vector $\boldsymbol{x}$, which is defined as follows:

$$h(\boldsymbol{x}; \boldsymbol{\theta}) = \mathbf{W}_L \text{ReLU}\left(\mathbf{W}_{L-1} \text{ReLU}\left(\cdots \text{ReLU}\left(\mathbf{W}_1 \boldsymbol{x}\right)\right)\right),$$

in which $\text{ReLU}(\boldsymbol{x}) = \max\{\boldsymbol{x}, 0\}$, $\mathbf{W}_1 \in \mathbb{R}^{m_{\text{NN}} \times d}$, $\mathbf{W}_l \in \mathbb{R}^{m_{\text{NN}} \times m_{\text{NN}}}$ for $2 \leq l < L$, $\mathbf{W}_L \in \mathbb{R}^{1 \times m_{\text{NN}}}$. We denote the parameters of NN by $\boldsymbol{\theta} = (\text{vec}\left(\mathbf{W}_1\right); \cdots \text{vec}\left(\mathbf{W}_L\right))$, where $\text{vec}\left(A\right)$ converts an $M \times N$ matrix A into a MN-dimensional vector. We use p to denote the total number of NN parameters: $p = dm_{\text{NN}} + m_{\text{NN}}^2(L-1) + m_{\text{NN}}$, and use $g(\boldsymbol{x}; \boldsymbol{\theta})$ to denote the gradient of $h(\boldsymbol{x}; \boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$.

The algorithmic design and analysis of neural bandit algorithms make use of the theory of the *neural tangent kernel* (NTK) [268]. We let all u users use the same initial NN parameters $\boldsymbol{\theta}_0$, and assume that the value of the *empircal NTK* is bounded: $\frac{1}{m_{\text{NN}}} \langle g(\boldsymbol{x}; \boldsymbol{\theta}_0), g(\boldsymbol{x}; \boldsymbol{\theta}_0) \rangle \leq 1, \forall \boldsymbol{x} \in \mathcal{X}$. This is a commonly adopted assumption in the analysis of neural bandits [269, 163]. Let T^j denote total number of rounds in which the users in cluster j is served. We use $\mathbf{H}_j$ to denote the *NTK matrix* [160] for cluster j, which is a $(T_j K) \times (T_j K)$-dimensional matrix. Similarly, we define $\mathbf{h}_j$ as the $(T_j K) \times 1$-dimensional vector containing the reward function values of all $T_j K$ arm feature vectors for cluster j. We provide the concrete definitions of $\mathbf{H}_j$ and $\mathbf{h}_j$ in App. A.9.1. We make the following assumptions which are commonly adopted by previous works on neural bandits [160, 161], for which we provide

justifications in App. A.9.1.

Assumption 9.5. *The reward functions for all users are bounded:* $|f_i(x)| \leq 1, \forall x \in \mathcal{X}, \forall i \in \mathcal{U}$. *There exists* $\lambda_0 > 0$ *s.t.* $\mathbf{H}_j \succeq \lambda_0 I, \forall j \in \mathcal{C}$. *All arm feature vectors satisfy* $\|x\|_2 = 1$ *and* $x_j = x_{j+d/2}$, $\forall x \in \mathcal{X}_t, \forall t \in [T]$.

Denote by f^j the common reward function of the users in cluster C_j, and let $j(i) \in [m]$ be the index of the cluster to which user i belongs. Same as Sec. 9.2.1, here all users in the same cluster share the same reawrd function. Therefore, for any $\ell \in \mathcal{U}$, if $\ell \in C_{j(i)}$, then $f_\ell(\boldsymbol{x}) = f_i(\boldsymbol{x}) = f^{j(i)}(\boldsymbol{x}), \forall \boldsymbol{x} \in \mathcal{X}$. The following lemma shows that when the NN is wide enough (i.e., m_{NN} is large), the reward function of every cluster can be modeled by a linear function.

Lemma 9.2.1 (Lemma B.3 of [161])**.** *As long as the width* m_{NN} *of the NN is large:* $m_{NN} \geq poly(T, L, K, 1/\kappa_\mu, L_\mu, 1/\lambda_0, 1/\lambda, \log(1/\delta))$, *then for all clusters* $j \in [m]$, *with probability of at least* $1 - \delta$, *there exits a* $\boldsymbol{\theta}_f^j$ *such that*

$$f^j(\boldsymbol{x}) = \langle g(\boldsymbol{x}; \boldsymbol{\theta}_0), \boldsymbol{\theta}_f^j - \boldsymbol{\theta}_0 \rangle,$$
$$\sqrt{m_{NN}} \left\| \boldsymbol{\theta}_f^j - \boldsymbol{\theta}_0 \right\|_2 \leq \sqrt{2\mathbf{h}_j^\top \mathbf{H}_j^{-1} \mathbf{h}_j} \leq B,$$

for all $\boldsymbol{x} \in \mathcal{X}_t$, $t \in [T]$ *with* $i_t \in C_j$.

We provide the detailed statement of Lemma 9.2.1 in Lemma A.9.1 (App. A.9.2). For a user i belonging to cluster $j(i)$, we let $\boldsymbol{\theta}_{f,i} = \boldsymbol{\theta}_f^{j(i)}$, then we have that $f_i(\boldsymbol{x}) = \langle g(\boldsymbol{x}; \boldsymbol{\theta}_0), \boldsymbol{\theta}_{f,i} - \boldsymbol{\theta}_0 \rangle, \forall \boldsymbol{x} \in \mathcal{X}$. As a result of Lemma 9.2.1, for any $\ell \in \mathcal{U}$, if $\ell \in C_{j(i)}$, we have that $\boldsymbol{\theta}_{f,\ell} = \boldsymbol{\theta}_{f,i} = \boldsymbol{\theta}^{j(i)}, \forall \boldsymbol{x} \in \mathcal{X}$.

The assumption below formalizes the gap between different clusters in a similar way to Assumption 9.2.

Assumption 9.6 (Cluster Separation)**.** *The reward functions of users from different clusters are separated by a constant gap γ':*

$$\left\| f^{j}(\boldsymbol{x}) - f^{j'}(\boldsymbol{x}) \right\|_2 \geq \gamma' > 0 \,, \forall j, j' \in [m] \,, j \neq j' \, \forall \boldsymbol{x} \in \mathcal{X}.$$

In neural bandits, we adopt $(1/\sqrt{m_{\mathrm{NN}}})g(\boldsymbol{x}; \boldsymbol{\theta}_0)$ as the feature mapping. Therefore, our item regularity assumption (Assumption 9.4) is also applicable here after plugging in $\phi(\boldsymbol{x}) = (1/\sqrt{m_{\mathrm{NN}}})g(\boldsymbol{x}; \boldsymbol{\theta}_0)$.

9.3 Algorithms

9.3.1 Clustering Of Linear Dueling Bandits (COLDB)

Our Clustering Of Linear Dueling Bandits (COLDB) algorithm is described in Algorithm 13. Here we elucidate the underlying principles and operational workflow of COLDB. COLDB maintains a dynamic graph $G_t = (\mathcal{U}, E_t)$ encompassing all users, whose connected components represent the inferred user clusters in round t. Throughout the learning process, COLDB adaptively removes edges to accurately cluster the users based on their estimated reward function parameters, thereby leveraging these clusters to enhance online learning efficiency. The operation of COLDB proceeds as follows:

Cluster Inference $\overline{C}_t$ for User i_t (Line 2-Line 5). Initially, COLDB constructs a complete undirected graph $G_0 = (\mathcal{U}, E_0)$

over the user set (Line 2). As learning progresses, edges are selectively removed to ensure that only users with similar preference profiles remain connected. At each round t, when a user i_t comes to the system with a feasible arm set $\mathcal{X}_t$ (Line 4), COLDB identifies the connected component $\overline{C}_t$ containing i_t in the maintained graph G_{t-1}, which serves as the current estimated cluster for this user (Line 5).

Estimating Shared Statistics for Cluster $\overline{C}_t$ (Line 6-Line 7). Once the cluster $\overline{C}_t$ is identified, COLDB estimates a common preference vector $\overline{\boldsymbol{\theta}}_t$ for all users within this cluster by aggregating the historical feedback from all members of $\overline{C}_t$. Specifically, in Line 6, the common preference vector is determined by minimizing the following loss function:

$$\overline{\boldsymbol{\theta}}_t = \arg\min_{\boldsymbol{\theta}} - \sum_{\substack{s\in[t-1]\\ i_s\in\overline{C}_t}} \Big(y_s \log \mu\left(\boldsymbol{\theta}^\top \left[\phi(\boldsymbol{x}_{s,1}) - \phi(\boldsymbol{x}_{s,2})\right]\right) + (1-y_s)\log\mu\left(\boldsymbol{\theta}^\top\left[\phi(\boldsymbol{x}_{s,2}) - \phi(\boldsymbol{x}_{s,1})\right]\right)\Big) + \frac{1}{2}\lambda \left\|\boldsymbol{\theta}\right\|_2^2 , \quad (9.1)$$

which corresponds to the Maximum Likelihood Estimation (MLE) using the data from all users in the cluster $\overline{C}_t$. Additionally, in Line 7, COLDB computes the aggregated information matrix for $\overline{C}_t$, which is subsequently utilized in selecting the second arm $\boldsymbol{x}_{t,2}$:

$$\boldsymbol{V}_{t-1} = \boldsymbol{V}_0 + \sum_{\substack{s\in[t-1]\\ i_s\in\overline{C}_t}} (\phi(\boldsymbol{x}_{s,1}) - \phi(\boldsymbol{x}_{s,2}))(\phi(\boldsymbol{x}_{s,1}) - \phi(\boldsymbol{x}_{s,2}))^\top \quad (9.2)$$

Arm Recommendation Based on Cluster Statistics (Line 8-Line 9). Leveraging the estimated common preference vector $\overline{\boldsymbol{\theta}}_t$ and the aggregated information matrix $\boldsymbol{V}_{t-1}$, COLDB proceeds to recommend two arms as follows:

- **First Arm Selection ($\boldsymbol{x}_{t,1}$).** In Line 8, COLDB selects the first arm by greedily choosing the arm that maximizes the estimated reward according to $\overline{\boldsymbol{\theta}}_t$:

$$\boldsymbol{x}_{t,1} = \arg\max_{\boldsymbol{x}\in\mathcal{X}_t} \overline{\boldsymbol{\theta}}_t^\top \phi(\boldsymbol{x}). \tag{9.3}$$

- **Second Arm Selection ($\boldsymbol{x}_{t,2}$).** Following the selection of $\boldsymbol{x}_{t,1}$, in Line 9, COLDB selects the second arm by maximizing an upper confidence bound (UCB):

$$\boldsymbol{x}_{t,2} = \arg\max_{\boldsymbol{x}\in\mathcal{X}_t} \overline{\boldsymbol{\theta}}_t^\top \phi(\boldsymbol{x}) + \frac{\beta_t}{\kappa_\mu} \|\phi(\boldsymbol{x}) - \phi(\boldsymbol{x}_{t,1})\|_{\boldsymbol{V}_{t-1}^{-1}} . \tag{9.4}$$

Intuitively, Eq.(9.4) encourages the selection of the arm which both (a) has a large predicted reward value and (b) is different from $\boldsymbol{x}_{t,1}$ and the arms selected in the previous $t-1$ rounds when the served user belongs to the currently estimated cluster $\overline{C}_t$. In other words, the second arm $\boldsymbol{x}_{t,2}$ is chosen by balancing exploration and exploitation.

Updating User Estimates and Interaction History (Line 10-Line 11). Upon recommending $\boldsymbol{x}_{t,1}$ and $\boldsymbol{x}_{t,2}$, the user receives binary feedback $y_t = \mathbb{1}(\boldsymbol{x}_{t,1} \succ \boldsymbol{x}_{t,2})$ from user i_t, and then updates the interaction history $\mathcal{D}_t = \{i_s, \boldsymbol{x}_{s,1}, \boldsymbol{x}_{s,2}, y_s\}_{s=1}^t$ (Line 10). Moreover, COLDB updates the preference vector estimate for user i_t while keeping the estimates for the other users unchanged (Line 11). Specifically, the preference vector estimate

$\hat{\boldsymbol{\theta}}_{i_t,t}$ is updated via MLE using the historical data from user i_t:

$$\hat{\boldsymbol{\theta}}_{i_t,t} = \arg\min_{\boldsymbol{\theta}} - \sum_{\substack{s\in[t-1]\\ i_s=i_t}} \Big(y_s \log \mu\big(\boldsymbol{\theta}^\top[\phi(\boldsymbol{x}_{s,1}) - \phi(\boldsymbol{x}_{s,2})]\big) + (1-y_s)\log\mu\big(\boldsymbol{\theta}^\top[\phi(\boldsymbol{x}_{s,2}) - \phi(\boldsymbol{x}_{s,1})]\big)\Big) + \frac{\lambda}{2}\|\boldsymbol{\theta}\|_2^2 . \tag{9.5}$$

Dynamic Graph Update (Line 12). Finally, based on the updated preference estimate $\hat{\boldsymbol{\theta}}_{i_t,t}$ for user i_t, COLDB reassesses the similarity between i_t and the other users. If the discrepancy between $\hat{\boldsymbol{\theta}}_{i_t,t}$ and $\hat{\boldsymbol{\theta}}_{\ell,t}$ for any user ℓ surpasses a predefined threshold (Line 12), the edge (i_t, ℓ) is removed from the graph G_{t-1}, effectively separating them into distinct clusters. The resultant graph $G_t = (\mathcal{U}, E_t)$ is then utilized in the subsequent rounds.

9.3.2 Clustering Of Neural Dueling Bandits (CONDB)

Our Clustering Of Neural Dueling Bandits (CONDB) algorithm is illustrated in Algorithm 19 (App. A.7), which adopts neural networks to model non-linear reward functions. Similar to COLDB, our CONDB algorithm also maintains a dynamic graph $G_t = (\mathcal{U}, E_t)$ in which every connected component denotes an inferred cluster, and adaptively removes the edges between users who are estimated to belong to different clusters.

Cluster Inference $\overline{C}_t$ for User i_t (Line 5). Similar to COLDB (Algo. 13), when a new user i_t arrives, our CONDB firstly identifies the connected component $\overline{C}_t$ in the maintained graph G_{t-1} which contains the user i_t and then uses it as the estimated cluster for i_t (Line 5).

Estimating Shared Statistics for Cluster $\overline{C}_t$ (Line 6). After

the cluster $\overline{C}_t$ is identified, our CONDB algorithm uses the history of preference feedback observations from all users in the cluster $\overline{C}_t$ to train a neural network (NN) to minimize the following loss function (Line 6):

$$\mathcal{L}_t(\boldsymbol{\theta}) = -\frac{1}{m} \sum_{\substack{s\in[t-1]\\ i_s\in\overline{C}_t}} \Big(y_s \log \mu\left(h(\boldsymbol{x}_{s,1};\boldsymbol{\theta}) - h(\boldsymbol{x}_{s,2};\boldsymbol{\theta})\right) + (1-y_s)\log\mu\left(h(\boldsymbol{x}_{s,2};\boldsymbol{\theta}) - h(\boldsymbol{x}_{s,1};\boldsymbol{\theta})\right)\Big) + \frac{\lambda}{2}\left\|\boldsymbol{\theta}-\boldsymbol{\theta}_0\right\|_2^2 \tag{9.9}$$

to yield parameters $\overline{\boldsymbol{\theta}}_t$. In addition, similar to COLDB (Algorithm 13), our CONDB computes the aggregated information matrix for the cluster $\overline{C}_t$ following Eq.(9.2) . Note that here we replace $\phi(\boldsymbol{x})$ from Eq.(9.2) by the NTK feature representation $\phi(\boldsymbol{x}) = (1/\sqrt{m})g(\boldsymbol{x};\boldsymbol{\theta}_0)$, in which $\boldsymbol{\theta}_0$ represents the initial parameters of the NN (Sec. 9.2.2).

Arm Recommendation Based on Cluster Statistics (Line 8-Line 9). Next, our CONDB algorithm leverages the trained NN with parameters $\overline{\boldsymbol{\theta}}_t$ and the aggregated information matrix $\boldsymbol{V}_{t-1}$ to select the pair of arms. The first arm is selected by greedily maximizing the reward prediction of the NN with parameters $\overline{\boldsymbol{\theta}}_t$ (Line 8):

$$\boldsymbol{x}_{t,1} = \arg\max_{\boldsymbol{x}\in\mathcal{X}_t} h(\boldsymbol{x};\overline{\boldsymbol{\theta}}_t). \tag{9.10}$$

The second arm is then selected optimistically (Line 9):

$$\boldsymbol{x}_{t,2} = \arg\max_{\boldsymbol{x}\in\mathcal{X}_t} h(\boldsymbol{x};\overline{\boldsymbol{\theta}}_t) + \nu_T \left\|(\phi(\boldsymbol{x}) - \phi(\boldsymbol{x}_{t,1}))\right\|_{\boldsymbol{V}_{t-1}^{-1}}, \tag{9.11}$$

in which $\nu_T \triangleq \beta_T + B\sqrt{\frac{\lambda}{\kappa_\mu}} + 1$, $\beta_T \triangleq \frac{1}{\kappa_\mu}\sqrt{\widetilde{d} + 2\log(u/\delta)}$ and B is defined in Lemma 9.2.1. Here $\widetilde{d}$ denotes the *effective dimenision* which we will introduce in detail in Sec. 9.4.2.

Updating User Estimates and Interaction History (Line 10-Line 11). After recommending the pair of arms $\boldsymbol{x}_{t,1}$ and $\boldsymbol{x}_{t,2}$, we collect the preference feedback $y_t = \mathbb{1}(\boldsymbol{x}_{t,1} \succ \boldsymbol{x}_{t,2})$ and update interaction history: $\mathcal{D}_t = \{i_s, \boldsymbol{x}_{s,1}, \boldsymbol{x}_{s,2}, y_s\}_{s=1}^{t}$ (Line 10). Next, we update the parameters of the NN used to predict the reward for user i_t by minimizing the following loss function (Line 11):

$$\mathcal{L}_{i_t,t}(\boldsymbol{\theta}) = -\frac{1}{m_{\text{NN}}} \sum_{\substack{s\in[t-1] \\ i_s=i_t}} \big(y_s \log \mu\left(h(\boldsymbol{x}_{s,1};\boldsymbol{\theta}) - h(\boldsymbol{x}_{s,2};\boldsymbol{\theta})\right) + (1-y_s) \log \mu\left(h(\boldsymbol{x}_{s,2};\boldsymbol{\theta}) - h(\boldsymbol{x}_{s,1};\boldsymbol{\theta})\right)\big) + \frac{\lambda}{2}\left\|\boldsymbol{\theta} - \boldsymbol{\theta}_0\right\|_2^2 \quad (9.12)$$

to yield parameters $\hat{\boldsymbol{\theta}}_{i_t,t}$. The NN parameters for the other users remain unchanged.

Dynamic Graph Update (Line 12). Finally, we use the updated NN parameters $\hat{\boldsymbol{\theta}}_{i_t,t}$ for user i_t to reassess the similarity between user i_t and the other users. We remove the edge between (i_t, ℓ) from the graph G_{t-1} if the difference between $\hat{\boldsymbol{\theta}}_{i_t,t}$ and $\hat{\boldsymbol{\theta}}_{\ell,t}$ is large enough (Line 12). Intuitively, if the estimated reward functions (represented by the respective parameters of their NNs for reward prediction) between two users are significantly different, we separate these two users into different clusters. The updated graph $G_t = (\mathcal{U}, E_t)$ is then used in the following rounds.

9.4 Theoretical Analysis

In this section, we present the theoretical results regarding the regret guarantees of our proposed algorithms and provide a detailed discussion of these findings.

9.4.1 Clustering Of Linear Dueling Bandits (COLDB)

The following theorem provides an upper bound on the expected regret achieved by the COLDB algorithm (Algo. 13) under the linear setting.

Theorem 9.4.1. *Suppose that Assumptions 9.1, 9.2, 9.3 and 9.4 are satisfied. Then the expected regret of the COLDB algorithm (Algo. 13) for T rounds satisfies*

$$R(T) = O\Big(u\big(\frac{d}{\kappa_\mu^2 \tilde{\lambda}_x \gamma^2} + \frac{1}{\tilde{\lambda}_x^2}\big) \log T + \frac{1}{\kappa_\mu} d\sqrt{mT}\Big) \tag{9.13}$$

$$= O\Big(\frac{1}{\kappa_\mu} d\sqrt{mT}\Big), \tag{9.14}$$

where $\tilde{\lambda}_x \triangleq \int_0^{\lambda_x} (1 - e^{-\frac{(\lambda_x - x)^2}{2\sigma^2}})^C dx$ is the problem instance dependent constant [4, 5].

The proof of this theorem can be found in Appendix A.8. The regret bound in Eq.(9.13) consists of two terms. The first term accounts for the number of rounds required to accumulate sufficient information to correctly cluster all users with high probability, and it scales only logarithmically with the number of time steps T. The second term captures the regret after successfully clustering the users, which depends on the number of clusters m, rather than the potentially huge total number of users u. Notably, the regret upper bound is not only sub-linear in T, but also *becomes tighter when there is a smaller number of clusters* m, i.e., when a larger number of users belong to the same cluster on average. This provides a formal justification for the advantage of cross-user collaboration in our problem setting where only preference feedback is available.

In the special case where there is only one user ($m = 1$), the regret bound simplifies to $O(d\sqrt{T}/\kappa_\mu)$, which aligns with the classical results in the single-user linear dueling bandit literature [155, 157, 159]. Compared to the previous works on clustering of bandits with linear reward functions [46, 4, 136], our regret upper bound has an extra dependency on $1/\kappa_\mu$. Since $\kappa_\mu < 0.25$ for the logistic function, this dependency makes our regret upper bound larger and hence captures the more challenging nature of the preference feedback compared to the numerical feedback in classical clustering of linear bandits.

9.4.2 Clustering Of Neural Dueling Bandits (CONDB)

Let $\mathbf{H}' = \sum_{t=1}^{T} \sum_{(i,j)\in C_K^2} z_j^i(t) z_j^i(t)^\top \frac{1}{m_{\text{NN}}}$, in which $z_j^i(t) = g(\boldsymbol{x}_{t,i}; \boldsymbol{\theta}_0) - g(\boldsymbol{x}_{t,j}; \boldsymbol{\theta}_0)$ and C_K^2 denotes all pairwise combinations of K arms. Then, the effective dimension $\widetilde{d}$ is defined as follows [60]:

$$\widetilde{d} = \log\det\left(\frac{\kappa_\mu}{\lambda}\mathbf{H}' + \mathbf{I}\right). \tag{9.15}$$

The definition of $\widetilde{d}$ considers the contexts from all users and in all T rounds. The theorem below gives an upper bound on the expected regret of our CONDB algorithm (Algo. 19).

Theorem 9.4.2. *Suppose that Assumptions 9.1, 9.4, 9.5 and 9.6 are satisfied (let $\phi(\boldsymbol{x}) = (1/\sqrt{m_{NN}})g(\boldsymbol{x}; \boldsymbol{\theta}_0)$ in Assumption 9.4). As long as $m_{NN} \geq poly(T, L, K, 1/\kappa_\mu, L_\mu, 1/\lambda_0, 1/\lambda, \log(1/\delta))$, then the expected regret of the CONDB algorithm (Algo. 19) for T*

rounds satisfies

$$R_T = O\Big(u\big(\frac{\widetilde{d}}{\kappa_\mu^2 \widetilde{\lambda}_x \gamma^2} + \frac{1}{\widetilde{\lambda}_x^2}\big)\log T + \big(\frac{\sqrt{\widetilde{d}}}{\kappa_\mu} + B\sqrt{\frac{\lambda}{\kappa_\mu}}\big)\sqrt{\widetilde{d}mT}\Big) \tag{9.16}$$

$$= O\Big(\big(\frac{\sqrt{\widetilde{d}}}{\kappa_\mu} + B\sqrt{\frac{\lambda}{\kappa_\mu}}\big)\sqrt{\widetilde{d}mT}\Big)\,. \tag{9.17}$$

The proof of this theorem can be found in Appendix A.9. The first term in the regret bound in Eq. 9.16 has the same form as the first term in the regret bound of COLDB in Eq.(9.13), except that the input dimension d for COLDB (Eq.(9.13)) is replaced by the effective dimension $\widetilde{d}$ for CONDB (Eq.(9.16)). As discussed in [60], $\widetilde{d}$ is usually larger than the effective dimension in classical neural bandits [160, 161]. This dependency, together with the extra dependency on $1/\kappa_\mu$, reflects the added difficulty from the preference feedback compared to the more informative numerical feedback in classical neural bandits.

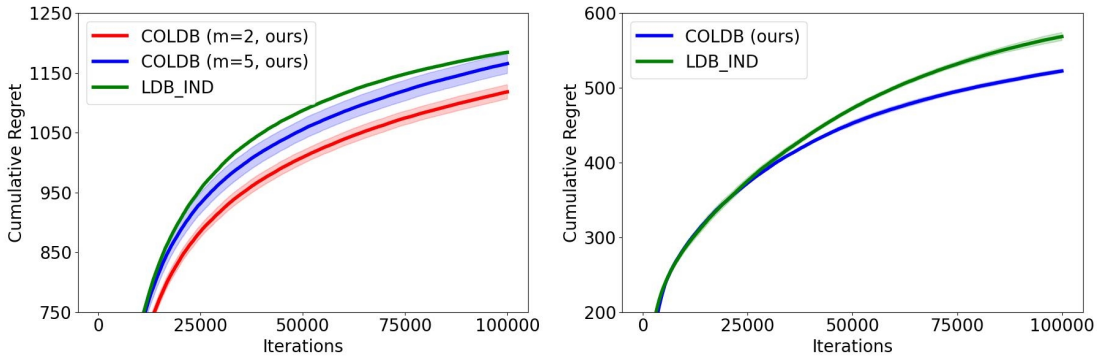
Similar to COLDB (Theorem 9.4.1), the first term in the regret upper bound of CONDB (Theorem 9.4.2) results from the number of rounds needed to collect enough observations to correctly identify the clustering structure. The second term corresponds to the regret of all users after the correct clustering structure is identified, which depends on the number of clusters m instead of the number of users u. Theorem 9.4.2 also shows that the regret upper bound of CONDB is sub-linear in T, and becomes improved as the number of users belonging to the same cluster is increased on average (i.e., when the number of clusters m is smaller). Moreover, in the special case where the number of clus-

ters is $m = 1$, the regret upper bound in Eq.(9.17) becomes the same as that of the standard neural dueling bandits [60].

9.5 Experimental Results

We use both synthetic and real-world experiments to evaluate the performance of our COLDB and CONDB algorithms. For both algorithms, we compare them with their corresponding single-user variant as the baseline. Specifically, for COLDB, we compare it with the baseline of LDB_IND, which refers to Linear Dueling Bandit (Independent) [157], meaning running independent classic linear dueling bandit algorithms for each user separately; similarly, for CONDB, we compare it with NDB_IND, which stands for Neural Dueling Bandit (Independent) [60].

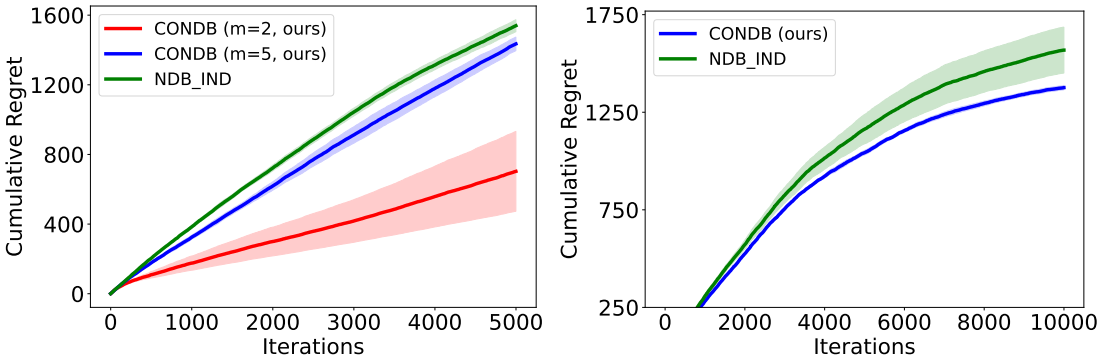
COLDB. Our experimental settings mostly follow the designs from the works on clustering of bandits [4, 136]. In our synthetic experiment for COLDB, we design a setting with linear reward functions: $f_i(\boldsymbol{x}) = \boldsymbol{\theta}_i^\top \boldsymbol{x}$. We choose $u = 200$ users, $K = 20$ arms and a feature dimension of $d = 20$, and construct two settings with $m = 2$ and $m = 5$ groundtruth clusters, respectively. In the experiment with the MovieLens dataset [228], we follow the experimental setting from [4], a setting with 200 users. Same as the synthetic experiment, we choose the number of arms in every round to be $K = 20$ and let the input feature dimension be $d = 20$. We construct a setting with $m = 5$ clusters. We repeat each experiment for three independent trials and report the mean $\pm$ standard error.



(a) Synthetic (b) MovieLens

Figure 9.1: Experimental results for our COLDB algorithm with a linear reward function.

Fig. 9.1 plots the cumulative regret of our COLDB and the baseline of LDB_IND. The results show that our COLDB algorithm significantly outperforms the baseline of LDB_IND in both the synthetic and real-world experiments. Moreover, Fig. 9.1 (a) demonstrates that when $m = 2$ (i.e., when a larger number of users belong to the same cluster on average), the performance of our COLDB is improved, which is *consisent with our theoretical results* (Sec. 9.4.1).



(a) Synthetic (b) MovieLens

Figure 9.2: Experimental results for our CONDB algorithm with a non-linear (square) reward function.

CONDB. We also construct both a synthetic and real-world experiment to evaluate our CONDB algorithm. Most of the experimental settings are the same as those of the COLDB algorithm described above. The major difference is that instead of using linear reward functions, here we adopt a non-linear reward function, i.e., a square function: $f_i(\boldsymbol{x}) = (\boldsymbol{\theta}_i^\top \boldsymbol{x})^2$. The results in this setting are plotted in Fig. 9.2. Our CONDB algorithm achieves significantly smaller cumulative regrets than the baseline algorithm of NDB_IND in both the synthetic and real-world experiments. Moreover, Fig. 9.2 (a) shows that the performance of our CONDB is improved when a larger number of users are in the same cluster on average, i.e., when $m = 2$. These results demonstrate the potential of our CONDB algorithm to excel in problems with complicated non-linear reward functions.

Algorithm 13 Clustering Of Linear Dueling Bandits (COLDB)

1: **Input:** $f(T_{i,t}) = \frac{\sqrt{\lambda/\kappa_\mu}+\sqrt{2\log(u/\delta)+d\log(1+4T_{i,t}\kappa_\mu/d\lambda)}}{\kappa_\mu\sqrt{2\tilde{\lambda}_x T_{i,t}}}$, regularization parameter $\lambda > 0$, confidence parameter $\beta_t \triangleq \sqrt{2\log(1/\delta)+d\log\left(1+tL^2\kappa_\mu/(d\lambda)\right)}$, $\kappa_\mu > 0$.

2: **Initialization:** $\boldsymbol{V}_0 = \boldsymbol{V}_{i,0} = \frac{\lambda}{\kappa_\mu}\mathbf{I}$, $\hat{\boldsymbol{\theta}}_{i,0} = \mathbf{0}$, $\forall i \in \mathcal{U}$, a complete Graph $G_0 = (\mathcal{U}, E_0)$ over $\mathcal{U}$.

3: **for** $t = 1, \ldots, T$ **do**

4: Receive the index of the current user $i_t \in \mathcal{U}$, and the current feasible arm set $\mathcal{X}_t$;

5: Find the connected component $\overline{C}_t$ for user i_t in the current graph G_{t-1} as the current cluster;

6: Estimate the common preference vector $\overline{\boldsymbol{\theta}}_t$ for the current cluster $\overline{C}_t$:

$$\overline{\boldsymbol{\theta}}_t = \text{argmin}_{\boldsymbol{\theta}} - \sum_{\substack{s\in[t-1]\\ i_s\in\overline{C}_t}} \left(y_s \log\mu\left(\boldsymbol{\theta}^\top[\phi(\boldsymbol{x}_{s,1})-\phi(\boldsymbol{x}_{s,2})]\right) + (1-y_s)\log\mu\left(\boldsymbol{\theta}^\top[\phi(\boldsymbol{x}_{s,2})-\phi(\boldsymbol{x}_{s,1})]\right)\right) + \frac{\lambda}{2}\|\boldsymbol{\theta}\|_2^2; \tag{9.6}$$

7: Calculate aggregated information matrix for cluster $\overline{C}_t$: $\boldsymbol{V}_{t-1} = \boldsymbol{V}_0 + \sum_{\substack{s\in[t-1]\\ i_s\in\overline{C}_t}} (\phi(\boldsymbol{x}_{s,1})-\phi(\boldsymbol{x}_{s,2}))(\phi(\boldsymbol{x}_{s,1})-\phi(\boldsymbol{x}_{s,2}))^\top$.

8: Choose the first arm $\boldsymbol{x}_{t,1} = \arg\max_{\boldsymbol{x}\in\mathcal{X}_t} \overline{\boldsymbol{\theta}}_t^\top \phi(\boldsymbol{x})$;

9: Choose the second arm $\boldsymbol{x}_{t,2} = \arg\max_{\boldsymbol{x}\in\mathcal{X}_t} \overline{\boldsymbol{\theta}}_t^\top (\phi(\boldsymbol{x})-\phi(\boldsymbol{x}_{t,1})) + \frac{\beta_t}{\kappa_\mu}\|\phi(\boldsymbol{x})-\phi(\boldsymbol{x}_{t,1})\|_{\boldsymbol{V}_{t-1}^{-1}}$;

10: Observe the preference feedback: $y_t = \mathbb{1}(\boldsymbol{x}_{t,1} \succ \boldsymbol{x}_{t,2})$, and update history: $\mathcal{D}_t = \{i_s, \boldsymbol{x}_{s,1}, \boldsymbol{x}_{s,2}, y_s\}_{s=1,\ldots,t}$;

11: Update the estimation for the current served user i_t:

$$\hat{\boldsymbol{\theta}}_{i_t,t} = \arg\min_{\boldsymbol{\theta}} - \sum_{\substack{s\in[t-1]\\ i_s=i_t}} \left(y_s \log\mu\left(\boldsymbol{\theta}^\top[\phi(\boldsymbol{x}_{s,1})-\phi(\boldsymbol{x}_{s,2})]\right) + (1-y_s)\log\mu\left(\boldsymbol{\theta}^\top[\phi(\boldsymbol{x}_{s,2})-\phi(\boldsymbol{x}_{s,1})]\right)\right) + \frac{\lambda}{2}\|\boldsymbol{\theta}\|_2^2, \tag{9.7}$$

keep the estimations of other users unchanged;

12: Delete the edge $(i_t, \ell) \in E_{t-1}$ if

$$\left\|\hat{\boldsymbol{\theta}}_{i_t,t} - \hat{\boldsymbol{\theta}}_{\ell,t}\right\|_2 > f(T_{i_t,t}) + f(T_{\ell,t}) \tag{9.8}$$

13: **end for**

Chapter 10

Conclusion and Future Work

In this chapter, we summarize the thesis and list some future directions that could inspire the follow-up works.

In Chapter 3, we presented a minimalist approach for achieving horizon-free and second-order regret bounds in RL: simply train transition models via Maximum Likelihood Estimation followed by optimistic or pessimistic planning, depending on whether we operate in the online or offline learning mode. Our horizon-free bounds for general function approximation look quite similar to the bounds in Contextual bandits, indicating that the need for long-horizon planning does not make RL harder than CB from a statistical perspective.

Our work has some limitations. First, when extending our result to continuous function class, we pay $\ln(H)$. This $\ln(H)$ is coming from a naive application of the ϵ-net/bracket argument to the generalization bounds of MLE. We conjecture that this $\ln(H)$ can be elimiated by using a more careful analysis that uses techiniques such as peeling/chaining [270, 271]. We leave this as an important future direction. Second, while our model-

based framework is quite general, it cannot capture problems that need to be solved via model-free approaches such as linear MDPs [272]. An interesting future work is to see if we can develop the corresponding model-free approaches that can achieve horizon-free and instance-dependent bounds for RL with general function approximation. Finally, the algorithms studied in this work are not computationally tractable. This is due to the need of performing optimism/pessimism planning for exploration. Deriving computationally tractacle RL algorithms for the rich function approximation setting is a long-standing question.

In Chapter 4, we study the zero-shot generalization (ZSG) performance of offline reinforcement learning (RL). We propose two offline RL frameworks, pessimistic empirical risk minimization and pessimistic proximal policy optimization, and show that both of them can find the optimal policy with ZSG ability. We also show that such a generalization property does not hold for offline RL without knowing the context information of the environment, which demonstrates the necessity of our proposed new algorithms. Currently, our theorems and algorithm design depend on the i.i.d. assumption of the environment selection. How to relax such an assumption remains an interesting future direction.

In Chapter 5, we present a new problem of clustering of bandits with misspecified user models (CBMUM), where the agent has to adaptively assign appropriate clusters for users under model misspecifications. We propose two robust CB algorithms, RCLUMB and RSCLUMB. Under milder assumptions than previous CB works, we prove the regret bounds of our algorithms, which match

the lower bound asymptotically in T up to logarithmic factors, and match the state-of-the-art results in several degenerate cases. It is challenging to bound the regret caused by *misclustering* users with close but not the same preference vectors and use inaccurate cluster-based information to select arms. Our analysis to bound this part of the regret is quite general and may be of independent interest. Experiments on synthetic and real-world data demonstrate the advantage of our algorithms. We would like to state some interesting future works: (1) Prove a tighter regret lower bound for CBMUM, (2) Incorporate recent model selection methods into our fundamental framework to design robust algorithms for CBMUM with unknown exact maximum model misspecification level, and (3) Consider the setting with misspecifications in the underlying user clustering structure rather than user models.

In Chapter 6, we are the first to propose the novel LOCUD problem, where there are many users with *unknown* preferences and *unknown* relations, and some corrupted users can occasionally perform disrupted actions to fool the agent. Hence, the agent not only needs to learn the *unknown* user preferences and relations robustly from potentially disrupted bandit feedback, balance the exploration-exploitation trade-off to minimize regret, but also needs to detect the corrupted users over time. To robustly learn and leverage the *unknown* user preferences and relations from corrupted behaviors, we propose a novel bandit algorithm RCLUB-WCU. To detect the corrupted users in the online bandit setting, based on the learned user relations of RCLUB-WCU, we propose a novel detection algorithm OCCUD. We prove a regret upper bound for RCLUB-WCU, which matches the lower bound

asymptotically in T up to logarithmic factors and matches the state-of-the-art results in degenerate cases. We also give a theoretical guarantee for the detection accuracy of OCCUD. Extensive experiments show that our proposed algorithms achieve superior performance over previous bandit algorithms and high corrupted user detection accuracy.

In Chapter 7, we introduce ConLinUCB, a general framework for conversational bandits with efficient information incorporation. Based on this framework, we propose ConLinUCB-BS and ConLinUCB-MCR, with explorative key-term selection strategies that can quickly elicit the user's potential interests. We prove tight regret bounds of our algorithms. Particularly, ConLinUCB-BS achieves a bound of $O(d\sqrt{T \log T})$, much better than $O(d\sqrt{T} \log T)$ of the classic ConUCB. In the empirical evaluations, our algorithms dramatically outperform the classic ConUCB. For future work, it would be interesting to consider the settings with knowledge graphs [182], hierarchy item trees [273], relative feedback [181] or different feedback selection strategies [274, 275], and use our framework and principles to improve the performance of existing algorithms.

In Chapter 8, we study non-stationary stochastic linear bandits in this work. We establish the first variance-dependent regret lower bound for non-stationary linear bandits, which captures the interplay between variance, non-stationarity, and dimensionality in the linear bandit setting, offering new insights into the complexity of this problem. We propose Restarted-WeightedOFUL$^+$ and Restarted SAVE$^+$, two algorithms that utilize the dynamic variance information of the dynamic reward distribution. We

show that both of our algorithms are able to achieve better dynamic regret compared with best existing results [194] under several parameter regimes, *e.g.*, when the total variance V_K is small. Experiment results backup our theoretical claim. It is worth noting there still exist gaps between our current obtained regret and the lower bound, and to fix such a gap leaves as our future work.

In Chapter 9, we introduce the first clustering of dueling bandit algorithms for both linear and non-linear latent reward functions, which enhance the performance of MAB with preference feedback via cross-user collaboraiton. Our algorithms estimates the clustering structure online based on the estimated reward function parameters, and employs the data from all users within the same cluster to select the pair of arms to query for preference feedback. We derive upper bounds on the cumulative regret of our algorithms, which show that our algorithms enjoy theoretically guaranteed improvement when a larger number of users belong to the same cluster on average. We also use synthetic and real-world experiments to validate our theoretical findings.

Appendix A

Appendices

A.1 Appendix for Chapter 3

A.1.1 Summary of Contents in the Appendix

The Appendix is organized as follows.

In Appendix A.1.2, we provide some new analyses for Eluder dimension, which we will use for proving the regret bounds for the online RL setting.

In Appendix A.1.3, we provide some other supporting lemmas that will be used in our proofs.

In Appendix A.1.4, we provide the detailed proofs for the online RL setting (Section 3.3). Specifically, in Appendix A.1.5 we give the proof of Theorem 3.3.2; in Appendix A.1.6, we show the proof of Corollary 3.2; in Appendix A.1.7, we give the proof of Corollary 3.3.

In Appendix A.1.8, we provide the detailed proofs for the offline RL setting (Section 3.4). Specifically, in Appendix A.1.9 we give the proof of Theorem 3.4.1; in Appendix A.1.10, we show the proof of Corollary 3.4; in Appendix A.1.11, we give the proof of Corollary 3.5; in Appendix A.1.12, we show the proof of the claim in Example 2.

A.1.2 Analysis regarding the Eluder Dimension

For simplicity, we denote $x_h^k = (s_h^k, a_h^k)$.

First we have two technical lemma. The first lemma bounds the summation of "self-normalization" terms by the Eluder dimension. Our result generalizes the previous result by [276] from the ℓ_2-Eluder dimension to the ℓ_1 case.

Lemma A.1.1. *Suppose for all $g \in \Psi, |g| \leq 1$ and $\lambda > 1$, then we have*

$$\sum_{k=1}^{K}\sum_{h=1}^{H}\min\left\{1, \sup_{g\in\Psi}\frac{|g(x_h^k)|}{\sum_{k'=1}^{k-1}\sum_{h'=1}^{H}|g(x_{h'}^{k'})| + \sum_{h'=1}^{h-1}|g(x_{h'}^{k})| + \lambda}\right\}$$
$$\leq 12\log^2(4\lambda KH)\cdot DE_1(\Psi, \mathcal{S}\times\mathcal{A}, 1/(8\lambda KH)) + \lambda^{-1}.$$

Proof [Proof of Lemma A.1.1] We follow the proof steps of Theorem 4.6 in [276]. For simplicity, we use $n = KH$, $i = kH + h$ to denote the indices and denote x_h^k by x_i. Then we need to prove

$$\sum_{i=1}^{n}\min\left\{1, \sup_{g\in\Psi}\frac{|g(x_i)|}{\sum_{t=1}^{i-1}|g(x_t)| + \lambda}\right\}$$
$$\leq 12\log^2(4\lambda n)\cdot DE_1(\Psi, \mathcal{S}\times\mathcal{A}, 1/(8\lambda n)) + \lambda^{-1} \tag{A.1}$$

Let

$$g_i = \operatorname*{argmax}_{g\in\Psi}\frac{|g(x_i)|}{\sum_{j=1}^{i-1}|g(x_j)| + \lambda} \tag{A.2}$$

For any $1/(\lambda n) \leq \rho \leq 1$ and $1 \leq j \leq \lceil\log(4\lambda n)\rceil$, we define

$$A_\rho^j = \left\{i \in [n] : 2^{-j} < |g_i(x_i)| \leq 2^{-j+1}, \frac{|g_i(x_i)|}{\sum_{t=1}^{i-1}|g_i(x_t)| + \lambda} \geq \rho/2\right\},$$
$$d_j := DE_1(\Psi, \mathcal{S}\times\mathcal{A}, 2^{-j}). \tag{A.3}$$

Next we only consider the set A_ρ^j where $|A_\rho^j| > d_j$. We denote $A_\rho^j = \{a_1, \ldots, a_A\}$, where $A = |A_\rho^j|$ and $\{a_i\}$ keeps the same order as $\{x_i\}$. Next we do the following constructions. We maintain $k = \lfloor(A-1)/d_j\rfloor$ number of queues $Q_1, \ldots, Q_k$, all of them initialized as emptysets. We put a_1 into Q_1. For $a_i, i \geq 2$, we put a_i into Q_l, where Q_l is the first queue where a_i is 2^{-j}-independent of all

elements in Q_l. Let $i_{\max}$ be the smallest i when we can not put a_i into any existing queue.

We claim that $i_{\max}$ indeed exists, i.e., our construction will stop before we put all elements in A_ρ^j into $Q_1, \ldots, Q_k$. In fact, note the fact that the length of each Q_l is always no more than d_j, which is due to the fact that any 2^{-j}-independent sequence's length is at most d_j. Meanwhile, since we only have $k = \lfloor (A-1)/d_j \rfloor$, then the amount of elements in $Q_1 \cup \cdots \cup Q_k$ will be upper bounded by $k \cdot d_j < A$. That suggests at least one element in A_ρ^j is not contained by $Q_1 \cup \cdots \cup Q_k$, i.e., $i_{\max}$ exists.

By the definition of $i_{\max}$, we know that $a_{i_{\max}}$ is 2^{-j}-dependent to each Q_l. Next we give a bound of A. First, note

$$\sum_{t=1}^{i_{\max}-1} |g_{i_{\max}}(x_t)| \geq \sum_{t \in Q_1 \cup \cdots \cup Q_k} |g_{i_{\max}}(a_t)| = \sum_{l=1}^{k} \sum_{t \in Q_l} |g_{i_{\max}}(a_t)| > k \cdot 2^{-j}, \tag{A.4}$$

where the first inequality holds since Q_l are the elements that appear before $a_{i_{\max}}$, the second one holds due to the following induction of Eluder dimension: since $a_{i_{\max}}$ is 2^{-j}-dependent to Q_l, then we have

$$\forall g \in \Psi, \sum_{t \in Q_l} |g(a_t)| \leq 2^{-j} \Rightarrow |g(a_{i_{\max}})| \leq 2^{-j}. \tag{A.5}$$

Therefore, given the fact $|g_{i_{\max}}(a_{i_{\max}})| > 2^{-j}$ (recall the definition of A_ρ^j), we must have $\sum_{t \in Q_l} |g_{i_{\max}}(a_t)| > 2^{-j}$ as well, which suggests the second inequality of Equation A.4 holds. Second, we have

$$\sum_{t=1}^{i_{\max}-1} |g_{i_{\max}}(x_t)| \leq 2/\rho \cdot |g_{i_{\max}}(a_{i_{\max}})| \leq 4 \cdot 2^{-j}/\rho, \tag{A.6}$$

where both inequalities hold due to the definition of A_ρ^j. Combining Equation A.4 and Equation A.6, we have

$$k < 4/\rho \Rightarrow A \leq 4d_j/\rho + d_j \leq 5d_j/\rho. \tag{A.7}$$

Therefore, we have that for all ρ, j, $|A_\rho^j| \leq 5d_j/\rho$.

Finally we prove Equation A.1. $1/(\lambda n) \leq \rho \leq 1$ and $1 \leq j \leq \lceil \log(4\lambda n) \rceil = J$. Denote

$$A_\rho = \left\{ i \in [n] : \frac{|g_i(x_i)|}{\sum_{t=1}^{i-1} |g_i(x_t)| + \lambda} \geq \rho/2 \right\}. \tag{A.8}$$

Then it is easy to notice that $|A_\rho| = \sum_j |A_\rho^j| \leq \lceil \log(4\lambda n) \rceil \cdot 5d_J/\rho$, where we use the fact that the Eluder dimension d_j is increasing. Therefore, by the standard peeling technique, we have

$$\begin{aligned}
&\sum_{i=1}^{n} \min \left\{ 1, \sup_{g \in \Psi} \frac{|g(x_i)|}{\sum_{t=1}^{i-1} |g(x_t)| + \lambda} \right\} \\
&= \sum_{j \in [\lceil \log(\lambda n) \rceil]} \sum_{i \in A_{2^{-j}} \setminus A_{2^{-j+1}}} + \sum_{j = \lceil \log(\lambda n) \rceil} \sum_{i \notin A_{2^{-j+1}}} \\
&\leq \sum_{j \in [\lceil \log(\lambda n) \rceil]} \sum_{i \in A_{2^{-j}} \setminus A_{2^{-j+1}}} 2^{-j-1} + n \cdot 1/(\lambda n) \\
&\leq \sum_{j \in [\lceil \log(\lambda n) \rceil]} \sum_{i \in A_{2^{-j}}} 2^{-j-1} + n \cdot 1/(\lambda n) \\
&\leq \lceil \log(\lambda n) \rceil \cdot \lceil \log(4\lambda n) \rceil \cdot 3d_J + \lambda^{-1},
\end{aligned}$$

which concludes our proof.

Next lemma gives a bound to bound the number of episodes where the behavior along these episodes are "bad". Intuitively speaking, our lemma suggests we only have limited number of bad episodes, therefore won't affect the final performance of our algorithm.

Lemma A.1.2. *Given $\lambda > 1$. There exists at most*

$$13\log^2(4\lambda KH)\cdot DE_1(\Psi, \mathcal{S}\times\mathcal{A}, 1/(8\lambda KH)) \tag{A.9}$$

number of $k \in [K]$ satisfying the following claim

$$\sup_{g\in\Psi}\frac{\lambda+\sum_{k'=1}^{k}\sum_{h'=1}^{H}|g(x_{h'}^{k'})|}{\lambda+\sum_{k'=1}^{k-1}\sum_{h'=1}^{H}|g(x_{h'}^{k'})|} > 4. \tag{A.10}$$

Proof[Proof of Lemma A.1.2]

Note that

$$\begin{aligned}
&\sum_{k=1}^{K}\min\left\{2, \log\sup_{g\in\Psi}\frac{\lambda+\sum_{k'=1}^{k}\sum_{h'=1}^{H}|g(x_{h'}^{k'})|}{\lambda+\sum_{k'=1}^{k-1}\sum_{h'=1}^{H}|g(x_{h'}^{k'})|}\right\}\\
&\le \sum_{k=1}^{K}\min\left\{2, \log\prod_{h=1}^{H}\sup_{g\in\Psi}\frac{\lambda+\sum_{k'=1}^{k-1}\sum_{h'=1}^{H}|g(x_{h'}^{k'})|+\sum_{h'=1}^{h}|g(x_{h'}^{k})|}{\lambda+\sum_{k'=1}^{k-1}\sum_{h'=1}^{H}|g(x_{h'}^{k'})|+\sum_{h'=1}^{h-1}|g(x_{h'}^{k})|}\right\}\\
&= \sum_{k=1}^{K}\min\left\{2, \sum_{h=1}^{H}\log\left(1+\sup_{g\in\Psi}\frac{|g(x_h^k)|}{\lambda+\sum_{k'=1}^{k-1}\sum_{h'=1}^{H}|g(x_{h'}^{k'})|+\sum_{h'=1}^{h-1}|g(x_{h'}^{k})|}\right)\right\}\\
&\le \sum_{k=1}^{K}\sum_{h=1}^{H}\min\left\{2, \sup_{g\in\Psi}\frac{|g(x_h^k)|}{\sum_{k'=1}^{k-1}\sum_{h'=1}^{H}|g(x_{h'}^{k'})|+\sum_{h'=1}^{h-1}|g(x_{h'}^{k})|+\lambda}\right\},\\
&\le 2\sum_{k=1}^{K}\sum_{h=1}^{H}\min\left\{1, \sup_{g\in\Psi}\frac{|g(x_h^k)|}{\sum_{k'=1}^{k-1}\sum_{h'=1}^{H}|g(x_{h'}^{k'})|+\sum_{h'=1}^{h-1}|g(x_{h'}^{k})|+\lambda}\right\}\\
&\le 24\log^2(4\lambda KH)\cdot DE_1(\Psi, \mathcal{S}\times\mathcal{A}, 1/(8\lambda KH)) + 2\lambda^{-1}\\
&\le 26\log^2(4\lambda KH)\cdot DE_1(\Psi, \mathcal{S}\times\mathcal{A}, 1/(8\lambda KH)).
\end{aligned} \tag{A.11}$$

where the first inequality holds since $\sup_g\prod f(g) \le \prod\sup_g f(g)$, the second one holds since $\log(1+x)\le x$, the fourth one holds due to Lemma A.1.1. Therefore, there are at most

$$26\log^2(4\lambda KH)\cdot DE_1(\Psi, \mathcal{S}\times\mathcal{A}, 1/(8\lambda KH))/2$$

number of k satisfying

$$\log \sup_{g\in\Psi} \frac{\lambda + \sum_{k'=1}^{k}\sum_{h'=1}^{H} |g(x_{h'}^{k'})|}{\lambda + \sum_{k'=1}^{k-1}\sum_{h'=1}^{H} |g(x_{h'}^{k'})|} > 2,$$

which concludes the proof.

We next have the following lemma, which bounds the regret by the Eluder dimension.

Lemma A.1.3 (Theorem 5.3, [208]). *Let $C := \sup_{(s,a)\in\mathcal{S}\times\mathcal{A}, f\in\Psi} |f((s,a))|$ be the envelope. For any sequences $f^{(1)}, \ldots, f^{(N)} \subseteq \Psi$, $(s,a)^{(1)}, \ldots, (s,a)^{(N)} \subseteq \mathcal{S}\times\mathcal{A}$, let β be a constant such that for all $n \in [N]$ we have, $\sum_{i=1}^{n-1} \left|f^{(n)}((s,a)^i)\right| \le \beta$. Then, for all $n \in [N]$, we have*

$$\sum_{t=1}^{n} \left|f^{(t)}((s,a)^t)\right| \le \inf_{0<\epsilon\le 1} \left\{DE_1(\Psi, \mathcal{S}\times\mathcal{A}, \epsilon)(2C + \beta\log(C/\epsilon)) + n\epsilon\right\}.$$

Given Lemma A.1.2 and Lemma Lemma A.1.3, we are able to prove the following key lemma.

Lemma A.1.4 (New Eluder Pigeon Lemma). *Let the event $\mathcal{E}$ be*

$$\mathcal{E} : \forall k \in [K],\ \sum_{i=1}^{k-1}\sum_{h=1}^{H} \mathbb{H}^2(\widehat{P}^k(s_h^i, a_h^i) || P^*(s_h^i, a_h^i)) \le \eta. \quad \text{(A.12)}$$

Then under event $\mathcal{E}$, there exists a set $\mathcal{K} \in [K]$ such that

- *We have $|\mathcal{K}| \le 13\log^2(4\eta KH)\cdot DE_1(\Psi, \mathcal{S}\times\mathcal{A}, 1/(8\eta KH))$.*
- *We have*

$$\begin{aligned}
&\sum_{k\in[K]\setminus\mathcal{K}}\sum_{h=1}^{H} \mathbb{H}^2\Big(P^\star(s_h^k, a_h^k) \parallel \widehat{P}^k\big(s_h^k, a_h^k\big)\Big) \\
&\le \inf_{0<\epsilon\le 1} \left\{DE_1(\Psi, \mathcal{S}\times\mathcal{A}, \epsilon)(2 + 7\eta\log(1/\epsilon)) + KH\epsilon\right\} \\
&\le DE_1(\Psi, \mathcal{S}\times\mathcal{A}, 1/KH)(2 + 7\eta\log(KH)) + 1\,, \quad \text{(A.13)}
\end{aligned}$$

where the function class $\Psi = \{(s,a) \mapsto \mathbb{H}^2(P^\star(s,a) \parallel P(s,a)) : P \in \mathcal{P}\}$.

Proof[Proof of Lemma A.1.4] We interchangeably use $n = kH+h$ to denote the indices of s_h^k, a_h^k. We set $f^{(n)}((s,a))$ in Lemma A.1.3 as $H^2(P^k(s,a)||P^*(s,a))$.

First, we prove that the β in Lemma A.1.3 can be selected as 7η under event $\mathcal{E}$. To show that, let $\mathcal{K}$ denote all the k stated in Lemma A.1.2. Then for all k such that $k+1 \notin \mathcal{K}$, $h = 2, ..., H$, let $n = kH + h$, we have

$$\begin{aligned}
\sum_{i=0}^{n-1} \left| f^{(n)}((s,a)^i) \right| &\leq \sum_{i=0}^{kH+H} \left| f^{(n)}((s,a)^i) \right| \\
&= \left(\lambda + \sum_{i=0}^{kH} \left| f^{(n)}((s,a)^i) \right| \right) \cdot \frac{\sum_{i=0}^{kH+H} \left| f^{(n)}((s,a)^i) \right| + \lambda}{\sum_{i=0}^{kH} \left| f^{(n)}((s,a)^i) \right| + \lambda} - \lambda \\
&\leq \left(\lambda + \sum_{i=0}^{kH} \left| f^{(n)}((s,a)^i) \right| \right) \cdot 4 - \lambda \\
&\leq 7\eta, \qquad \text{(A.14)}
\end{aligned}$$

where the second inequality holds due to Lemma A.1.2, the last one holds due to the definition of $\mathcal{E}$. Therefore, we prove our lemma by the conclusion of Lemma A.1.3 with $\beta = 7\eta$.

A.1.3 Other Supporting Lemmas

Lemma A.1.5 (Simulation Lemma ([277]))**.** *We have*

$$V_{0;P^\star}^\pi - V_{0;\hat{P}}^\pi \leq \sum_{h=0}^{H-1} \mathbb{E}_{s,a \sim d_h^\pi} \left[\left| \mathbb{E}_{s' \sim P^\star(s,a)} V_{h+1;\widehat{P}}^\pi(s') - \mathbb{E}_{s' \sim \widehat{P}(s,a)} V_{h+1;\widehat{P}}^\pi(s') \right| \right].$$

Lemma A.1.6 (Change of Variance Lemma (Lemma C.5 in [278]))**.**

$$\sum_{h=0}^{H-1} \mathbb{E}_{s,a\sim d_h^\pi}\left[\left(\mathbb{V}_{P^\star} V_{h+1;P^\star}^\pi\right)(s,a)\right] = \mathrm{VaR}_\pi.$$

Lemma A.1.7 (Generalization bounds of MLE for finite model class (Theorem E.4 in [208]))**.** *Let $\mathcal{X}$ be the context/feature space and $\mathcal{Y}$ be the label space, and we are given a dataset $D = \{(x_i, y_i)\}_{i\in[n]}$ from a martingale process: for $i = 1, 2, ..., n$, sample $x_i \sim \mathcal{D}_i(x_{1:i-1}, y_{1:i-1})$ and $y_i \sim p(\cdot \mid x_i)$. Let $f^\star(x, y) = p(y \mid x)$ and we are given a realizable,* i.e., *$f^\star \in \mathcal{F}$, function class $\mathcal{F} : \mathcal{X} \times \mathcal{Y} \to \Delta(\mathbb{R})$ of distributions. Suppose $\mathcal{F}$ is finite. Fix any $\delta \in (0, 1)$, set $\beta = \log(|\mathcal{F}|/\delta)$ and define*

$$\widehat{\mathcal{F}} = \left\{ f \in \mathcal{F} : \sum_{i=1}^n \log f(x_i, y_i) \geq \max_{\widetilde{f}\in\mathcal{F}} \sum_{i=1}^n \log \widetilde{f}(x_i, y_i) - 4\beta \right\}.$$

Then w.p. at least $1 - \delta$, the following holds:

(1) The true distribution is in the version space, i.e., *$f^\star \in \widehat{\mathcal{F}}$.*

(2) Any function in the version space is close to the ground truth data-generating distribution, i.e., *for all $f \in \widehat{\mathcal{F}}$*

$$\sum_{i=1}^n \mathbb{E}_{x\sim\mathcal{D}_i}\left[\mathbb{H}^2(f(x,\cdot) \parallel f^\star(x,\cdot))\right] \leq 22\beta.$$

Lemma A.1.8 (Generalization bounds of MLE for infinite model class (Theorem E.5 in [208]))**.** *Let $\mathcal{X}$ be the context/feature space and $\mathcal{Y}$ be the label space, and we are given a dataset $D = \{(x_i, y_i)\}_{i\in[n]}$ from a martingale process: for $i = 1, 2, ..., n$, sample $x_i \sim \mathcal{D}_i(x_{1:i-1}, y_{1:i-1})$ and $y_i \sim p(\cdot \mid x_i)$. Let $f^\star(x, y) = p(y \mid x)$ and we are given*

a realizable, i.e., $f^\star \in \mathcal{F}$, *function class* $\mathcal{F} : \mathcal{X} \times \mathcal{Y} \to \Delta(\mathbb{R})$ *of distributions. Suppose* $\mathcal{F}$ *is finite. Fix any* $\delta \in (0,1)$, *set* $\beta = \log(\mathcal{N}_{[]}((n|\mathcal{Y}|)^{-1}, \mathcal{F}, \|\cdot\|_\infty)/\delta)$ *(where* $\mathcal{N}_{[]}((n|\mathcal{Y}|)^{-1}, \mathcal{F}, \|\cdot\|_\infty)$ *is the bracketing number defined in Definition 3.2) and define*

$$\widehat{\mathcal{F}} = \left\{ f \in \mathcal{F} : \sum_{i=1}^{n} \log f(x_i, y_i) \geq \max_{\widetilde{f} \in \mathcal{F}} \sum_{i=1}^{n} \log \widetilde{f}(x_i, y_i) - 7\beta \right\}.$$

Then w.p. at least $1-\delta$, *the following holds:*

(1) The true distribution is in the version space, i.e., $f^\star \in \widehat{\mathcal{F}}$.

(2) Any function in the version space is close to the ground truth data-generating distribution, i.e., *for all* $f \in \widehat{\mathcal{F}}$

$$\sum_{i=1}^{n} \mathbb{E}_{x \sim \mathcal{D}_i} \left[\mathbb{H}^2(f(x,\cdot) \parallel f^\star(x,\cdot)) \right] \leq 28\beta.$$

Lemma A.1.9 (Recursion Lemma)**.** *Let* $G > 0$ *be a positive constant,* $a < G/2$ *is also a positive constant, and let* $\{C_m\}_{m=0}^{N=\lceil \log_2(\frac{KH}{G}) \rceil}$ *be a sequence of positive real numbers satisfying:*

[1] $C_m \leq 2^m G + \sqrt{aC_{m+1}} + a$ *for all* $m \geq 0$,

[2] $C_m \leq KH$ *for all* $m \geq 0$, *where* $K > 0$ *and* $H > 0$ *are positive constants.*

Then, it holds that:

$$C_0 \leq 4G.$$

Proof[Proof of Lemma A.1.9] We will prove by induction that for all $m \geq 0$,

$$C_m \leq 2^{m+2} G.$$

Then, for $m = 0$, this would immediately show $C_0 \leq 4G$.

1. The base case $m = N$:

Since $N = \lceil \log_2(\frac{KH}{G}) \rceil$, it is obvious that $2^{N+2}G \geq KH$. Thus, $C_N \leq KH \leq 2^{N+2}G$, the inequality holds for $m = N$.

2. The induction step:

Assume that for some $m \geq 0$, for C_{m+1}, we have:

$$C_{m+1} \leq 2^{m+1+2}G = 2^{m+3}G.$$

Then, we have

$$\begin{aligned} C_m &\leq 2^m G + \sqrt{aC_{m+1}} + a \\ &\leq 2^m G + \sqrt{a2^{m+3}G} + a \\ &\leq 2^m G + \sqrt{\frac{G}{2} \cdot 2^{m+3}G} + \frac{G}{2} \\ &= G \cdot (2^m + 2^{m/2+1} + 2^{-1}) \\ &\leq G \cdot (2^m + 2^{m+1} + 2^m) \\ &= 2^{m+2}G\,. \end{aligned} \tag{A.15}$$

Therefore, by induction, we have for all $m \geq 0$,

$$C_m \leq 2^{m+2}G.$$

And the proof follows by setting $m = 0$.

A.1.4 Detailed Proofs for the online setting in Section 3.3

A.1.5 Proof of Theorem 3.3.2

The following is the full proof of Theorem 3.3.2.

For notational simplicity, throughout this whole section, we denote

$$A := \sum_{k\in[K-1]\setminus\mathcal{K}}\sum_{h=0}^{H-1}\left[\left(\mathbb{V}_{P^\star}V^{\pi^k}_{h+1;\widehat{P}^k}\right)(s^k_h,a^k_h)\right]$$

$$B := \sum_{k\in[K-1]\setminus\mathcal{K}}\sum_{h=0}^{H-1}\left[\left(\mathbb{V}_{P^\star}V^{\pi^k}_{h+1}\right)(s^k_h,a^k_h)\right],$$

$$C_m := \sum_{k\in[K-1]\setminus\mathcal{K}}\sum_{h=0}^{H-1}\left[\left(\mathbb{V}_{P^\star}(V^{\pi^k}_{h+1;\widehat{P}^k}-V^{\pi^k}_{h+1})^{2^m}\right)(s^k_h,a^k_h)\right]$$

$$G := \sqrt{\sum_{k\in[K-1]\setminus\mathcal{K}}\sum_{h=0}^{H-1}\left[\left(\mathbb{V}_{P^\star}V^{\pi^k}_{h+1;\widehat{P}^k}\right)(s^k_h,a^k_h)\right]\cdot \mathrm{DE}_1(\Psi,\mathcal{S}\times\mathcal{A},1/KH)\cdot\log(K\left|\mathcal{P}\right|/\delta)\log(KH)}$$
$$+\,\mathrm{DE}_1(\Psi,\mathcal{S}\times\mathcal{A},1/KH)\cdot\log(K\left|\mathcal{P}\right|/\delta)\log(KH)$$

$$I^k_h := \mathbb{E}_{s'\sim P^*(s^k_h,a^k_h)}V^{\pi^k}_{h+1;\widehat{P}^k}(s') - V^{\pi^k}_{h+1;\widehat{P}^k}(s^k_{h+1}) \tag{A.16}$$

We use $\mathbb{I}\{\cdot\}$ to denote the indicator function. We define the following events which we will later show that they happen with high probability.

$$\mathcal{E}_1 := \{\forall k\in[K-1]: P^\star\in\widehat{\mathcal{P}}^k, \text{and}\sum_{i=0}^{k-1}\sum_{h=0}^{H-1}\mathbb{H}^2(P^\star(s^i_h,a^i_h)||\widehat{P}^k(s^i_h,a^i_h))\le 22\log(K\left|\mathcal{P}\right|/\delta).\}, \tag{A.17}$$

$$\mathcal{E}_2 := \Big\{\sum_{k\in[K-1]\setminus\mathcal{K}}\sum_{h=0}^{H-1}I^k_h \lesssim \sqrt{\sum_{k\in[K-1]\setminus\mathcal{K}}\sum_{h=0}^{H-1}\left(\mathbb{V}_{P^\star}V^{\pi^k}_{h+1;\widehat{P}^k}\right)(s^k_h,a^k_h)\log(1/\delta)}+\log(1/\delta)\Big\}, \tag{A.18}$$

$$\mathcal{E}_3 := \mathcal{E}_1\cap\{\forall m\in[0,\lceil\log_2(\frac{KH}{G})\rceil]: C_m\lesssim 2^m G+\sqrt{\log(1/\delta)\cdot C_{m+1}}+\log(1/\delta)\}, \tag{A.19}$$

$$\mathcal{E}_4 := \{\sum_{k=0}^{K-1}\sum_{h=0}^{H-1}\left[\left(\mathbb{V}_{P^\star}V^{\pi^k}_{h+1}\right)(s^k_h,a^k_h)\right]\lesssim\sum_{k=0}^{K-1}\mathrm{VaR}_{\pi^k}+\log(1/\delta)\}, \tag{A.20}$$

$$\mathcal{E}_5 := \{\sum_{k=0}^{K-1}\sum_{h=1}^{H}r(s^k_h,a^k_h)-\sum_{k=0}^{K-1}V^{\pi^k}_{0;P^*}\lesssim\sqrt{\sum_{k=0}^{K-1}\mathrm{VaR}_{\pi^k}\log(1/\delta)}+\log(1/\delta)\}, \tag{A.21}$$

$$\mathcal{E} := \mathcal{E}_2\cap\mathcal{E}_3\cap\mathcal{E}_4\cap\mathcal{E}_5. \tag{A.22}$$

First, by the realizability assumption, the standard generaliza-

tion bound for MLE (Lemma A.1.7) with simply setting D_i to be the delta distribution on the realized (s_h^k, a_h^k) pairs, and a union bound over K episodes, we have that w.p. at least $1-\delta$, for any $k \in [0, K-1]$:

(1) $P^\star \in \widehat{\mathcal{P}}^k$;

(2)

$$\sum_{i=0}^{k-1}\sum_{h=0}^{H-1} \mathbb{H}^2(P^\star(s_h^i, a_h^i)||\widehat{P}^k(s_h^i, a_h^i)) \leq 22\log(K\left|\mathcal{P}\right|/\delta). \tag{A.23}$$

This directly indicates that

$$P(\mathbb{I}\{\mathcal{E}_1\}) \geq 1-\delta\,. \tag{A.24}$$

Under event $\mathcal{E}_1$, with the realizability in above (1), and by the optimistic algorithm design $(\pi^k, \widehat{P}^k) \leftarrow \text{argmax}_{\pi\in\Pi, P\in\widehat{\mathcal{P}}^k} V_{0;P}^{\pi}(s_0)$, for any $k \in [0, K-1]$, we have the following optimism guarantee

$$V_{0;P^\star}^{\star} \leq \max_{\pi\in\Pi, P\in\widehat{\mathcal{P}}^k} V_{0;P}^{\pi} = V_{0;\widehat{P}^k}^{\pi^k}.$$

Then, under event $\mathcal{E}_1$, we use Lemma A.1.4 and Equation A.23 to get the following:

There exists a set $\mathcal{K} \subseteq [K-1]$ such that

- $|\mathcal{K}| \leq 13\log^2(88\log(K\left|\mathcal{P}\right|/\delta)KH)\cdot DE_1(\Psi, \mathcal{S}\times\mathcal{A}, 1/(176\log(K\left|\mathcal{P}\right|/\delta)KH))$
- And

$$\begin{aligned}
&\sum_{k\in[K-1]\setminus\mathcal{K}}\sum_{h=0}^{H-1} \mathbb{H}^2\Big(P^\star(s_h^k, a_h^k) \parallel \widehat{P}^k\big(s_h^k, a_h^k\big)\Big) \\
&\leq \mathrm{DE}_1(\Psi, \mathcal{S}\times\mathcal{A}, 1/KH)\cdot(2+154\log(K\left|\mathcal{P}\right|/\delta)\log(KH)) + 1 \\
&\lesssim \mathrm{DE}_1(\Psi, \mathcal{S}\times\mathcal{A}, 1/KH)\cdot\log(K\left|\mathcal{P}\right|/\delta)\log(KH)\,. \quad \text{(A.25)}
\end{aligned}$$

We upper bound the regret with optimism, and by dividing $k \in [K-1]$ into $\mathcal{K}$ and $[K-1] \setminus \mathcal{K}$ with the assumption that the trajectory-wise cumulative reward is normalized in [0,1], as follows

$$
\begin{aligned}
&\sum_{k=0}^{K-1} V^{\star}_{0;P^{\star}} - \sum_{k=0}^{K-1}\sum_{h=1}^{H} r(s_h^k, a_h^k) \\
&\leq |\mathcal{K}| + \sum_{k\in[K-1]\setminus\mathcal{K}} \left(V^{\pi^k}_{0;\widehat{P}^k} - \sum_{h=0}^{H-1} r(s_h^k, a_h^k) \right) \\
&\lesssim \log^2(\log(K\,|\mathcal{P}|\,/\delta)KH) \cdot DE_1(\Psi, \mathcal{S}\times\mathcal{A}, 1/(\log(K\,|\mathcal{P}|\,/\delta)KH)) \\
&+ \sum_{k\in[K-1]\setminus\mathcal{K}} \left(V^{\pi^k}_{0;\widehat{P}^k} - \sum_{h=0}^{H-1} r(s_h^k, a_h^k) \right) . \qquad \text{(A.26)}
\end{aligned}
$$

We then do the following decomposition. Note that for any $k \in [K-1]$, policy π^k is deterministic. We have that for any $k \in [K-1]$

$$
\begin{aligned}
&V^{\pi^k}_{0;\widehat{P}^k}(s_0^k) - \sum_{h=0}^{H-1} r(s_h^k, a_h^k) \\
&= Q^{\pi^k}_{0;\widehat{P}^k}(s_0^k, a_0^k) - \sum_{h=0}^{H-1} r(s_h^k, a_h^k) \\
&= r(s_0^k, a_0^k) + \mathbb{E}_{s'\sim\widehat{P}^k(s_0^k,a_0^k)} V^{\pi^k}_{1;\widehat{P}^k}(s') - \sum_{h=0}^{H-1} r(s_h^k, a_h^k) \\
&= \mathbb{E}_{s'\sim\widehat{P}^k(s_0^k,a_0^k)} V^{\pi^k}_{1;\widehat{P}^k}(s') - \sum_{h=1}^{H} r(s_h^k, a_h^k) \\
&= \mathbb{E}_{s'\sim P^*(s_0^k,a_0^k)} V^{\pi^k}_{1;\widehat{P}^k}(s') - \sum_{h=1}^{H} r(s_h^k, a_h^k) + \mathbb{E}_{s'\sim\widehat{P}^k(s_0^k,a_0^k)} V^{\pi^k}_{1;\widehat{P}^k}(s') - \mathbb{E}_{s'\sim P^*(s_0^k,a_0^k)} V^{\pi^k}_{1;\widehat{P}^k}(s')
\end{aligned}
$$

$$= V^{\pi^k}_{1;\widehat{P}^k}(s^k_1) - \sum_{h=1}^{H-1} r(s^k_h, a^k_h) + \underbrace{\mathbb{E}_{s'\sim P^*(s^k_0,a^k_0)} V^{\pi^k}_{1;\widehat{P}^k}(s') - V^{\pi^k}_{1;\widehat{P}^k}(s^k_1)}_{I^k_0}$$
$$+ \mathbb{E}_{s'\sim \widehat{P}^k(s^k_0,a^k_0)} V^{\pi^k}_{1;\widehat{P}^k}(s') - \mathbb{E}_{s'\sim P^*(s^k_0,a^k_0)} V^{\pi^k}_{1;\widehat{P}^k}(s')\,,$$

where we use the Bellman equation for several times.

Then, by doing this recursively, we can get for any $k \in [K-1]$

$$V^{\pi^k}_{0;\widehat{P}^k}(s^k_h) - \sum_{h=0}^{H-1} r(s^k_h, a^k_h)$$
$$\leq \sum_{h=0}^{H-1} I^k_h + \sum_{h=0}^{H-1} \left| \mathbb{E}_{s'\sim \widehat{P}^k(s^k_h,a^k_h)} V^{\pi^k}_{h+1;\widehat{P}^k}(s') - \mathbb{E}_{s'\sim P^*(s^k_h,a^k_h)} V^{\pi^k}_{h+1;\widehat{P}^k}(s') \right| \tag{A.27}$$

Therefore,

$$\sum_{k\in[K-1]\setminus\mathcal{K}} (V^{\pi^k}_{0;\widehat{P}^k}(s^k_h) - \sum_{h=0}^{H-1} r(s^k_h, a^k_h))$$
$$\leq \sum_{k\in[K-1]\setminus\mathcal{K}} \sum_{h=0}^{H-1} I^k_h + \sum_{k\in[K-1]\setminus\mathcal{K}} \sum_{h=0}^{H-1} \left| \mathbb{E}_{s'\sim \widehat{P}^k(s^k_h,a^k_h)} V^{\pi^k}_{h+1;\widehat{P}^k}(s') - \mathbb{E}_{s'\sim P^*(s^k_h,a^k_h)} V^{\pi^k}_{h+1;\widehat{P}^k}(s') \right| \tag{A.28}$$

Next we bound $\sum_{k\in[K-1]\setminus\mathcal{K}} \sum_{h=0}^{H-1} I^k_h$. Note that by Azuma Bernstein's inequality, with probability at least $1-\delta$

$$\sum_{k\in[K-1]\setminus\mathcal{K}} \sum_{h=0}^{H-1} I^k_h \leq \sqrt{2 \sum_{k\in[K-1]\setminus\mathcal{K}} \sum_{h=0}^{H-1} \left(\mathbb{V}_{P^\star} V^{\pi^k}_{h+1;\widehat{P}^k}\right)(s^k_h, a^k_h) \log(1/\delta)} + \frac{2}{3}\log(1/\delta) \tag{A.29}$$

This directly indicates that

$$P(\mathbb{I}\{\mathcal{E}_2\}) \geq 1-\delta\,. \tag{A.30}$$

Then, we propose the following lemma.

Lemma A.1.10 (Bound of sum of mean value differences for online RL)**.** *Under event $\mathcal{E}_1$, we have*

$$\begin{aligned}
&\sum_{k\in[K-1]\backslash\mathcal{K}}\sum_{h=0}^{H-1}\left|\mathbb{E}_{s'\sim\widehat{P}^k(s_h^k,a_h^k)}V^{\pi^k}_{h+1;\widehat{P}^k}(s')-\mathbb{E}_{s'\sim P^*(s_h^k,a_h^k)}V^{\pi^k}_{h+1;\widehat{P}^k}(s')\right|\\
&\lesssim\sqrt{\sum_{k\in[K-1]\backslash\mathcal{K}}\sum_{h=0}^{H-1}\left[\left(\mathbb{V}_{P^\star}V^{\pi^k}_{h+1;\widehat{P}^k}\right)(s_h^k,a_h^k)\right]\cdot DE_1(\Psi,\mathcal{S}\times\mathcal{A},1/KH)\cdot\log(K\left|\mathcal{P}\right|/\delta)\log(KH)}\\
&\quad+DE_1(\Psi,\mathcal{S}\times\mathcal{A},1/KH)\cdot\log(K\left|\mathcal{P}\right|/\delta)\log(KH).
\end{aligned}$$

Proof [Proof of Lemma A.1.10] Under event $\mathcal{E}_1$, we have

$$\begin{aligned}
&\sum_{k\in[K-1]\backslash\mathcal{K}}\sum_{h=0}^{H-1}\left|\mathbb{E}_{s'\sim\widehat{P}^k(s_h^k,a_h^k)}V^{\pi^k}_{h+1;\widehat{P}^k}(s')-\mathbb{E}_{s'\sim P^*(s_h^k,a_h^k)}V^{\pi^k}_{h+1;\widehat{P}^k}(s')\right|\\
&\leq 4\sum_{k\in[K-1]\backslash\mathcal{K}}\sum_{h=0}^{H-1}\left[\sqrt{\left(\mathbb{V}_{P^\star}V^{\pi^k}_{h+1;\widehat{P}^k}\right)(s_h^k,a_h^k)D_\triangle\Big(V^{\pi^k}_{h+1;\widehat{P}^k}\big(s'\sim P^\star(s_h^k,a_h^k)\big)\parallel V^{\pi^k}_{h+1;\widehat{P}^k}(s'\sim\widehat{P}^k\big(s_h^k,a_h^k\big))\Big)}\right]\\
&\quad+5\sum_{k\in[K-1]\backslash\mathcal{K}}\sum_{h=0}^{H-1}\left[D_\triangle\Big(V^{\pi^k}_{h+1;\widehat{P}^k}\big(s'\sim P^\star(s_h^k,a_h^k)\big)\parallel V^{\pi^k}_{h+1;\widehat{P}^k}(s'\sim\widehat{P}^k\big(s_h^k,a_h^k\big))\Big)\right]\\
&\leq 8\sum_{k\in[K-1]\backslash\mathcal{K}}\sum_{h=0}^{H-1}\left[\sqrt{\left(\mathbb{V}_{P^\star}V^{\pi^k}_{h+1;\widehat{P}^k}\right)(s_h^k,a_h^k)\mathbb{H}^2\Big(V^{\pi^k}_{h+1;\widehat{P}^k}\big(s'\sim P^\star(s_h^k,a_h^k)\big)\parallel V^{\pi^k}_{h+1;\widehat{P}^k}(s'\sim\widehat{P}^k\big(s_h^k,a_h^k\big))\Big)}\right]\\
&\quad+20\sum_{k\in[K-1]\backslash\mathcal{K}}\sum_{h=0}^{H-1}\left[\mathbb{H}^2\Big(V^{\pi^k}_{h+1;\widehat{P}^k}\big(s'\sim P^\star(s_h^k,a_h^k)\big)\parallel V^{\pi^k}_{h+1;\widehat{P}^k}(s'\sim\widehat{P}^k\big(s_h^k,a_h^k\big))\Big)\right]\\
&\leq 8\sum_{k\in[K-1]\backslash\mathcal{K}}\sum_{h=0}^{H-1}\left[\sqrt{\left(\mathbb{V}_{P^\star}V^{\pi^k}_{h+1;\widehat{P}^k}\right)(s_h^k,a_h^k)\mathbb{H}^2\Big(P^\star(s_h^k,a_h^k)\parallel\widehat{P}^k\big(s_h^k,a_h^k\big)\Big)}\right]\\
&\quad+20\sum_{k\in[K-1]\backslash\mathcal{K}}\sum_{h=0}^{H-1}\left[\mathbb{H}^2\Big(P^\star(s_h^k,a_h^k)\parallel\widehat{P}^k\big(s_h^k,a_h^k\big)\Big)\right]\\
&\leq 8\sqrt{\sum_{k\in[K-1]\backslash\mathcal{K}}\sum_{h=0}^{H-1}\left[\left(\mathbb{V}_{P^\star}V^{\pi^k}_{h+1;\widehat{P}^k}\right)(s_h^k,a_h^k)\right]\cdot\sum_{k\in[K-1]\backslash\mathcal{K}}\sum_{h=0}^{H-1}\left[\mathbb{H}^2\Big(P^\star(s_h^k,a_h^k)\parallel\widehat{P}^k\big(s_h^k,a_h^k\big)\Big)\right]}\\
&\quad+20\sum_{k\in[K-1]\backslash\mathcal{K}}\sum_{h=0}^{H-1}\left[\mathbb{H}^2\Big(P^\star(s_h^k,a_h^k)\parallel\widehat{P}^k\big(s_h^k,a_h^k\big)\Big)\right]\\
&\lesssim\sqrt{\sum_{k\in[K-1]\backslash\mathcal{K}}\sum_{h=0}^{H-1}\left[\left(\mathbb{V}_{P^\star}V^{\pi^k}_{h+1;\widehat{P}^k}\right)(s_h^k,a_h^k)\right]\cdot\mathrm{DE}_1(\Psi,\mathcal{S}\times\mathcal{A},1/KH)\cdot\log(K\left|\mathcal{P}\right|/\delta)\log(KH)}
\end{aligned}$$

$$+ \mathrm{DE}_1(\Psi, \mathcal{S}\times\mathcal{A}, 1/KH)\cdot\log(K\left|\mathcal{P}\right|/\delta)\log(KH)\,, \tag{A.31}$$

where in the first inequality, we use Lemma 3.2.1 to bound the difference of two means $\mathbb{E}_{s'\sim P^\star(s_h^k,a_h^k)}V^{\pi^k}_{h+1;\widehat{P}^k}(s') - \mathbb{E}_{s'\sim\widehat{P}^k(s_h^k,a_h^k)}V^{\pi^*}_{h+1;\widehat{P}^k}(s')$ using variances and the triangle discrimination; in the second inequality we use the fact that that triangle discrimination is equivalent to squared Hellinger distance, i.e., $D_\triangle \le 4\mathbb{H}^2$; the third inequality is via data processing inequality on the squared Hellinger distance; the fourth inequality is by the Cauchy–Schwarz inequality; the last inequality holds under $\mathcal{E}_1$ by Equation A.25.

The next lemma shows that the event $\mathcal{E}_3$ happens with high probability.

Lemma A.1.11 (Recursion Event Lemma)**.** *Event $\mathcal{E}_3$ happens with high probability. Specifically, we have*

$$P(\mathbb{I}\{\mathcal{E}_3\}) \ge 1 - (1 + \lceil\log_2(\frac{KH}{G})\rceil)\delta. \tag{A.32}$$

Proof[Proof of Lemma A.1.11] Let $\Delta^{\pi^k}_{h+1} := V^{\pi^k}_{h+1;\widehat{P}^k} - V^{\pi^k}_{h+1}$. First, under event $\mathcal{E}_1$, with happens with probability at least $1-\delta$ by Equation A.24, and also note that π^k is deterministic for any $k\in[K-1]$, we can prove the following

$$\begin{aligned}
&\sum_{k\in[K-1]\setminus\mathcal{K}}\sum_{h=0}^{H-1}\Big[\Big|(\Delta^{\pi^k}_h)(s_h^k) - \big(P^\star\Delta^{\pi^k}_{h+1}\big)(s_h^k,a_h^k)\Big|\Big] \qquad\qquad (A.33)\\
&= \sum_{k\in[K-1]\setminus\mathcal{K}}\sum_{h=0}^{H-1}\Big[\Big|(V^{\pi^k}_{h;\widehat{P}^k})(s_h^k) - \big(P^\star V^{\pi^k}_{h+1;\widehat{P}^k}\big)(s_h^k,a_h^k) - \Big((V^{\pi^k}_h)(s_h^k) - \big(P^\star V^{\pi^k}_{h+1}\big)(s_h^k,a_h^k)\Big)\Big|\Big]\\
&= \sum_{k\in[K-1]\setminus\mathcal{K}}\sum_{h=0}^{H-1}\Big[\Big|r(s_h^k,a_h^k) + \big(\widehat{P}^k V^{\pi^k}_{h+1;\widehat{P}^k}\big)(s_h^k,a_h^k) - \big(P^\star V^{\pi^k}_{h+1;\widehat{P}^k}\big)(s_h^k,a_h^k) - r(s_h^k,a_h^k)\Big|\Big]\\
&= \sum_{k\in[K-1]\setminus\mathcal{K}}\sum_{h=0}^{H-1}\Big[\Big|\big(\widehat{P}^k V^{\pi^k}_{h+1;\widehat{P}^k}\big)(s_h^k,a_h^k) - \big(P^\star V^{\pi^k}_{h+1;\widehat{P}^k}\big)(s_h^k,a_h^k)\Big|\Big]
\end{aligned}$$

$$
\begin{aligned}
&= \sum_{k\in[K-1]\setminus\mathcal{K}} \sum_{h=0}^{H-1} \left[\left| \mathbb{E}_{s'\sim P^\star(s_h^k,a_h^k)} \left[V_{h+1;\widehat{P}^k}^{\pi^k}(s') \right] - \mathbb{E}_{s'\sim \widehat{P}^k(s_h^k,a_h^k)} \left[V_{h+1;\widehat{P}^k}^{\pi^k}(s') \right] \right| \right] \\
&\lesssim \sqrt{\sum_{k\in[K-1]\setminus\mathcal{K}} \sum_{h=0}^{H-1} \left[\left(\mathbb{V}_{P^\star} V_{h+1;\widehat{P}^k}^{\pi^k}\right)(s_h^k,a_h^k) \right] \cdot \mathrm{DE}_1(\Psi, \mathcal{S}\times\mathcal{A}, 1/KH) \cdot \log(K|\mathcal{P}|/\delta)\log(KH)} \\
&\quad + \mathrm{DE}_1(\Psi, \mathcal{S}\times\mathcal{A}, 1/KH) \cdot \log(K|\mathcal{P}|/\delta)\log(KH) \\
&= G
\end{aligned} \tag{A.34}
$$

where the first equality is by the definition of $\Delta_{h+1}^{\pi^k}$, the inequality holds under $\mathcal{E}_1$ by Lemma A.1.10, and the last equality is by definition of A and G.

Under event $\mathcal{E}_1$, with probability at least $1 - \lceil \log_2(\frac{KH}{G}) \rceil \delta$, for any $m \in [0, \lceil \log_2(\frac{KH}{G}) \rceil]$

$$
\begin{aligned}
C_m &= \sum_{k\in[K-1]\setminus\mathcal{K}} \sum_{h=0}^{H-1} \left[\left(\mathbb{V}_{P^\star} (V_{h+1;\widehat{P}^k}^{\pi^k} - V_{h+1}^{\pi^k})^{2^m}\right)(s_h^k,a_h^k) \right] \\
&= \sum_{k\in[K-1]\setminus\mathcal{K}} \sum_{h=0}^{H-1} \left[\left(P^\star (\Delta_{h+1}^{\pi^k})^{2^{m+1}}\right)(s_h^k,a_h^k) - \left((P^\star (\Delta_{h+1}^{\pi^k})^{2^m})(s_h^k,a_h^k)\right)^2 \right] \\
&= \sum_{k\in[K-1]\setminus\mathcal{K}} \sum_{h=0}^{H-1} \left[(\Delta_{h+1}^{\pi^k})^{2^{m+1}}(s_{h+1}^k) \right] - \sum_{k\in[K-1]\setminus\mathcal{K}} \sum_{h=0}^{H-1} \left[\left((P^\star (\Delta_{h+1}^{\pi^k})^{2^m})(s_h^k,a_h^k)\right)^2 \right] \\
&\quad + \sum_{k\in[K-1]\setminus\mathcal{K}} \sum_{h=0}^{H-1} \left(\mathbb{E}_{s\sim P^*(s_h^k,a_h^k)} \left[(\Delta_{h+1}^{\pi^k})^{2^{m+1}}(s) \right] - (\Delta_{h+1}^{\pi^k})^{2^{m+1}}(s_{h+1}^k) \right) \\
&\le \sum_{k\in[K-1]\setminus\mathcal{K}} \sum_{h=0}^{H-1} \left[(\Delta_h^{\pi^k})^{2^{m+1}}(s_h^k) - \left((P^\star (\Delta_{h+1}^{\pi^k})^{2^m})(s_h^k,a_h^k)\right)^2 \right] \\
&\quad + \sum_{k\in[K-1]\setminus\mathcal{K}} \sum_{h=0}^{H-1} \left(\mathbb{E}_{s\sim P^*(s_h^k,a_h^k)} \left[(\Delta_{h+1}^{\pi^k})^{2^{m+1}}(s) \right] - (\Delta_{h+1}^{\pi^k})^{2^{m+1}}(s_h^k) \right) \\
&\lesssim \sum_{k\in[K-1]\setminus\mathcal{K}} \sum_{h=0}^{H-1} \left[(\Delta_h^{\pi^k})^{2^{m+1}}(s_h^k) - \left((P^\star (\Delta_{h+1}^{\pi^k})^{2^m})(s_h^k,a_h^k)\right)^2 \right] + \log(1/\delta) \\
&\quad + \sqrt{\sum_{k\in[K-1]\setminus\mathcal{K}} \sum_{h=0}^{H-1} \mathbb{V}_{P^*} \left((V_{h+1;\widehat{P}^k}^{\pi^k} - V_{h+1}^{\pi^k})^{2^{m+1}} \right)(s_h^k,a_h^k) \log(1/\delta)} \\
&= \sum_{k\in[K-1]\setminus\mathcal{K}} \sum_{h=0}^{H-1} \left[\left((\Delta_h^{\pi^k})^{2^m}(s_h^k) + (P^\star (\Delta_{h+1}^{\pi^k})^{2^m})(s_h^k,a_h^k) \right) \cdot \left((\Delta_h^{\pi^k})^{2^m}(s_h^k) - (P^\star (\Delta_{h+1}^{\pi^k})^{2^m})(s_h^k,a_h^k) \right) \right]
\end{aligned}
$$

$$
\begin{aligned}
&+\sqrt{\log(1/\delta)\cdot C_{m+1}}+\log(1/\delta)\\
&=\sum_{k\in[K-1]\setminus\mathcal{K}}\sum_{h=0}^{H-1}\left[\left((\Delta_h^{\pi^k})^{2^m}(s_h^k)+(P^\star(\Delta_{h+1}^{\pi^k})^{2^m})(s_h^k,a_h^k)\right)\cdot\left((\Delta_h^{\pi^k})^{2^m}(s_h^k)-(P^\star((\Delta_{h+1}^{\pi^k})^2)^{2^{m-1}})(s_h^k,a_h^k)\right)\right]\\
&+\sqrt{\log(1/\delta)\cdot C_{m+1}}+\log(1/\delta)\\
&\le\sum_{k\in[K-1]\setminus\mathcal{K}}\sum_{h=0}^{H-1}\left[\left((\Delta_h^{\pi^k})^{2^m}(s_h^k)+(P^\star(\Delta_{h+1}^{\pi^k})^{2^m})(s_h^k,a_h^k)\right)\cdot\left((\Delta_h^{\pi^k})^{2^m}(s_h^k)-((P^\star(\Delta_{h+1}^{\pi^k})^2)(s_h^k,a_h^k))^{2^{m-1}}\right)\right]\\
&+\sqrt{\log(1/\delta)\cdot C_{m+1}}+\log(1/\delta)\\
&\le 2^m\sum_{k\in[K-1]\setminus\mathcal{K}}\sum_{h=0}^{H-1}\left[\left|(\Delta_h^{\pi^k})^2(s_h^k)-((P^\star\Delta_{h+1}^{\pi^k})(s_h^k,a_h^k))^2\right|\right]+\sqrt{\log(1/\delta)\cdot C_{m+1}}+\log(1/\delta)\\
&=2^m\sum_{k\in[K-1]\setminus\mathcal{K}}\sum_{h=0}^{H-1}\left[\left|\left((\Delta_h^{\pi^k})(s_h^k)+(P^\star\Delta_{h+1}^{\pi^k})(s_h^k,a_h^k)\right)\cdot\left((\Delta_h^{\pi^k})(s_h^k)-(P^\star\Delta_{h+1}^{\pi^k})(s_h^k,a_h^k)\right)\right|\right]\\
&+\sqrt{\log(1/\delta)\cdot C_{m+1}}+\log(1/\delta)\\
&\le 2\cdot 2^m\sum_{k\in[K-1]\setminus\mathcal{K}}\sum_{h=0}^{H-1}\left[\left|(\Delta_h^{\pi^k})(s_h^k)-\left(P^\star\Delta_{h+1}^{\pi^k}\right)(s_h^k,a_h^k)\right|\right]+\sqrt{\log(1/\delta)\cdot C_{m+1}}+\log(1/\delta)\\
&\lesssim 2^m G+\sqrt{\log(1/\delta)\cdot C_{m+1}}+\log(1/\delta)\,, \qquad \text{(A.35)}
\end{aligned}
$$

where in the first inequality we change the index, the second inequality holds with probability at least $1-\delta$ by Azuma Bernstain's inequality, the third inequality holds because that $E[X^{2^{m-1}}] \ge (E[X])^{2^{m-1}}$ for $m \ge 1$ and $X \ge 0$, the fourth inequality holds by keep using $a^2-b^2=(a+b)(a-b)$, then with $E[X^2] \ge E[X]^2$, and the assumption that the trajectory-wise total reward is normalized in $[0,1]$, the last inequality holds under $\mathcal{E}_1$ by Equation A.34, and we take a union bound to get this hold for all $m \in [0, \lceil \log_2(\frac{KH}{G})\rceil]$ with probability at least $1-\lceil \log_2(\frac{KH}{G})\rceil\delta$ (because for each $m \in [0, \lceil \log_2(\frac{KH}{G})\rceil]$ we need to apply the Azuma Bernstain's inequality once).

The above reasoning directly implies that

$$
P(\mathbb{I}\{\mathcal{E}_3\}) \ge 1-(1+\lceil \log_2(\frac{KH}{G})\rceil)\delta. \qquad \text{(A.36)}
$$

Under the event $\mathcal{E}_3$, we prove the following lemma to bound $\sum_{k\in[K-1]\backslash\mathcal{K}}\sum_{h=0}^{H-1}\left[\left(\mathbb{V}_{P^\star}V^{\pi^k}_{h+1;\widehat{P}^k}\right)(s_h^k,a_h^k)\right]$.

Lemma A.1.12 (Variance Conversion Lemma for online RL)**.** *Under event $\mathcal{E}_3$, we have*

$$\sum_{k\in[K-1]\backslash\mathcal{K}}\sum_{h=0}^{H-1}\left[\left(\mathbb{V}_{P^\star}V^{\pi^k}_{h+1;\widehat{P}^k}\right)(s_h^k,a_h^k)\right]$$
$$\leq O\Big(\sum_{k\in[K-1]\backslash\mathcal{K}}\sum_{h=0}^{H-1}\left[\left(\mathbb{V}_{P^\star}V^{\pi^k}_{h+1}\right)(s_h^k,a_h^k)\right]+DE_1(\Psi,\mathcal{S}\times\mathcal{A},1/KH)\cdot\log(K\left|\mathcal{P}\right|/\delta)\log(KH)\Big).$$

Proof[Proof of Lemma A.1.12] Under $\mathcal{E}_3$, we have for any $m\in[0,\lceil\log_2(\frac{KH}{G})\rceil]$

$$C_m\lesssim 2^mG+\sqrt{\log(1/\delta)\cdot C_{m+1}}+\log(1/\delta)\,. \tag{A.37}$$

Then, by Lemma A.1.9, we have

$$C_0\lesssim G\,. \tag{A.38}$$

Also note that we have $A\leq 2B+2C_0$ since $\mathbb{V}_{P^\star}(a+b)\leq 2\mathbb{V}_{P^\star}(a)+2\mathbb{V}_{P^\star}(b)$. Therefore, we have

$$\begin{aligned}A&\leq 2B+2C_0\\&\lesssim B+G\\&=B+\sqrt{A\cdot\mathrm{DE}_1(\Psi,\mathcal{S}\times\mathcal{A},1/KH)\cdot\log(K\left|\mathcal{P}\right|/\delta)\log(KH)}\\&\quad+\mathrm{DE}_1(\Psi,\mathcal{S}\times\mathcal{A},1/KH)\cdot\log(K\left|\mathcal{P}\right|/\delta)\log(KH)\end{aligned} \tag{A.39}$$

Then, with the fact that $x\leq 2a+b^2$ if $x\leq a+b\sqrt{x}$, we have

$$A\leq O\left(B+\mathrm{DE}_1(\Psi,\mathcal{S}\times\mathcal{A},1/KH)\cdot\log(K\left|\mathcal{P}\right|/\delta)\log(KH)\right), \tag{A.40}$$

which is

$$\sum_{k\in[K-1]\setminus\mathcal{K}}\sum_{h=0}^{H-1}\Big[\big(\mathbb{V}_{P^\star}V^{\pi^k}_{h+1;\widehat{P}^k}\big)(s^k_h,a^k_h)\Big]$$
$$\leq O\Big(\sum_{k\in[K-1]\setminus\mathcal{K}}\sum_{h=0}^{H-1}\Big[\big(\mathbb{V}_{P^\star}V^{\pi^k}_{h+1}\big)(s^k_h,a^k_h)\Big]+\mathrm{DE}_1(\Psi,\mathcal{S}\times\mathcal{A},1/KH)\cdot\log(K\,|\mathcal{P}|\,/\delta)\log(KH)\Big) \tag{A.41}$$

By the same reasoning in Lemma 26 of [279], we have that with probability at least $1-\delta$

$$\sum_{k\in[K-1]\setminus\mathcal{K}}\sum_{h=0}^{H-1}\Big[\big(\mathbb{V}_{P^\star}V^{\pi^k}_{h+1}\big)(s^k_h,a^k_h)\Big]\leq\sum_{k=0}^{K-1}\sum_{h=0}^{H-1}\Big[\big(\mathbb{V}_{P^\star}V^{\pi^k}_{h+1}\big)(s^k_h,a^k_h)\Big]$$
$$\leq O(\sum_{k=0}^{K-1}\mathrm{VaR}_{\pi^k}+\log(1/\delta))\,. \tag{A.42}$$

This indicates that

$$P(\mathbb{I}\{\mathcal{E}_4\})\geq 1-\delta\,. \tag{A.43}$$

We can use the Azuma Bernstain's inequality to get that with probability at least $1-\delta$:

$$\sum_{k=0}^{K-1}\sum_{h=1}^{H}r(s^k_h,a^k_h)-\sum_{k=0}^{K-1}V^{\pi^k}_{0;P^*}\lesssim\sqrt{\sum_{k=0}^{K-1}\mathrm{VaR}_{\pi^k}\log(1/\delta)}+\log(1/\delta)\,. \tag{A.44}$$

This indicates that

$$P(\mathbb{I}\{\mathcal{E}_5\})\geq 1-\delta\,. \tag{A.45}$$

Then, together with Lemma A.1.11, Equation A.24 and Equation A.30, we have

$$P(\mathbb{I}\{\mathcal{E}\})\geq 1-(5+\lceil\log_2(\frac{KH}{G})\rceil)\delta\geq 1-5KH\delta\,. \tag{A.46}$$

Finally, under event $\mathcal{E}$, with all the things above (Equation A.26, Equation A.28, Equation A.29,Lemma A.1.10, Lemma A.1.12), we have

$$
\begin{aligned}
&\sum_{k=0}^{K-1} V_{0;P^\star}^{\star} - \sum_{k=0}^{K-1} V_{0;P^*}^{\pi^k} \\
&= \sum_{k=0}^{K-1} V_{0;P^\star}^{\star} - \sum_{k=0}^{K-1}\sum_{h=1}^{H} r(s_h^k, a_h^k) + \sum_{k=0}^{K-1}\sum_{h=1}^{H} r(s_h^k, a_h^k) - \sum_{k=0}^{K-1} V_{0;P^*}^{\pi^k} \\
&\lesssim |\mathcal{K}| + \sum_{k\in[K-1]\setminus\mathcal{K}} \left(V_{0;\widehat{P}^k}^{\pi^k} - \sum_{h=0}^{H-1} r(s_h^k, a_h^k) \right) + \sqrt{\sum_{k=0}^{K-1} \mathrm{VaR}_{\pi^k} \log(1/\delta)} + \log(1/\delta) \\
&\lesssim \log^2(\log(K|\mathcal{P}|/\delta)KH) \cdot DE_1(\Psi, \mathcal{S}\times\mathcal{A}, 1/(\log(K|\mathcal{P}|/\delta)KH)) + \sum_{k\in[K-1]\setminus\mathcal{K}} \left(V_{0;\widehat{P}^k}^{\pi^k} - \sum_{h=0}^{H-1} r(s_h^k, a_h^k) \right) \\
&\quad + \sqrt{\sum_{k=0}^{K-1} \mathrm{VaR}_{\pi^k} \log(1/\delta)} + \log(1/\delta) \\
&\lesssim \log^2(\log(K|\mathcal{P}|/\delta)KH) \cdot DE_1(\Psi, \mathcal{S}\times\mathcal{A}, 1/(\log(K|\mathcal{P}|/\delta)KH)) + \log(1/\delta) \\
&\quad + \sqrt{\sum_{k\in[K-1]\setminus\mathcal{K}} \sum_{h=0}^{H-1} \left(\mathbb{V}_{P^\star} V_{h+1;\widehat{P}^k}^{\pi^k}\right)(s_h^k, a_h^k) \log(1/\delta)} + \sqrt{\sum_{k=0}^{K-1} \mathrm{VaR}_{\pi^k} \log(1/\delta)} \\
&\quad + \sum_{k\in[K-1]\setminus\mathcal{K}} \sum_{h=0}^{H-1} \left| \mathbb{E}_{s'\sim\widehat{P}^k(s_1^k, A^k)} V_{1;\widehat{P}^k}^{\pi^k}(s') - \mathbb{E}_{s'\sim P^*(s_1^k, A^k)} V_{1;\widehat{P}^k}^{\pi^k}(s') \right| \\
&\lesssim \log^2(\log(K|\mathcal{P}|/\delta)KH) \cdot DE_1(\Psi, \mathcal{S}\times\mathcal{A}, 1/(\log(K|\mathcal{P}|/\delta)KH)) + \sqrt{\sum_{k=0}^{K-1} \mathrm{VaR}_{\pi^k} \log(1/\delta)} + \log(1/\delta) \\
&\quad + \sqrt{\left(\sum_{k\in[K-1]\setminus\mathcal{K}} \sum_{h=0}^{H-1} \left[\left(\mathbb{V}_{P^\star} V_{h+1}^{\pi^k}\right)(s_h^k, a_h^k) \right] + \mathrm{DE}_1(\Psi, \mathcal{S}\times\mathcal{A}, 1/KH) \cdot \log(K|\mathcal{P}|/\delta) \log(KH) \right) \cdot \log(1/\delta)} \\
&\quad + \sqrt{\sum_{k\in[K-1]\setminus\mathcal{K}} \sum_{h=0}^{H-1} \left[\left(\mathbb{V}_{P^\star} V_{h+1;\widehat{P}^k}^{\pi^k}\right)(s_h^k, a_h^k) \right] \cdot \mathrm{DE}_1(\Psi, \mathcal{S}\times\mathcal{A}, 1/KH) \cdot \log(K|\mathcal{P}|/\delta) \log(KH)} \\
&\lesssim \log^2(\log(K|\mathcal{P}|/\delta)KH) \cdot DE_1(\Psi, \mathcal{S}\times\mathcal{A}, 1/(\log(K|\mathcal{P}|/\delta)KH)) + \sqrt{\sum_{k=0}^{K-1} \mathrm{VaR}_{\pi^k} \log(1/\delta)} + \log(1/\delta) \\
&\quad + \sqrt{\left(\sum_{k\in[K-1]\setminus\mathcal{K}} \sum_{h=0}^{H-1} \left[\left(\mathbb{V}_{P^\star} V_{h+1}^{\pi^k}\right)(s_h^k, a_h^k) \right] + \mathrm{DE}_1(\Psi, \mathcal{S}\times\mathcal{A}, 1/KH) \cdot \log(K|\mathcal{P}|/\delta) \log(KH) \right) \cdot \log(1/\delta)}
\end{aligned}
$$

$$
\begin{aligned}
&+\sqrt{\Big(\sum_{k\in[K-1]\setminus\mathcal{K}}\sum_{h=0}^{H-1}\Big[\big(\mathbb{V}_{P^\star}V_{h+1}^{\pi^k}\big)(s_h^k,a_h^k)\Big]+\mathrm{DE}_1(\Psi,\mathcal{S}\times\mathcal{A},1/KH)\cdot\log(K\left|\mathcal{P}\right|/\delta)\log(KH)\Big)}\\
&\quad\times\sqrt{\mathrm{DE}_1(\Psi,\mathcal{S}\times\mathcal{A},1/KH)\cdot\log(K\left|\mathcal{P}\right|/\delta)\log(KH))}\\
&\lesssim\log^2(\log(K\left|\mathcal{P}\right|/\delta)KH)\cdot DE_1(\Psi,\mathcal{S}\times\mathcal{A},1/(\log(K\left|\mathcal{P}\right|/\delta)KH))+\sqrt{\sum_{k=0}^{K-1}\mathrm{VaR}_{\pi^k}\log(1/\delta)}+\log(1/\delta)\\
&+\sqrt{\Big(\sum_{k=0}^{K-1}\mathrm{VaR}_{\pi^k}+\log(1/\delta)+\mathrm{DE}_1(\Psi,\mathcal{S}\times\mathcal{A},1/KH)\cdot\log(K\left|\mathcal{P}\right|/\delta)\log(KH)\Big)\cdot\log(1/\delta)}\\
&+\sqrt{\left(\sum_{k=0}^{K-1}\mathrm{VaR}_{\pi^k}+\log\left(\frac{1}{\delta}\right)+\mathrm{DE}_1\left(\Psi,\mathcal{S}\times\mathcal{A},\frac{1}{KH}\right)\cdot\log\left(\frac{K\left|\mathcal{P}\right|}{\delta}\right)\log(KH)\right)}\\
&\quad\times\sqrt{\mathrm{DE}_1\left(\Psi,\mathcal{S}\times\mathcal{A},\frac{1}{KH}\right)\log\left(\frac{K\left|\mathcal{P}\right|}{\delta}\right)\log(KH)}\\
&\le O\Big(\sqrt{\sum_{k=0}^{K-1}\mathrm{VaR}_{\pi^k}\cdot\mathrm{DE}_1(\Psi,\mathcal{S}\times\mathcal{A},1/KH)\cdot\log(K\left|\mathcal{P}\right|/\delta)\log(KH)}\\
&\quad+\mathrm{DE}_1(\Psi,\mathcal{S}\times\mathcal{A},1/KH)\cdot\log(K\left|\mathcal{P}\right|/\delta)\log(KH)\Big).
\end{aligned}
\tag{A.47}
$$

The final result follows by replacing δ to be $\delta/(5KH)$ to make the event $\mathcal{E}$ happen with probability at least $1-\delta$.

A.1.6 Proof of Corollary 3.2

Proof[Proof of Corollary 3.2] By Lemma A.1.6, we have

$$
\mathrm{VaR}_{\pi^k}=\sum_{h=0}^{H-1}\mathbb{E}_{s,a\sim d_h^{\pi^k}}\left[\big(\mathbb{V}_{P^\star}V_{h+1}^{\pi^k}\big)(s,a)\right]
\tag{A.48}
$$

Therefore, when $P^\star$ is deterministic, the $\mathbb{E}_{s,a\sim d_h^{\pi^k}}\left[\big(\mathbb{V}_{P^\star}V_{h+1}^{\pi^k}\big)(s,a)\right]$ terms are all 0 for any $k\in[K-1]$ and $h\in[H-1]$, and then the $\sum_{k=0}^{K-1}\mathrm{VaR}_{\pi^k}$ term in the higher order term in Theorem 3.3.2 is 0.

A.1.7 Proof of Corollary 3.3

Proof[Proof of Corollary 3.3] We follow the MLE guarantee for the infinite model class in Lemma A.1.8 and the same proof steps in the proof of Theorem 3.3.2 in Appendix A.1.5.

A.1.8 Detailed Proofs for the Offline RL setting in Section 3.4

A.1.9 Proof of Theorem 3.4.1

The following is the full proof of Theorem 3.4.1.

Proof[Proof of Theorem 3.4.1] First, by the realizability assumption, the standard generalization bound for MLE (Lemma A.1.7) with simply setting D_i to be the delta distribution on the (s_h^k, a_h^k) pairs in the offline dataset $\mathcal{D}$, we have that w.p. at least $1-\delta$:

(1) $P^\star \in \widehat{\mathcal{P}}$;

(2)

$$\frac{1}{K}\sum_{k=1}^{K}\sum_{h=0}^{H-1}\mathbb{H}^2(P^\star(s_h^k,a_h^k)||\widehat{P}(s_h^k,a_h^k)) \leq \frac{22\log(|\mathcal{P}|/\delta)}{K}. \quad \text{(A.49)}$$

Then, with the above realizability in (1), and by the pessimistic algorithm design $\hat{\pi} \leftarrow \operatorname{argmax}_{\pi\in\Pi}\min_{P\in\widehat{\mathcal{P}}} V_{0;P}^{\pi}(s_0)$, $\widehat{P} \leftarrow \operatorname{argmin}_{P\in\widehat{\mathcal{P}}} V_{0;P}^{\hat{\pi}}(s_0)$, we have that for any $\pi^\star \in \Pi$

$$\begin{aligned} V_{0;P^\star}^{\pi^\star} - V_{0;P^\star}^{\hat{\pi}} &= V_{0;P^\star}^{\pi^\star} - V_{0;\widehat{P}}^{\pi^\star} + V_{0;\widehat{P}}^{\pi^\star} - V_{0;P^\star}^{\hat{\pi}} \\ &\leq V_{0;P^\star}^{\pi^\star} - V_{0;\widehat{P}}^{\pi^\star} + V_{0;\widehat{P}}^{\hat{\pi}} - V_{0;P^\star}^{\hat{\pi}} \\ &\leq V_{0;P^\star}^{\pi^\star} - V_{0;\widehat{P}}^{\pi^\star}. \end{aligned} \quad \text{(A.50)}$$

We can then bound $V^{\pi^\star}_{0;P^\star} - V^{\pi^\star}_{0;\widehat{P}}$ using the simulation lemma (Lemma A.1.5):

$$V^{\pi^\star}_{0;P^\star} - V^{\pi^\star}_{0;\widehat{P}} \leq \sum_{h=0}^{H-1} \mathbb{E}_{s,a\sim d_h^{\pi^\star}} \left[\left| \mathbb{E}_{s'\sim P^\star(s,a)} V^{\pi^\star}_{h+1;\widehat{P}}(s') - \mathbb{E}_{s'\sim \widehat{P}(s,a)} V^{\pi^\star}_{h+1;\widehat{P}}(s') \right| \right] . \tag{A.51}$$

Then, we prove the following lemma to bound the RHS of Equation A.51.

Lemma A.1.13 (Bound of sum of mean value differences for offline RL)**.** *With probability at least* $1-\delta$*, we have*

$$\begin{aligned}
&\sum_{h=0}^{H-1} \mathbb{E}_{s,a\sim d_h^{\pi^\star}} \left[\left| \mathbb{E}_{s'\sim P^\star(s,a)} V^{\pi^\star}_{h+1;\widehat{P}}(s') - \mathbb{E}_{s'\sim \widehat{P}(s,a)} V^{\pi^\star}_{h+1;\widehat{P}}(s') \right| \right] \\
&\leq 8\sqrt{\sum_{h=0}^{H-1} \mathbb{E}_{s,a\sim d_h^{\pi^\star}} \left[\left(\mathbb{V}_{P^\star} V^{\pi^\star}_{h+1;\widehat{P}}\right)(s,a) \right] \cdot \frac{22C^{\pi^\star}\log(|\mathcal{P}|/\delta)}{K}} + \frac{440C^{\pi^\star}\log(|\mathcal{P}|/\delta)}{K}.
\end{aligned}$$

Proof[Proof of Lemma A.1.13] We have

$$\begin{aligned}
&\sum_{h=0}^{H-1} \mathbb{E}_{s,a\sim d_h^{\pi^\star}} \left[\left| \mathbb{E}_{s'\sim P^\star(s,a)} V^{\pi^\star}_{h+1;\widehat{P}}(s') - \mathbb{E}_{s'\sim \widehat{P}(s,a)} V^{\pi^\star}_{h+1;\widehat{P}}(s') \right| \right] \\
&\leq 4 \sum_{h=0}^{H-1} \mathbb{E}_{s,a\sim d_h^{\pi^\star}} \left[\sqrt{\left(\mathbb{V}_{P^\star} V^{\pi^\star}_{h+1;\widehat{P}}\right)(s,a) D_\triangle\Big(V^{\pi^\star}_{h+1;\widehat{P}}\big(s'\sim P^\star(s,a)\big) \parallel V^{\pi^\star}_{h+1;\widehat{P}}\big(s'\sim \widehat{P}(s,a)\big)\Big)} \right] \\
&\quad + 5 \sum_{h=0}^{H-1} \mathbb{E}_{s,a\sim d_h^{\pi^\star}} \left[D_\triangle\Big(V^{\pi^\star}_{h+1;\widehat{P}}\big(s'\sim P^\star(s,a)\big) \parallel V^{\pi^\star}_{h+1;\widehat{P}}\big(s'\sim \widehat{P}(s,a)\big)\Big) \right] \\
&\leq 8 \sum_{h=0}^{H-1} \mathbb{E}_{s,a\sim d_h^{\pi^\star}} \left[\sqrt{\left(\mathbb{V}_{P^\star} V^{\pi^\star}_{h+1;\widehat{P}}\right)(s,a) \mathbb{H}^2\Big(V^{\pi^\star}_{h+1;\widehat{P}}\big(s'\sim P^\star(s,a)\big) \parallel V^{\pi^\star}_{h+1;\widehat{P}}\big(s'\sim \widehat{P}(s,a)\big)\Big)} \right] \\
&\quad + 20 \sum_{h=0}^{H-1} \mathbb{E}_{s,a\sim d_h^{\pi^\star}} \left[\mathbb{H}^2\Big(V^{\pi^\star}_{h+1;\widehat{P}}\big(s'\sim P^\star(s,a)\big) \parallel V^{\pi^\star}_{h+1;\widehat{P}}\big(s'\sim \widehat{P}(s,a)\big)\Big) \right] \\
&\leq 8 \sum_{h=0}^{H-1} \mathbb{E}_{s,a\sim d_h^{\pi^\star}} \left[\sqrt{\left(\mathbb{V}_{P^\star} V^{\pi^\star}_{h+1;\widehat{P}}\right)(s,a) \mathbb{H}^2\Big(P^\star(s,a) \parallel \widehat{P}\big(s,a\big)\Big)} \right] + 20 \sum_{h=0}^{H-1} \mathbb{E}_{s,a\sim d_h^{\pi^\star}} \left[\mathbb{H}^2\Big(P^\star(s,a) \parallel \widehat{P}\big(s,a\big)\Big) \right]
\end{aligned} \tag{A.52}$$

where in the first inequality, we use Lemma 3.2.1 to bound the difference of two means $\mathbb{E}_{s'\sim P^\star(s,a)} V^{\pi^\star}_{h+1;\widehat{P}}(s') - \mathbb{E}_{s'\sim \widehat{P}(s,a)} V^{\pi^*}_{h+1;\widehat{P}}(s')$ using variances and the triangle discrimination; in the second inequality we use the fact that that triangle discrimination is equivalent to squared Hellinger distance, i.e., $D_\triangle \leq 4\mathbb{H}^2$; the third inequality is via data processing inequality on the squared Hellinger distance. Next, starting from Equation A.52, with probability at least $1-\delta$, we have

$$
\begin{aligned}
&8\sum_{h=0}^{H-1}\mathbb{E}_{s,a\sim d_h^{\pi^\star}}\left[\sqrt{\left(\mathbb{V}_{P^\star}V^{\pi^\star}_{h+1;\widehat{P}}\right)(s,a)\mathbb{H}^2\Big(P^\star(s,a)\parallel\widehat{P}\big(s,a\big)\Big)}\right]+20\sum_{h=0}^{H-1}\mathbb{E}_{s,a\sim d_h^{\pi^\star}}\left[\mathbb{H}^2\Big(P^\star(s,a)\parallel\widehat{P}\big(s,a\big)\Big)\right]\\
&\leq 8\sqrt{\sum_{h=0}^{H-1}\mathbb{E}_{s,a\sim d_h^{\pi^\star}}\left[\left(\mathbb{V}_{P^\star}V^{\pi^\star}_{h+1;\widehat{P}}\right)(s,a)\right]\cdot\sum_{h=0}^{H-1}\mathbb{E}_{s,a\sim d_h^{\pi^\star}}\left[\mathbb{H}^2\Big(P^\star(s,a)\parallel\widehat{P}\big(s,a\big)\Big)\right]}\\
&\quad+20\sum_{h=0}^{H-1}\mathbb{E}_{s,a\sim d_h^{\pi^\star}}\left[\mathbb{H}^2\Big(P^\star(s,a)\parallel\widehat{P}\big(s,a\big)\Big)\right]\\
&\leq 8\sqrt{\sum_{h=0}^{H-1}\mathbb{E}_{s,a\sim d_h^{\pi^\star}}\left[\left(\mathbb{V}_{P^\star}V^{\pi^\star}_{h+1;\widehat{P}}\right)(s,a)\right]\cdot C^{\pi^\star}\frac{1}{K}\sum_{k=1}^{K}\sum_{h=0}^{H-1}\mathbb{H}^2(P^\star(s_h^k,a_h^k)||\widehat{P}(s_h^k,a_h^k))}\\
&\quad+20C^{\pi^\star}\frac{1}{K}\sum_{k=1}^{K}\sum_{h=0}^{H-1}\mathbb{H}^2(P^\star(s_h^k,a_h^k)||\widehat{P}(s_h^k,a_h^k))\\
&\leq 8\sqrt{\sum_{h=0}^{H-1}\mathbb{E}_{s,a\sim d_h^{\pi^\star}}\left[\left(\mathbb{V}_{P^\star}V^{\pi^\star}_{h+1;\widehat{P}}\right)(s,a)\right]\cdot\frac{22C^{\pi^\star}\log(|\mathcal{P}|/\delta)}{K}}+\frac{440C^{\pi^\star}\log(|\mathcal{P}|/\delta)}{K},
\end{aligned}
\tag{A.53}
$$

where the first inequality is by the Cauchy–Schwarz inequality; the second inequality is by the definition of single policy coverage (Definition 3.3); the last inequality holds with probability at least $1-\delta$ with Equation A.49. Substituting Equation A.53 into Equation A.52 ends our proof.

We denote $\mathcal{E}$ as the event that Lemma A.1.13 holds. Under the event $\mathcal{E}$, we prove the following lemma to bound $\sum_{h=0}^{H-1}\mathbb{E}_{s,a\sim d_h^{\pi^\star}}\left[\left(\mathbb{V}_{P^\star}V^{\pi^\star}_{h+1;\widehat{P}}\right)(s,a)\right]$

with $\widetilde{O}(\sum_{h=0}^{H-1}\mathbb{E}_{s,a\sim d_h^{\pi^*}}\left[\left(\mathbb{V}_{P^\star}V_{h+1}^{\pi^*}\right)(s,a)\right]+C^{\pi^*}\log(|\mathcal{P}|/\delta)/K)$.

Lemma A.1.14 (Variance Conversion Lemma for offline RL)**.** *Under event $\mathcal{E}$, we have*

$$\sum_{h=0}^{H-1}\mathbb{E}_{s,a\sim d_h^{\pi^\star}}\left[\left(\mathbb{V}_{P^\star}V_{h+1;\widehat{P}}^{\pi^\star}\right)(s,a)\right]\le O\Big(\sum_{h=0}^{H-1}\mathbb{E}_{s,a\sim d_h^{\pi^*}}\left[\left(\mathbb{V}_{P^\star}V_{h+1}^{\pi^*}\right)(s,a)\right]+C^{\pi^*}\frac{\log(|\mathcal{P}|/\delta)}{K}\Big).$$

Proof[Proof of Lemma A.1.14] For notational simplicity, we denote $A:=\sum_{h=0}^{H-1}\mathbb{E}_{s,a\sim d_h^{\pi^\star}}\left[\left(\mathbb{V}_{P^\star}V_{h+1;\widehat{P}}^{\pi^\star}\right)(s,a)\right]$, and we denote
$B:=\sum_{h=0}^{H-1}\mathbb{E}_{s,a\sim d_h^{\pi^\star}}\left[\left(\mathbb{V}_{P^\star}V_{h+1}^{\pi^\star}\right)(s,a)\right]$,
$C:=\sum_{h=0}^{H-1}\mathbb{E}_{s,a\sim d_h^{\pi^\star}}\left[\left(\mathbb{V}_{P^\star}(V_{h+1;\widehat{P}}^{\pi^\star}-V_{h+1}^{\pi^\star})\right)(s,a)\right]$, then we have

$$A\le 2B+2C,$$

since $\mathbb{V}_{P^\star}(a+b)\le 2\mathbb{V}_{P^\star}(a)+2\mathbb{V}_{P^\star}(b)$.

Let $\Delta_{h+1}^{\pi^\star}:=V_{h+1;\widehat{P}}^{\pi^\star}-V_{h+1}^{\pi^\star}$. Then, w.p. at least $1-\delta$, we have

$$\begin{aligned}
C&=\sum_{h=0}^{H-1}\mathbb{E}_{s,a\sim d_h^{\pi^\star}}\left[\left(P^\star(\Delta_{h+1}^{\pi^\star})^2\right)(s,a)-\left(P^\star\Delta_{h+1}^{\pi^\star}\right)^2(s,a)\right]\\
&=\sum_{h=0}^{H-1}\mathbb{E}_{s\sim d_{h+1}^{\pi^\star}}\left[(\Delta_{h+1}^{\pi^\star})^2(s)\right]-\sum_{h=0}^{H-1}\mathbb{E}_{s,a\sim d_h^{\pi^\star}}\left[\left(P^\star\Delta_{h+1}^{\pi^\star}\right)^2(s,a)\right]\\
&\le\sum_{h=0}^{H-1}\mathbb{E}_{s,a\sim d_h^{\pi^\star}}\left[(\Delta_h^{\pi^\star})^2(s)-\left(P^\star\Delta_{h+1}^{\pi^\star}\right)^2(s,a)\right]\\
&=\sum_{h=0}^{H-1}\mathbb{E}_{s,a\sim d_h^{\pi^\star}}\left[\left((\Delta_h^{\pi^\star})(s)+\left(P^\star\Delta_{h+1}^{\pi^\star}\right)(s,a)\right)\cdot\left((\Delta_h^{\pi^\star})(s)-\left(P^\star\Delta_{h+1}^{\pi^\star}\right)(s,a)\right)\right],
\end{aligned}\tag{A.54}$$

where the first equality is by the definition of variance, the second equality holds as $d_h^{\pi^\star}$ is the occupancy measure also generated under $P^\star$, the first inequality is just changing the index, the third

equality holds as $a^2 - b^2 = (a+b)\cdot(a-b)$. Starting from Equation A.54, we have

$$\begin{aligned}
&\sum_{h=0}^{H-1} \mathbb{E}_{s,a\sim d_h^{\pi^\star}} \left[\left((\Delta_h^{\pi^\star})(s) + \left(P^\star \Delta_{h+1}^{\pi^\star}\right)(s,a) \right) \cdot \left((\Delta_h^{\pi^\star})(s) - \left(P^\star \Delta_{h+1}^{\pi^\star}\right)(s,a) \right) \right] \\
&\leq 2 \sum_{h=0}^{H-1} \mathbb{E}_{s,a\sim d_h^{\pi^\star}} \left[\left| (\Delta_h^{\pi^\star})(s) - \left(P^\star \Delta_{h+1}^{\pi^\star}\right)(s,a) \right| \right] \\
&= 2 \sum_{h=0}^{H-1} \mathbb{E}_{s,a\sim d_h^{\pi^\star}} \left[\left| (V_{h;\widehat{P}}^{\pi^\star})(s) - \left(P^\star V_{h+1;\widehat{P}}^{\pi^\star}\right)(s,a) - \left((V_h^{\pi^\star})(s) - \left(P^\star V_{h+1}^{\pi^\star}\right)(s,a) \right) \right| \right] \\
&= 2 \sum_{h=0}^{H-1} \mathbb{E}_{s,a\sim d_h^{\pi^\star}} \left[\left| r(s,a) + \left(\widehat{P} V_{h+1;\widehat{P}}^{\pi^\star}\right)(s,a) - \left(P^\star V_{h+1;\widehat{P}}^{\pi^\star}\right)(s,a) - r(s,a) \right| \right] \\
&= 2 \sum_{h=0}^{H-1} \mathbb{E}_{s,a\sim d_h^{\pi^\star}} \left[\left| \left(\widehat{P} V_{h+1;\widehat{P}}^{\pi^\star}\right)(s,a) - \left(P^\star V_{h+1;\widehat{P}}^{\pi^\star}\right)(s,a) \right| \right], \quad \text{(A.55)}
\end{aligned}$$

where the inequality holds as the value functions are all bounded by 1 by the assumption that the total reward over any trajectory is bounded by 1, the first equality is by the definition of $\Delta_{h+1}^{\pi^\star}$, the second equality is because a is drawn from $\pi^\star$. Starting from Equation A.55, we have

$$\begin{aligned}
&2 \sum_{h=0}^{H-1} \mathbb{E}_{s,a\sim d_h^{\pi^\star}} \left[\left| \left(\widehat{P} V_{h+1;\widehat{P}}^{\pi^\star}\right)(s,a) - \left(P^\star V_{h+1;\widehat{P}}^{\pi^\star}\right)(s,a) \right| \right] \\
&= 2 \sum_{h=0}^{H-1} \mathbb{E}_{s,a\sim d_h^{\pi^\star}} \left[\left| \mathbb{E}_{s'\sim P^\star(s,a)} \left[V_{h+1;\widehat{P}}^{\pi^\star}(s') \right] - \mathbb{E}_{s'\sim \widehat{P}(\cdot|s,a)} \left[V_{h+1;\widehat{P}}^{\pi^\star}(s') \right] \right| \right] \\
&\leq 16 \sqrt{ \sum_{h=0}^{H-1} \mathbb{E}_{s,a\sim d_h^{\pi^\star}} \left[\left(\mathbb{V}_{P^\star} V_{h+1;\widehat{P}}^{\pi^\star} \right)(s,a) \right] \cdot \frac{22 C^{\pi^\star} \log(|\mathcal{P}|/\delta)}{K} } + \frac{880 C^{\pi^\star} \log(|\mathcal{P}|/\delta)}{K} \\
&= 16 \sqrt{ A \cdot \frac{22 C^{\pi^\star} \log(|\mathcal{P}|/\delta)}{K} } + \frac{880 C^{\pi^\star} \log(|\mathcal{P}|/\delta)}{K} \qquad \text{(A.56)}
\end{aligned}$$

where the inequality holds with probability at least $1-\delta$ by Lemma A.1.13, and the second equality is by definition of A.

Then combining Equation A.54, Equation A.55 and Equation A.56, we obtain an upper bound for C, which suggests

$$\begin{aligned} A &\leq 2B + 2C \\ &\leq 2B + \frac{1760C^{\pi^\star}\log(|\mathcal{P}|/\delta)}{K} + 32\sqrt{\frac{22C^{\pi^\star}\log(|\mathcal{P}|/\delta)}{K}}\cdot\sqrt{A}. \end{aligned}$$

Then, with the fact that $x \leq 2a + b^2$ if $x \leq a + b\sqrt{x}$, we have

$$A \leq 4B + \frac{3520C^{\pi^\star}\log(|\mathcal{P}|/\delta)}{K} + \frac{22528C^{\pi^\star}\log(|\mathcal{P}|/\delta)}{K} \leq O(B + \frac{C^{\pi^\star}\log(|\mathcal{P}|/\delta)}{K}).$$

With the above lemmas, we can now prove the final results of Theorem 3.4.1. We have that w.p. at least $1-\delta$

$$\begin{aligned} V_{0;P^\star}^{\pi^\star} - V_{0;P^\star}^{\hat{\pi}} &\leq O\Big(\sqrt{A\cdot\frac{C^{\pi^\star}\log(|\mathcal{P}|/\delta)}{K}} + \frac{C^{\pi^\star}\log(|\mathcal{P}|/\delta)}{K}\Big) \\ &\leq O\Big(\sqrt{(B + \frac{C^{\pi^\star}\log(|\mathcal{P}|/\delta)}{K})\cdot\frac{C^{\pi^\star}\log(|\mathcal{P}|/\delta)}{K}} + \frac{C^{\pi^\star}\log(|\mathcal{P}|/\delta)}{K}\Big) \\ &\leq O\Big(\sqrt{B\cdot\frac{C^{\pi^\star}\log(|\mathcal{P}|/\delta)}{K}} + \sqrt{\frac{C^{\pi^\star}\log(|\mathcal{P}|/\delta)}{K}\cdot\frac{C^{\pi^\star}\log(|\mathcal{P}|/\delta)}{K}} + \frac{C^{\pi^\star}\log(|\mathcal{P}|/\delta)}{K}\Big) \\ &= O\Big(\sqrt{\sum_{h=0}^{H-1}\mathbb{E}_{s,a\sim d_h^{\pi^\star}}\left[(\mathbb{V}_{P^\star}V_{h+1}^{\pi^\star})(s,a)\right]\cdot\frac{C^{\pi^\star}\log(|\mathcal{P}|/\delta)}{K}} + \frac{C^{\pi^\star}\log(|\mathcal{P}|/\delta)}{K}\Big) \\ &= O\Big(\sqrt{\frac{\mathrm{VaR}_{\pi^\star}C^{\pi^\star}\log(|\mathcal{P}|/\delta)}{K}} + \frac{C^{\pi^\star}\log(|\mathcal{P}|/\delta)}{K}\Big), \end{aligned} \tag{A.57}$$

where in the last equation we use Lemma A.1.6, and $\mathrm{VaR}_{\pi^\star} := \mathbb{E}\left[\left(\sum_{h=0}^{H-1} r(s_h,\pi^\star(s_h)) - V_0^{\pi^\star}\right)^2\right]$.

A.1.10 Proof of Corollary 3.4

Proof[Proof of Corollary 3.4] By Lemma A.1.6, we have

$$\mathrm{VaR}_{\pi^*} = \sum_{h=0}^{H-1} \mathbb{E}_{s,a\sim d_h^{\pi^*}} \left[\left(\mathbb{V}_{P^\star} V_{h+1}^{\pi^*} \right)(s,a) \right] \tag{A.58}$$

Therefore, when $P^\star$ is deterministic, the $\mathbb{E}_{s,a\sim d_h^{\pi^*}} \left[\left(\mathbb{V}_{P^\star} V_{h+1}^{\pi^*} \right)(s,a) \right]$ terms are all 0 for any $k \in [K-1]$ and $h \in [H-1]$, and then the VaR_{π^*} term in the higher order term in Theorem 3.4.1 is 0.

A.1.11 Proof of Corollary 3.5

Proof[Proof of Corollary 3.5] This claim follows the proof of Theorem 3.4.1, while we take a different choice of β that depends on the bracketing number and follow the MLE guarantee in Lemma A.1.8 for infinite model class.

A.1.12 Proof of the claim in Example 2

Proof Recall that in Definition 3.3, we have

$$C_{\mathcal{D}}^{\pi^*} := \max_{h,P\in\mathcal{P}} \frac{\mathbb{E}_{s,a\sim d_h^{\pi^\star}} \mathbb{H}^2 \left(P(s,a) \parallel P^\star(s,a) \right)}{1/K \sum_{k=1}^K \mathbb{H}^2 \left(P(s_h^k, a_h^k) \parallel P^\star(s_h^k, a_h^k) \right)} .$$

For each step h, define two distributions, p_h, q_h, where $p_h(s,a) = d^{\pi^*}(s,a)$, $q_h(s,a) = \frac{1}{K} \sum_{k=1}^K \mathbb{I}\{(s,a) = (s_h^k, a_h^k)\}$, and we define

$f(s,a,P) = \mathbb{H}^2(P(s,a) \parallel P^\star(s,a))$, then we have

$$\begin{aligned} C_{\mathcal{D}}^{\pi^*} &= \max_{h,P\in\mathcal{P}} \frac{\mathbb{E}_{s,a\sim p_h} f(s,a,P)}{\mathbb{E}_{s,a\sim q_h} f(s,a,P)} \\ &= \max_{h,P\in\mathcal{P}} \frac{\mathbb{E}_{s,a\sim q_h} \frac{p_h(s,a)}{q_h(s,a)} f(s,a,P)}{\mathbb{E}_{s,a\sim q_h} f(s,a,P)} \\ &\leq \max_{h,s,a} \frac{p_h(s,a)}{q_h(s,a)} \\ &\leq \max_{h,s,a} \frac{1}{q_h(s,a)} \,. \end{aligned} \tag{A.59}$$

Note that for all h, $\{(s_h^k, a_h^k)\}_{k=1}^K$ are i.i.d. samples drawn from $d_h^{\pi^b}$, therefore, $\mathbb{E}[\mathbb{I}\{(s_h^k, a_h^k) = (s,a)\}] = d_h^{\pi^b}(s,a)$. By Hoeffding's inequality and with a union bound over s, a, h, and for $K \geq \frac{2\log(\frac{|\mathcal{S}||\mathcal{A}|H}{\delta})}{\rho_{\min}^2}$, w.p. at least $1-\delta$, we have

$$\begin{aligned} q_h(s,a) &= \frac{1}{K}\sum_{k=1}^K \mathbb{I}\{(s_h^k, a_h^k) = (s,a)\} \\ &\geq d_h^{\pi^b}(s,a) - \sqrt{\frac{\log(\frac{|\mathcal{S}||\mathcal{A}|H}{\delta})}{2K}} \\ &\geq \frac{d_h^{\pi^b}(s,a)}{2} \,, \end{aligned} \tag{A.60}$$

where in the last inequality we use the assumption that $d_h^{\pi^b}(s,a) \geq \rho_{\min}, \forall s,a,h$, which gives us $K \geq \frac{2\log(\frac{|\mathcal{S}||\mathcal{A}|H}{\delta})}{\rho_{\min}^2} \geq \max_{s,a,h} \frac{2\log(\frac{|\mathcal{S}||\mathcal{A}|H}{\delta})}{(d_h^{\pi^b}(s,a))^2}$, so $K \geq \frac{2\log(\frac{|\mathcal{S}||\mathcal{A}|H}{\delta})}{(d_h^{\pi^b}(s,a))^2}$ for any s,a,h.

Therefore, with $K \geq \frac{2\log(\frac{|\mathcal{S}||\mathcal{A}|H}{\delta})}{\rho_{\min}^2}$, we have that w.p. at least $1-\delta$

$$C_{\mathcal{D}}^{\pi^*} \leq \max_{h,s,a} \frac{1}{q_h(s,a)} \leq \max_{h,s,a} \frac{2}{d_h^{\pi^b}(s,a)} \leq \frac{2}{\rho_{min}} \,. \tag{A.61}$$

A.2 Appendix for Chapter 4

We provide missing proofs and theoretical results of our paper in the Appendix sections:

- In Appendix A.2.1, we provide the missing results of Section 4.3. We first provide the proof of Proposition 4.1, then we analyze the suboptimality gap of the Pessimistic Value Iteration (PEVI) ([91]) in the contextual linear MDP setting without context information.
- In Appendix A.2.2, we provide the proofs of our main theorems on the suboptimality bounds of PERM and PPPO in Section 4.4.
- In Appendix A.2.3, we state and prove the suboptimality bounds we promised in Remarks 9 and 11, where we merge the sampled contexts into m groups ($m < n$) to reduce the computational complexity in practical settings.
- In Appendix A.2.4, we provide the proofs of results in Section 4.5 on linear MDPs. Namely, we provide proof of Theorem 4.5.1, proof of Corollary 4.1.

A.2.1 Results in Section 4.3

A.2.1.1 Proof of Proposition 4.1

Let $\mathcal{D}' = \{(x^{\tau}_{c_\tau,h}, a^{\tau}_{c_\tau,h}, r^{\tau}_{c_\tau,h})\}_{h=1,\tau=1}^{H,K}$ denote the merged dataset, where each trajectory belongs to a context c_τ. For simplicity, let $\mathcal{D}_c$ denote the collection of trajectories that belong to MDP $\mathcal{M}_c$. Then each trajectory in $\mathcal{D}'$ is generated by the following steps:

- The experimenter randomly samples an environment $c \sim C$.
- The experimenter collect a trajectory from the episodic MDP $\mathcal{M}_c$.

Then for any x', r', τ we have

$$\begin{aligned}
&\mathbb{P}_{\mathcal{D}'}(r^\tau_{c_\tau,h} = r', x^\tau_{c_\tau,h+1} = x' | \{(x^j_{c_j,h}, a^j_{c_j,h})\}^\tau_{j=1}, \{r^j_{c_j,h}, x^j_{c_j,h+1}\}^{\tau-1}_{j=1}) \\
&= \frac{\mathbb{P}_{\mathcal{D}'}(r^\tau_{c_\tau,h} = r', x^\tau_{c_\tau,h+1} = x', \{(x^j_{c_j,h}, a^j_{c_j,h})\}^\tau_{j=1}, \{r^j_{c_j,h}, x^j_{c_j,h+1}\}^{\tau-1}_{j=1})}{\mathbb{P}_{\mathcal{D}'}(\{(x^j_{c_j,h}, a^j_{c_j,h})\}^\tau_{j=1}, \{r^j_{c_j,h}, x^j_{c_j,h+1}\}^{\tau-1}_{j=1})} \\
&= \sum_{c\in C} \mathbb{P}_{\mathcal{D}'}(r^\tau_{c_\tau,h} = r', x^\tau_{c_\tau,h+1} = x' | \{(x^j_{c_j,h}, a^j_{c_j,h})\}^\tau_{j=1}, \{r^j_{c_j,h}, x^j_{c_j,h+1}\}^{\tau-1}_{j=1}, c_\tau = c) q(c),
\end{aligned} \tag{A.62}$$

where

$$q(c') := \frac{\mathbb{P}_{\mathcal{D}'}(\{(x^j_{c_j,h}, a^j_{c_j,h})\}^\tau_{j=1}, \{r^j_{c_j,h}, x^j_{c_j,h+1}\}^{\tau-1}_{j=1}, c_\tau = c')}{\sum_{c\in C} \mathbb{P}_{\mathcal{D}'}(\{(x^j_{c_j,h}, a^j_{c_j,h})\}^\tau_{j=1}, \{r^j_{c_j,h}, x^j_{c_j,h+1}\}^{\tau-1}_{j=1}, c_\tau = c)}.$$

Next, we further have

$$\begin{aligned}
&\text{(A.62)} \\
&= \sum_{c\in C} \mathbb{P}_c(r_{c,h}(s_h) = r', s_{h+1} = x' | s_h = x^\tau_{c_\tau,h}, a_h = a^\tau_{c_\tau,h}) q(c) \\
&= \sum_{c\in C} \frac{\mathbb{P}_c(r_{c,h}(s_h) = r', s_{h+1} = x' | s_h = x^\tau_{c_\tau,h}, a_h = a^\tau_{c_\tau,h}) \mathbb{P}_{\mathcal{D}'}(s_h = x^\tau_{c_\tau,h}, a_h = a^\tau_{c_\tau,h}, c_\tau = c)}{\sum_{c\in C} \mathbb{P}_{\mathcal{D}'}(s_h = x^\tau_{c_\tau,h}, a_h = a^\tau_{c_\tau,h}, c_\tau = c)} \\
&= \sum_{c\in C} p(c) \cdot \frac{\mathbb{P}_c(r_{c,h}(s_h) = r', s_{h+1} = x' | s_h = x^\tau_{c_\tau,h}, a_h = a^\tau_{c_\tau,h}) \mathbb{P}_c(s_h = x^\tau_{c_\tau,h}, a_h = a^\tau_{c_\tau,h})}{\sum_{c\in C} p(c) \cdot \mathbb{P}_c(s_h = x^\tau_{c_\tau,h}, a_h = a^\tau_{c_\tau,h})} \\
&= \mathbb{E}_{c\sim C} \frac{\mathbb{P}_c(r_{c,h}(s_h) = r', s_{h+1} = x' | s_h = x^\tau_{c_\tau,h}, a_h = a^\tau_{c_\tau,h}) \mu_{c,h}(x^\tau_{c_\tau,h}, a^\tau_{c_\tau,h})}{\mathbb{E}_{c\sim C} \mu_{c,h}(x^\tau_{c_\tau,h}, a^\tau_{c_\tau,h})},
\end{aligned}$$

where the first equality holds since for all trajectories τ satisfying $c_\tau = c$, they are compliant with $\mathcal{M}_c$, the second one holds since all trajectories are independent of each other, the third and fourth ones hold due to the definition of $\mu_{c,h}(\cdot, \cdot)$.

A.2.1.2 PEVI algorithm

Algorithm 14 [91] Pessimistic Value Iteration (PEVI)

Require: Dataset $\mathcal{D} = \{(x^\tau_{c_\tau,h}, a^\tau_{c_\tau,h}, r^\tau_{c_\tau,h})_{h=1}^H\}_{\tau=1}^K$, confidence probability $\delta \in (0,1)$.

Initialization: Set $\hat{V}_{H+1}(\cdot) \leftarrow 0$.
for step $h = H, H-1, \ldots, 1$ **do**
  Set $\Lambda_h \leftarrow \sum_{\tau=1}^K \phi(x_h^\tau, a_h^\tau)\phi(x_h^\tau, a_h^\tau)^\top + \lambda \cdot I$.
  Set $\hat{w}_h \leftarrow \Lambda_h^{-1}(\sum_{\tau=1}^K \phi(x_h^\tau, a_h^\tau) \cdot (r_h^\tau + \hat{V}_{h+1}(x_{h+1}^\tau)))$.
  Set $\Gamma_h(\cdot,\cdot) \leftarrow \beta(\delta) \cdot (\phi(\cdot,\cdot)^\top \Lambda_h^{-1} \phi(\cdot,\cdot))^{1/2}$.
  Set $\hat{Q}_h(\cdot,\cdot) \leftarrow \min\{\phi(\cdot,\cdot)^\top \hat{w}_h - \Gamma_h(\cdot,\cdot), H-h+1\}^+$.
  Set $\hat{\pi}_h(\cdot\,|\,\cdot) \leftarrow \operatorname{argmax}_{\pi_h} \langle \hat{Q}_h(\cdot,\cdot), \pi_h(\cdot\,|\,\cdot)\rangle_{\mathcal{A}}$.
  Set $\hat{V}_h(\cdot) \leftarrow \langle \hat{Q}_h(\cdot,\cdot), \hat{\pi}_h(\cdot\,|\,\cdot)\rangle_{\mathcal{A}}$.
end for
return $\pi^{\text{PEVI}} = \{\hat{\pi}_h\}_{h=1}^H$.

We analyze the suboptimality gap of the Pessimistic Value Iteration (PEVI) ([91]) in the contextual linear MDP setting without context information to demonstrate that by finding the optimal policy for $\bar{\mathcal{M}}$ is not enough to find the policy that performs well on MDPs with context information.

Pessimistic Value Iteration (PEVI). Let $\overline{\pi}^*$ be the optimal policy w.r.t. the average MDP $\bar{\mathcal{M}}$. We analyze the performance of the Pessimistic Value Iteration (PEVI) [91] under the unknown context information setting. The details of PEVI is in Algo.14.

Suppose that $\bar{\mathcal{D}}$ consists of K number of trajectories generated i.i.d. following by a fixed behavior policy $\bar{\pi}$. Then the following theorem shows the suboptimality gap for Algo.14 does not converge to 0 even when the data size grows to infinity.

Theorem A.2.1. *Assume that $\bar{\pi}$ In Algo.6, we set*

$$\lambda = 1, \quad \beta(\delta) = c' \cdot dH\sqrt{\log(4dHK/\delta)}\,, \tag{A.63}$$

where $c' > 0$ is a positive constant. Suppose we have $K \geq \tilde{c} \cdot d\log(4dH/\xi)$, where $\tilde{c} > 0$ is a sufficiently large positive constant that depends on c. Then we have: w.p. at least $1-\delta$, for the output policy π^{PEVI} of Algo.14,

$$\sup_{\pi} V^{\pi}_{\bar{\mathcal{M}},1} - V^{\pi^{PEVI}}_{\bar{\mathcal{M}},1} \leq c'' \cdot d^{3/2}H^2K^{-1/2}\sqrt{\log(4dHK/\delta)}, \tag{A.64}$$

and the suboptimality gap satisfies

$$\begin{aligned}\mathrm{SubOpt}(\pi^{\mathrm{PEVI}}) \leq\ & c'' \cdot d^{3/2}H^2K^{-1/2}\sqrt{\log(4dHK/\delta)} \\ & + 2\sup_{\pi} \left|V^{\pi}_{\bar{\mathcal{M}},1}(x_1) - \mathbb{E}_{c\sim C}V^{\pi}_{c,1}(x_1)\right|, \end{aligned} \tag{A.65}$$

where $c'' > 0$ is a positive constant that only depends on c and c'.

Proof of Theorem A.2.1. First, we define the value function on the average MDP $\bar{\mathcal{M}}$ as follows.

$$\overline{V}^{\pi}_h(x) = \mathbb{E}_{\pi,\bar{\mathcal{M}}}\Big[\sum_{i=h}^{H} r_i(s_i,a_i) \,\Big|\, s_h = x\Big]. \tag{A.66}$$

We then decompose the suboptimality gap as follows.

$$\begin{aligned}&\mathrm{SubOpt}(\pi^{\mathrm{PEVI}}) \\ &= \mathbb{E}_{c\sim C}\big[V^{\pi^*}_{c,1}(x_1)\big] - \mathbb{E}_{c\sim C}\big[V^{\pi^{\mathrm{PEVI}}}_{c,1}(x_1)\big] \\ &= \overline{V}^{\overline{\pi}^*}_1(x_1) - \overline{V}^{\pi^{\mathrm{PEVI}}}_1(x_1) + \big(\mathbb{E}_{c\sim C}\big[V^{\pi^*}_{c,1}(x_1)\big] - \overline{V}^{\overline{\pi}^*}_1(x_1)\big) \\ &+ \big(\overline{V}^{\pi^{\mathrm{PEVI}}}_1(x_1) - \mathbb{E}_{c\sim C}\big[V^{\pi^{\mathrm{PEVI}}}_{c,1}(x_1)\big]\big) \\ &\leq \overline{V}^{\overline{\pi}^*}_1(x_1) - \overline{V}^{\pi^{\mathrm{PEVI}}}_1(x_1) + 2\sup_{\pi}\left|V^{\pi}_{\bar{\mathcal{M}},1}(x_1) - \mathbb{E}_{c\sim C}V^{\pi}_{c,1}(x_1)\right|. \end{aligned} \tag{A.67}$$

Then, applying Corollary 4.6 in [91], we can get that w.p. at least $1-\delta$

$$\overline{V}_1^{\bar{\pi}^*}(x_1) - \overline{V}_1^{\pi^{\mathrm{PEVI}}}(x_1) \leq c'' \cdot d^{3/2} H^2 K^{-1/2} \sqrt{\log(4dHK/\delta)}\,, \quad \text{(A.68)}$$

which, together with Eq.(A.67) completes the proof.

□

Theorem A.2.1 shows that by adapting the standard pessimistic offline RL algorithm over the offline dataset without context information, the learned policy π^{PEVI} converges to the optimal policy $\bar{\pi}^*$ over the average MDP $\bar{\mathcal{M}}$.

A.2.2 Proof of Theorems in Section 4.4

A.2.2.1 Proof of Theorem 4.4.1

We define the model estimation error as

$$\iota_{i,h}^{\pi}(x,a) = (\mathbb{B}_{i,h}\hat{V}_{i,h+1}^{\pi})(x,a) - \hat{Q}_{i,h}^{\pi}(x,a). \quad \text{(A.69)}$$

And we define the following condition

$$\begin{aligned} &\left|(\hat{\mathbb{B}}_{i,h}\hat{V}_{i,h+1}^{\pi})(x,a) - (\mathbb{B}_{i,h}\hat{V}_{i,h+1}^{\pi})(x,a)\right| \leq \Gamma_{i,h}(x,a) \\ &\text{for all } i \in [n], \pi \in \Pi, (x,a) \in \mathcal{S}\times\mathcal{A}, h \in [H]\,. \end{aligned} \quad \text{(A.70)}$$

We introduce the following lemma to bound the model estimation error.

Lemma A.2.2 (Model estimation error bound (Adapted from Lemma 5.1 in [91])**.** *Under the condition of Eq.(A.70), we have*

$$0 \leq \iota_{i,h}^{\pi}(x,a) \leq 2\Gamma_{i,h}(x,a), \quad \textit{for all } i \in [n],\ \pi \in \Pi,\ (x,a) \in \mathcal{S}\times\mathcal{A},\ h \in [H]. \quad \text{(A.71)}$$

Then, we prove the following lemma for pessimism in V values.

Lemma A.2.3 (Pessimism for Estimated V Values)**.** *Under the condition of Eq.(A.70), for any $i \in [n], \pi \in \Pi, x \in \mathcal{S}$, we have*

$$V_{i,h}^{\pi}(x) \geq \hat{V}_{i,h}^{\pi}(x)\,. \tag{A.72}$$

Proof. For any $i \in [n], \pi \in \Pi, x \in \mathcal{S}, a \in \mathcal{A}$, we have

$$\begin{aligned}
&Q_{i,h}^{\pi}(x,a) - \hat{Q}_{i,h}^{\pi}(x,a) \\
&\geq r_{i,h}(x,a) + (\mathbb{B}_{i,h}V_{i,h+1}^{\pi})(x,a) - \big(r_{i,h}(s,a) + (\hat{\mathbb{B}}_{i,h}\hat{V}_{i,h+1}^{\pi})(x,a) - \Gamma_{i,h}(x,a)\big) \\
&= (\mathbb{B}_{i,h}V_{i,h+1}^{\pi})(x,a) - (\mathbb{B}_{i,h}\hat{V}_{i,h+1}^{\pi})(x,a) + \Gamma_{i,h}(x,a) \\
&\quad - \big((\hat{\mathbb{B}}_{i,h}\hat{V}_{i,h+1}^{\pi})(x,a) - \mathbb{B}_{i,h}\hat{V}_{i,h+1}^{\pi})(x,a)\big) \\
&\geq (\mathbb{B}_{i,h}V_{i,h+1}^{\pi})(x,a) - (\mathbb{B}_{i,h}\hat{V}_{i,h+1}^{\pi})(x,a) \\
&= \big(P_{i,h}(V_{i,h+1}^{\pi} - \hat{V}_{i,h+1}^{\pi})\big)(x,a)\,,
\end{aligned}$$

where the second inequality is because of Eq.(A.70). And since in the $H+1$ step we have $V_{i,H+1}^{\pi} = \hat{V}_{i,h+1}^{\pi} = 0$, we can get $Q_{i,H}^{\pi}(x,a) - \hat{Q}_{i,H}^{\pi}(x,a)$. Then we use induction to prove $Q_{i,h}^{\pi}(x,a) \geq \hat{Q}_{i,h}^{\pi}(x,a)$ for all h. Given $Q_{i,h+1}^{\pi}(x,a) \geq \hat{Q}_{i,h+1}^{\pi}(x,a)$, we have

$$\begin{aligned}
&Q_{i,h}^{\pi}(x,a) - \hat{Q}_{i,h}^{\pi}(x,a) \\
&\geq \big(P_{i,h}(V_{i,h+1}^{\pi} - \hat{V}_{i,h+1}^{\pi})\big)(x,a) \\
&= \mathbb{E}\langle Q_{i,h+1}^{\pi}(s_{h+1},\cdot) - \hat{Q}_{i,h+1}^{\pi}(s_{h+1},\cdot), \pi_{h+1}(\cdot|s_{h+1})\rangle_{\mathcal{A}} | s_h = x, a_h = a \\
&\geq 0\,.
\end{aligned} \tag{A.73}$$

Then we have

$$V_{i,h}^{\pi}(x) - \hat{V}_{i,h}^{\pi}(x) = \langle Q_{i,h}^{\pi}(x,\cdot) - \hat{Q}_{i,h}^{\pi}(x,\cdot), \pi_h(\cdot\,|\,x)\rangle_{\mathcal{A}} \geq 0\,.$$

□

Then we start our proof.

Proof of Theorem 4.4.1. First, we decompose the suboptimality gap as follows

$$\begin{aligned}
&\mathrm{SubOpt}(\pi^{\mathrm{PERM}}) \\
&= \mathbb{E}_{c\sim C} V_{c,1}^{\pi^*}(x_1) - V_{c,1}^{\hat{\pi}^*}(x_1) \\
&= \mathbb{E}_{c\sim C} V_{c,1}^{\pi^*}(x_1) - \frac{1}{n}\sum_{i=1}^{n} V_{i,1}^{\pi^*}(x_1) + \frac{1}{n}\sum_{i=1}^{n} V_{i,1}^{\pi^{\mathrm{PERM}}}(x_1) - \mathbb{E}_{c\sim C} V_{c,1}^{\pi^{\mathrm{PERM}}}(x_1) \\
&\quad + \frac{1}{n}\sum_{i=1}^{n} \left(V_{i,1}^{\pi^*}(x_1) - V_{i,1}^{\pi^{\mathrm{PERM}}}(x_1)\right). \qquad \text{(A.74)}
\end{aligned}$$

For the first two terms, we can bound them following the standard generalization techniques ([126]), *i.e.*, we use the covering argument, Chernoff bound,and union bound.

Define the distance between policies $d(\pi^1, \pi^2) \triangleq \max_{s\in\mathcal{S}, h\in[H]} \|\pi_h^1(\cdot|s) - \pi_h^2(\cdot|s)\|_1$. We construct the ϵ-covering set $\tilde{\Pi}$ w.r.t. d such that

$$\forall \pi \in \Pi, \exists \tilde{\pi} \in \tilde{\Pi}, s.t. \quad d(\pi, \tilde{\pi}) \leq \epsilon. \qquad \text{(A.75)}$$

Then we have

$$\forall i \in [n], \pi \in \Pi, \exists \tilde{\pi} \in \tilde{\Pi}, s.t. V_{i,1}^{\pi}(x_1) - V_{i,1}^{\tilde{\pi}}(x_1) \leq H\epsilon. \qquad \text{(A.76)}$$

By the definition of the covering number, $\left|\tilde{\Pi}\right| = \mathcal{N}_\epsilon^{\Pi}$. By Chernoff bound and union bound over the policy set $\tilde{\Pi}$, we have with prob. at least $1 - \frac{\delta}{3}$, for any $\tilde{\pi} \in \tilde{\Pi}$,

$$\left|\frac{1}{n}\sum_{i=1}^{n} V_{i,1}^{\tilde{\pi}}(x_1) - \mathbb{E}_{c\sim C} V_{c,1}^{\tilde{\pi}}(x_1)\right| \leq \sqrt{\frac{2\log(6\mathcal{N}_\epsilon^{\Pi}/\delta)}{n}}. \qquad \text{(A.77)}$$

By Eq.(A.76) and Eq.(A.77), $\forall i \in [n], \pi \in \Pi, \exists \tilde{\pi} \in \tilde{\Pi}$ with $\left|\tilde{\Pi}\right| = \mathcal{N}_\epsilon^\Pi,\ s.t. V_{i,1}^\pi(x_1) - V_{i,1}^{\tilde{\pi}}(x_1) \leq H\epsilon$, and with probability at least $1-\delta/3$, we have

$$\begin{aligned}
&\left|\frac{1}{n}\sum_{i=1}^n V_{i,1}^\pi(x_1) - \mathbb{E}_{c\sim C} V_{c,1}^\pi(x_1)\right| \\
&\leq \left|\frac{1}{n}\sum_{i=1}^n V_{i,1}^{\tilde{\pi}}(s_1) - \mathbb{E}_{c\sim C} V_{c,1}^{\tilde{\pi}}(x_1)\right| \\
&+ \left|\frac{1}{n}\sum_{i=1}^n V_{i,1}^\pi(s_1) - \frac{1}{n}\sum_{i=1}^n V_{i,1}^{\tilde{\pi}}(s_1)\right| + \left|\mathbb{E}_{c\sim C} V_{c,1}^{\tilde{\pi}}(x_1) - \mathbb{E}_{c\sim C} V_{c,1}^\pi(x_1)\right| \\
&\leq \sqrt{\frac{2\log(6\mathcal{N}_\epsilon^\Pi/\delta)}{n}} + 2H\epsilon\,. \qquad \text{(A.78)}
\end{aligned}$$

Therefore, we have for the first two terms, w.p. at least $1-\frac{2}{3}\delta$ we can upper bound them with $4H\epsilon + 2\sqrt{\frac{2\log(6\mathcal{N}_\epsilon^\Pi/\delta)}{n}}$.

Then, what remains is to bound the term $\frac{1}{n}\sum_{i=1}^n \left(V_{i,1}^{\pi^*}(x_1) - V_{i,1}^{\pi^{\text{PERM}}}(x_1)\right)$.

First, by similar arguments, we have

$$\begin{aligned}
V_{i,1}^{\pi^*}(x_1) - V_{i,1}^{\pi^{\text{PERM}}}(x_1) &\leq \left(V_{i,1}^{\pi^*}(x_1) - V_{i,1}^{\tilde{\pi}^{\text{PERM}}}(x_1)\right) + |V_{i,1}^{\tilde{\pi}^{\text{PERM}}}(x_1) - V_{i,1}^{\pi^{\text{PERM}}}(x_1)| \\
&\leq H\epsilon + V_{i,1}^{\pi^*}(x_1) - V_{i,1}^{\tilde{\pi}^{\text{PERM}}}(x_1)\,, \qquad \text{(A.79)}
\end{aligned}$$

where $\tilde{\pi}^{\text{PERM}} \in \tilde{\Pi}$ such that $|V_{i,1}^{\tilde{\pi}^{\text{PERM}}}(x_1) - V_{i,1}^{\pi^{\text{PERM}}}(x_1)| \leq H\epsilon$.

By the definition of the oracle in Definition.4.2, the algorithm design of Algo.3 (e.g., we call oracle $\mathbb{O}(\mathcal{D}_h, \hat{V}_{h+1}, \delta/(3nH\mathcal{N}_{(Hn)^{-1}}^\Pi))$), and use a union bound over H steps, n contexts, and $\mathcal{N}_{(Hn)^{-1}}^\Pi$ policies, we have: with probability at least $1-\delta/3$, the condition in Eq.(A.70) holds (with the policy class Π replaced by $\tilde{\Pi}$ (and $\epsilon = 1/(Hn)$).

Then, we have

$$
\begin{aligned}
&\frac{1}{n}\sum_{i=1}^{n}\left(V_{i,1}^{\pi^*}(x_1)-V_{i,1}^{\tilde{\pi}^{\mathrm{PERM}}}(x_1)\right)\\
&\le \frac{1}{n}\sum_{i=1}^{n}\left(V_{i,1}^{\pi^*}(x_1)-\hat{V}_{i,1}^{\tilde{\pi}^{\mathrm{PERM}}}(x_1)\right)\\
&= \frac{1}{n}\sum_{i=1}^{n}\left(V_{i,1}^{\pi^*}(x_1)-\hat{V}_{i,1}^{\pi^{\mathrm{PERM}}}(x_1)\right)+\frac{1}{n}\sum_{i=1}^{n}\left(\hat{V}_{i,1}^{\pi^{\mathrm{PERM}}}(x_1)-\hat{V}_{i,1}^{\tilde{\pi}^{\mathrm{PERM}}}(x_1)\right)\\
&\le \frac{1}{n}\sum_{i=1}^{n}\left(V_{i,1}^{\pi^*}(x_1)-\hat{V}_{i,1}^{\pi^{\mathrm{PERM}}}(x_1)\right)+H\cdot\frac{1}{Hn}\\
&\le \frac{1}{n}\sum_{i=1}^{n}\left(V_{i,1}^{\pi^*}(x_1)-\hat{V}_{i,1}^{\pi^*}(x_1)\right)+1/n\,, \qquad \text{(A.80)}
\end{aligned}
$$

where the first inequality holds because of the pessimism in Lemma A.2.3, the second inequality holds because $|\hat{V}_{i,1}^{\tilde{\pi}^{\mathrm{PERM}}}(x_1)-\hat{V}_{i,1}^{\pi^{\mathrm{PERM}}}(x_1)|\le H\epsilon$ with ϵ here specified as $1/(Hn)$, and the last inequality holds because that in the algorithm design of Algo.4 we set $\pi^{\mathrm{PERM}}=\mathrm{argmax}_{\pi\in\Pi}\frac{1}{n}\sum_{i=1}^{n}\hat{V}_{i,1}^{\pi}(x_1)$.

Then what left is to bound $V_{i,1}^{\pi^*}(x_1)-\hat{V}_{i,1}^{\pi^*}(x_1)$.

And using Lemma A.1 in [91], we have

$$
\begin{aligned}
&V_{i,1}^{\pi^*}(x_1)-\hat{V}_{i,1}^{\pi^*}(x_1)\\
&=-\sum_{h=1}^{H}\mathbb{E}_{\hat{\pi}^*,\mathcal{M}_i}\left[\iota_{i,h}^{\pi^*}(s_h,a_h)\,\middle|\,s_1=x\right]+\sum_{h=1}^{H}\mathbb{E}_{\pi^*,\mathcal{M}_i}\left[\iota_{i,h}^{\pi^*}(s_h,a_h)\,\middle|\,s_1=x\right]\\
&\quad+\sum_{h=1}^{H}\mathbb{E}_{\pi^*,\mathcal{M}_i}\left[\langle\hat{Q}_{i,h}^{\pi^*}(s_h,\cdot),\pi_h^*(\cdot\,|\,s_h)-\pi_h^*(\cdot\,|\,s_h)\rangle_{\mathcal{A}}\,\middle|\,s_1=x\right]\\
&\le 2\sum_{h=1}^{H}\mathbb{E}_{\pi^*,\mathcal{M}_i}\left[\Gamma_{i,h}(s_h,a_h)\,\middle|\,s_1=x\right], \qquad \text{(A.81)}
\end{aligned}
$$

where in the last inequality we use Lemma A.2.2.

Finally, with Eq.(A.209), Eq.(A.78), Eq.(A.79), Eq.(A.80), and Eq.(A.81), with ϵ set as $\frac{1}{nH}$, we can get w.p. at least $1-\delta$

$$\begin{aligned}
&\mathbb{E}_{c\sim C} V_{c,1}^{\pi^*}(x_1) - V_{c,1}^{\pi^{\text{PERM}}}(x_1) \\
&\leq \frac{5}{n} + 2\sqrt{\frac{2\log(6\mathcal{N}^{\Pi}_{(Hn)^{-1}}/\delta)}{n}} + \frac{2}{n}\sum_{i=1}^{n}\sum_{h=1}^{H} \mathbb{E}\pi^*, \mathcal{M}_i\Gamma_{i,h}(s_h,a_h)|s_1 = x_1 \\
&\leq 7\sqrt{\frac{2\log(6\mathcal{N}^{\Pi}_{(Hn)^{-1}}/\delta)}{n}} + \frac{2}{n}\sum_{i=1}^{n}\sum_{h=1}^{H} \mathbb{E}\pi^*, \mathcal{M}_i\Gamma_{i,h}(s_h,a_h)|s_1 = x_1\,.
\end{aligned}$$

□

A.2.2.2 Proof of Theorem 4.4.2

Our proof has two steps. First, we define that

$$\iota_{i,h}(x,a) := \mathbb{B}_{i,h}V_{i,h+1}(x,a) - Q_{i,h}(x,a) \tag{A.82}$$

Then we have the following lemma from [91]:

Lemma A.2.4. *Define the event $\mathcal{E}$ as*

$$\mathcal{E} = \left\{ \left|(\hat{\mathbb{B}}\hat{V}^{\pi_i}_{i,h+1})(x,a) - (\mathbb{B}_{i,h}\hat{V}^{\pi_i}_{i,h+1})(x,a)\right| \leq \Gamma_{i,h}(x,a)\ \forall (x,a)\in\mathcal{S}\times\mathcal{A}, \forall h\in[H], \forall i\in[n] \right\},$$

Then by selecting the input parameter $\xi = \delta/(Hn)$ in $\mathbb{O}$, we have $\mathbb{P}(\mathcal{E}) \geq 1-\delta$ and

$$0 \leq \iota_{i,h}(x,a) \leq 2\Gamma_{i,h}(x,a).$$

Proof. The proof is the same as [Lemma 5.1, [91]] with the probability assigned as $\delta/(Hn)$ and a union bound over $h\in[H], i\in[n]$. □

Next lemma shows the difference between the value of the optimal policy π^* and number n of different policies π_i for n MDPs.

Lemma A.2.5. *Let π be an arbitrary policy. Then we have*

$$\sum_{i=1}^{n}[V_{i,1}^{\pi}(x_1) - V_{i,1}^{\pi^i}(x_1)] = \sum_{i=1}^{n}\sum_{h=1}^{H}\mathbb{E}_{i,\pi}[\langle Q_{i,h}(\cdot,\cdot), \pi_h(\cdot|\cdot) - \pi_{i,h}(\cdot|\cdot)\rangle_{\mathcal{A}}] + \sum_{i=1}^{n}\sum_{h=1}^{H}(\mathbb{E}_{i,\pi}[\iota_{i,h}(x_h,a_h)] - \mathbb{E}_{i,\pi_i}[\iota_{i,h}(x_h,a_h)]) \tag{A.83}$$

Proof. The proof is the same as Lemma 3.1 in [91] except substituting π into the lemma. □

We also have the following one-step lemma:

Lemma A.2.6 (Lemma 3.3, [217])**.** *For any distribution $p^*, p \in \Delta(\mathcal{A})$, if $p'(\cdot) \propto p(\cdot)\cdot\exp(\alpha\cdot Q(x,\cdot))$, then*

$$\langle Q(x,\cdot), p^*(\cdot) - p(\cdot)\rangle \leq \alpha H^2/2 + \alpha^{-1}\cdot\Big(KL(p^*(\cdot)\|p(\cdot)) - KL(p^*(\cdot)\|p'(\cdot))\Big).$$

Given the above lemmas, we begin our proof of Theorem 4.4.2.

Proof of Theorem 4.4.2. Combining Lemma A.2.4 and Lemma A.2.5,

we have

$$\begin{aligned}
&\sum_{i=1}^{n}[V_{i,1}^{\pi^*}(x_1) - V_{i,1}^{\pi^i}(x_1)] \\
&\leq \sum_{i=1}^{n}\sum_{h=1}^{H}\mathbb{E}_{i,\pi^*}[\langle Q_{i,h}, \pi_h^* - \pi_{i,h}\rangle] + 2\sum_{i=1}^{n}\sum_{h=1}^{H}\mathbb{E}_{i,\pi^*}[\Gamma_{i,h}(x_h, a_h)] \\
&\leq \sum_{i=1}^{n}\sum_{h=1}^{H}\alpha H^2/2 + \alpha^{-1}\mathbb{E}_{i,\pi^*}[\mathrm{KL}(\pi_h^*(\cdot|x_h)\|\pi_{i,h}(\cdot|x_h)) - \mathrm{KL}(\pi_h^*(\cdot|x_h)\|\pi_{i+1,h}(\cdot|x_h))] \\
&\quad + 2\sum_{i=1}^{n}\sum_{h=1}^{H}\mathbb{E}_{i,\pi^*}[\Gamma_{i,h}(x_h, a_h)] \\
&\leq \alpha H^3 n/2 + \alpha^{-1}\cdot\sum_{h=1}^{H}\mathbb{E}_{i,\pi^*}[\mathrm{KL}(\pi_h^*(\cdot|x_h)\|\pi_{1,h}(\cdot|x_h))] + 2\sum_{i=1}^{n}\sum_{h=1}^{H}\mathbb{E}_{i,\pi^*}[\Gamma_{i,h}(x_h, a_h)] \\
&\leq \alpha H^3 n/2 + \alpha^{-1} H\log|A| + 2\sum_{i=1}^{n}\sum_{h=1}^{H}\mathbb{E}_{i,\pi^*}[\Gamma_{i,h}(x_h, a_h)],
\end{aligned}$$

where the last inequality holds since $\pi_{1,h}$ is the uniform distribution over $\mathcal{A}$. Then, selecting $\alpha = 1/\sqrt{H^2 n}$, we have

$$\sum_{i=1}^{n}[V_{i,1}^{\pi^*}(x_1) - V_{i,1}^{\pi^i}(x_1)] \leq 2\sqrt{n\log|A|H^2} + 2\sum_{i=1}^{n}\sum_{h=1}^{H}\mathbb{E}_{i,\pi^*}[\Gamma_{i,h}(s_h, a_h)],$$

which holds for the random selection of $\mathcal{D}$ with probability at least $1-\delta$. Meanwhile, note that each MDP M_i is drawn i.i.d. from C. Meanwhile, note that π_i only depends on MDP $M_1, ..., M_{i-1}$. Therefore, by the standard online-to-batch conversion, we have

$$\mathbb{P}\left(\frac{1}{n}\sum_{i=1}^{n}[V_{i,1}^{\pi^*}(x_1) - V_{i,1}^{\pi_i}(x_1)] + \left(\frac{1}{n}\sum_{i=1}^{n}\mathbb{E}_{c\sim C}V_{c,1}^{\pi_i}(x_1) - \mathbb{E}_{c\sim C}V_{c,1}^{\pi^*}(x_1)\right) \leq 2H\sqrt{\frac{2\log 1/\delta}{n}}\right) \geq 1-\delta,$$

which suggests that with probability at least $1-2\delta$,

$$\mathbb{E}_{c\sim C}V_{c,1}^{\pi^*}(x_1) - \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}_{c\sim C}V_{c,1}^{\pi_i}(x_1)$$
$$\leq 2\sqrt{\frac{\log|A|H^2}{n}} + \frac{2}{n}\sum_{i=1}^{n}\sum_{h=1}^{H}\mathbb{E}_{\pi^*}[\Gamma_{i,h}(x_h,a_h)] + 2\sqrt{\frac{2H\log 1/\delta}{n}}.$$

Therefore, by selecting $\pi^{\text{PPPO}} := \text{random}(\pi_1, ..., \pi_n)$ and applying the Markov inequality, setting $\delta = 1/8$, we have our bound holds. □

A.2.3 Suboptimality bounds for real-world setups

In this section we state and prove the suboptimality bounds we promised in Remarks 9 and 11, where we merge the sampled contexts into m groups (generally, $m << n$) to reduce the computational complexity in practical settings.

Assume $m|n$ and the n contexts from offline dataset are equally partitioned into m groups. We write the resulting average MDPs (see Proposition 4.1) for each group as $\bar{\mathcal{M}}_1, \ldots, \bar{\mathcal{M}}_m$. For each $\bar{\mathcal{M}}_j$, we regard it as an individual context in the sense of (A.70) and denote the resulting uncertainty quantifier and value function as ${\Gamma'}_{j,h}, {V'}_{j,h}^{\pi}$.

Theorem A.2.7 (Suboptimality bound for Remark 9)**.** *Assume the same setting as Theorem 4.4.1 with the original n contexts grouped as m contexts, and denote the resulting algorithm as PERM-m V. Then w.p. at least $1-\delta$, the output π' of PERM-m V*

satisfies

$$\text{SubOpt}(\pi') \leq \underbrace{2\sqrt{\frac{2\log(6\mathcal{N}^{\Pi}_{(Hm)^{-1}}/\delta)}{n}}}_{I_1:\textit{Supervised learning (SL) error}} + \underbrace{\frac{2}{m}\sum_{j=1}^{m}\sum_{h=1}^{H}\mathbb{E}\pi^*, \bar{\mathcal{M}}_j {\Gamma'}_{j,h}(s_h,a_h)|s_1=x_1}_{I_2:\textit{Reinforcement learning (RL) error}}$$
$$+ \underbrace{\frac{5}{m} + 2\sup_{\pi}\left|\frac{1}{n}\sum_{i=1}^{n}V^{\pi}_{i,1}(x_1) - \frac{1}{m}\sum_{j=1}^{m}{V'}^{\pi}_{j,1}(x_1)\right|}_{\textit{Additional approximation error}},$$

where $\mathbb{E}_{j,\pi^*}$ is w.r.t. the trajectory induced by π^* with the transition $\bar{\mathcal{P}}_j$ in the underlying average MDP $\bar{\mathcal{M}}_j$.

Proof of Theorem A.2.7. Similar to the proof of Theorem 4.4.1, we decompose the suboptimality gap as follows

$$\begin{aligned}
&\text{SubOpt}(\pi') \\
&= \mathbb{E}_{c\sim C}V^{\pi^*}_{c,1}(x_1) - V^{\pi'}_{c,1}(x_1) \\
&= \mathbb{E}_{c\sim C}V^{\pi^*}_{c,1}(x_1) - \frac{1}{n}\sum_{i=1}^{n}V^{\pi^*}_{i,1}(x_1) + \frac{1}{n}\sum_{i=1}^{n}V^{\pi'}_{i,1}(x_1) - \mathbb{E}_{c\sim C}V^{\pi'}_{c,1}(x_1) \\
&\quad + \frac{1}{n}\sum_{i=1}^{n}V^{\pi^*}_{i,1}(x_1) - \frac{1}{m}\sum_{j=1}^{m}{V'}^{\pi^*}_{j,1}(x_1) + \frac{1}{m}\sum_{j=1}^{m}{V'}^{\pi'}_{j,1}(x_1) - \frac{1}{n}\sum_{i=1}^{n}V^{\pi'}_{i,1}(x_1) \\
&\quad + \frac{1}{m}\sum_{j=1}^{m}\left({V'}^{\pi^*}_{j,1}(x_1) - {V'}^{\pi'}_{j,1}(x_1)\right). \qquad \text{(A.84)}
\end{aligned}$$

Note that we can bound the first and third lines of (A.84) with the exactly same arguments as the proof of Theorem 4.4.1, the only notation-wise difference is that the uncertainty quantifier becomes Γ' as we are operating on the level of average MDP $\bar{\mathcal{M}}_j$.

The only thing left is to bound the second line of (A.84). This is the same in spirit of the bound (A.67), so that we can express

the bound as follows

$$\frac{1}{n}\sum_{i=1}^{n} V_{i,1}^{\pi^*}(x_1) - \frac{1}{m}\sum_{j=1}^{m} V'^{\pi^*}_{j,1}(x_1) + \frac{1}{m}\sum_{j=1}^{m} V'^{\pi'}_{j,1}(x_1) - \frac{1}{n}\sum_{i=1}^{n} V_{i,1}^{\pi'}(x_1)$$
$$\leq 2\sup_{\pi}\left|\frac{1}{n}\sum_{i=1}^{n} V_{i,1}^{\pi}(x_1) - \frac{1}{m}\sum_{j=1}^{m} V'^{\pi}_{j,1}(x_1)\right|.$$

To conclude, our final bound can be expressed as: with ϵ set as $\frac{1}{mH}$, we can get w.p. at least $1-\delta$

$$\text{SubOpt}(\pi')$$
$$\leq 2\sqrt{\frac{2\log(6\mathcal{N}^{\Pi}_{(Hm)^{-1}}/\delta)}{n}} + \frac{2}{m}\sum_{j=1}^{m}\sum_{h=1}^{H}\mathbb{E}\pi^*, \bar{\mathcal{M}}_j \Gamma'_{j,h}(s_h, a_h)|s_1 = x_1$$
$$+ \frac{5}{m} + 2\sup_{\pi}\left|\frac{1}{n}\sum_{i=1}^{n} V_{i,1}^{\pi}(x_1) - \frac{1}{m}\sum_{j=1}^{m} V'^{\pi}_{j,1}(x_1)\right|.$$

□

To prove the suboptimality bound for Remark 11, we denote that the policies produced by PPPO after merging dataset to m groups to be $\pi_1, \ldots, \pi_m$, and the original PPPO algorithm would produce the policies as $\pi'_1, \ldots, \pi'_n$. We assume that the merging of dataset from n to m groups is only to combine the consecutive n/m terms from $\pi'_1, \ldots, \pi'_n$ and preserves the order.

Theorem A.2.8 (Suboptimality bound for Remark 11)**.** *Assume the same setting as Theorem 4.4.2 with the original n contexts grouped as m contexts, and denote the resulting algorithm as PPPO-mV. Let $\Gamma'_{j,h}$ be the uncertainty quantifier returned by $\mathbb{O}$ through the PPPO-mV algorithm. Selecting $\alpha = 1/\sqrt{H^2 m}$. Then*

selecting $\delta = 1/8$, w.p. at least $2/3$, we have

$$\begin{aligned}
SubOpt(\pi^{PPPO-mV}) \leq 10\Big(& \underbrace{\sqrt{\frac{\log|\mathcal{A}|H^2}{m}}}_{I_1:SL\ error} + \underbrace{\frac{1}{m}\sum_{j=1}^{m}\sum_{h=1}^{H}\mathbb{E}j, \pi^*\Gamma'_{j,h}(s_h,a_h)|s_1 = x_1}_{I_2:RL\ error} \\
& + \sup_{\pi}\left|\frac{1}{n}\sum_{i=1}^{n}V_{i,1}^{\pi}(x_1) - \frac{1}{m}\sum_{j=1}^{m}V'^{\pi}_{j,1}(x_1)\right| + \frac{1}{n}\sum_{i=1}^{n}\sup_{\pi}\left|\mathbb{E}_c[V_{c,1}^{\pi}(x_1)] - V_{i,1}^{\pi}(x_1)\right| \\
& + \frac{1}{m}\sum_{j=1}^{m}\sup_{\pi}\left|\mathbb{E}_c[V'^{\pi}_{c,1}(x_1)] - V'^{\pi}_{j,1}(x_1)\right|\Big).
\end{aligned}$$

where $\mathbb{E}_{j,\pi^*}$ is w.r.t. the trajectory induced by π^* with the transition $\bar{\mathcal{P}}_j$ in the underlying MDP $\bar{\mathcal{M}}_j$.

Proof of Theorem A.2.8. Using the same arguments as in the proof of Theorem 4.4.2 with $\alpha = 1/\sqrt{H^2 m}$, we can derive the bound

$$\sum_{j=1}^{m}[V'^{\pi^*}_{j,1}(x_1) - V'^{\pi_j}_{j,1}(x_1)] \leq 2\sqrt{m\log|A|H^2} + 2\sum_{j=1}^{m}\sum_{h=1}^{H}\mathbb{E}_{j,\pi^*}[\Gamma'_{j,h}(s_h,a_h)].$$

Leveraging this bound and online-to-batch, we obtain the following estimation

$$\begin{aligned}
& \mathbb{E}_c[V_{c,1}^{\pi^*}(x_1)] - \frac{1}{m}\sum_{j=1}^{m}\mathbb{E}_c[V_{c,1}^{\pi_j}(x_1)] \\
=& \mathbb{E}_c[V_{c,1}^{\pi^*}(x_1)] - \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}_c[V_{c,1}^{\pi'_i}(x_1)] + \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}_c[V_{c,1}^{\pi'_i}(x_1)] - \frac{1}{m}\sum_{j=1}^{m}\mathbb{E}_c[V_{c,1}^{\pi_j}(x_1)] \\
\leq & 2H\sqrt{\frac{2\log 1/\delta}{n}} + \frac{1}{n}\sum_{i=1}^{n}\Big(\mathbb{E}_c[V_{c,1}^{\pi'_i}(x_1)] - V_{i,1}^{\pi'_i}(x_1)\Big) \\
& + \frac{1}{n}\sum_{i=1}^{n}V_{i,1}^{\pi^*}(x_1) - \frac{1}{m}\sum_{j=1}^{m}\mathbb{E}_c[V_{c,1}^{\pi_j}(x_1)]
\end{aligned}$$

$$
\begin{aligned}
=&2H\sqrt{\frac{2\log 1/\delta}{n}}+\frac{1}{n}\sum_{i=1}^{n}V_{i,1}^{\pi^*}(x_1)-\frac{1}{m}\sum_{j=1}^{m}{V'}_{j,1}^{\pi^*}(x_1)\\
&+\frac{1}{m}\sum_{j=1}^{m}{V'}_{j,1}^{\pi^*}(x_1)-\frac{1}{m}\sum_{j=1}^{m}{V'}_{j,1}^{\pi_j}(x_1)\\
&+\frac{1}{n}\sum_{i=1}^{n}\Big(\mathbb{E}_c[V_{c,1}^{\pi_i'}(x_1)]-V_{i,1}^{\pi_i'}(x_1)\Big)+\frac{1}{m}\sum_{j=1}^{m}{V'}_{j,1}^{\pi_j}(x_1)-\frac{1}{m}\sum_{j=1}^{m}\mathbb{E}_c[V_{c,1}^{\pi_j}(x_1)]\\
\le&2H\sqrt{\frac{2\log 1/\delta}{n}}+\sup_{\pi}\left|\frac{1}{n}\sum_{i=1}^{n}V_{i,1}^{\pi}(x_1)-\frac{1}{m}\sum_{j=1}^{m}{V'}_{j,1}^{\pi}(x_1)\right|\\
&+2\sqrt{\frac{\log|A|H^2}{m}}+\frac{2}{m}\sum_{j=1}^{m}\sum_{h=1}^{H}\mathbb{E}_{j,\pi^*}[{\Gamma'}_{j,h}(s_h,a_h)]\\
&+\frac{1}{n}\sum_{i=1}^{n}\sup_{\pi}\left|\mathbb{E}_c[V_{c,1}^{\pi}(x_1)]-V_{i,1}^{\pi}(x_1)\right|+\frac{1}{m}\sum_{j=1}^{m}\sup_{\pi}\left|\mathbb{E}_c[{V'}_{c,1}^{\pi}(x_1)]-{V'}_{j,1}^{\pi}(x_1)\right|.
\end{aligned}
$$

Finally we apply Markov inequality and take $\delta = 1/8$ as in the proof of Theorem 4.4.2. □

A.2.4 Results in Section 4.5

A.2.4.1 Proof of Theorem 4.5.1

By [91], the parameters specified as $\lambda = 1, \quad \beta(\delta) = c{\cdot}dH\sqrt{\log(2dHK/\delta)}$, and applying union bound, we can get: for Algo.6, with probability at least $1-\delta/3$

$$
\left|(\hat{\mathbb{B}}_{i,h}\hat{V}_{i,h+1}^{\pi})(x,a)-(\mathbb{B}_{i,h}\hat{V}_{i,h+1}^{\pi})(x,a)\right|\le\beta\big(\frac{\delta}{3nH\mathcal{N}_{(Hn)^{-1}}^{\Pi}}\big)\big(\phi(x,a)^{\top}\Lambda_{i,h}^{-1}\phi(x,a)\big)^{1/2},
$$

$$
\text{for all } i\in[n],\pi\in\tilde{\Pi},(x,a)\in\mathcal{S}\times\mathcal{A},h\in[H], \tag{A.85}
$$

where $\tilde{\Pi}$ is the $\frac{1}{Hn}$-covering set of the policy space Π w.r.t. distance $\mathrm{d}(\pi^1,\pi^2)=\max_{s\in\mathcal{S},h\in[H]}\|\pi_h^1(\cdot|s)-\pi_h^2(\cdot|s)\|_1$.

Therefore, we can specify the $\Gamma_{i,h}(\cdot,\cdot)$ in Theorem 4.4.1 with $\beta\big(\frac{\delta}{3nH\mathcal{N}^{\Pi}_{(Hn)^{-1}}}\big)\big(\phi(x,a)^\top\Lambda_{i,h}^{-1}\phi(x,a)\big)^{1/2}$, and follow the same process as the proof of Theorem 4.4.1 to get the result for Algo.4 with subroutine Algo.6.

Similarly, we can get: we can get: for Algo.6, with probability at least $1-1/4$

$$\big|(\hat{\mathbb{B}}_{i,h}\hat{V}_{i,h+1})(x,a)-(\mathbb{B}_{i,h}\hat{V}_{i,h+1})(x,a)\big|\le\beta\big(\frac{\delta}{4nH}\big)\big(\phi(x,a)^\top\Lambda_{i,h}^{-1}\phi(x,a)\big)^{1/2},$$
$$\text{for all } i\in[n],(x,a)\in\mathcal{S}\times\mathcal{A},h\in[H]\,. \tag{A.86}$$

Therefore, we can specify the $\Gamma_{i,h}(\cdot,\cdot)$ in Theorem 4.4.2 with $\beta\big(\frac{\delta}{4nH}\big)\big(\phi(x,a)^\top\Lambda_{i,h}^{-1}\phi(x,a)\big)^{1/2}$ and follow the same process as the proof of Theorem 4.4.2 to get the result for Algo.5 with subroutine Algo.6.

A.2.4.2 Proof of Corollary 4.1

By the assumption that $\mathcal{D}_i$ is generated by behavior policy $\bar{\pi}_i$ which well-explores MDP $\mathcal{M}_i$ with constant c_i (where the well-explore is defined in Def.4.3), the proof of Corollary 4.6 in [91], and applying a union bound over n contexts, we have that for Algo.4 with subroutine Algo.6 w.p. at least $1-\delta/2$

$$\|\phi(x,a)\|_{\Lambda_{i,h}^{-1}}\le\sqrt{\frac{2d}{c_iK}}$$
$$\text{for all } i\in[n],\ (x,a)\in\mathcal{S}\times\mathcal{A} \text{ and all } h\in[H]\,, \tag{A.87}$$

and for Algo.4 with subroutine Algo.6 w.p. at least $1-\delta/2$

$$\|\phi(x,a)\|_{\Lambda_{i,h}^{-1}}\le\sqrt{\frac{2dH}{c_iK}}$$
$$\text{for all } i\in[n],\ (x,a)\in\mathcal{S}\times\mathcal{A} \text{ and all } h\in[H]\,, \tag{A.88}$$

because we use the data splitting technique and we only utilize each trajectory once for one data tuple at some stage h, so we replace K with K/H.

Then, the result follows by plugging the results above into Theorem 4.5.1.

A.3 Appendix for Chapter 5

A.3.1 More Discussions on Related Work

In this section, we will give more comparisons and discussions on some previous works that are related to our work to some extent.

There are some other works on bandits leveraging user (or task) relations, which have some relations with the clustering of bandits (CB) works to some extent, but are in different lines of research from CB, and are quite different from our work. First, besides CB, the work [179] also leverages user relations. Specifically, it utilizes a *known* user adjacency graph to share context and payoffs among neighbors, whereas in CB, the user relations are *unknown* and need to be learnt, thus the setting differs a lot from CB. Second, there are lines of works on multi-task learning [280, 281, 282, 283, 284, 285], meta-learning [286, 287, 288] and federated learning [289, 290], where multiple different tasks are solved jointly and share information. Note that all of these works do not assume an underlying *unknown* user clustering structure which needs to be inferred by the agent to speed up learning. For works on multi-task learning [280, 281, 282, 283, 284, 285], they assume the tasks are related but no user clustering structures, and to the best of our knowledge, none of them consider model misspefications, thus differing a lot from ours. For some recent works on meta-learning [286, 287, 291], they propose general Bayesian hierarchical models to share knowledge across tasks, and design Thompson-Sampling-based algorithms to optimize the Bayes regret, which are quite different from the line of CB works, and differ a lot from ours. And additionally, as supported by the

discussions in the works [288, 285], multi-task learning and meta-learning are different lines of research from CB. For the works on federated learning [289, 290], they consider the privacy and communication costs among multiple servers, whose setting is also very different from the previous CB works and our work.

Remark. Again, we emphasize that the goal of this work is to initialize the study of the important CBMUM problem, and propose general design ideas for dealing with model misspecifications in CB problems. Therefore, our study is based on fundamental models on CB [46, 136] and MLB [138], and the algorithm design ideas and theoretical analysis are pretty general. We leave incorporating the more recent model selection methods [140, 141] into our framework to address the unknown exact maximum model misspecification level as an interesting future work. It would also be interesting to consider incorporating our methods and ideas of tackling model misspecifications into the studies of multi-task learning, meta learning and federated learning.

A.3.2 More Discussions on Assumptions

All the assumptions (Assumptions 9.2,9.3,9.4,9.1)in this work are natural and basically follow (or less strigent than) previous works on CB and MLB [46, 135, 136, 137, 138].

A.3.2.1 Less Strigent Assumption on on the Generating Distribution of Arm Vectors

We also make some contributions to relax a widely-used but stringent assumption on the generating distribution of arm vectors.

Specifically, our Assumption 9.4 on item regularity relaxes the previous one used in previous CB works [46, 135, 136, 137] by removing the condition that the variance should be upper bounded by $\frac{\lambda^2}{8\log(4|\mathcal{A}_t|)}$. For technical details on this, please refer to the theoretical analysis and discussions in Appendix A.3.10.

A.3.2.2 Discussions on Assumption 9.1 about Bounded Misspecification Level

This assumption follows [138]. Note that this ϵ_* can be an upper bound on the maximum misspecification level, not the exact maximum itself. In real-world applications, the deviations are usually small [48], and we can set a relatively big ϵ_* (e.g., 0.2) to be the upper bound. Our experimental results support this claim. As shown in our experimental results on real-data case 2, even when ϵ_* is unknown, our algorithms still perform well by setting $\epsilon_* = 0.2$. Some recent studies [140, 141] use model selection methods to theoretically deal with unknown exact maximum misspecification level in the single-user case, which is not the emphasis of this work. Additionally, the work [141] assumes that the learning agent has access to a regression oracle. And for the work [140], though their regret bound is dependent on the exact maximum misspecification level that needs not to be known by the agent, an upper bound of the exact maximum misspecification level is still needed. We leave incorporating their methods to deal with unknown exact maximum misspecification level as an interesting future work.

A.3.2.3 Discussions on Assumption 9.3 about the Theoretical Results under General User Arrival Distributions

The uniform arrival in Assumption 9.3 follows previous CB works [46, 135, 137], it only affects the T_0 term, which is the time after which the algorithm maintains a "good partition" and is of $O(u \log T)$. For an arbitrary arrival distribution, T_0 becomes $O(1/p_{min} \log T)$, where p_{min} is the minimal arrival probability of a user. And since it is a lower-order term (of $O(\log T)$), it will not affect the main order of our regret upper bound which is of $O(\epsilon_* T\sqrt{md \log T} + d\sqrt{mT} \log T)$. The work [136] studies arbitrary arrivals and aims to remove the $1/p_{min}$ factor in this term, but their setting is different. They make an additional assumption that users in the same cluster not only have the same preference vector, but also the same arrival probability, which is different from our setting and other classic CB works [46, 135, 137] where we only assume users in the same cluster share the same preference vector.

A.3.3 Highlight of the Theoretical Analysis

Our proof flow and methodologies are novel in clustering of bandits (CB), which are expected to inspire future works on model misspecifications and CB. The main challenge of the regret analysis in CBMUM is that due to the estimation inaccuracy caused by misspecifications, it is impossible to cluster all users exactly correctly, and it is highly non-trivial to bound the regret caused by **"misclustering" ζ-close users**.

To the best of our knowledge, the common proof flow of pre-

vious CB works (e.g., [46, 135, 137]) can be summarized in two steps: The first is to prove a sufficient time T_0' after which the algorithms can cluster all users **exactly correctly** with high probability. Note that the inferred clustering structure remains static after T_0', making the analysis easy. Second, after the **correct static clustering**, the regret can be trivially bounded by bounding m (number of underlying clusters) independent linear bandit algorithms, resulting in a $O(d\sqrt{mT}\log T)$ regret.

The above common proof flow is straightforward in CB with perfectly linear models, but it would fail to get a non-vacuous regret bound for CBMUM. In CBMUM, it is impossible to learn an exactly correct static clustering structure with model misspecifications. In particular, we prove that we can only expect the algorithm to cluster ζ-close users together rather than cluster all users exactly correctly. Therefore, the previous flow can not be applied to the more challenging CBMUM problem.

We do the following to address the challenges in obtaining a tight regret bound for CBMUM. With the carefully-designed novel key components of RCLUMB, we can prove a sufficient time T_0 after which RCLUMB can get a "good partition" (Definition 5.4) with high probability, which means the cluster $\overline{V_t}$ assigned to i_t contains all users in the same ground-truth cluster as i_t, and possibly some other i_t's ζ-close users. Intuitively, after T_0, the algorithm can leverage all the information from the users' ground-truth clusters but may misuse some information from other ζ-close users with preference gaps up to ζ, causing a regret of **"misclustering" ζ-close users**. It is highly non-trivial to bound this part of regret, and the proof methods would

be beneficial for future studies in CB in challenging cases when it is impossible to cluster all users exactly correctly. For details, please refer to the discussions "(ii) Bounding the term of misclustering it's ζ-close users" in Section 5.4, the key Lemma 5.4.4 (Bound of error caused by misclustering), its proof and tightness discussion in Appendix A.3.7. Also, a more subtle analysis is needed to handle the time-varying inferred clustering structure since the "good partition" may change over time, whereas in the previous CB works, the clustering structure remains static after T_0'. For theoretical details on this, please refer to Appendix A.3.5.

A.3.4 Discussions on why Trivially Combining Existing CB and MLB Works Could Not Achieve a Non-vacuous Regret Upper Bound

We consider discussing regret upper bounds for CB without considering misspecifications for three cases: (1) neither the clustering process nor the decision process considers misspecifications (previous CB algorithms); (2) the decision process does not consider misspecifications; (3) the clustering process does not consider misspecifications.

For cases (1) and (2), the decision process could contribute to the leading regret. We consider the case where there are m underlying clusters, with each cluster's arrival being T/m, and the agent knows the underlying clustering structure. For this case, there exist some instances where the regret upper bound $R(T)$ is strictly larger than $\epsilon_* T\sqrt{m \log T}$ asymptotically in T. Formally, in the discussion of "Failure of unmodified algorithm"

in Appendix E in [138], they give an example to show that in the single-user case, the regret $R_1(T)$ of the classic linear bandit algorithms without considering misspecifications will have: $\lim_{T\to+\infty} \frac{R_1(T)}{\epsilon_* T\sqrt{m\log T}} = +\infty$. In our problem with multiple users and m underlying clusters, even if we know the underlying clustering structure and keep m independent linear bandit algorithms with T_i for the cluster $i \in [m]$ to leverage the common information of clusters, the best we can get is $R_2(T) = \sum_{i\in[m]} R_1(T_i)$. By the above results, if the decision process does not consider misspecifications, we have $\lim_{T\to+\infty} \frac{R_2(T)}{\epsilon_* T\sqrt{m\log T}} = \lim_{T\to+\infty} \frac{mR_1(T/m)}{\epsilon_* T\sqrt{m\log T}} = +\infty$. Recall that the regret upper bound $R(T)$ of our proposed algorithms is of $O(\epsilon_* T\sqrt{md\log T} + d\sqrt{mT}\log T)$ (thus, we have $\lim_{T\to+\infty} \frac{R(T)}{\epsilon_* T\sqrt{m\log T}} < +\infty$), which gives a proof that that the regret upper bound of our proposed algorithms is asymptotically much better than CB algorithms in cases (1)(2).

For case (3), if the clustering process does not use the more tolerant deletion rule in Line 10 of Algo.7, the gap between users linked by edges would possibly exceed ζ ($\zeta = 2\epsilon_*\sqrt{\frac{2}{\tilde{\lambda}_x}}$) even after T_0, which will result in a regret upper bound no better than $O(\epsilon_* u\sqrt{d}T)$. As the number of users u is usually huge in practice, this result is vacuous. The reasons for getting the above claim are as follows. Even if the clustering process further uses our deletion rule considering misspecifications, and the users linked by edges are within ζ distance, failing to extract 1-hop users (Line 5 in Algo.7) would cause the leading $O(\epsilon_* u\sqrt{d}T)$ regret term, as in the worst case, the preference vector θ of the user in $\tilde{V}_t$ who is h-hop away from user i_t could deviate by $h\zeta$ from θ_{i_t}, where h can

be as large as u, and it would make the second term in Eq.(5.8) a $O(\epsilon_* u\sqrt{d}T)$ term. If we completely do not consider the misspecifications in the clustering process, the above user gap between users linked by edges would possibly exceed ζ, which will cause a regret upper bound worse than $O(\epsilon_* u\sqrt{d}T)$.

A.3.5 Proof of Theorem 6.4.3

We first prove the result in the case when γ_1 defined in Definition 5.2 is not infinity, i.e., $4\epsilon_*\sqrt{\frac{2}{\tilde{\lambda}_x}} < \gamma_1 < \infty$. The proof of the special case when $\gamma_1 = \infty$ will directly follow the proof of this case.

For the instantaneous regret R_t at round t, with probability at least $1-5\delta$ for some $\delta \in (0, \frac{1}{5})$, at $\forall t \geq T_0$:

$$\begin{aligned}
R_t &= (\boldsymbol{x}_{a_t^*}^\top \boldsymbol{\theta}_{i_t} + \boldsymbol{\epsilon}_{a_t^*}^{i_t,t}) - (\boldsymbol{x}_{a_t}^\top \boldsymbol{\theta}_{i_t} + \boldsymbol{\epsilon}_{a_t}^{i_t,t}) \\
&= \boldsymbol{x}_{a_t^*}^\top (\boldsymbol{\theta}_{i_t} - \hat{\boldsymbol{\theta}}_{\overline{V}_t,t-1}) + (\boldsymbol{x}_{a_t^*}^\top \hat{\boldsymbol{\theta}}_{\overline{V}_t,t-1} + C_{a_t^*,t}) - (\boldsymbol{x}_{a_t}^\top \hat{\boldsymbol{\theta}}_{\overline{V}_t,t-1} + C_{a_t,t}) \\
&\quad + \boldsymbol{x}_{a_t}^\top (\hat{\boldsymbol{\theta}}_{\overline{V}_t,t-1} - \boldsymbol{\theta}_{i_t}) + C_{a_t,t} - C_{a_t^*,t} + (\boldsymbol{\epsilon}_{a_t^*}^{i_t,t} - \boldsymbol{\epsilon}_{a_t}^{i_t,t}) \\
&\leq 2C_{a_t,t} + \frac{2\epsilon_*\sqrt{2d}}{\tilde{\lambda}_x^{\frac{3}{2}}} \mathbb{I}\{\overline{V}_t \notin \mathcal{V}\} + 2\epsilon_* \,,
\end{aligned} \tag{A.89}$$

where the last inequality holds by the UCB arm selection strategy in Eq.(6.3), the concentration bound given in Lemma 6.4.2, and the fact that $\left\|\boldsymbol{\epsilon}^{i,t}\right\|_\infty \leq \epsilon_*, \forall i \in \mathcal{U}, \forall t$.

We define the following events. Let

$$\mathcal{E}_0 = \{R_t \leq 2C_{a_t,t} + \frac{2\epsilon_*\sqrt{2d}}{\tilde{\lambda}_x^{\frac{3}{2}}} \mathbb{I}\{\overline{V}_t \notin \mathcal{V}\} + 2\epsilon_*,$$

for all $\{t : t \geq T_0, \text{and the algorithm maintains a "good partition" at } t\}\}$,

$$\mathcal{E}_1 = \{\text{the algorithm maintains a "good partition" for all } t \geq T_0\}\,,$$

$$\mathcal{E} = \mathcal{E}_0 \cap \mathcal{E}_1 \,.$$

$\mathbb{P}(\mathcal{E}_0) \geq 1 - 2\delta$. According to Lemma A.3.2, $\mathbb{P}(\mathcal{E}_1) \geq 1 - 3\delta$. Thus, $\mathbb{P}(\mathcal{E}) \geq 1 - 5\delta$ for some $\delta \in (0, \frac{1}{5})$. Take $\delta = \frac{1}{T}$, we can get that

$$\begin{aligned}\mathbb{E}[R(T)] &= \mathbb{P}(\mathcal{E})\mathbb{I}\{\mathcal{E}\}R(T) + \mathbb{P}(\overline{\mathcal{E}})\mathbb{I}\{\overline{\mathcal{E}}\}R(T) \\ &\leq \mathbb{I}\{\mathcal{E}\}R(T) + 5 \times \frac{1}{T} \times T \qquad \text{(A.90)} \\ &= \mathbb{I}\{\mathcal{E}\}R(T) + 5\,,\end{aligned}$$

where $\overline{\mathcal{E}}$ denotes the complementary event of $\mathcal{E}$, $\mathbb{I}\{\mathcal{E}\}R(T)$ denotes $R(T)$ under event $\mathcal{E}$, $\mathbb{I}\{\overline{\mathcal{E}}\}R(T)$ denotes $R(T)$ under event $\overline{\mathcal{E}}$, and we use $R(T) \leq T$ to bound $R(T)$ under event $\overline{\mathcal{E}}$.

Then it remains to bound $\mathbb{I}\{\mathcal{E}\}R(T)$:

$$\begin{aligned}\mathbb{I}\{\mathcal{E}\}R(T) &\leq R(T_0) + \mathbb{E}[\mathbb{I}\{\mathcal{E}\} \sum_{t=T_0+1}^{T} R_t] \\ &\leq T_0 + 2\mathbb{E}[\mathbb{I}\{\mathcal{E}\} \sum_{t=T_0+1}^{T} C_{a_t,t}] + \frac{2\epsilon_* \sqrt{2d}}{\tilde{\lambda}_x^{\frac{3}{2}}} \sum_{t=T_0+1}^{T} \mathbb{E}[\mathbb{I}\{\mathcal{E}, \overline{V}_t \notin \mathcal{V}\}] + 2\epsilon_* T \qquad \text{(A.91)} \\ &= T_0 + 2\mathbb{E}[\mathbb{I}\{\mathcal{E}\} \sum_{t=T_0+1}^{T} C_{a_t,t}] + \frac{2\epsilon_* \sqrt{2d}}{\tilde{\lambda}_x^{\frac{3}{2}}} \sum_{t=T_0+1}^{T} \mathbb{P}(\mathbb{I}\{\mathcal{E}, \overline{V}_t \notin \mathcal{V}\}) + 2\epsilon_* T \\ &\leq T_0 + 2\mathbb{E}[\mathbb{I}\{\mathcal{E}\} \sum_{t=T_0+1}^{T} C_{a_t,t}] + \frac{2\epsilon_* \sqrt{2d}}{\tilde{\lambda}_x^{\frac{3}{2}}} \times \frac{\tilde{u}}{u} T + 2\epsilon_* T \,, \qquad \text{(A.92)}\end{aligned}$$

where Eq.(A.91) follows from Eq.(A.195). Eq.(A.92) holds since under Assumption 9.3 about user arrival uniformness and by Definition 5.4 of "good partition", $\mathbb{P}(\mathbb{I}\{\mathcal{E}, \overline{V}_t \notin \mathcal{V}\}) \leq \frac{\tilde{u}}{u}, \forall t \geq T_0$, where $\tilde{u}$ is defined in Definition 5.3.

Then we need to bound $\mathbb{E}[\mathbb{I}\{\mathcal{E}\}\sum_{t=T_0+1}^{T} C_{a_t,t}]$:

$$\mathbb{I}\{\mathcal{E}\}\sum_{t=T_0+1}^{T} C_{a_t,t} = \Big(\sqrt{\lambda}+\sqrt{2\log(\frac{1}{\delta})+d\log(1+\frac{T}{\lambda d})}\Big)\mathbb{I}\{\mathcal{E}\}\sum_{t=T_0+1}^{T}\|\boldsymbol{x}_{a_t}\|_{\overline{\boldsymbol{M}}^{-1}_{\overline{V}_t,t-1}} + \mathbb{I}\{\mathcal{E}\}\epsilon_*\sum_{t=T_0+1}^{T}\sum_{\substack{s\in[t-1]\\ i_s\in\overline{V}_t}}\left|\boldsymbol{x}_{a_t}^\top\overline{\boldsymbol{M}}^{-1}_{\overline{V}_t,t-1}\boldsymbol{x}_{a_s}\right| . \tag{A.93}$$

Next, we bound the $\mathbb{I}\{\mathcal{E}\}\sum_{t=T_0+1}^{T}\|\boldsymbol{x}_{a_t}\|_{\overline{\boldsymbol{M}}^{-1}_{\overline{V}_t,t-1}}$ term in Eq.(A.93):

$$\begin{aligned}
&\mathbb{I}\{\mathcal{E}\}\sum_{t=T_0+1}^{T}\|\boldsymbol{x}_{a_t}\|_{\overline{\boldsymbol{M}}^{-1}_{\overline{V}_t,t-1}}\\
&=\mathbb{I}\{\mathcal{E}\}\sum_{t=T_0+1}^{T}\sum_{k=1}^{m_t}\mathbb{I}\{i_t\in\tilde{V}'_{t,k}\}\|\boldsymbol{x}_{a_t}\|_{\overline{\boldsymbol{M}}^{-1}_{\overline{V}'_{t,k},t-1}}\\
&\le\mathbb{I}\{\mathcal{E}\}\sum_{t=T_0+1}^{T}\sum_{j=1}^{m}\mathbb{I}\{i_t\in V_j\}\|\boldsymbol{x}_{a_t}\|_{\overline{\boldsymbol{M}}^{-1}_{V_j,t-1}} &\text{(A.94)}\\
&\le\mathbb{I}\{\mathcal{E}\}\sum_{j=1}^{m}\sqrt{\sum_{t=T_0+1}^{T}\mathbb{I}\{i_t\in V_j\}\sum_{t=T_0+1}^{T}\mathbb{I}\{i_t\in V_j\}\|\boldsymbol{x}_{a_t}\|^2_{\overline{\boldsymbol{M}}^{-1}_{V_j,t-1}}} &\text{(A.95)}\\
&\le\mathbb{I}\{\mathcal{E}\}\sum_{j=1}^{m}\sqrt{2T_{V_j,T}d\log(1+\frac{T}{\lambda d})} &\text{(A.96)}\\
&\le\mathbb{I}\{\mathcal{E}\}\sqrt{2\sum_{j=1}^{m}1\sum_{j=1}^{m}T_{V_j,T}d\log(1+\frac{T}{\lambda d})}\\
&=\mathbb{I}\{\mathcal{E}\}\sqrt{2mdT\log(1+\frac{T}{\lambda d})}\,, &\text{(A.97)}
\end{aligned}$$

where we use m_t to denote the number of connected components partitioned by the algorithm at t, $\tilde{V}'_{t,k}, k\in[m_t]$ to denote the connected components partitioned by the algorithm at

t, $\overline{V}'_{t,k} \subseteq \tilde{V}'_{t,k}$ to denote the subset extracted to be the cluster $\overline{V}_t$ for i_t from $\tilde{V}'_{t,k}$ conditioned on $i_t \in \tilde{V}'_{t,k}$, and $T_{V_j,T}$ to denote the number of times that the served users lie in the *ground-truth cluster* V_j up to time T, i.e., $T_{V_j,T} = \sum_{t\in[T]} \mathbb{I}\{i_t \in V_j\}$.

The reasons for having Eq.(A.94) are as follows. Under event $\mathcal{E}$, the algorithm will always have a "good partition" after T_0. By Definition 5.4 and the proof process of Lemma A.3.2 about the edge deletion conditions, we can get $m_t \leq m$ and if $i_t \in \tilde{V}'_{t,k}, i_t \in V_j$, then $V_j \subseteq \overline{V}'_{t,k}$ since $\overline{V}'_{t,k}$ contains V_j and possibly other *ground-truth clusters* $V_n, n \in [m]$, whose preference vectors are ζ-close to $\boldsymbol{\theta}^j$. Therefore, by the definition of the regularized Gramian matrix, we can get $M_{\overline{V}'_{t,k},t-1} \succeq M_{V_j,t-1}, \forall t \geq T_0+1$. Thus by the above reasoning, $\sum_{k=1}^{m_t} \mathbb{I}\{i_t \in \tilde{V}'_{t,k}\} \|\boldsymbol{x}_{a_t}\|_{\overline{\boldsymbol{M}}^{-1}_{\overline{V}'_{t,k},t-1}} \leq \sum_{j=1}^{m} \mathbb{I}\{i_t \in V_j\} \|\boldsymbol{x}_{a_t}\|_{\overline{\boldsymbol{M}}^{-1}_{V_j,t-1}}, \forall t \geq T_0+1$. Eq.(A.223) holds by the Cauchy–Schwarz inequality; Eq.(A.96) follows by the following technical Lemma A.4.2. Eq.(A.97) is from the Cauchy–Schwarz inequality and the fact that $\sum_{j=1}^{m} T_{V_j,T} = T$.

We then bound the last term in Eq.(A.93):

$$\mathbb{I}\{\mathcal{E}\}\epsilon_* \sum_{t=T_0+1}^{T} \sum_{\substack{s\in[t-1]\\ i_s\in\overline{V}_t}} \left|\boldsymbol{x}_{a_t}^\top \overline{\boldsymbol{M}}^{-1}_{\overline{V}_t,t-1} \boldsymbol{x}_{a_s}\right|$$

$$= \mathbb{I}\{\mathcal{E}\}\epsilon_* \sum_{t=T_0+1}^{T} \sum_{k=1}^{m_t} \mathbb{I}\{i_t \in \tilde{V}'_{t,k}\} \sum_{\substack{s\in[t-1]\\ i_s\in\overline{V}'_{t,k}}} \left|\boldsymbol{x}_{a_t}^\top \overline{\boldsymbol{M}}^{-1}_{\overline{V}'_{t,k},t-1} \boldsymbol{x}_{a_s}\right|$$

$$\leq \mathbb{I}\{\mathcal{E}\}\epsilon_* \sum_{t=T_0+1}^{T} \sum_{k=1}^{m_t} \mathbb{I}\{i_t \in \tilde{V}'_{t,k}\} \sqrt{\sum_{\substack{s\in[t-1] \\ i_s\in\overline{V}'_{t,k}}} 1 \sum_{\substack{s\in[t-1] \\ i_s\in\overline{V}'_{t,k}}} \left| \boldsymbol{x}_{a_t}^\top \overline{\boldsymbol{M}}^{-1}_{\overline{V}'_{t,k},t-1} \boldsymbol{x}_{a_s} \right|^2} \tag{A.98}$$

$$\leq \mathbb{I}\{\mathcal{E}\}\epsilon_* \sum_{t=T_0+1}^{T} \sum_{k=1}^{m_t} \mathbb{I}\{i_t \in \tilde{V}'_{t,k}\} \sqrt{T_{\overline{V}'_{t,k},t-1} \left\|\boldsymbol{x}_{a_t}\right\|^2_{\overline{\boldsymbol{M}}^{-1}_{\overline{V}'_{t,k},t-1}}} \tag{A.99}$$

$$\leq \mathbb{I}\{\mathcal{E}\}\epsilon_* \sum_{t=T_0+1}^{T} \sqrt{\sum_{k=1}^{m_t} \mathbb{I}\{i_t \in \tilde{V}'_{t,k}\} \sum_{k=1}^{m_t} \mathbb{I}\{i_t \in \tilde{V}'_{t,k}\} T_{\overline{V}'_{t,k},t-1} \left\|\boldsymbol{x}_{a_t}\right\|^2_{\overline{\boldsymbol{M}}^{-1}_{\overline{V}'_{t,k},t-1}}} \tag{A.100}$$

$$\leq \mathbb{I}\{\mathcal{E}\}\epsilon_* \sqrt{T} \sum_{t=T_0+1}^{T} \sqrt{\sum_{k=1}^{m_t} \mathbb{I}\{i_t \in \tilde{V}'_{t,k}\} \left\|\boldsymbol{x}_{a_t}\right\|^2_{\overline{\boldsymbol{M}}^{-1}_{\overline{V}'_{t,k},t-1}}} \tag{A.101}$$

$$\leq \mathbb{I}\{\mathcal{E}\}\epsilon_* \sqrt{T} \sqrt{\sum_{t=T_0+1}^{T} 1 \sum_{t=T_0+1}^{T} \sum_{k=1}^{m_t} \mathbb{I}\{i_t \in \tilde{V}'_{t,k}\} \left\|\boldsymbol{x}_{a_t}\right\|^2_{\overline{\boldsymbol{M}}^{-1}_{\overline{V}'_{t,k},t-1}}} \tag{A.102}$$

$$\leq \mathbb{I}\{\mathcal{E}\}\epsilon_* \sqrt{T} \sqrt{T \sum_{t=T_0+1}^{T} \sum_{j=1}^{m} \mathbb{I}\{i_t \in V_j\} \left\|\boldsymbol{x}_{a_t}\right\|^2_{\overline{\boldsymbol{M}}^{-1}_{V_j,t-1}}} \tag{A.103}$$

$$= \mathbb{I}\{\mathcal{E}\}\epsilon_* T \sqrt{\sum_{j=1}^{m} \sum_{t=T_0+1}^{T} \mathbb{I}\{i_t \in V_j\} \left\|\boldsymbol{x}_{a_t}\right\|^2_{\overline{\boldsymbol{M}}^{-1}_{V_j,t-1}}}$$

$$\leq \mathbb{I}\{\mathcal{E}\}\epsilon_* T \sqrt{2md \log(1 + \frac{T}{\lambda d})}\,, \tag{A.104}$$

where Eq.(A.98), Eq.(A.100) and Eq.(A.102) hold because of the Cauchy–Schwarz inequality, Eq.(A.99) holds since $\overline{\boldsymbol{M}}_{\overline{V}'_{t,k},t-1} \succeq \sum_{\substack{s\in[t-1] \\ i_s\in\overline{V}'_{t,k}}} \boldsymbol{x}_{a_s}\boldsymbol{x}_{a_s}^\top$, Eq.(A.101) is because $T_{\overline{V}'_{t,k},t-1} \leq T$, Eq. (A.103) follows from the same reasoning as Eq.(A.94), and Eq.(A.104)

comes from the following technical Lemma A.4.2.

Finally, plugging Eq.(A.97) and Eq.(A.104) into Eq.(A.93), take expectation and plug it into Eq.(A.92), we can get:

$$R(T) \leq 5 + T_0 + \frac{\tilde{u}}{u} \times \frac{2\epsilon_* \sqrt{2d}T}{\tilde{\lambda}_x^{\frac{3}{2}}} + 2\epsilon_* T\left(1 + \sqrt{2md\log(1+\frac{T}{\lambda d})}\right) + 2\left(\sqrt{\lambda} + \sqrt{2\log(T) + d\log(1+\frac{T}{\lambda d})}\right) \times \sqrt{2mdT\log(1+\frac{T}{\lambda d})}\,, \tag{A.105}$$

where

$$T_0 = 16u\log(\frac{u}{\delta}) + 4u\max\max\{\frac{8d}{\tilde{\lambda}_x(\frac{\gamma_1}{4} - \epsilon_*\sqrt{\frac{1}{2\tilde{\lambda}_x}})^2}\log(\frac{u}{\delta}), \frac{16}{\tilde{\lambda}_x^2}\log(\frac{8d}{\tilde{\lambda}_x^2\delta})\}$$

is given in the following Lemma A.3.2 in Appendix A.3.8.

A.3.6 Proof and Discussions of Theorem 6.4.4

In the work [138], they give a lower bound for misspecified linear bandits with a single user. The lower bound of $R(T)$ is given by: $R_3(T) \geq \epsilon_* T\sqrt{d}$. Therefore, suppose our problem with multiple users and m underlying clusters where the arrival times are T_i for each cluster, then for any algorithms, even if they know the underlying clustering structure and keep m independent linear bandit algorithms to leverage the common information of clusters, the best they can get is $R(T) = \sum_{i\in[m]} R_3(T_i) \geq \epsilon_* \sum_{i\in[m]} T_i\sqrt{d} = \epsilon_* T\sqrt{d}$, which gives a lower bound of $O(\epsilon_* T\sqrt{d})$ for the CBMUM problem. Recall that the regret upper bound of our algorithms is of $O(\epsilon_* T\sqrt{md\log T} + d\sqrt{mT}\log T)$, asymptotically matching this lower bound with respect to T up to logarithmic factors and

with respect to m up to $O(\sqrt{m})$ factors, showing the tightness of our theoretical results (where m are typically very small for real applications).

We conjecture that the gap for the m factor is due to the strong assumption that cluster structures are known to prove our lower bound, and whether there exists a tighter lower bound will be left for future work.

A.3.7 Proof of the key Lemma 5.4.4

In Lemma 5.4.4, we want to bound $\left|\boldsymbol{x}_a^\top \overline{\boldsymbol{M}}_{\overline{V}_t,t-1}^{-1} \sum_{\substack{s\in[t-1]\\ i_s\in\overline{V}_t}} \boldsymbol{x}_{a_s}\boldsymbol{x}_{a_s}^\top(\boldsymbol{\theta}_{i_s}-\boldsymbol{\theta}_{i_t})\right|$. By the definition of "good partition", we have $\|\boldsymbol{\theta}_{i_s}-\boldsymbol{\theta}_{i_t}\|_2 \leq \zeta\,, \forall i_s \in \overline{V}_t$. It is an easy-to-be-made mistake to directly drag $\|\boldsymbol{\theta}_{i_s}-\boldsymbol{\theta}_{i_t}\|_2$ out to upper bound it by $\left\|\boldsymbol{x}_a^\top \overline{\boldsymbol{M}}_{\overline{V}_t,t-1}^{-1} \sum_{\substack{s\in[t-1]\\ i_s\in\overline{V}_t}} \boldsymbol{x}_{a_s}\boldsymbol{x}_{a_s}^\top\right\|_2 \times \zeta$ and then proceed. We need more careful analysis.

We first prove the following general lemma.

Lemma A.3.1. *For vectors $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_k \in \mathbb{R}^d, \|\boldsymbol{x}_i\|_2 \leq 1, \forall i \in [k]$, and vectors $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_k \in \mathbb{R}^d, \|\boldsymbol{\theta}_i\|_2 \leq C, \forall i \in [k]$, where $C > 0$ is a constant, we have:*

$$\left\|\sum_{i=1}^k \boldsymbol{x}_i\boldsymbol{x}_i^\top\boldsymbol{\theta}_i\right\|_2 \leq C\sqrt{d}\left\|\sum_{i=1}^k \boldsymbol{x}_i\boldsymbol{x}_i^\top\right\|_2.$$

Proof. Let $\boldsymbol{X} \in \mathbb{R}^{d\times k}$ be a matrix such that it has $\boldsymbol{x}_i$ s as its columns, i.e., $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k] = \begin{bmatrix} \boldsymbol{x}_{11} & x_{21} & \cdots & \boldsymbol{x}_{k1} \\ \boldsymbol{x}_{12} & x_{22} & \cdots & \boldsymbol{x}_{k2} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{x}_{1d} & x_{2d} & \cdots & \boldsymbol{x}_{kd} \end{bmatrix}$.

Let $\boldsymbol{y} \in \mathbb{R}^{k\times 1}$ be a vector that has $\boldsymbol{x}_i^\top \boldsymbol{\theta}_i$ s as its elements, i.e., $\boldsymbol{y} = [\boldsymbol{x}_1^\top \boldsymbol{\theta}_1, \ldots, \boldsymbol{x}_k^\top \boldsymbol{\theta}_k]^\top$. Then we have:

$$\left\| \sum_{i=1}^{k} \boldsymbol{x}_i \boldsymbol{x}_i^\top \boldsymbol{\theta}_i \right\|_2^2 = \|\boldsymbol{X}\boldsymbol{y}\|_2^2 \leq \|\boldsymbol{X}\|_2^2 \|\boldsymbol{y}\|_2^2 \tag{A.106}$$

$$= \|\boldsymbol{X}\|_2^2 \sum_{i=1}^{k} (\boldsymbol{x}_i^\top \boldsymbol{\theta}_i)^2$$

$$\leq \|\boldsymbol{X}\|_2^2 \sum_{i=1}^{k} \|\boldsymbol{x}_i\|_2^2 \|\boldsymbol{\theta}_i\|_2^2 \tag{A.107}$$

$$\leq C^2 \|\boldsymbol{X}\|_2^2 \sum_{i=1}^{k} \|\boldsymbol{x}_i\|_2^2$$

$$= C^2 \|\boldsymbol{X}\|_2^2 \|\boldsymbol{X}\|_F^2$$

$$\leq C^2 d \|\boldsymbol{X}\|_2^4 \tag{A.108}$$

$$= C^2 d \left\|\boldsymbol{X}\boldsymbol{X}^\top\right\|_2^2 \tag{A.109}$$

$$= C^2 d \left\| \sum_{i=1}^{k} \boldsymbol{x}_i \boldsymbol{x}_i^\top \right\|_2^2, \tag{A.110}$$

where Eq. (A.106) follows by the matrix operator norm inequality, Eq. (A.107) follows by the Cauchy–Schwarz inequality, Eq. (A.108) follows by $\|\boldsymbol{X}\|_F \leq \sqrt{d}\,\|\boldsymbol{X}\|_2$, Eq. (A.109) follows from $\|\boldsymbol{X}\|_2^2 = \left\|\boldsymbol{X}\boldsymbol{X}^\top\right\|_2$. □

The above result is tight. We can show that the lower bound

of $\left\|\sum_{i=1}^{k} \boldsymbol{x}_i \boldsymbol{x}_i^\top \boldsymbol{\theta}_i\right\|_2$ under the conditions in the lemma is exactly $C\sqrt{d}\left\|\sum_{i=1}^{k} \boldsymbol{x}_i \boldsymbol{x}_i^\top\right\|_2$. Specifically, let $k = 2$, $C = 1$, $d = 2$, $\boldsymbol{x}_1 = [0, 1]^\top$, $\boldsymbol{x}_2 = [1, 0]^\top$, $\boldsymbol{\theta}_1 = [1, 0]^\top$, $\boldsymbol{\theta}_2 = [0, 1]^\top$, then we have $\left\|\sum_{i=1}^{2} \boldsymbol{x}_i \boldsymbol{x}_i^\top \boldsymbol{\theta}_i\right\|_2 = \left\|[1, 1]^\top\right\|_2 = \sqrt{2}$, and $C\sqrt{d}\left\|\sum_{i=1}^{2} \boldsymbol{x}_i \boldsymbol{x}_i^\top\right\|_2 = 1 \times \sqrt{2} \times \left\|\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right\|_2 = \sqrt{2}$. Therefore, we have that the upper bound given in Lemma A.3.1 matches the lower bound.

We are now ready to prove the key Lemma 5.4.4 with the above Lemma A.3.1.

At any $t \geq T_0$, if the current partition is a "good partition", and $\overline{V}_t \notin \mathcal{V}$, then for all $\boldsymbol{x}_a \in \mathbb{R}^d$, $\|\boldsymbol{x}_a\|_2 \leq 1$, with probability at least $1 - \delta$:

$$
\begin{aligned}
&\left|\boldsymbol{x}_a^\top \overline{\boldsymbol{M}}_{\overline{V}_t, t-1}^{-1} \sum_{\substack{s \in [t-1] \\ i_s \in \overline{V}_t}} \boldsymbol{x}_{a_s} \boldsymbol{x}_{a_s}^\top (\boldsymbol{\theta}_{i_s} - \boldsymbol{\theta}_{i_t})\right| \\
&\leq \|\boldsymbol{x}_a\|_2 \left\|\overline{\boldsymbol{M}}_{\overline{V}_t, t-1}^{-1} \sum_{\substack{s \in [t-1] \\ i_s \in \overline{V}_t}} \boldsymbol{x}_{a_s} \boldsymbol{x}_{a_s}^\top (\boldsymbol{\theta}_{i_s} - \boldsymbol{\theta}_{i_t})\right\|_2 && \text{(A.111)} \\
&\leq \left\|\overline{\boldsymbol{M}}_{\overline{V}_t, t-1}^{-1}\right\|_2 \left\|\sum_{\substack{s \in [t-1] \\ i_s \in \overline{V}_t}} \boldsymbol{x}_{a_s} \boldsymbol{x}_{a_s}^\top (\boldsymbol{\theta}_{i_s} - \boldsymbol{\theta}_{i_t})\right\|_2 && \text{(A.112)} \\
&\leq 2\epsilon_* \sqrt{\frac{2d}{\tilde{\lambda}_x}} \times \left\|\overline{\boldsymbol{M}}_{\overline{V}_t, t-1}^{-1}\right\|_2 \left\|\sum_{\substack{s \in [t-1] \\ i_s \in \overline{V}_t}} \boldsymbol{x}_{a_s} \boldsymbol{x}_{a_s}^\top\right\|_2 && \text{(A.113)} \\
&\leq 2\epsilon_* \sqrt{\frac{2d}{\tilde{\lambda}_x}} \times \frac{\lambda_{max}(\sum_{\substack{s \in [t-1] \\ i_s \in \overline{V}_t}} \boldsymbol{x}_{a_s} \boldsymbol{x}_{a_s}^\top)}{\lambda_{\min}(\overline{\boldsymbol{M}}_{\overline{V}_t, t-1})}
\end{aligned}
$$

$$\leq 2\epsilon_* \sqrt{\frac{2d}{\tilde{\lambda}_x}} \times \frac{T_{\overline{V}_t,t-1}}{2T_{\overline{V}_t,t-1}\tilde{\lambda}_x + \lambda} \tag{A.114}$$
$$\leq \frac{\epsilon_* \sqrt{2d}}{\tilde{\lambda}_x^{\frac{3}{2}}},$$

where Eq.(A.111) follows by the Cauchy–Schwarz inequality, Eq.(A.112) follows from the inequality of matrix's operator norm, Eq.(A.113) follows from the fact that in a "good partition", $\|\boldsymbol{\theta}_{i_t} - \boldsymbol{\theta}_l\|_2 \leq 2\epsilon_* \sqrt{\frac{2}{\tilde{\lambda}_x}}, \forall l \in \overline{V}_t$ and Lemma A.3.1, Eq.(A.114) follows by Eq.(A.348) with probability $\geq 1 - \delta$.

A.3.8 Lemma A.3.2 of the sufficient time T_0 and its proof

The following lemma gives a sufficient time T_0 for the algorithm to get a "good partition".

Lemma A.3.2. *With the carefully designed edge deletion rule, after*

$$T_0 \triangleq 16u \log(\frac{u}{\delta}) + 4u \max \max\{\frac{8d}{\tilde{\lambda}_x(\frac{\gamma_1}{4} - \epsilon_* \sqrt{\frac{1}{2\tilde{\lambda}_x}})^2} \log(\frac{u}{\delta}), \frac{16}{\tilde{\lambda}_x^2} \log(\frac{8d}{\tilde{\lambda}_x^2 \delta})\}$$
$$= O\left(u \left(\frac{d}{\tilde{\lambda}_x(\gamma_1 - \zeta)^2} + \frac{1}{\tilde{\lambda}_x^2}\right) \log \frac{1}{\delta}\right)$$

rounds, with probability at least $1 - 3\delta$ *for some* $\delta \in (0, \frac{1}{3})$*, RCLUMB can always get a "good partition".*

Below is the detailed proof of Lemma A.3.2.

Proof. We first prove the following result:
With probability at least $1 - \delta$ for some $\delta \in (0, 1)$, at any $t \in [T]$:

$$\left\| \hat{\boldsymbol{\theta}}_{i,t} - \boldsymbol{\theta}^{j(i)} \right\|_2 \leq \frac{\beta(T_{i,t}, \frac{\delta}{u}) + \epsilon_* \sqrt{T_{i,t}}}{\sqrt{\lambda + \lambda_{\min}(\boldsymbol{M}_{i,t})}}, \forall i \in \mathcal{U}, \tag{A.115}$$

where $\beta(T_{i,t}, \frac{\delta}{u}) \triangleq \sqrt{\lambda} + \sqrt{2\log(\frac{u}{\delta}) + d\log(1 + \frac{T_{i,t}}{\lambda d})}$.

$$\begin{aligned}
\hat{\boldsymbol{\theta}}_{i,t} - \boldsymbol{\theta}^{j(i)} &= (\sum_{\substack{s\in[t]\\ i_s=i}} \boldsymbol{x}_{a_s}\boldsymbol{x}_{a_s}^\top + \lambda \boldsymbol{I})^{-1}\Bigg(\sum_{\substack{s\in[t]\\ i_s=i}} \boldsymbol{x}_{a_s}(\boldsymbol{x}_{a_s}^\top \boldsymbol{\theta}^{j(i)} + \boldsymbol{\epsilon}_{a_s}^{i_s,s} + \eta_s)\Bigg) - \boldsymbol{\theta}^{j(i)} \qquad \text{(A.116)}\\
&= (\sum_{\substack{s\in[t]\\ i_s=i}} \boldsymbol{x}_{a_s}\boldsymbol{x}_{a_s}^\top + \lambda \boldsymbol{I})^{-1}[(\sum_{\substack{s\in[t]\\ i_s=i}} \boldsymbol{x}_{a_s}\boldsymbol{x}_{a_s}^\top + \lambda \boldsymbol{I})\boldsymbol{\theta}^{j(i)} - \lambda\boldsymbol{\theta}^{j(i)} + \sum_{\substack{s\in[t]\\ i_s=i}} \boldsymbol{x}_{a_s}\boldsymbol{\epsilon}_{a_s}^{i_s,s}\\
&+ \sum_{\substack{s\in[t]\\ i_s=i}} \boldsymbol{x}_{a_s}\eta_s] - \boldsymbol{\theta}^{j(i)}\\
&= -\lambda\tilde{\boldsymbol{M}}_{i,t}^{-1}\boldsymbol{\theta}^{j(i)} + \tilde{\boldsymbol{M}}_{i,t}^{-1}\sum_{\substack{s\in[t]\\ i_s=i}} \boldsymbol{x}_{a_s}\boldsymbol{\epsilon}_{a_s}^{i_s,s} + \tilde{\boldsymbol{M}}_{i,t}^{-1}\sum_{\substack{s\in[t]\\ i_s=i}} \boldsymbol{x}_{a_s}\eta_s\,,
\end{aligned}$$

where we denote $\tilde{\boldsymbol{M}}_{i,t} = \boldsymbol{M}_{i,t} + \lambda\boldsymbol{I}$, and Eq.(A.116) holds by definition.

Therefore,

$$\left\|\hat{\boldsymbol{\theta}}_{i,t} - \boldsymbol{\theta}^{j(i)}\right\|_2 \le \lambda\left\|\tilde{\boldsymbol{M}}_{i,t}^{-1}\boldsymbol{\theta}^{j(i)}\right\|_2 + \left\|\tilde{\boldsymbol{M}}_{i,t}^{-1}\sum_{\substack{s\in[t]\\ i_s=i}} \boldsymbol{x}_{a_s}\boldsymbol{\epsilon}_{a_s}^{i_s,s}\right\|_2 + \left\|\tilde{\boldsymbol{M}}_{i,t}^{-1}\sum_{\substack{s\in[t]\\ i_s=i}} \boldsymbol{x}_{a_s}\eta_s\right\|_2 . \qquad \text{(A.117)}$$

We then bound the three terms in Eq.(A.162) one by one. For the first term:

$$\lambda\left\|\tilde{\boldsymbol{M}}_{i,t}^{-1}\boldsymbol{\theta}^{j(i)}\right\|_2 \le \lambda\left\|\tilde{\boldsymbol{M}}_{i,t}^{-\frac{1}{2}}\right\|_2^2\left\|\boldsymbol{\theta}^{j(i)}\right\|_2 \le \frac{\sqrt{\lambda}}{\sqrt{\lambda_{\min}(\tilde{\boldsymbol{M}}_{i,t})}}\,, \qquad \text{(A.118)}$$

where we use the Cauchy–Schwarz inequality, the inequality for the operator norm of matrices, and the fact that $\lambda_{\min}(\tilde{\boldsymbol{M}}_{i,t}) \ge \lambda$.

For the second term in Eq.(A.162):

$$\begin{aligned}
\left\| \tilde{\boldsymbol{M}}_{i,t}^{-1} \sum_{\substack{s\in[t]\\ i_s=i}} \boldsymbol{x}_{a_s} \boldsymbol{\epsilon}_{a_s}^{i_s,s} \right\|_2 &= \max_{\boldsymbol{x}\in S^{d-1}} \sum_{\substack{s\in[t]\\ i_s=i}} \boldsymbol{x}^\top \tilde{\boldsymbol{M}}_{i,t}^{-1} \boldsymbol{x}_{a_s} \boldsymbol{\epsilon}_{a_s}^{i_s,s} \\
&\le \max_{\boldsymbol{x}\in S^{d-1}} \sum_{\substack{s\in[t]\\ i_s=i}} \left| \boldsymbol{x}^\top \tilde{\boldsymbol{M}}_{i,t}^{-1} \boldsymbol{x}_{a_s} \boldsymbol{\epsilon}_{a_s}^{i_s,s} \right| \\
&\le \max_{\boldsymbol{x}\in S^{d-1}} \sum_{\substack{s\in[t]\\ i_s=i}} \left| \boldsymbol{x}^\top \tilde{\boldsymbol{M}}_{i,t}^{-1} \boldsymbol{x}_{a_s} \right| \left\| \boldsymbol{\epsilon}_{a_s}^{i_s,s} \right\|_\infty \quad \text{(A.119)} \\
&\le \epsilon_* \max_{\boldsymbol{x}\in S^{d-1}} \sum_{\substack{s\in[t]\\ i_s=i}} \left| \boldsymbol{x}^\top \tilde{\boldsymbol{M}}_{i,t}^{-1} \boldsymbol{x}_{a_s} \right| \\
&\le \epsilon_* \max_{\boldsymbol{x}\in S^{d-1}} \sqrt{\sum_{\substack{s\in[t]\\ i_s=i}} 1 \sum_{\substack{s\in[t]\\ i_s=i}} \left| \boldsymbol{x}^\top \tilde{\boldsymbol{M}}_{i,t}^{-1} \boldsymbol{x}_{a_s} \right|^2} \quad \text{(A.120)} \\
&\le \epsilon_* \sqrt{T_{i,t}} \sqrt{\max_{\boldsymbol{x}\in S^{d-1}} \boldsymbol{x}^\top \tilde{\boldsymbol{M}}_{i,t}^{-1} \boldsymbol{x}} \quad \text{(A.121)} \\
&= \frac{\epsilon_* \sqrt{T_{i,t}}}{\sqrt{\lambda_{\min}(\tilde{\boldsymbol{M}}_{i,t})}}, \quad \text{(A.122)}
\end{aligned}$$

where we denote $S^{d-1} = \{\boldsymbol{x} \in \mathbb{R}^d : \|\boldsymbol{x}\|_2 = 1\}$, Eq.(A.119) follows from Holder's inequality, Eq.(A.120) follows by the Cauchy–Schwarz inequality, Eq.(A.121) holds because $\tilde{\boldsymbol{M}}_{i,t} \succeq \sum_{\substack{s\in[t]\\ i_s=i}} \boldsymbol{x}_{a_s} \boldsymbol{x}_{a_s}^\top$, Eq.(A.122) follows from the Courant-Fischer theorem.

For the last term in Eq.(A.162)

$$\left\| \tilde{\boldsymbol{M}}_{i,t}^{-1} \sum_{\substack{s\in[t]\\ i_s=i}} \boldsymbol{x}_{a_s} \eta_s \right\|_2 \le \left\| \tilde{\boldsymbol{M}}_{i,t}^{-\frac{1}{2}} \sum_{\substack{s\in[t]\\ i_s=i}} \boldsymbol{x}_{a_s} \eta_s \right\|_2 \left\| \tilde{\boldsymbol{M}}_{i,t}^{-\frac{1}{2}} \right\|_2 \quad \text{(A.123)}$$

$$= \frac{\left\| \sum_{\substack{s\in[t]\\ i_s=i}} \boldsymbol{x}_{a_s}\eta_s \right\|_{\tilde{\boldsymbol{M}}_{i,t}^{-1}}}{\sqrt{\lambda_{\min}(\tilde{\boldsymbol{M}}_{i,t})}}\,, \tag{A.124}$$

where Eq.(A.164) follows by the Cauchy–Schwarz inequality and the inequality for the operator norm of matrices, and Eq.(A.165) follows by the Courant-Fischer theorem.

Following Theorem 1 in [43], with probability at least $1-\delta$ for some $\delta \in (0,1)$, for any $i \in \mathcal{U}$, we have:

$$\begin{aligned}\left\| \sum_{\substack{s\in[t]\\ i_s=i}} \boldsymbol{x}_{a_s}\eta_s \right\|_{\tilde{\boldsymbol{M}}_{i,t}^{-1}} &\leq \sqrt{2\log(\frac{u}{\delta}) + \log(\frac{\det(\tilde{\boldsymbol{M}}_{i,t})}{\det(\lambda \boldsymbol{I})})} \\ &\leq \sqrt{2\log(\frac{u}{\delta}) + d\log(1+\frac{T_{i,t}}{\lambda d})}\,, \end{aligned} \tag{A.125}$$

where $\det(\boldsymbol{M})$ denotes the determinant of matrix $\boldsymbol{M}$, Eq.(A.166) is because $\det(\tilde{\boldsymbol{M}}_{i,t}) \leq \left(\frac{\mathrm{trace}(\lambda \boldsymbol{I} + \sum_{\substack{s\in[t]\\ i_s=i}} \boldsymbol{x}_{a_s}\boldsymbol{x}_{a_s}^\top)}{d} \right)^d \leq \left(\frac{\lambda d + T_{i,t}}{d}\right)^d$, and $\det(\lambda \boldsymbol{I}) = \lambda^d$.

Plugging Eq.(A.166) into Eq. (A.165), then plugging Eq. (A.163), Eq.(A.122) and Eq.(A.165) into Eq.(A.162), we can get that Eq.(A.161) holds with probability $\geq 1-\delta$.

Then, with the item regularity assumption stated in Assumption 9.4, the technical Lemma A.4.1, together with Lemma 7 in [135], with probability at least $1-\delta$, for a particular user i, at any t such that $T_{i,t} \geq \frac{16}{\tilde{\lambda}_x^2}\log(\frac{8d}{\tilde{\lambda}_x^2\delta})$, we have:

$$\lambda_{\min}(\tilde{\boldsymbol{M}}_{i,t}) \geq 2\tilde{\lambda}_x T_{i,t} + \lambda\,. \tag{A.126}$$

Based on the above reasoning, we have: if $T_{i,t} \geq \frac{16}{\tilde{\lambda}_x^2}\log(\frac{8d}{\tilde{\lambda}_x^2\delta})$,

then with probability $\geq 1-2\delta$, we have:

$$\begin{aligned}\left\|\hat{\boldsymbol{\theta}}_{i,t}-\boldsymbol{\theta}^{j(i)}\right\|_2 &\leq \frac{\beta(T_{i,t},\frac{\delta}{u})+\epsilon_*\sqrt{T_{i,t}}}{\sqrt{\lambda_{\min}(\tilde{\boldsymbol{M}}_{i,t})}}\\ &\leq \frac{\beta(T_{i,t},\frac{\delta}{u})+\epsilon_*\sqrt{T_{i,t}}}{\sqrt{2\tilde{\lambda}_x T_{i,t}+\lambda}}\\ &\leq \frac{\sqrt{\lambda}+\sqrt{2\log(\frac{u}{\delta})+d\log(1+\frac{T_{i,t}}{\lambda d})}}{\sqrt{2\tilde{\lambda}_x T_{i,t}+\lambda}}+\epsilon_*\sqrt{\frac{1}{2\tilde{\lambda}_x}},\end{aligned} \tag{A.127}$$

for any $i\in\mathcal{U}$.

Let

$$\frac{\sqrt{\lambda}+\sqrt{2\log(\frac{u}{\delta})+d\log(1+\frac{T_{i,t}}{\lambda d})}}{\sqrt{2\tilde{\lambda}_x T_{i,t}+\lambda}}+\epsilon_*\sqrt{\frac{1}{2\tilde{\lambda}_x}}<\frac{\gamma_1}{4}, \tag{A.128}$$

which is equivalent to

$$\frac{\sqrt{\lambda}+\sqrt{2\log(\frac{u}{\delta})+d\log(1+\frac{T_{i,t}}{\lambda d})}}{\sqrt{2\tilde{\lambda}_x T_{i,t}+\lambda}}<\frac{\gamma_1}{4}-\epsilon_*\sqrt{\frac{1}{2\tilde{\lambda}_x}}, \tag{A.129}$$

where γ_1 is given in Definition 5.2.

Assume $\lambda\leq 2\log(\frac{u}{\delta})+d\log(1+\frac{T_{i,t}}{\lambda d})$, which is typically held, then a sufficient condition for Eq. (A.129) is:

$$\frac{2\log(\frac{u}{\delta})+d\log(1+\frac{T_{i,t}}{\lambda d})}{2\tilde{\lambda}_x T_{i,t}}<\frac{1}{4}(\frac{\gamma_1}{4}-\epsilon_*\sqrt{\frac{1}{2\tilde{\lambda}_x}})^2. \tag{A.130}$$

To satisfy the condition in Eq.(A.130), it is sufficient to show

$$\frac{2\log(\frac{u}{\delta})}{2\tilde{\lambda}_x T_{i,t}}<\frac{1}{8}(\frac{\gamma_1}{4}-\epsilon_*\sqrt{\frac{1}{2\tilde{\lambda}_x}})^2 \tag{A.131}$$

and

$$\frac{d\log(1+\frac{T_{i,t}}{\lambda d})}{2\tilde{\lambda}_x T_{i,t}} < \frac{1}{8}(\frac{\gamma_1}{4} - \epsilon_*\sqrt{\frac{1}{2\tilde{\lambda}_x}})^2 \,. \tag{A.132}$$

From Eq.(A.319), we can get:

$$T_{i,t} \geq \frac{8\log(\frac{u}{\delta})}{\tilde{\lambda}_x(\frac{\gamma_1}{4} - \epsilon_*\sqrt{\frac{1}{2\tilde{\lambda}_x}})^2} \,. \tag{A.133}$$

Following Lemma 9 in [135], we can get the following sufficient condition for Eq.(A.320):

$$T_{i,t} \geq \frac{8d\log(\frac{4}{\lambda\tilde{\lambda}_x(\frac{\gamma_1}{4} - \epsilon_*\sqrt{\frac{1}{2\tilde{\lambda}_x}})^2})}{\tilde{\lambda}_x(\frac{\gamma_1}{4} - \epsilon_*\sqrt{\frac{1}{2\tilde{\lambda}_x}})^2} \,. \tag{A.134}$$

Assume $\frac{u}{\delta} \geq \frac{4}{\lambda\tilde{\lambda}_x(\frac{\gamma_1}{4} - \epsilon_*\sqrt{\frac{1}{2\tilde{\lambda}_x}})^2}$, which is typically held, we can get that

$$T_{i,t} \geq \frac{8d}{\tilde{\lambda}_x(\frac{\gamma_1}{4} - \epsilon_*\sqrt{\frac{1}{2\tilde{\lambda}_x}})^2}\log(\frac{u}{\delta}) \tag{A.135}$$

is a sufficient condition for Eq.(A.128). Together with the condition that $T_{i,t} \geq \frac{16}{\tilde{\lambda}_x^2}\log(\frac{8d}{\tilde{\lambda}_x^2\delta})$, we can get that if

$$T_{i,t} \geq \max\{\frac{8d}{\tilde{\lambda}_x(\frac{\gamma_1}{4} - \epsilon_*\sqrt{\frac{1}{2\tilde{\lambda}_x}})^2}\log(\frac{u}{\delta}), \frac{16}{\tilde{\lambda}_x^2}\log(\frac{8d}{\tilde{\lambda}_x^2\delta})\}, \forall i \in \mathcal{U} \,, \tag{A.136}$$

then with probability $\geq 1 - 2\delta$:

$$\left\|\hat{\boldsymbol{\theta}}_{i,t} - \boldsymbol{\theta}^{j(i)}\right\|_2 < \frac{\gamma_1}{4} \,, \forall i \in \mathcal{U} \,.$$

By Lemma 8 in [135], and Assumption 9.3 of user arrival uniformness, we have that for all

$$t \geq T_0 \triangleq 16u\log(\frac{u}{\delta}) + 4u\max\{\frac{8d}{\tilde{\lambda}_x(\frac{\gamma_1}{4} - \epsilon_*\sqrt{\frac{1}{2\tilde{\lambda}_x}})^2}\log(\frac{u}{\delta}), \frac{16}{\tilde{\lambda}_x^2}\log(\frac{8d}{\tilde{\lambda}_x^2\delta})\} \,, \tag{A.137}$$

with probability at least $1-\delta$, condition in Eq.(A.136) is satisfied.

Therefore we have that for all $t \geq T_0$, with probability $\geq 1-3\delta$:

$$\left\|\hat{\boldsymbol{\theta}}_{i,t} - \boldsymbol{\theta}^{j(i)}\right\|_2 < \frac{\gamma_1}{4}, \forall i \in \mathcal{U}. \tag{A.138}$$

Next, we show that with Eq.(A.325), we can get that the RCLUMB keeps a "good partition". First, if we delete the edge (i,l), then user i and user j belong to different *ground-truth clusters*, i.e., $\|\boldsymbol{\theta}_i - \boldsymbol{\theta}_l\|_2 > 0$. This is because by the deletion rule of the algorithm, the concentration bound, and triangle inequality, $\|\boldsymbol{\theta}_i - \boldsymbol{\theta}_l\|_2 = \left\|\boldsymbol{\theta}^{j(i)} - \boldsymbol{\theta}^{j(l)}\right\|_2 \geq \left\|\hat{\boldsymbol{\theta}}_{i,t} - \hat{\boldsymbol{\theta}}_{l,t}\right\|_2 - \left\|\boldsymbol{\theta}^{j(l)} - \boldsymbol{\theta}_{l,t}\right\|_2 - \left\|\boldsymbol{\theta}^{j(i)} - \boldsymbol{\theta}_{i,t}\right\|_2 > 0$. Second, we show that if $\|\boldsymbol{\theta}_i - \boldsymbol{\theta}_l\| \geq \gamma_1 > 2\epsilon_* \sqrt{\frac{2}{\tilde{\lambda}_x}}$, the RCLUMB algorithm will delete the edge (i,l). This is because if $\|\boldsymbol{\theta}_i - \boldsymbol{\theta}_l\| \geq \gamma_1$, then by the triangle inequality, and $\left\|\hat{\boldsymbol{\theta}}_{i,t} - \boldsymbol{\theta}^{j(i)}\right\|_2 < \frac{\gamma_1}{4}$, $\left\|\hat{\boldsymbol{\theta}}_{l,t} - \boldsymbol{\theta}^{j(l)}\right\|_2 < \frac{\gamma_1}{4}$, $\boldsymbol{\theta}_i = \boldsymbol{\theta}^{j(i)}$, $\boldsymbol{\theta}_l = \boldsymbol{\theta}^{j(l)}$, we have $\left\|\hat{\boldsymbol{\theta}}_{i,t} - \hat{\boldsymbol{\theta}}_{l,t}\right\|_2 \geq \|\boldsymbol{\theta}_i - \boldsymbol{\theta}_l\| - \left\|\hat{\boldsymbol{\theta}}_{i,t} - \boldsymbol{\theta}^{j(i)}\right\|_2 - \left\|\hat{\boldsymbol{\theta}}_{l,t} - \boldsymbol{\theta}^{j(l)}\right\|_2 > \gamma_1 - \frac{\gamma_1}{4} - \frac{\gamma_1}{4} = \frac{\gamma_1}{2} > \frac{\sqrt{\lambda}+\sqrt{2\log(\frac{u}{\delta})+d\log(1+\frac{T_{i,t}}{\lambda d})}}{\sqrt{\lambda+2\tilde{\lambda}_x T_{i,t}}} + \epsilon_* \sqrt{\frac{1}{2\tilde{\lambda}_x}} + \frac{\sqrt{\lambda}+\sqrt{2\log(\frac{u}{\delta})+d\log(1+\frac{T_{l,t}}{\lambda d})}}{\sqrt{\lambda+2\tilde{\lambda}_x T_{l,t}}} + \epsilon_* \sqrt{\frac{1}{2\tilde{\lambda}_x}}$, which will trigger the deletion condition Line 10 in Algo.7.

From the above reasoning, we can get that at round t, any user within $\overline{V}_t$ is ζ-close to i_t, and all the users belonging to $V_{j(i)}$ are contained in $\overline{V}_t$, which means the algorithm has done a "good partition" at t by Definition 5.4. □

A.3.9 Proof of Lemma 6.4.2

We prove the result in two situations: when $\overline{V}_t \in \mathcal{V}$ and when $\overline{V}_t \notin \mathcal{V}$.

(1) Situation 1: for any $t \geq T_0$ and $\overline{V}_t \in \mathcal{V}$, which means that

the current user i_t is clustered completely correctly, i.e., $\overline{V}_t = V_{j(i_t)}$, therefore $\boldsymbol{\theta}_l = \boldsymbol{\theta}_{i_t}, \forall l \in \overline{V}_t$, then we have:

$$\begin{aligned}
\hat{\boldsymbol{\theta}}_{\overline{V}_t,t-1} - \boldsymbol{\theta}_{i_t} &= (\sum_{\substack{s\in[t-1]\\ i_s\in\overline{V}_t}} \boldsymbol{x}_{a_s}\,\boldsymbol{x}_{a_s}^\top + \lambda \boldsymbol{I})^{-1}(\sum_{\substack{s\in[t-1]\\ i_s\in\overline{V}_t}} \boldsymbol{x}_{a_s} r_s) - \boldsymbol{\theta}_{i_t} \\
&= (\sum_{\substack{s\in[t-1]\\ i_s\in\overline{V}_t}} \boldsymbol{x}_{a_s}\,\boldsymbol{x}_{a_s}^\top + \lambda \boldsymbol{I})^{-1}\Bigg(\sum_{\substack{s\in[t-1]\\ i_s\in\overline{V}_t}} \boldsymbol{x}_{a_s}(\boldsymbol{x}_{a_s}^\top \boldsymbol{\theta}_{i_s} + \boldsymbol{\epsilon}_{a_s}^{i_s,s} + \eta_s)\Bigg) - \boldsymbol{\theta}_{i_t} \\
&= (\sum_{\substack{s\in[t-1]\\ i_s\in\overline{V}_t}} \boldsymbol{x}_{a_s}\,\boldsymbol{x}_{a_s}^\top + \lambda \boldsymbol{I})^{-1}\Bigg(\sum_{\substack{s\in[t-1]\\ i_s\in\overline{V}_t}} \boldsymbol{x}_{a_s}(\boldsymbol{x}_{a_s}^\top \boldsymbol{\theta}_{i_t} + \boldsymbol{\epsilon}_{a_s}^{i_s,s} + \eta_s)\Bigg) - \boldsymbol{\theta}_{i_t} \\
&= (\sum_{\substack{s\in[t-1]\\ i_s\in\overline{V}_t}} \boldsymbol{x}_{a_s}\,\boldsymbol{x}_{a_s}^\top + \lambda \boldsymbol{I})^{-1}[(\sum_{\substack{s\in[t-1]\\ i_s\in\overline{V}_t}} \boldsymbol{x}_{a_s}\boldsymbol{x}_{a_s}^\top + \lambda \boldsymbol{I})\boldsymbol{\theta}_{i_t} - \lambda\boldsymbol{\theta}_{i_t} \\
&+ \sum_{\substack{s\in[t-1]\\ i_s\in\overline{V}_t}} \boldsymbol{x}_{a_s}\boldsymbol{\epsilon}_{a_s}^{i_s,s} + \sum_{\substack{s\in[t-1]\\ i_s\in\overline{V}_t}} \boldsymbol{x}_{a_s}\eta_s] - \boldsymbol{\theta}_{i_t} \\
&= -\lambda \overline{\boldsymbol{M}}^{-1}_{\overline{V}_t,t-1}\boldsymbol{\theta}_{i_t} + \sum_{\substack{s\in[t-1]\\ i_s\in\overline{V}_t}} \overline{\boldsymbol{M}}^{-1}_{\overline{V}_t,t-1}\boldsymbol{x}_{a_s}\boldsymbol{\epsilon}_{a_s}^{i_s,s} + \sum_{\substack{s\in[t-1]\\ i_s\in\overline{V}_t}} \overline{\boldsymbol{M}}^{-1}_{\overline{V}_t,t-1}\boldsymbol{x}_{a_s}\eta_s\,.
\end{aligned}$$

Therefore we have

$$\begin{aligned}
\left|\boldsymbol{x}_a^\top(\hat{\boldsymbol{\theta}}_{\overline{V}_t,t-1} - \boldsymbol{\theta}_{i_t})\right| \le \lambda\left|\boldsymbol{x}_a^\top \overline{\boldsymbol{M}}^{-1}_{\overline{V}_t,t-1}\boldsymbol{\theta}_{i_t}\right| &+ \left|\sum_{\substack{s\in[t-1]\\ i_s\in\overline{V}_t}} \boldsymbol{x}_a^\top \overline{\boldsymbol{M}}^{-1}_{\overline{V}_t,t-1}\boldsymbol{x}_{a_s}\boldsymbol{\epsilon}_{a_s}^{i_s,s}\right| \\
&+ \left|\boldsymbol{x}_a^\top \overline{\boldsymbol{M}}^{-1}_{\overline{V}_t,t-1}\sum_{\substack{s\in[t-1]\\ i_s\in\overline{V}_t}} \boldsymbol{x}_{a_s}\eta_s\right|. \qquad \text{(A.139)}
\end{aligned}$$

Next, we bound the three terms in Eq.(A.139). For the first term:

$$\lambda\left|\boldsymbol{x}_a^\top \overline{\boldsymbol{M}}^{-1}_{\overline{V}_t,t-1}\boldsymbol{\theta}_{i_t}\right| \le \lambda \left\|\boldsymbol{x}_a\right\|_{\overline{\boldsymbol{M}}^{-1}_{\overline{V}_t,t-1}} \sqrt{\lambda_{\max}(\overline{\boldsymbol{M}}^{-1}_{\overline{V}_t,t-1})}\left\|\boldsymbol{\theta}_{i_t}\right\|_2 \le \sqrt{\lambda}\left\|\boldsymbol{x}_a\right\|_{\overline{\boldsymbol{M}}^{-1}_{\overline{V}_t,t-1}}, \qquad \text{(A.140)}$$

where we use the inequality of matrix norm, the Cauchy–Schwarz inequality, $\|\boldsymbol{\theta}_{i_t}\|_2 \leq 1$, and the fact that $\lambda_{\max}(\overline{\boldsymbol{M}}^{-1}_{\overline{V}_t,t-1}) = \frac{1}{\lambda_{\min}(\overline{\boldsymbol{M}}_{\overline{V}_t,t-1})} \leq \frac{1}{\lambda}$.

For the second term in Eq.(A.139):

$$\begin{aligned}\left|\sum_{\substack{s\in[t-1]\\ i_s\in\overline{V}_t}} \boldsymbol{x}_a^\top \overline{\boldsymbol{M}}^{-1}_{\overline{V}_t,t-1}\boldsymbol{x}_{a_s}\boldsymbol{\epsilon}_{a_s}^{i_s,s}\right| &\leq \sum_{\substack{s\in[t-1]\\ i_s\in\overline{V}_t}} \left|\boldsymbol{x}_a^\top \overline{\boldsymbol{M}}^{-1}_{\overline{V}_t,t-1}\boldsymbol{x}_{a_s}\boldsymbol{\epsilon}_{a_s}^{i_s,s}\right| \\ &\leq \sum_{\substack{s\in[t-1]\\ i_s\in\overline{V}_t}} \left\|\boldsymbol{\epsilon}_{a_s}^{i_s,s}\right\|_\infty \left|\boldsymbol{x}_a^\top \overline{\boldsymbol{M}}^{-1}_{\overline{V}_t,t-1}\boldsymbol{x}_{a_s}\right| \\ &\leq \epsilon_* \sum_{\substack{s\in[t-1]\\ i_s\in\overline{V}_t}} \left|\boldsymbol{x}_a^\top \overline{\boldsymbol{M}}^{-1}_{\overline{V}_t,t-1}\boldsymbol{x}_{a_s}\right|, \quad \text{(A.141)}\end{aligned}$$

where in the second inequality we use the Holder's inequality.

For the last term, with probability at least $1-\delta$:

$$\left|\boldsymbol{x}_a^\top \overline{\boldsymbol{M}}^{-1}_{\overline{V}_t,t-1} \sum_{\substack{s\in[t-1]\\ i_s\in\overline{V}_t}} \boldsymbol{x}_{a_s}\eta_s\right| \leq \|\boldsymbol{x}_a\|_{\overline{\boldsymbol{M}}^{-1}_{\overline{V}_t,t-1}} \left\|\sum_{\substack{s\in[t-1]\\ i_s\in\overline{V}_t}} \boldsymbol{x}_{a_s}\eta_s\right\|_{\overline{\boldsymbol{M}}^{-1}_{\overline{V}_t,t-1}} \quad \text{(A.142)}$$

$$\leq \|\boldsymbol{x}_a\|_{\overline{\boldsymbol{M}}^{-1}_{\overline{V}_t,t-1}} \sqrt{2\log(\frac{1}{\delta}) + \log(\frac{\det(\overline{\boldsymbol{M}}_{\overline{V}_t,t-1})}{\det(\lambda\boldsymbol{I})})}$$

$$\leq \|\boldsymbol{x}_a\|_{\overline{\boldsymbol{M}}^{-1}_{\overline{V}_t,t-1}} \sqrt{2\log(\frac{1}{\delta}) + d\log(1+\frac{T}{\lambda d})}, \quad \text{(A.143)}$$

where the second inequality follows by Theorem 1 in [43], Eq.(A.143) is because $\det(\overline{\boldsymbol{M}}_{\overline{V}_t,t-1}) \leq \left(\frac{\mathrm{trace}(\lambda\boldsymbol{I}+\sum_{\substack{s\in[t]\\ i_s\in\overline{V}_t}} \boldsymbol{x}_{a_s}\boldsymbol{x}_{a_s}^\top)}{d}\right)^d \leq \left(\frac{\lambda d+T_{\overline{V}_t,t}}{d}\right)^d \leq \left(\frac{\lambda d+T}{d}\right)^d$, and $\det(\lambda\boldsymbol{I}) = \lambda^d$.

Plugging Eq.(A.140), Eq.(A.141) and Eq.(A.143) into Eq.(A.139), we can prove Lemma 6.4.2 in situation 1, i.e., for any $t \geq T_0$ and $\overline{V}_t \in V$, with probability at least $1-\delta$:

$$\left|\boldsymbol{x}_a^\top(\hat{\boldsymbol{\theta}}_{\overline{V}_t,t-1} - \boldsymbol{\theta}_{i_t})\right| \leq \epsilon_* \sum_{\substack{s\in[t-1]\\ i_s\in\overline{V}_t}} \left|\boldsymbol{x}_a^\top \overline{\boldsymbol{M}}^{-1}_{\overline{V}_t,t-1}\boldsymbol{x}_{a_s}\right| + \|\boldsymbol{x}_a\|_{\overline{\boldsymbol{M}}^{-1}_{\overline{V}_t,t-1}} \left(\sqrt{\lambda} + \sqrt{2\log(\frac{1}{\delta}) + d\log(1+\frac{T}{\lambda d})}\right). \tag{A.144}$$

(2) Situation 2: for any $t \geq T_0$ and $\overline{V}_t \notin \mathcal{V}$, which means that the current user is *misclustered* by the algorithm, i.e., $\overline{V}_t \neq V_{j(i_t)}$, but with Lemma A.3.2, with probability at least $1-3\delta$, the current partition is a "good partition", i.e., $\|\boldsymbol{\theta}_l - \boldsymbol{\theta}_{i_t}\|_2 \leq 2\epsilon_*\sqrt{\frac{2}{\tilde{\lambda}_x}}, \forall l \in \overline{V}_t$, we have:

$$\begin{aligned}
&\hat{\boldsymbol{\theta}}_{\overline{V}_t,t-1} - \boldsymbol{\theta}_{i_t}\\
&= (\sum_{\substack{s\in[t-1]\\ i_s\in\overline{V}_t}} \boldsymbol{x}_{a_s}\,\boldsymbol{x}_{a_s}^\top + \lambda\boldsymbol{I})^{-1}(\sum_{\substack{s\in[t-1]\\ i_s\in\overline{V}_t}} \boldsymbol{x}_{a_s} r_s) - \boldsymbol{\theta}_{i_t}\\
&= (\sum_{\substack{s\in[t-1]\\ i_s\in\overline{V}_t}} \boldsymbol{x}_{a_s}\,\boldsymbol{x}_{a_s}^\top + \lambda\boldsymbol{I})^{-1}\Big(\sum_{\substack{s\in[t-1]\\ i_s\in\overline{V}_t}} \boldsymbol{x}_{a_s}(\boldsymbol{x}_{a_s}^\top\boldsymbol{\theta}_{i_s} + \boldsymbol{\epsilon}_{a_s}^{i_s,s} + \eta_s)\Big) - \boldsymbol{\theta}_{i_t}\\
&= \overline{\boldsymbol{M}}^{-1}_{\overline{V}_t,t-1}\sum_{\substack{s\in[t-1]\\ i_s\in\overline{V}_t}} \boldsymbol{x}_{a_s}\boldsymbol{\epsilon}_{a_s}^{i_s,s} + \overline{\boldsymbol{M}}^{-1}_{\overline{V}_t,t-1}\sum_{\substack{s\in[t-1]\\ i_s\in\overline{V}_t}} \boldsymbol{x}_{a_s}\eta_s + \overline{\boldsymbol{M}}^{-1}_{\overline{V}_t,t-1}\sum_{\substack{s\in[t-1]\\ i_s\in\overline{V}_t}} \boldsymbol{x}_{a_s}\boldsymbol{x}_{a_s}^\top\boldsymbol{\theta}_{i_s} - \boldsymbol{\theta}_{i_t}\\
&= \overline{\boldsymbol{M}}^{-1}_{\overline{V}_t,t-1}\sum_{\substack{s\in[t-1]\\ i_s\in\overline{V}_t}} \boldsymbol{x}_{a_s}\boldsymbol{\epsilon}_{a_s}^{i_s,s} + \overline{\boldsymbol{M}}^{-1}_{\overline{V}_t,t-1}\sum_{\substack{s\in[t-1]\\ i_s\in\overline{V}_t}} \boldsymbol{x}_{a_s}\eta_s + \overline{\boldsymbol{M}}^{-1}_{\overline{V}_t,t-1}\sum_{\substack{s\in[t-1]\\ i_s\in\overline{V}_t}} \boldsymbol{x}_{a_s}\boldsymbol{x}_{a_s}^\top(\boldsymbol{\theta}_{i_s} - \boldsymbol{\theta}_{i_t})\\
&\quad + \overline{\boldsymbol{M}}^{-1}_{\overline{V}_t,t-1}(\sum_{\substack{s\in[t-1]\\ i_s\in\overline{V}_t}} \boldsymbol{x}_{a_s}\boldsymbol{x}_{a_s}^\top + \lambda\boldsymbol{I})\boldsymbol{\theta}_{i_t} - \lambda\overline{\boldsymbol{M}}^{-1}_{\overline{V}_t,t-1}\boldsymbol{\theta}_{i_t} - \boldsymbol{\theta}_{i_t}\\
&= \overline{\boldsymbol{M}}^{-1}_{\overline{V}_t,t-1}\sum_{\substack{s\in[t-1]\\ i_s\in\overline{V}_t}} \boldsymbol{x}_{a_s}\boldsymbol{\epsilon}_{a_s}^{i_s,s} + \overline{\boldsymbol{M}}^{-1}_{\overline{V}_t,t-1}\sum_{\substack{s\in[t-1]\\ i_s\in\overline{V}_t}} \boldsymbol{x}_{a_s}\eta_s + \overline{\boldsymbol{M}}^{-1}_{\overline{V}_t,t-1}\sum_{\substack{s\in[t-1]\\ i_s\in\overline{V}_t}} \boldsymbol{x}_{a_s}\boldsymbol{x}_{a_s}^\top(\boldsymbol{\theta}_{i_s} - \boldsymbol{\theta}_{i_t})\\
&\quad - \lambda\overline{\boldsymbol{M}}^{-1}_{\overline{V}_t,t-1}\boldsymbol{\theta}_{i_t}\,.
\end{aligned}$$

Thus, with Lemma 5.4.4 and with the previous reasoning, with

probability at least $1-5\delta$, we have:

$$\begin{aligned}
&\left|\boldsymbol{x}_a^\top(\hat{\boldsymbol{\theta}}_{\overline{V}_t,t-1}-\boldsymbol{\theta}_{i_t})\right| \\
&\leq \lambda\left|\boldsymbol{x}_a^\top\overline{\boldsymbol{M}}_{\overline{V}_t,t-1}^{-1}\boldsymbol{\theta}_{i_t}\right| + \left|\sum_{\substack{s\in[t-1]\\ i_s\in\overline{V}_t}}\boldsymbol{x}_a^\top\overline{\boldsymbol{M}}_{\overline{V}_t,t-1}^{-1}\boldsymbol{x}_{a_s}\boldsymbol{\epsilon}_{a_s}^{i_s,s}\right| + \left|\boldsymbol{x}_a^\top\overline{\boldsymbol{M}}_{\overline{V}_t,t-1}^{-1}\sum_{\substack{s\in[t-1]\\ i_s\in\overline{V}_t}}\boldsymbol{x}_{a_s}\eta_s\right| \\
&\quad + \left|\boldsymbol{x}_a^\top\overline{\boldsymbol{M}}_{\overline{V}_t,t-1}^{-1}\sum_{\substack{s\in[t-1]\\ i_s\in\overline{V}_t}}\boldsymbol{x}_{a_s}\boldsymbol{x}_{a_s}^\top(\boldsymbol{\theta}_{i_s}-\boldsymbol{\theta}_{i_t})\right| \\
&\leq \epsilon_*\sum_{\substack{s\in[t-1]\\ i_s\in\overline{V}_t}}\left|\boldsymbol{x}_a^\top\overline{\boldsymbol{M}}_{\overline{V}_t,t-1}^{-1}\boldsymbol{x}_{a_s}\right| + \|\boldsymbol{x}_a\|_{\overline{\boldsymbol{M}}_{\overline{V}_t,t-1}^{-1}}\left(\sqrt{\lambda}+\sqrt{2\log(\frac{1}{\delta})+d\log(1+\frac{T}{\lambda d})}\right) \\
&\quad + \frac{\epsilon_*\sqrt{2d}}{\tilde{\lambda}_x^{\frac{3}{2}}}\,.
\end{aligned}$$

Therefore, combining situation 1 and situation 2, the result of Lemma 6.4.2 then follows.

A.3.10 Technical Lemmas and Their Proofs

We first prove the following technical lemma which is used to prove Lemma A.3.2.

Lemma A.3.3. *Under Assumption 9.4, at any time t, for any fixed unit vector $\boldsymbol{\theta}\in\mathbb{R}^d$*

$$\mathbb{E}_t[(\boldsymbol{\theta}^\top\boldsymbol{x}_{a_t})^2|\,|\mathcal{A}_t|] \geq \tilde{\lambda}_x \triangleq \int_0^{\lambda_x}(1-e^{-\frac{(\lambda_x-x)^2}{2\sigma^2}})^C dx\,. \tag{A.145}$$

Proof. The proof of this lemma mainly follows the proof of Claim 1 in [46], but with more careful analysis, since their assumption is more stringent than ours.

Denote the feasible arms at round t by $\mathcal{A}_t=\{\boldsymbol{x}_{t,1},\boldsymbol{x}_{t,2},\ldots,\boldsymbol{x}_{t,|\mathcal{A}_t|}\}$. Consider the corresponding i.i.d. random variables $\theta_i=(\boldsymbol{\theta}^\top\boldsymbol{x}_{t,i})^2-$

$\mathbb{E}_t[(\boldsymbol{\theta}^\top \boldsymbol{x}_{t,i})^2 | \, |\mathcal{A}_t|], i = 1, 2, \ldots, |\mathcal{A}_t|$. By Assumption 9.4, θ_i s are sub-Gaussian random variables with variance bounded by σ^2. Therefore, we have that for any $\alpha > 0$ and any $i \in [|\mathcal{A}_t|]$:

$$\mathbb{P}_t(\theta_i < -\alpha | \, |\mathcal{A}_t|) \leq e^{-\frac{\alpha^2}{2\sigma^2}} ,$$

where $\mathbb{P}_t(\cdot)$ is the shorthand for the conditional probability $\mathbb{P}(\cdot | (i_1, \mathcal{A}_1, r_1), \ldots, (i_{t-1}, \mathcal{A}_{t-1}, r_{t-1}), i_t)$.

We also have that $\mathbb{E}_t[(\boldsymbol{\theta}^\top \boldsymbol{x}_{t,i})^2 | \, |\mathcal{A}_t| = \mathbb{E}_t[\boldsymbol{\theta}^\top \boldsymbol{x}_{t,i} \boldsymbol{x}_{t,i}^\top \boldsymbol{\theta} | \, |\mathcal{A}_t|] \geq \lambda_{\min}(\mathbb{E}_{\boldsymbol{x} \sim \rho}[\boldsymbol{x}\boldsymbol{x}^\top]) \geq \lambda_x$ by Assumption 9.4. With the above inequalities, we can get

$$\mathbb{P}_t(\min_{i=1,\ldots,|\mathcal{A}_t|} (\boldsymbol{\theta}^\top \boldsymbol{x}_{t,i})^2 \geq \lambda_x - \alpha | \, |\mathcal{A}_t|) \geq (1 - e^{-\frac{\alpha^2}{2\sigma^2}})^C ,$$

where C is the upper bound of $|\mathcal{A}_t|$.

Therefore, we have

$$\begin{aligned}
\mathbb{E}_t[(\boldsymbol{\theta}^\top \boldsymbol{x}_{a_t})^2 | \, |\mathcal{A}_t|] &\geq \mathbb{E}_t[\min_{i=1,\ldots,|\mathcal{A}_t|} (\boldsymbol{\theta}^\top \boldsymbol{x}_{t,i})^2 | \, |\mathcal{A}_t|] \\
&\geq \int_0^\infty \mathbb{P}_t(\min_{i=1,\ldots,|\mathcal{A}_t|} (\boldsymbol{\theta}^\top \boldsymbol{x}_{t,i})^2 \geq x | \, |\mathcal{A}_t|) dx \\
&\geq \int_0^{\lambda_x} (1 - e^{-\frac{(\lambda_x - x)^2}{2\sigma^2}})^C dx \triangleq \tilde{\lambda}_x
\end{aligned}$$

□

Finally, we prove the following lemma which is used in the proof of Theorem 6.4.3.

Lemma A.3.4.

$$\sum_{t=T_0+1}^{T} \min\{. \, \|\boldsymbol{x}_{a_t}\|^2_{\boldsymbol{M}^{-1}_{V_j,t-1}}, 1\} \leq 2d \log(1 + \frac{T}{\lambda d}), \forall j \in [m] . \quad \text{(A.146)}$$

Proof.

$$\begin{aligned}
\det(\overline{\boldsymbol{M}}_{V_j,T}) &= \det\left(\overline{\boldsymbol{M}}_{V_j,T-1} + \mathbb{I}\{i_T \in V_j\}\boldsymbol{x}_{a_T}\boldsymbol{x}_{a_T}^\top\right) \\
&= \det(\overline{\boldsymbol{M}}_{V_j,T-1})\det\left(\boldsymbol{I} + \mathbb{I}\{i_T \in V_j\}\overline{\boldsymbol{M}}_{V_j,T-1}^{-\frac{1}{2}}\boldsymbol{x}_{a_T}\boldsymbol{x}_{a_T}^\top\overline{\boldsymbol{M}}_{V_j,T-1}^{-\frac{1}{2}}\right) \\
&= \det(\overline{\boldsymbol{M}}_{V_j,T-1})\left(1 + \mathbb{I}\{i_T \in V_j\}\left\|\boldsymbol{x}_{a_T}\right\|^2_{\overline{\boldsymbol{M}}^{-1}_{V_j,T-1}}\right) \\
&= \det(\overline{\boldsymbol{M}}_{V_j,T_0})\prod_{t=T_0+1}^{T}\left(1 + \mathbb{I}\{i_t \in V_j\}\left\|\boldsymbol{x}_{a_t}\right\|^2_{\overline{\boldsymbol{M}}^{-1}_{V_j,t-1}}\right) \\
&\geq \det(\lambda\boldsymbol{I})\prod_{t=T_0+1}^{T}\left(1 + \mathbb{I}\{i_t \in V_j\}\left\|\boldsymbol{x}_{a_t}\right\|^2_{\overline{\boldsymbol{M}}^{-1}_{V_j,t-1}}\right).
\end{aligned} \tag{A.147}$$

$\forall x \in [0,1]$, we have $x \leq 2\log(1+x)$. Therefore

$$\begin{aligned}
&\sum_{t=T_0+1}^{T}\min\{\mathbb{I}\{i_t \in V_j\}\left\|\boldsymbol{x}_{a_t}\right\|^2_{\overline{\boldsymbol{M}}^{-1}_{V_j,t-1}}, 1\} \\
&\leq 2\sum_{t=T_0+1}^{T}\log\left(1 + \mathbb{I}\{i_t \in V_j\}\left\|\boldsymbol{x}_{a_t}\right\|^2_{\overline{\boldsymbol{M}}^{-1}_{V_j,t-1}}\right) \\
&= 2\log\left(\prod_{t=T_0+1}^{T}\left(1 + \mathbb{I}\{i_t \in V_j\}\left\|\boldsymbol{x}_{a_t}\right\|^2_{\overline{\boldsymbol{M}}^{-1}_{V_j,t-1}}\right)\right) \\
&\leq 2[\log(\det(\overline{\boldsymbol{M}}_{V_j,T})) - \log(\det(\lambda\boldsymbol{I}))] \\
&\leq 2\log\left(\frac{\mathrm{trace}(\lambda\boldsymbol{I} + \sum_{t=1}^{T}\mathbb{I}\{i_t \in V_j\}\boldsymbol{x}_{a_t}\boldsymbol{x}_{a_t}^\top)}{\lambda d}\right)^d \\
&\leq 2d\log(1 + \frac{T}{\lambda d}).
\end{aligned} \tag{A.148}$$

□

A.3.11 Algorithms of RSCLUMB

This section introduces the Robust Set-based Clustering of Misspecified Bandits Algorithm (RSCLUMB). Unlike RCLUMB, which maintains a graph-based clustering structure, RSCLUMB maintains a set-based clustering structure. Besides, RCLUMB only splits clusters during the learning process, while RSCLUMB allows both split and merge operations. A brief illustration is that the agent will split a user out of its current set(cluster) if it finds an inconsistency between the user and its set, and if there are two clusters whose estimated preferences are close enough, the agent will merge them. A detailed discussion of the connection between the graph structure and the set structure can be found in [136].

Now we introduce the details of RSCLUMB. The algorithm first initializes a single set $\boldsymbol{S}_1$ containing all users and updates it during the learning process. The whole learning process consists of phases (Algo. 15 Line 3), where the $s-th$ phase contains 2^s rounds. At the beginning of each phase, the agent marks all users as "unchecked", and if a user comes later, it will be marked as "checked". If all users in a cluster are checked, then this cluster will be marked as "checked" meaning it is an accurate cluster in the current phase. With this mechanism, every phase can maintain an accuracy level, and the agent can put the accurate clusters aside and focus on exploring the inaccurate ones. For each cluster V_j, the algorithm maintains two estimated vectors $\hat{\boldsymbol{\theta}}_{V_j}$ and $\tilde{\boldsymbol{\theta}}_{V_j}$, where the $\hat{\boldsymbol{\theta}}_{V_j}$ is similar to the $\hat{\boldsymbol{\theta}}_{\overline{V}_j}$ in RCLUMB and is used for the recommendation, while the $\tilde{\boldsymbol{\theta}}_{V_j}$ is the average of all the estimated user preference vectors in this cluster and is used

for the split and merge operations.

At time t in phase s, the user i_τ comes with the item set $\mathcal{D}_\tau$, where τ represents the index of total time steps. Then the algorithm determines the cluster and makes a cluster-based recommendation. This process is similar to RCLUMB. After updating the information (Algo. 15 Line12), the agent checks if a split or a merge is possible (Algo. 15 Line13-17).

By our assumption, users in the same cluster have the same vectors. So a cluster can be regarded as a good cluster only when all the estimated user vectors are close to the estimated cluster vector. We call a user is consistent with the cluster if their estimated vectors are close enough. If a user is inconsistent with its current cluster, the agent will split it out. Two clusters are consistent when their estimated vectors are close, and the agent will merge them.

RSCLUMB maintains two sets of estimated cluster vectors: (i) cluster-level estimation with integrated user information, which is for recommendations (Line 12 and Line 10 in Algo.15); (ii) the average of estimated user vectors, which is used for robust clustering (Line 3 in Algo.16 and Line 2 in Algo.17). The previous set-based CB work [136] only uses (i) for both recommendations and clustering, which would lead to erroneous clustering under misspecifications, and cannot get any non-vacuous regret bound in CBMUM.

A.3.12 Main Theorem and Lemmas of RSCLUMB

Theorem A.3.5 (main result on regret bound for RSCLUMB)**.** *With the same assumptions in Theorem 6.4.3, the expected regret*

of the RSCLUMB algorithm for T rounds satisfies:

$$R(T) \leq O\bigg(u\left(\frac{d}{\tilde{\lambda}_x(\gamma_1-\zeta_1)^2}+\frac{1}{\tilde{\lambda}_x^2}\right)\log T+\frac{\epsilon_*\sqrt{d}T}{\tilde{\lambda}_x^{1.5}}+\epsilon_* T\sqrt{md\log T}+d\sqrt{mT}\log T+\epsilon_*\sqrt{\frac{1}{\tilde{\lambda}_x}}T\bigg) \tag{A.149}$$

$$\leq O(\epsilon_* T\sqrt{md\log T}+d\sqrt{mT}\log T) \tag{A.150}$$

Lemma A.3.6. *For RSCLUMB, we use T_1 to represent the corresponding T_0 of RCLUMB. Then :*

$$T_1 \triangleq 16u\log(\frac{u}{\delta})+4u\max\{\frac{16}{\tilde{\lambda}_x^2}\log(\frac{8d}{\tilde{\lambda}_x^2\delta}),\frac{8d}{\tilde{\lambda}_x(\frac{\gamma_1}{6}-\epsilon_*\sqrt{\frac{1}{2\tilde{\lambda}_x}})^2}\log(\frac{u}{\delta})\}$$

$$= O\bigg(u\left(\frac{d}{\tilde{\lambda}_x(\gamma_1-\zeta_1)^2}+\frac{1}{\tilde{\lambda}_x^2}\right)\log\frac{1}{\delta}\bigg)$$

Lemma A.3.7. *For RSCLUMB, after $2T_1+1$ rounds: in each phase, after the first u rounds, with probability at least $1-5\delta$:*

$$\left|\boldsymbol{x}_a^\top(\boldsymbol{\theta}_{i_t}-\hat{\boldsymbol{\theta}}_{\overline{V}_t,t-1})\right|$$
$$\leq (\frac{3\epsilon_*\sqrt{2d}}{2\tilde{\lambda}_x^{\frac{3}{2}}}+6\epsilon_*\sqrt{\frac{1}{2\tilde{\lambda}_x}})\mathbb{I}\{\overline{V}_t\notin V\}+\beta\left\|\boldsymbol{x}_a\right\|_{\overline{\boldsymbol{M}}^{-1}_{\overline{V}_t,t-1}}+\epsilon_*\sum_{\substack{s\in[t-1]\\ i_s\in\overline{V}_t}}\left|\boldsymbol{x}_a^\top\overline{\boldsymbol{M}}^{-1}_{\overline{V}_t,t-1}\boldsymbol{x}_{a_s}\right|$$
$$\triangleq (\frac{3\epsilon_*\sqrt{2d}}{2\tilde{\lambda}_x^{\frac{3}{2}}}+6\epsilon_*\sqrt{\frac{1}{2\tilde{\lambda}_x}})\mathbb{I}\{\overline{V}_t\notin V\}+C_{a,t}$$

A.3.13 Proof of Lemma A.3.7

$$\begin{aligned}|\boldsymbol{x}_a^{\mathrm{T}}(\boldsymbol{\theta}_i-\hat{\boldsymbol{\theta}}_{\overline{V}_t,t})| &= |\boldsymbol{x}_a^{\mathrm{T}}(\boldsymbol{\theta}_i-\boldsymbol{\theta}_{V_t})|+|\boldsymbol{x}_a^{\mathrm{T}}(\hat{\boldsymbol{\theta}}_{\overline{V}_t,t}-\boldsymbol{\theta}_{V_t})|\\ &\leq \left\|\boldsymbol{x}_a^{\mathrm{T}}\right\|\left\|\boldsymbol{\theta}_i-\boldsymbol{\theta}_{V_t}\right\|+|\boldsymbol{x}_a^{\mathrm{T}}(\hat{\boldsymbol{\theta}}_{\overline{V}_t,t}-\boldsymbol{\theta}_{V_t})|\\ &\leq 6\epsilon_*\sqrt{\frac{1}{2\tilde{\lambda}_x}}+|\boldsymbol{x}_a^{\mathrm{T}}(\hat{\boldsymbol{\theta}}_{\overline{V}_t,t}-\boldsymbol{\theta}_{V_t})|\end{aligned} \tag{A.151}$$

where the last inequality holds due to the fact $\|\boldsymbol{x}_a\| \leq 1$ and the condition of "split" and "merge". For $|\boldsymbol{x}_a^{\mathrm{T}}(\hat{\boldsymbol{\theta}}_{\overline{V}_t,t} - \boldsymbol{\theta}_{V_t})|$:

$$
\begin{aligned}
&\hat{\boldsymbol{\theta}}_{\overline{V}_t,t-1} - \boldsymbol{\theta}_{V_t} \\
&= (\sum_{\substack{s\in[t-1]\\ i_s\in\overline{V}_t}} \boldsymbol{x}_{a_s}\, \boldsymbol{x}_{a_s}^{\top} + \lambda \boldsymbol{I})^{-1} (\sum_{\substack{s\in[t-1]\\ i_s\in\overline{V}_t}} \boldsymbol{x}_{a_s} r_s) - \boldsymbol{\theta}_{V_t} \\
&= (\sum_{\substack{s\in[t-1]\\ i_s\in\overline{V}_t}} \boldsymbol{x}_{a_s}\, \boldsymbol{x}_{a_s}^{\top} + \lambda \boldsymbol{I})^{-1} \left(\sum_{\substack{s\in[t-1]\\ i_s\in\overline{V}_t}} \boldsymbol{x}_{a_s} (\boldsymbol{x}_{a_s}^{\top} \boldsymbol{\theta}_{i_s} + \boldsymbol{\epsilon}_{a_s}^{i_s,s} + \eta_s) \right) - \boldsymbol{\theta}_{V_t} \\
&= \overline{\boldsymbol{M}}_{\overline{V}_t,t-1}^{-1} \sum_{\substack{s\in[t-1]\\ i_s\in\overline{V}_t}} \boldsymbol{x}_{a_s} \boldsymbol{\epsilon}_{a_s}^{i_s,s} + \overline{\boldsymbol{M}}_{\overline{V}_t,t-1}^{-1} \sum_{\substack{s\in[t-1]\\ i_s\in\overline{V}_t}} \boldsymbol{x}_{a_s} \eta_s + \overline{\boldsymbol{M}}_{\overline{V}_t,t-1}^{-1} \sum_{\substack{s\in[t-1]\\ i_s\in\overline{V}_t}} \boldsymbol{x}_{a_s} \boldsymbol{x}_{a_s}^{\top} \boldsymbol{\theta}_{i_s} - \boldsymbol{\theta}_{V_t} \\
&= \overline{\boldsymbol{M}}_{\overline{V}_t,t-1}^{-1} \sum_{\substack{s\in[t-1]\\ i_s\in\overline{V}_t}} \boldsymbol{x}_{a_s} \boldsymbol{\epsilon}_{a_s}^{i_s,s} + \overline{\boldsymbol{M}}_{\overline{V}_t,t-1}^{-1} \sum_{\substack{s\in[t-1]\\ i_s\in\overline{V}_t}} \boldsymbol{x}_{a_s} \eta_s + \overline{\boldsymbol{M}}_{\overline{V}_t,t-1}^{-1} \sum_{\substack{s\in[t-1]\\ i_s\in\overline{V}_t}} \boldsymbol{x}_{a_s} \boldsymbol{x}_{a_s}^{\top} (\boldsymbol{\theta}_{i_s} - \boldsymbol{\theta}_{V_t}) \\
&\quad + \overline{\boldsymbol{M}}_{\overline{V}_t,t-1}^{-1} (\sum_{\substack{s\in[t-1]\\ i_s\in\overline{V}_t}} \boldsymbol{x}_{a_s} \boldsymbol{x}_{a_s}^{\top} + \lambda \boldsymbol{I}) \boldsymbol{\theta}_{V_t} - \lambda \overline{\boldsymbol{M}}_{\overline{V}_t,t-1}^{-1} \boldsymbol{\theta}_{V_t} - \boldsymbol{\theta}_{V_t} \\
&= \overline{\boldsymbol{M}}_{\overline{V}_t,t-1}^{-1} \sum_{\substack{s\in[t-1]\\ i_s\in\overline{V}_t}} \boldsymbol{x}_{a_s} \boldsymbol{\epsilon}_{a_s}^{i_s,s} + \overline{\boldsymbol{M}}_{\overline{V}_t,t-1}^{-1} \sum_{\substack{s\in[t-1]\\ i_s\in\overline{V}_t}} \boldsymbol{x}_{a_s} \eta_s + \overline{\boldsymbol{M}}_{\overline{V}_t,t-1}^{-1} \sum_{\substack{s\in[t-1]\\ i_s\in\overline{V}_t}} \boldsymbol{x}_{a_s} \boldsymbol{x}_{a_s}^{\top} (\boldsymbol{\theta}_{i_s} - \boldsymbol{\theta}_{V_t}) \\
&\quad - \lambda \overline{\boldsymbol{M}}_{\overline{V}_t,t-1}^{-1} \boldsymbol{\theta}_{V_t} \,.
\end{aligned}
$$

Thus, with the same method in Lemma 5.4.4 but replace $\zeta = 4\epsilon_* \sqrt{\frac{1}{2\tilde{\lambda}_x}}$ with $\zeta_1 = 6\epsilon_* \sqrt{\frac{1}{2\tilde{\lambda}_x}}$, and with the previous reasoning, with probability at least $1 - 5\delta$, we have:

$$
|\boldsymbol{x}_a^{\mathrm{T}} (\hat{\boldsymbol{\theta}}_{\overline{V}_t,t} - \boldsymbol{\theta}_{V_t})| \leq C_{a_t} + \frac{3\epsilon_* \sqrt{2d}}{2\tilde{\lambda}_x^{\frac{3}{2}}} \tag{A.152}
$$

The lemma can be concluded.

A.3.14 Proof of Lemma A.3.6

With the analysis in the proof of Lemma A.3.2, with probability at least $1-\delta$:

$$\left\|\hat{\boldsymbol{\theta}}_{i,t}-\boldsymbol{\theta}^{j(i)}\right\|_2 \leq \frac{\beta(T_{i,t},\frac{\delta}{u})+\epsilon_*\sqrt{T_{i,t}}}{\sqrt{\lambda+\lambda_{\min}(\boldsymbol{M}_{i,t})}}, \forall i \in \mathcal{U}\,, \tag{A.153}$$

and the estimated error of the current cluster $\left\|\tilde{\boldsymbol{\theta}}^{j(i)}-\boldsymbol{\theta}^{j(i)}\right\|$ also satisfies this inequality. For set-based clustering structure, to ensure for each user there is only one ζ-close cluster, we let:

$$\frac{\beta(T_{i,t},\frac{\delta}{u})+\epsilon_*\sqrt{T_{i,t}}}{\sqrt{\lambda+\lambda_{\min}(\boldsymbol{M}_{i,t})}} \leq \frac{\gamma_1}{6} \tag{A.154}$$

By assuming $\lambda < 2\log(\frac{u}{\delta})+d\log(1+\frac{T_{i,t}}{\lambda d})$, we can simplify it to

$$\frac{2\log(\frac{u}{\delta})+d\log(1+\frac{T_{i,t}}{\lambda d})}{2\tilde{\lambda}_x T_{i,t}} < \frac{1}{4}(\frac{\gamma_1}{6}-\epsilon_*\sqrt{\frac{1}{2\tilde{\lambda}_x}})^2 \tag{A.155}$$

which can be proved by $\frac{2\log(\frac{u}{\delta})}{2\tilde{\lambda}_x T_{i,t}} \leq \frac{1}{8}(\frac{\gamma_1}{6}-\epsilon_*\sqrt{\frac{1}{2\tilde{\lambda}_x}})^2$ and $\frac{d\log(1+\frac{T_{i,t}}{\lambda d})}{2\tilde{\lambda}_x T_{i,t}} \leq \frac{1}{8}(\frac{\gamma_1}{6}-\epsilon_*\sqrt{\frac{1}{2\tilde{\lambda}_x}})^2$. It's obvious that the former one can be satisfied by $T_{i,t} \geq \frac{8\log(u/\delta)}{\tilde{\lambda}_x(\frac{\gamma_1}{6}-\epsilon_*\sqrt{1/2\tilde{\lambda}_x})^2}$. As for the latter one, by [135] Lemma 9, we can get $T_{i,t} \geq \frac{8d\log(\frac{16}{\tilde{\lambda}_x\lambda(\frac{\gamma_1}{6}-\epsilon_*\sqrt{1/2\tilde{\lambda}_x})^2})}{4\tilde{\lambda}_x(\frac{\gamma_1}{6}-\epsilon_*\sqrt{1/2\tilde{\lambda}_x})^2}$. By assuming $\frac{u}{\delta} \geq \frac{16}{4\tilde{\lambda}_x\lambda(\frac{\gamma_1}{6}-\epsilon_*\sqrt{2/4\tilde{\lambda}_x})^2}$, the lemma is proved.

A.3.15 Proof of Theorem A.3.5

After $2T_1$ rounds,in each phase, at most u times split operations will happen, we use $u\log(T)$ to bound the regret generated in these rounds. Then in the remained rounds the cluster num will

be no more than m.

For the instantaneous regret R_t at round t, with probability at least $1-2\delta$ for some $\delta \in (0, \frac{1}{2})$:

$$\begin{aligned} R_t &= (\boldsymbol{x}_{a_t^*}^{\mathrm{T}} \boldsymbol{\theta}_{i_t} + \boldsymbol{\epsilon}_{a_t^*}^{i_t,t}) - (\boldsymbol{x}_{a_t}^{\mathrm{T}} \boldsymbol{\theta}_{i_t} + \boldsymbol{\epsilon}_{a_t}^{i_t,t}) \\ &= \boldsymbol{x}_{a_t^*}^{\top}(\boldsymbol{\theta}_{i_t} - \hat{\boldsymbol{\theta}}_{\overline{V}_t,t-1}) + (\boldsymbol{x}_{a_t^*}^{\top} \hat{\boldsymbol{\theta}}_{\overline{V}_t,t-1} + C_{a_t^*,t}) - (\boldsymbol{x}_{a_t}^{\top} \hat{\boldsymbol{\theta}}_{\overline{V}_t,t-1} + C_{a_t,t}) \\ &\quad + \boldsymbol{x}_{a_t}^{\top}(\hat{\boldsymbol{\theta}}_{\overline{V}_t,t-1} - \boldsymbol{\theta}_{i_t}) + C_{a_t,t} - C_{a_t^*,t} + (\boldsymbol{\epsilon}_{a_t^*}^{i_t,t} - \boldsymbol{\epsilon}_{a_t}^{i_t,t}) \\ &\leq 2C_{a_t} + 2\epsilon_* + (12\epsilon_* \sqrt{\frac{1}{2\tilde{\lambda}_x}} + \frac{3\epsilon_* \sqrt{2d}}{\tilde{\lambda}_x^{\frac{3}{2}}}) \mathbb{I}(\overline{V}_t \notin V) \end{aligned} \tag{A.156}$$

where the last inequality holds due to the UCB arm selection strategy, the concentration bound given in LemmaA.3.7 and the fact that $\left\|\epsilon^{i,t}\right\|_\infty \leq \epsilon_*$.

Define such events. Let:

$$\begin{aligned} \mathcal{E}_2 = \{&\text{All clusters } \overline{V}_t \text{ only contain users who satisfy} \\ &\left\|\tilde{\boldsymbol{\theta}}_i - \tilde{\boldsymbol{\theta}}_{\overline{V}_t}\right\| \leq \alpha_1 (\sqrt{\frac{1+\log(1+T_{i,t})}{1+T_{i,t}}} + \sqrt{\frac{1+\log(1+T_{\overline{V}_t,t})}{1+T_{\overline{V}_t,t}}}) + \alpha_2 \epsilon_* \} \end{aligned} \tag{A.157}$$

$$\mathcal{E}_3 = \{r_t \leq 2C_{a_t} + 2\epsilon_* + 12\epsilon_* \sqrt{\frac{1}{2\tilde{\lambda}_x}} + \frac{3\epsilon_* \sqrt{2d}}{\tilde{\lambda}_x^{\frac{3}{2}}}\}$$

$$\mathcal{E}' = \mathcal{E}_2 \cap \mathcal{E}_3$$

From previous analysis, we can know that $\mathbb{P}(\mathcal{E}_2) \geq 1-3\delta$ and $\mathbb{P}(\mathcal{E}_3) \geq 1-2\delta$, thus $\mathbb{P}(\mathcal{E}' \geq 1-5\delta)$. By taking $\delta = \frac{1}{T}$, we can

get:

$$\begin{aligned} E(R_t) &= P(\mathcal{E})\mathbb{I}\{\mathcal{E}\}R_t + P(\bar{\mathcal{E}})\mathbb{I}\{\bar{\mathcal{E}}\}R_t \\ &\leq \mathbb{I}\{\mathcal{E}\}R_t + 5 \\ &\leq 2T_1 + 2\epsilon_* T + (12\epsilon_*\sqrt{\frac{1}{2\tilde{\lambda}_x}} + \frac{3\epsilon_*\sqrt{2d}}{\tilde{\lambda}_x^{\frac{3}{2}}})T + 2\sum_{2T_1}^{T} C_{a_t} + 5 \end{aligned} \tag{A.158}$$

Now we need to bound $2\sum_{2T_1}^{T} C_{a_t}$. We already know that after $2T_1$ rounds, in each phase k after the first u rounds,there will be at most m clusters

Consider phase k, for simplicity, ignore the fist u rounds. For the first term in C_{a_t}:

$$\begin{aligned} \sum_{t=T_{k-1}}^{T_k} \|\boldsymbol{x}_{a_t}\|^{-1}_{\overline{\boldsymbol{M}}_{\overline{V}_t,t-1}} &= \sum_{t=T_{k-1}}^{T_k}\sum_{j=1}^{m_t} \mathbb{I}\{i \in \overline{V}_{t,j}\} \|\boldsymbol{x}_{a_t}\|_{\overline{\boldsymbol{M}}^{-1}_{\overline{V}_{t,j}}} \\ &\leq \sum_{j=1}^{m_t} \sqrt{\sum_{t=T_{k-1}}^{T_k} \mathbb{I}\{i \in V_{t,j}\} \sum_{t=T_{k-1}}^{T_k} \mathbb{I}\{i \in V_{t,j}\} \|\boldsymbol{x}_{a_t}\|^2_{\boldsymbol{M}^{-1}_{\overline{V}_{t,j}}}} \\ &\leq \sum_{j=1}^{m_t} \sqrt{2T_{k,j} d \log(1 + \frac{T}{\lambda d})} \\ &\leq \sqrt{2m(T_k - T_{k-1}) d \log(1 + \frac{T}{\lambda d})} \end{aligned} \tag{A.159}$$

For all phases:

$$\begin{aligned} \sum_{k=1}^{s} \sqrt{2m(T_{k+1} - T_k) d \log(1 + \frac{T}{\lambda d})} &\leq \sqrt{2\sum_{k=1}^{s} 1 \sum_{k=1}^{s} (T_{k+1} - T_k) md \log(1 + \frac{T}{\lambda d})} \\ &\leq \sqrt{2mdT \log(T) \log(1 + \frac{T}{\lambda d})} \end{aligned} \tag{A.160}$$

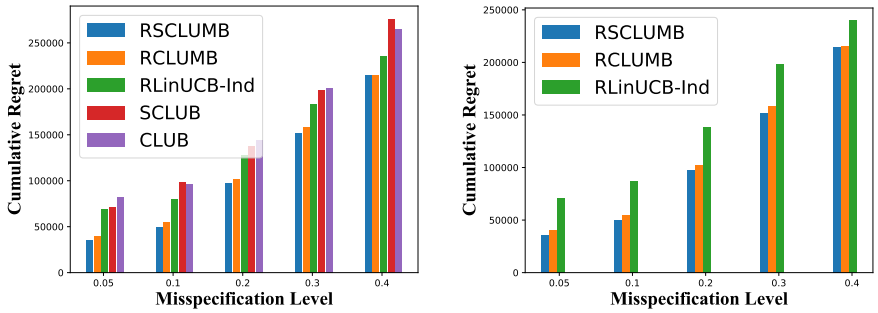
Similarly, for the second term in C_{a_t}:

$$\begin{aligned}
&\sum_{t=T_{k-1}}^{T_k} \sum_{\substack{s\in[t-1]\\ i_s\in\overline{V}_t}} \epsilon_* |\boldsymbol{x}_{a_t}^{\mathrm{T}} \overline{\boldsymbol{M}}_{\overline{V}_{t,t-1}}^{-1} \boldsymbol{x}_{a_s}| \\
&= \sum_{t=T_{k-1}}^{T_k} \sum_{j=1}^{m_t} \mathbb{I}\{i \in \overline{V}_{t,j}\} \sum_{\substack{s\in[t-1]\\ i_s\in\overline{V}_{t,j}}} \epsilon_* |\boldsymbol{x}_{a_t}^{\mathrm{T}} \overline{\boldsymbol{M}}_{\overline{V}_{t,j}^{-1}} \boldsymbol{x}_{a_s}| \\
&\le \epsilon_* \sum_{t=T_{k-1}}^{T_k} \sum_{j=1}^{m_t} \mathbb{I}\{i \in \overline{V}_{t,j}\} \sqrt{\sum_{\substack{s\in[t-1]\\ i_s\in\overline{V}_{t,j}}} 1 \sum_{\substack{s\in[t-1]\\ i_s\in\overline{V}_{t,j}}} |\boldsymbol{x}_{a_t}^T \overline{\boldsymbol{M}}_{\overline{V}_{t,j}^{-1}} \boldsymbol{x}_{a_s}|^2} \\
&\le \epsilon_* \sum_{t=T_{k-1}}^{T_k} \sum_{j=1}^{m_t} \mathbb{I}\{i \in \overline{V}_{t,j}\} \sqrt{T_{k,j} \left\|\boldsymbol{x}_{a_t}\right\|^2_{\overline{\boldsymbol{M}}^{-1}_{\overline{V}_{t,j}}}} \\
&\le \epsilon_* \sum_{t=T_{k-1}}^{T_k} \sqrt{\sum_{j=1}^{m_t} \mathbb{I}\{i \in \overline{V}_{t,j}\} \sum_{j=1}^{m_t} \mathbb{I}\{i \in \overline{V}_{t,j}\} T_{k,j} \left\|\boldsymbol{x}_{a_t}\right\|^2_{\overline{\boldsymbol{M}}^{-1}_{\overline{V}_{t,j}}}} \\
&\le \epsilon_* \sqrt{(T_k - T_{k-1})} \sum_{t=T_{k-1}}^{T_k} \sqrt{\sum_{j=1}^{m_t} \mathbb{I}\{i \in \overline{V}_{t,j}\} \left\|\boldsymbol{x}_{a_t}\right\|^2_{\overline{\boldsymbol{M}}^{-1}_{\overline{V}_{t,j}}}} \\
&\le \epsilon_* (T_k - T_{k-1}) \sqrt{2md \log(1 + \frac{T}{\lambda d})}
\end{aligned}$$

Then for all phases this term can be bounded by $\epsilon_* T \sqrt{2md \log(1 + \frac{T}{\lambda d})}$. Thus the total regret can be bounded by:

$$\begin{aligned}
R_t \le\ & 2\sqrt{2mTd \log(T) \log(1 + \frac{T}{\lambda d})} (\sqrt{2\log(T) + d\log(1 + \frac{T}{\lambda d})} + 2\sqrt{\lambda}) \\
& + 2\epsilon_* T \sqrt{2md \log(1 + \frac{T}{\lambda d})} + 2\epsilon_* T + 12\epsilon_* \sqrt{\frac{1}{2\tilde{\lambda}_x}} T + \frac{3\epsilon_* \sqrt{2d}}{\tilde{\lambda}_x^{\frac{3}{2}}} T + 2T_1 + u\log(T) + 5
\end{aligned}$$

$$\text{where } T_1 = 16u \log(\tfrac{u}{\delta}) + 4u \max\{\tfrac{16}{\tilde{\lambda}_x^2} \log(\tfrac{8d}{\tilde{\lambda}_x^2 \delta}), \tfrac{8d}{\tilde{\lambda}_x (\frac{\gamma_1}{6} - \epsilon_* \sqrt{\frac{1}{2\tilde{\lambda}_x}})^2} \log(\tfrac{u}{\delta})\}$$



(a) Known Misspecification Level (b) Unknown Misspecification Level

Figure A.1: The cumulative regret of the algorithms under different scales of misspecification level.

A.3.16 More Experiments

For ablation study, we test our algorithms' performance under different scales of deviation. We test RCLUMB and RSCLUMB when $\epsilon^* = 0.05, 0.1, 0.2, 0.3$ and 0.4 in both misspecification level known and unknown cases. In the known case, we set ϵ^* according to the real misspecification level, and we compare our algorithms' performance to the baselines except LinUCB and CW-OFUL which perform worst; in the unknown case, we keep $\epsilon^* = 0.2$, and we compare our algorithms to RLinUCB-Ind as only it has the pre-spicified parameter ϵ^* among the baselines. The results are shown in Fig.A.1. We plot each algorithm's final cumulative regret under different misspecification levels. All the algorithms' performances get worse when the deviation gets larger, and our two algorithms always perform better than the baselines. Besides, the regrets in the unknown case are only slightly larger than the known case. These results can match our theoretical results and again show our algorithms' effectiveness, as well as verify that our algorithm can handle the unknown misspecification level.

Algorithm 15 Robust Set-based Clustering of Misspecified Bandits Algorithm (RSCLUMB)

Input: Deletion parameter $\alpha_1, \alpha_2 > 0$, $f(T) = \sqrt{\frac{1+\ln(1+T)}{1+T}}$, $\lambda, \beta, \epsilon_* > 0$.

Initialization:

- $\boldsymbol{M}_{i,0} = 0_{d\times d}, \boldsymbol{b}_{i,0} = 0_{d\times 1}, T_{i,0} = 0$, $\forall i \in \mathcal{U}$;
- Initialize the set of cluster indexes by $J = \{1\}$ and the single cluster $\boldsymbol{S}_1$ by $\boldsymbol{M}_1 = 0_{d\times d}$, $\boldsymbol{b}_1 = 0_{d\times 1}$, $T_1 = 0$, $C_1 = \mathcal{U}$, $j(i) = 1$, $\forall i$.

for all $s = 1, 2, \ldots$ **do**
  Mark every user unchecked for each cluster.
  For each cluster V_j, compute $\tilde{T}_{V_j} = T_{V_j}$, $\hat{\boldsymbol{\theta}}_{V_j} = (\lambda \boldsymbol{I} + \boldsymbol{M}_{V_j})^{-1}\boldsymbol{b}_{V_j}$, $\tilde{\boldsymbol{\theta}}_{V_j} = \frac{\sum_{i\in V_j}\hat{\boldsymbol{\theta}}_i}{[V_j]}$
  for all $t = 1, 2, \ldots, T$ **do**
    Compute $\tau = 2^s - 2 + t$
    Receive the user i_τ and the decision set $\mathcal{D}_\tau$
    Determine the cluster index $j = j(i_\tau)$
    Recommend item a_τ with the largest UCB index as shown in Eq. (6.3)
    Received the feedback r_τ.
    Update the information:

$$\begin{aligned}
\boldsymbol{M}_{i_\tau,\tau} &= \boldsymbol{M}_{i_\tau,\tau-1} + \boldsymbol{x}_{a_\tau}\boldsymbol{x}_{a_\tau}^{\mathrm{T}}, \boldsymbol{b}_{i_\tau,\tau} = \boldsymbol{b}_{i_\tau,\tau-1} + r_\tau \boldsymbol{x}_{a_\tau},\\
T_{i_\tau,\tau} &= T_{i_\tau,\tau-1} + 1, \hat{\boldsymbol{\theta}}_{i_\tau,\tau} = (\lambda \boldsymbol{I} + \boldsymbol{M}_{i_\tau,\tau})^{-1}\boldsymbol{b}_{i_\tau,\tau}\\
\boldsymbol{M}_{V_j,\tau} &= \boldsymbol{M}_{V_j,\tau-1} + \boldsymbol{x}_{a_\tau}\boldsymbol{x}_{a_\tau}^{\mathrm{T}}, \boldsymbol{b}_{V_j,\tau} = \boldsymbol{b}_{V_j,\tau-1} + r_\tau \boldsymbol{x}_\tau,\\
T_{V_j,\tau} &= T_{V_j,\tau-1} + 1, \hat{\boldsymbol{\theta}}_{V_j,\tau} = (\lambda \boldsymbol{I} + \boldsymbol{M}_{V_j,\tau})^{-1}\boldsymbol{b}_{V_j,\tau},\\
\tilde{\boldsymbol{\theta}}_{V_j,\tau} &= \frac{\sum_{i\in V_j}\hat{\boldsymbol{\theta}}_i, \tau}{[V_j]}
\end{aligned}$$

    if i_τ is unchecked **then**
      Run **Split**
      Mark user i_τ has been checked
      Run **Merge**
    end if
  end for
end for

Algorithm 16 Split

1: Define $F(T) = \sqrt{\frac{1+\ln(1+T)}{1+T}}$

2: **if** $\left\|\hat{\boldsymbol{\theta}}_{i_\tau,\tau} - \tilde{\boldsymbol{\theta}}_{V_j,\tau}\right\| > \alpha_1(F(T_{i_\tau,\tau}) + F(T_{V_j,\tau})) + \alpha_2\epsilon_*$ **then**

3: Split user i_τ from cluster V_j and form a new cluster $V_j^{'}$ of user i_τ

$$\begin{aligned}
\boldsymbol{M}_{V_j,\tau} &= \boldsymbol{M}_{V_j,\tau} - \boldsymbol{M}_{i_\tau,\tau}, \boldsymbol{b}_{V_j} = \boldsymbol{b}_{V_j} - \boldsymbol{b}_{i_\tau,\tau}, \\
T_{V_j,\tau} &= T_{V_j,\tau} - T_{i_\tau,\tau}, C_{j,\tau} = C_{j,\tau} - \{i_\tau\}, \\
\boldsymbol{M}_{V_j',\tau} &= \boldsymbol{M}_{i_\tau,\tau}, \boldsymbol{b}_{V_j',\tau} = \boldsymbol{b}_{i_\tau,\tau}, \\
T_{V_j',\tau} &= T_{i_\tau,\tau}, C_{j',\tau} = \{i_\tau\}
\end{aligned}$$

4: **end if**

Algorithm 17 Merge

1: **for** any two checked clustersV_{j_1}, V_{j_2} satisfying

$$\left\|\tilde{\boldsymbol{\theta}}_{j_1} - \tilde{\boldsymbol{\theta}}_{j_2}\right\| < \frac{\alpha_1}{2}(F(T_{V_{j_1}}) + F(T_{V_{j_2}})) + \frac{\alpha_2}{2}\epsilon_*$$

do

2: Merge them:

$$\begin{aligned}
\boldsymbol{M}_{V_{j_1}} &= \boldsymbol{M}_{j_1} + \boldsymbol{M}_{j_2}, \boldsymbol{b}_{V_{j_1}} = \boldsymbol{b}_{V_{j_1}} + \boldsymbol{b}_{V_{j_2}}, \\
T_{V_{j_1}} &= T_{V_{j_1}} + T_{V_{j_2}}, C_{V_{j_1}} = C_{V_{j_1}} \cup C_{V_{j_2}}
\end{aligned}$$

3: Set $j(i) = j_1, \forall i \in j_2$, delete V_{j_2}

4: **end for**

A.4 Appendix for chapter 6

A.4.1 Proof of Lemma A.3.2

We first prove the following result:
With probability at least $1-\delta$ for some $\delta \in (0,1)$, at any $t \in [T]$:

$$\left\|\hat{\boldsymbol{\theta}}_{i,t} - \boldsymbol{\theta}^{j(i)}\right\|_2 \leq \frac{\beta(T_{i,t}, \frac{\delta}{u})}{\sqrt{\lambda + \lambda_{\min}(\boldsymbol{M}_{i,t})}}, \forall i \in \mathcal{U}\,, \tag{A.161}$$

where $\beta(T_{i,t}, \frac{\delta}{u}) \triangleq \sqrt{2\log(\frac{u}{\delta}) + d\log(1+\frac{T_{i,t}}{\lambda d})} + \sqrt{\lambda} + \alpha C$.

$$\begin{aligned}
\hat{\boldsymbol{\theta}}_{i,t} - \boldsymbol{\theta}^{j(i)} &= (\lambda \boldsymbol{I} + \boldsymbol{M}_{i,t})^{-1}\boldsymbol{b}_{i,t} - \boldsymbol{\theta}^{j(i)} \\
&= (\lambda \boldsymbol{I} + \sum_{\substack{s\in[t]\\ i_s=i}} w_{i_s,s}\boldsymbol{x}_{a_s}\boldsymbol{x}_{a_s}^\top)^{-1} \sum_{\substack{s\in[t]\\ i_s=i}} w_{i_s,s}\boldsymbol{x}_{a_s} r_s - \boldsymbol{\theta}^{j(i)} \\
&= (\lambda \boldsymbol{I} + \sum_{\substack{s\in[t]\\ i_s=i}} w_{i_s,s}\boldsymbol{x}_{a_s}\boldsymbol{x}_{a_s}^\top)^{-1} \left(\sum_{\substack{s\in[t]\\ i_s=i}} w_{i_s,s}\boldsymbol{x}_{a_s}(\boldsymbol{x}_{a_s}^\top \boldsymbol{\theta}_{i_s} + \eta_s + c_s) \right) - \boldsymbol{\theta}^{j(i)} \\
&= (\lambda \boldsymbol{I} + \sum_{\substack{s\in[t]\\ i_s=i}} w_{i_s,s}\boldsymbol{x}_{a_s}\boldsymbol{x}_{a_s}^\top)^{-1} \Bigg[(\lambda \boldsymbol{I} + \sum_{\substack{s\in[t]\\ i_s=i}} w_{i_s,s}\boldsymbol{x}_{a_s}\boldsymbol{x}_{a_s}^\top)\boldsymbol{\theta}^{j(i)} - \lambda \boldsymbol{\theta}^{j(i)} \\
&\qquad + \sum_{\substack{s\in[t]\\ i_s=i}} w_{i_s,s}\boldsymbol{x}_{a_s}\eta_s + \sum_{\substack{s\in[t]\\ i_s=i}} w_{i_s,s}\boldsymbol{x}_{a_s} c_s \Bigg] - \boldsymbol{\theta}^{j(i)} \\
&= -\lambda \boldsymbol{M}_{i,t}'^{-1}\boldsymbol{\theta}^{j(i)} + \boldsymbol{M}_{i,t}'^{-1} \sum_{\substack{s\in[t]\\ i_s=i}} w_{i_s,s}\boldsymbol{x}_{a_s}\eta_s + \boldsymbol{M}_{i,t}'^{-1} \sum_{\substack{s\in[t]\\ i_s=i}} w_{i_s,s}\boldsymbol{x}_{a_s} c_s\,,
\end{aligned}$$

where we denote $\boldsymbol{M}'_{i,t} = \boldsymbol{M}_{i,t} + \lambda \boldsymbol{I}$, and the above equations hold by definition.

Therefore, we have

$$\left\|\hat{\boldsymbol{\theta}}_{i,t} - \boldsymbol{\theta}^{j(i)}\right\|_2 \leq \lambda \left\|\boldsymbol{M}_{i,t}'^{-1}\boldsymbol{\theta}^{j(i)}\right\|_2 + \left\|\boldsymbol{M}_{i,t}'^{-1} \sum_{\substack{s\in[t]\\ i_s=i}} w_{i_s,s}\boldsymbol{x}_{a_s}\eta_s\right\|_2 + \left\|\boldsymbol{M}_{i,t}'^{-1} \sum_{\substack{s\in[t]\\ i_s=i}} w_{i_s,s}\boldsymbol{x}_{a_s} c_s\right\|_2 . \tag{A.162}$$

We then bound the three terms in Eq.(A.162) one by one. For the first term:

$$\lambda \left\| \boldsymbol{M}_{i,t}'^{-1} \boldsymbol{\theta}^{j(i)} \right\|_2 \leq \lambda \left\| \boldsymbol{M}_{i,t}'^{-\frac{1}{2}} \right\|_2^2 \left\| \boldsymbol{\theta}^{j(i)} \right\|_2 \leq \frac{\sqrt{\lambda}}{\sqrt{\lambda_{\min}(\boldsymbol{M}_{i,t}')}}, \quad \text{(A.163)}$$

where we use the Cauchy–Schwarz inequality, the inequality for the operator norm of matrices, and the fact that $\lambda_{\min}(\boldsymbol{M}_{i,t}') \geq \lambda$.

For the second term in Eq.(A.162), we have

$$\left\| \boldsymbol{M}_{i,t}'^{-1} \sum_{\substack{s \in [t] \\ i_s = i}} w_{i_s,s} \boldsymbol{x}_{a_s} \eta_s \right\|_2 \leq \left\| \boldsymbol{M}_{i,t}'^{-\frac{1}{2}} \sum_{\substack{s \in [t] \\ i_s = i}} w_{i_s,s} \boldsymbol{x}_{a_s} \eta_s \right\|_2 \left\| \boldsymbol{M}_{i,t}'^{-\frac{1}{2}} \right\|_2 \quad \text{(A.164)}$$

$$= \frac{\left\| \sum_{\substack{s \in [t] \\ i_s = i}} w_{i_s,s} \boldsymbol{x}_{a_s} \eta_s \right\|_{\boldsymbol{M}_{i,t}'^{-1}}}{\sqrt{\lambda_{\min}(\boldsymbol{M}_{i,t}')}}, \quad \text{(A.165)}$$

where Eq.(A.164) follows by the Cauchy–Schwarz inequality and the inequality for the operator norm of matrices, and Eq.(A.165) follows by the Courant-Fischer theorem.

Let $\tilde{\boldsymbol{x}}_s \triangleq \sqrt{w_{i_s,s}} \boldsymbol{x}_{a_s}$, $\tilde{\eta}_s \triangleq \sqrt{w_{i_s,s}} \eta_s$, then we have: $\|\tilde{\boldsymbol{x}}_s\|_2 \leq \left\|\sqrt{w_{i_s,s}}\right\|_2 \|\boldsymbol{x}_{a_s}\|_2 \leq 1$, $\tilde{\eta}_s$ is still 1-sub-gaussian (since η_s is 1-sub-gaussian and $\sqrt{w_{i_s,s}} \leq 1$), $\boldsymbol{M}_{i,t}' = \lambda \boldsymbol{I} + \sum_{\substack{s \in [t] \\ i_s = i}} \tilde{\boldsymbol{x}}_s \tilde{\boldsymbol{x}}_s^\top$, and the nominator in Eq.(A.165) becomes $\left\| \sum_{\substack{s \in [t] \\ i_s = i}} \tilde{\boldsymbol{x}}_s \tilde{\eta}_s \right\|_{\boldsymbol{M}_{i,t}'^{-1}}$. Then, following Theorem 1 in [43] and by union bound, with probability at least $1 - \delta$ for some $\delta \in (0, 1)$, for any $i \in \mathcal{U}$, we have:

$$\left\| \sum_{\substack{s \in [t] \\ i_s = i}} w_{i_s,s} \boldsymbol{x}_{a_s} \eta_s \right\|_{\boldsymbol{M}_{i,t}'^{-1}} = \left\| \sum_{\substack{s \in [t] \\ i_s = i}} \tilde{\boldsymbol{x}}_s \tilde{\eta}_s \right\|_{\boldsymbol{M}_{i,t}'^{-1}}$$

$$\leq \sqrt{2\log(\frac{u}{\delta}) + \log(\frac{\det(\boldsymbol{M}'_{i,t})}{\det(\lambda \boldsymbol{I})})}$$
$$\leq \sqrt{2\log(\frac{u}{\delta}) + d\log(1 + \frac{T_{i,t}}{\lambda d})}, \quad \text{(A.166)}$$

where $\det(\cdot)$ denotes the determinant of matrix arguement, Eq.(A.166) is because $\det(\boldsymbol{M}'_{i,t}) \leq \left(\frac{\mathrm{trace}(\lambda \boldsymbol{I} + \sum_{\substack{s\in[t]\\ i_s=i}} w_{i_s,s}\boldsymbol{x}_{a_s}\boldsymbol{x}_{a_s}^\top)}{d}\right)^d \leq \left(\frac{\lambda d + T_{i,t}}{d}\right)^d$, and $\det(\lambda \boldsymbol{I}) = \lambda^d$.

For the third term in Eq.(A.162), we have

$$\left\| \boldsymbol{M}'^{-1}_{i,t} \sum_{\substack{s\in[t]\\ i_s=i}} w_{i_s,s}\boldsymbol{x}_{a_s} c_s \right\|_2 \leq \left\| \boldsymbol{M}'^{-\frac{1}{2}}_{i,t} \sum_{\substack{s\in[t]\\ i_s=i}} w_{i_s,s}\boldsymbol{x}_{a_s} c_s \right\|_2 \left\| \boldsymbol{M}'^{-\frac{1}{2}}_{i,t} \right\|_2 \quad \text{(A.167)}$$
$$= \frac{\left\| \sum_{\substack{s\in[t]\\ i_s=i}} w_{i_s,s}\boldsymbol{x}_{a_s} c_s \right\|_{\boldsymbol{M}'^{-1}_{i,t}}}{\sqrt{\lambda_{\min}(\boldsymbol{M}'_{i,t})}} \quad \text{(A.168)}$$
$$\leq \frac{\sum_{\substack{s\in[t]\\ i_s=i}} |c_s|\, w_{i,s} \left\|\boldsymbol{x}_{a_s}\right\|_{\boldsymbol{M}'^{-1}_{i,t}}}{\sqrt{\lambda_{\min}(\boldsymbol{M}'_{i,t})}}$$
$$\leq \frac{\alpha C}{\sqrt{\lambda_{\min}(\boldsymbol{M}'_{i,t})}} \quad \text{(A.169)}$$

where Eq.(A.167) follows by the Cauchy–Schwarz inequality and the inequality for the operator norm of matrices, Eq.(A.168) follows by the Courant-Fischer theorem, and Eq.(A.169) is because by definition $w_{i,s} \leq \frac{\alpha}{\|\boldsymbol{x}_{a_s}\|_{\boldsymbol{M}'^{-1}_{i,s}}} \leq \frac{\alpha}{\|\boldsymbol{x}_{a_s}\|_{\boldsymbol{M}'^{-1}_{i,t}}}$ (since $\boldsymbol{M}'_{i,t} \succeq \boldsymbol{M}'_{i,s}$, $\boldsymbol{M}'^{-1}_{i,s} \succeq \boldsymbol{M}'^{-1}_{i,t}$, $\|\boldsymbol{x}_{a_s}\|_{\boldsymbol{M}'^{-1}_{i,s}} \geq \|\boldsymbol{x}_{a_s}\|_{\boldsymbol{M}'^{-1}_{i,t}}$), $\sum_{t=1}^T |c_t| \leq C$.

Combining the above bounds of these three terms, we can get

that Eq.(A.161) holds.

We then prove the following technical lemma.

Lemma A.4.1. *Under Assumption 9.4, at any time t, for any fixed unit vector $\boldsymbol{\theta} \in \mathbb{R}^d$*

$$\mathbb{E}_t[(\boldsymbol{\theta}^\top \boldsymbol{x}_{a_t})^2 | \, |\mathcal{A}_t|] \geq \tilde{\lambda}_x \triangleq \int_0^{\lambda_x} (1 - e^{-\frac{(\lambda_x - x)^2}{2\sigma^2}})^K dx \,, \qquad \text{(A.170)}$$

where K is the upper bound of $|\mathcal{A}_t|$ for any t.

Proof. The proof of this lemma mainly follows the proof of Claim 1 in [46], but with more careful analysis, since their assumption on the arm generation distribution is more stringent than our Assumption 9.4 by putting more restrictions on the variance upper bound σ^2 (specifically, they require $\sigma^2 \leq \frac{\lambda^2}{8\log(4K)}$).

Denote the feasible arms at round t by $\mathcal{A}_t = \{\boldsymbol{x}_{t,1}, \boldsymbol{x}_{t,2}, \ldots, \boldsymbol{x}_{t,|\mathcal{A}_t|}\}$. Consider the corresponding i.i.d. random variables $\theta_i = (\boldsymbol{\theta}^\top \boldsymbol{x}_{t,i})^2 - \mathbb{E}_t[(\boldsymbol{\theta}^\top \boldsymbol{x}_{t,i})^2 | \, |\mathcal{A}_t|], i = 1, 2, \ldots, |\mathcal{A}_t|$. By Assumption 9.4, θ_i s are sub-Gaussian random variables with variance bounded by σ^2. Therefore, for any $\alpha > 0$ and any $i \in [|\mathcal{A}_t|]$, we have:

$$\mathbb{P}_t(\theta_i < -\alpha | \, |\mathcal{A}_t|) \leq e^{-\frac{\alpha^2}{2\sigma^2}} \,,$$

where we use $\mathbb{P}_t(\cdot)$ to be the shorthand for the conditional probability $\mathbb{P}(\cdot | (i_1, \mathcal{A}_1, r_1), \ldots, (i_{t-1}, \mathcal{A}_{t-1}, r_{t-1}), i_t)$.

By Assumption 9.4, we can also get that $\mathbb{E}_t[(\boldsymbol{\theta}^\top \boldsymbol{x}_{t,i})^2 | \, |\mathcal{A}_t| = \mathbb{E}_t[\boldsymbol{\theta}^\top \boldsymbol{x}_{t,i} \boldsymbol{x}_{t,i}^\top \boldsymbol{\theta} | \, |\mathcal{A}_t|] \geq \lambda_{\min}(\mathbb{E}_{\boldsymbol{x} \sim \rho}[\boldsymbol{x}\boldsymbol{x}^\top]) \geq \lambda_x$. With these inequalities above, we can get

$$\mathbb{P}_t(\min_{i=1,\ldots,|\mathcal{A}_t|} (\boldsymbol{\theta}^\top \boldsymbol{x}_{t,i})^2 \geq \lambda_x - \alpha | \, |\mathcal{A}_t|) \geq (1 - e^{-\frac{\alpha^2}{2\sigma^2}})^K \,.$$

Therefore, we can get

$$\begin{aligned}\mathbb{E}_t[(\boldsymbol{\theta}^\top \boldsymbol{x}_{a_t})^2|\,|\mathcal{A}_t|] &\geq \mathbb{E}_t[\min_{i=1,\ldots,|\mathcal{A}_t|}(\boldsymbol{\theta}^\top \boldsymbol{x}_{t,i})^2|\,|\mathcal{A}_t|] \\ &\geq \int_0^\infty \mathbb{P}_t(\min_{i=1,\ldots,|\mathcal{A}_t|}(\boldsymbol{\theta}^\top \boldsymbol{x}_{t,i})^2 \geq x|\,|\mathcal{A}_t|)dx \\ &\geq \int_0^{\lambda_x}(1-e^{-\frac{(\lambda_x-x)^2}{2\sigma^2}})^K dx \triangleq \tilde{\lambda}_x\end{aligned}$$

□

Note that $w_{i,s} = \min\{1, \frac{\alpha}{\|\boldsymbol{x}_{a_s}\|_{\boldsymbol{M}'^{-1}_{i,t}}}\}$, and we have

$$\frac{\alpha}{\|\boldsymbol{x}_{a_s}\|_{\boldsymbol{M}'^{-1}_{i,t}}} = \frac{\alpha}{\sqrt{\boldsymbol{x}_{a_s}^\top \boldsymbol{M}'^{-1}_{i,t}\boldsymbol{x}_{a_s}}} \geq \frac{\alpha}{\sqrt{\lambda_{\min}(\boldsymbol{M}'^{-1}_{i,t})}} = \alpha\sqrt{\lambda_{\min}(\boldsymbol{M}'_{i,t})} \geq \alpha\sqrt{\lambda}.$$

Since $\alpha\sqrt{\lambda} < 1$ typically holds, we have $w_{i,s} \geq \alpha\sqrt{\lambda}$.

Then, with the item regularity assumption stated in Assumption 9.4, the technical Lemma A.4.1, together with Lemma 7 in [135], with probability at least $1-\delta$, for a particular user i, at any t such that $T_{i,t} \geq \frac{16}{\tilde{\lambda}_x^2}\log(\frac{8d}{\tilde{\lambda}_x^2\delta})$, we have:

$$\lambda_{\min}(\boldsymbol{M}'_{i,t}) \geq 2\alpha\sqrt{\lambda}\tilde{\lambda}_x T_{i,t} + \lambda\,. \tag{A.171}$$

With this result, together with Eq.(A.161), we can get that for any t such that $T_{i,t} \geq \frac{16}{\tilde{\lambda}_x^2}\log(\frac{8d}{\tilde{\lambda}_x^2\delta})$, with probability at least $1-\delta$ for some $\delta \in (0,1)$, $\forall i \in \mathcal{U}$, we have:

$$\begin{aligned}\left\|\hat{\boldsymbol{\theta}}_{i,t} - \boldsymbol{\theta}^{j(i)}\right\|_2 &\leq \frac{\beta(T_{i,t}, \frac{\delta}{u})}{\sqrt{\lambda_{\min}(\boldsymbol{M}'_{i,t})}} \\ &\leq \frac{\beta(T_{i,t}, \frac{\delta}{u})}{\sqrt{2\alpha\sqrt{\lambda}\tilde{\lambda}_x T_{i,t} + \lambda}}\end{aligned}$$

$$\leq \frac{\beta(T_{i,t}, \frac{\delta}{u})}{\sqrt{2\alpha\sqrt{\lambda}\tilde{\lambda}_x T_{i,t}}}$$
$$= \frac{\sqrt{2\log(\frac{u}{\delta}) + d\log(1+\frac{T_{i,t}}{\lambda d})} + \sqrt{\lambda} + \alpha C}{\sqrt{2\alpha\sqrt{\lambda}\tilde{\lambda}_x T_{i,t}}}. \tag{A.172}$$

Then, we want to find a sufficient time $T_{i,t}$ for a fixed user i such that

$$\left\| \hat{\boldsymbol{\theta}}_{i,t} - \boldsymbol{\theta}^{j(i)} \right\|_2 < \frac{\gamma}{4}. \tag{A.173}$$

To do this, with Eq.(A.172), we can get it by letting

$$\frac{\sqrt{\lambda}}{\sqrt{2\alpha\sqrt{\lambda}\tilde{\lambda}_x T_{i,t}}} < \frac{\gamma}{12}, \tag{A.174}$$

$$\frac{\alpha C}{\sqrt{2\alpha\sqrt{\lambda}\tilde{\lambda}_x T_{i,t}}} < \frac{\gamma}{12}, \tag{A.175}$$

$$\frac{\sqrt{2\log(\frac{u}{\delta}) + d\log(1+\frac{T_{i,t}}{\lambda d})}}{\sqrt{2\alpha\sqrt{\lambda}\tilde{\lambda}_x T_{i,t}}} < \frac{\gamma}{12}. \tag{A.176}$$

For Eq.(A.174), we can get

$$T_{i,t} > \frac{72\sqrt{\lambda}}{\alpha\gamma^2\tilde{\lambda}_x}. \tag{A.177}$$

For Eq.(A.175), we can get

$$T_{i,t} > \frac{72\alpha C^2}{\gamma^2\sqrt{\lambda}\tilde{\lambda}_x}. \tag{A.178}$$

For Eq.(A.176), we have

$$\frac{2\log(\frac{u}{\delta}) + d\log(1+\frac{T_{i,t}}{\lambda d})}{2\alpha\sqrt{\lambda}\tilde{\lambda}_x T_{i,t}} < \frac{\gamma^2}{144}. \tag{A.179}$$

Then it is sufficient to get Eq.(A.179) if the following holds

$$\frac{2\log(\frac{u}{\delta})}{2\alpha\sqrt{\lambda}\tilde{\lambda}_x T_{i,t}} < \frac{\gamma^2}{288}\,, \tag{A.180}$$

$$\frac{d\log(1+\frac{T_{i,t}}{\lambda d})}{2\alpha\sqrt{\lambda}\tilde{\lambda}_x T_{i,t}} < \frac{\gamma^2}{288}\,. \tag{A.181}$$

For Eq.(A.180), we can get

$$T_{i,t} > \frac{288\log(\frac{u}{\delta})}{\gamma^2\alpha\sqrt{\lambda}\tilde{\lambda}_x} \tag{A.182}$$

For Eq.(A.181), we can get

$$T_{i,t} > \frac{144d}{\gamma^2\alpha\sqrt{\lambda}\tilde{\lambda}_x}\log(1+\frac{T_{i,t}}{\lambda d})\,. \tag{A.183}$$

Following Lemma 9 in [135], we can get the following sufficient condition for Eq.(A.183):

$$T_{i,t} > \frac{288d}{\gamma^2\alpha\sqrt{\lambda}\tilde{\lambda}_x}\log(\frac{288}{\gamma^2\alpha\sqrt{\lambda}\tilde{\lambda}_x})\,. \tag{A.184}$$

Then, since typically $\frac{u}{\delta} > \frac{288}{\gamma^2\alpha\sqrt{\lambda}\tilde{\lambda}_x}$, we can get the following sufficient condition for Eq.(A.182) and Eq.(A.184)

$$T_{i,t} > \frac{288d}{\gamma^2\alpha\sqrt{\lambda}\tilde{\lambda}_x}\log(\frac{u}{\delta})\,. \tag{A.185}$$

Together with Eq.(A.177), Eq.(A.178), and the condition for Eq.(A.348) we can get the following sufficient condition for Eq.(A.173) to hold

$$T_{i,t} > \max\{\frac{288d}{\gamma^2\alpha\sqrt{\lambda}\tilde{\lambda}_x}\log(\frac{u}{\delta}), \frac{16}{\tilde{\lambda}_x^2}\log(\frac{8d}{\tilde{\lambda}_x^2\delta}), \frac{72\sqrt{\lambda}}{\alpha\gamma^2\tilde{\lambda}_x}, \frac{72\alpha C^2}{\gamma^2\sqrt{\lambda}\tilde{\lambda}_x}\}\,. \tag{A.186}$$

Then, with Assumption 9.3 on the uniform arrival of users, following Lemma 8 in [135], and by union bound, we can get that

with probability at least $1-\delta$, for all

$$t \geq T_0 \triangleq 16u\log(\frac{u}{\delta})+4u\max\{\frac{288d}{\gamma^2\alpha\sqrt{\lambda}\tilde{\lambda}_x}\log(\frac{u}{\delta}), \frac{16}{\tilde{\lambda}_x^2}\log(\frac{8d}{\tilde{\lambda}_x^2\delta}), \frac{72\sqrt{\lambda}}{\alpha\gamma^2\tilde{\lambda}_x}, \frac{72\alpha C^2}{\gamma^2\sqrt{\lambda}\tilde{\lambda}_x}\}, \tag{A.187}$$

Eq.(A.185) holds for all $i \in \mathcal{U}$, and therefore Eq.(A.173) holds for all $i \in \mathcal{U}$. With this, we can show that RCLUB-WCU will cluster all the users correctly after T_0. First, if RCLUB-WCU deletes the edge (i,l), then user i and user j belong to different ground-truth clusters, i.e., $\|\boldsymbol{\theta}_i - \boldsymbol{\theta}_l\|_2 > 0$. This is because by the deletion rule of the algorithm, the concentration bound, and triangle inequality, $\|\boldsymbol{\theta}_i - \boldsymbol{\theta}_l\|_2 = \left\|\boldsymbol{\theta}^{j(i)} - \boldsymbol{\theta}^{j(l)}\right\|_2 \geq \left\|\hat{\boldsymbol{\theta}}_{i,t} - \hat{\boldsymbol{\theta}}_{l,t}\right\|_2 - \left\|\boldsymbol{\theta}^{j(l)} - \boldsymbol{\theta}_{l,t}\right\|_2 - \left\|\boldsymbol{\theta}^{j(i)} - \boldsymbol{\theta}_{i,t}\right\|_2 > 0$. Second, we show that if $\|\boldsymbol{\theta}_i - \boldsymbol{\theta}_l\| \geq \gamma$, RCLUB-WCU will delete the edge (i,l). This is because if $\|\boldsymbol{\theta}_i - \boldsymbol{\theta}_l\| \geq \gamma$, then by the triangle inequality, and $\left\|\hat{\boldsymbol{\theta}}_{i,t} - \boldsymbol{\theta}^{j(i)}\right\|_2 < \frac{\gamma}{4}$, $\left\|\hat{\boldsymbol{\theta}}_{l,t} - \boldsymbol{\theta}^{j(l)}\right\|_2 < \frac{\gamma}{4}$, $\boldsymbol{\theta}_i = \boldsymbol{\theta}^{j(i)}$, $\boldsymbol{\theta}_l = \boldsymbol{\theta}^{j(l)}$, we have $\left\|\hat{\boldsymbol{\theta}}_{i,t} - \hat{\boldsymbol{\theta}}_{l,t}\right\|_2 \geq \|\boldsymbol{\theta}_i - \boldsymbol{\theta}_l\| - \left\|\hat{\boldsymbol{\theta}}_{i,t} - \boldsymbol{\theta}^{j(i)}\right\|_2 - \left\|\hat{\boldsymbol{\theta}}_{l,t} - \boldsymbol{\theta}^{j(l)}\right\|_2 > \gamma - \frac{\gamma}{4} - \frac{\gamma}{4} = \frac{\gamma}{2} > \frac{\sqrt{\lambda}+\sqrt{2\log(\frac{u}{\delta})+d\log(1+\frac{T_{i,t}}{\lambda d})}}{\sqrt{\lambda+2\tilde{\lambda}_x T_{i,t}}} + \frac{\sqrt{\lambda}+\sqrt{2\log(\frac{u}{\delta})+d\log(1+\frac{T_{l,t}}{\lambda d})}}{\sqrt{\lambda+2\tilde{\lambda}_x T_{l,t}}}$, which will trigger the deletion condition Line 10 in Algo.8.

A.4.2 Proof of Lemma 6.4.2

After T_0, if the clustering structure is correct, *i.e.*, $V_t = V_{j(i_t)}$, then we have

$$\begin{aligned}\hat{\boldsymbol{\theta}}_{V_t,t-1} - \boldsymbol{\theta}_{i_t} &= \boldsymbol{M}_{V_t,t-1}^{-1}\boldsymbol{b}_{V_t,t-1} - \boldsymbol{\theta}_{i_t} \\ &= (\lambda\boldsymbol{I} + \sum_{\substack{s\in[t-1]\\ i_s\in V_t}} w_{i_s,s}\boldsymbol{x}_{a_s}\boldsymbol{x}_{a_s}^\top)^{-1}(\sum_{\substack{s\in[t-1]\\ i_s\in V_t}} w_{i_s,s}\boldsymbol{x}_{a_s}r_s) - \boldsymbol{\theta}_{i_t}\end{aligned}$$

$$
\begin{aligned}
&= (\lambda \boldsymbol{I} + \sum_{\substack{s\in[t-1]\\ i_s\in V_t}} w_{i_s,s}\boldsymbol{x}_{a_s}\boldsymbol{x}_{a_s}^\top)^{-1}\big(\sum_{\substack{s\in[t-1]\\ i_s\in V_t}} w_{i_s,s}\boldsymbol{x}_{a_s}(\boldsymbol{x}_{a_s}^\top\boldsymbol{\theta}_{i_t}+\eta_s+c_s)\big) - \boldsymbol{\theta}_{i_t} \qquad \text{(A.188)}\\
&= (\lambda \boldsymbol{I} + \sum_{\substack{s\in[t-1]\\ i_s\in V_t}} w_{i_s,s}\boldsymbol{x}_{a_s}\boldsymbol{x}_{a_s}^\top)^{-1}\Bigg(\sum_{\substack{s\in[t-1]\\ i_s\in V_t}} (w_{i_s,s}\boldsymbol{x}_{a_s}\boldsymbol{x}_{a_s}^\top+\lambda\boldsymbol{I})\boldsymbol{\theta}_{i_t} - \lambda\boldsymbol{\theta}_{i_t}\\
&\qquad + \sum_{\substack{s\in[t-1]\\ i_s\in V_t}} w_{i_s,s}\boldsymbol{x}_{a_s}\eta_s + \sum_{\substack{s\in[t-1]\\ i_s\in V_t}} w_{i_s,s}\boldsymbol{x}_{a_s}c_s)\Bigg) - \boldsymbol{\theta}_{i_t}\\
&= -\lambda\boldsymbol{M}'^{-1}_{V_t,t-1}\boldsymbol{\theta}_{i_t} - \boldsymbol{M}'^{-1}_{V_t,t-1}\sum_{\substack{s\in[t-1]\\ i_s\in V_t}} w_{i_s,s}\boldsymbol{x}_{a_s}\eta_s + \boldsymbol{M}'^{-1}_{V_t,t-1}\sum_{\substack{s\in[t-1]\\ i_s\in V_t}} w_{i_s,s}\boldsymbol{x}_{a_s}c_s\,,
\end{aligned}
$$

where we denote $\boldsymbol{M}'_{V_t,t-1} = \boldsymbol{M}_{V_t,t-1} + \lambda\boldsymbol{I}$, and Eq.(A.188) is because $V_t = V_{j(i_t)}$ thus $\boldsymbol{\theta}_{i_s} = \boldsymbol{\theta}_{i_t}, \forall i_s \in V_t$.

Therefore, we have

$$
\begin{aligned}
&\left|\boldsymbol{x}_a^\top(\hat{\boldsymbol{\theta}}_{V_t,t-1} - \boldsymbol{\theta}_{i_t})\right|\\
&\le \lambda\left|\boldsymbol{x}_a^\top\boldsymbol{M}'^{-1}_{V_t,t-1}\boldsymbol{\theta}_{i_t}\right| + \left|\boldsymbol{x}_a^\top\boldsymbol{M}'^{-1}_{V_t,t-1}\sum_{\substack{s\in[t-1]\\ i_s\in V_t}} w_{i_s,s}\boldsymbol{x}_{a_s}\eta_s\right| + \left|\boldsymbol{x}_a^\top\boldsymbol{M}'^{-1}_{V_t,t-1}\sum_{\substack{s\in[t-1]\\ i_s\in V_t}} w_{i_s,s}\boldsymbol{x}_{a_s}c_s\right|\\
&\le \|\boldsymbol{x}_a\|_{\boldsymbol{M}'^{-1}_{V_t,t-1}}\left(\sqrt{\lambda} + \left\|\sum_{\substack{s\in[t-1]\\ i_s\in V_t}} w_{i_s,s}\boldsymbol{x}_{a_s}\eta_s\right\|_{\boldsymbol{M}'^{-1}_{V_t,t-1}} + \left\|\sum_{\substack{s\in[t-1]\\ i_s\in V_t}} w_{i_s,s}\boldsymbol{x}_{a_s}c_s\right\|_{\boldsymbol{M}'^{-1}_{V_t,t-1}}\right),
\end{aligned}
\qquad \text{(A.189)}
$$

where Eq.(A.189) is by Cauchy–Schwarz inequality, matrix operator inequality, and $\left|\boldsymbol{x}_a^\top\boldsymbol{M}'^{-1}_{V_t,t-1}\boldsymbol{\theta}_{i_t}\right| \le \lambda\left\|\boldsymbol{M}'^{-\frac{1}{2}}_{V_t,t-1}\right\|_2\|\boldsymbol{\theta}_{i_t}\|_2 = \lambda\frac{1}{\sqrt{\lambda_{\min}(\boldsymbol{M}_{V_t,t-1})}}\|\boldsymbol{\theta}_{i_t}\|_2 \le \sqrt{\lambda}$ since $\lambda_{\min}(\boldsymbol{M}_{V_t,t-1}) \ge \lambda$ and $\|\boldsymbol{\theta}_{i_t}\|_2 \le 1$.

Let $\tilde{\boldsymbol{x}}_s \triangleq \sqrt{w_{i_s,s}}\boldsymbol{x}_{a_s}$, $\tilde{\eta}_s \triangleq \sqrt{w_{i_s,s}}\eta_s$, then we have: $\|\tilde{\boldsymbol{x}}_s\|_2 \le \left\|\sqrt{w_{i_s,s}}\right\|_2\|\boldsymbol{x}_{a_s}\|_2 \le 1$, $\tilde{\eta}_s$ is still 1-sub-gaussian (since η_s is 1-sub-gaussian and $\sqrt{w_{i_s,s}} \le 1$), $\boldsymbol{M}'_{i,t} = \lambda\boldsymbol{I} + \sum_{\substack{s\in[t]\\ i_s=i}}\tilde{\boldsymbol{x}}_s\tilde{\boldsymbol{x}}_s^\top$, and

$\left\|\sum_{\substack{s\in[t-1]\\ i_s\in V_t}} w_{i_s,s}\boldsymbol{x}_{a_s}\eta_s\right\|_{\boldsymbol{M}'^{-1}_{V_t,t-1}}$ becomes $\left\|\sum_{\substack{s\in[t]\\ i_s=i}} \tilde{\boldsymbol{x}}_s\tilde{\eta}_s\right\|_{\boldsymbol{M}'^{-1}_{V_t,t-1}}$. Then, following Theorem 1 in [43], with probability at least $1-\delta$ for some $\delta\in(0,1)$, we have:

$$\begin{aligned}\left\|\sum_{\substack{s\in[t-1]\\ i_s\in V_t}} w_{i_s,s}\boldsymbol{x}_{a_s}\eta_s\right\|_{\boldsymbol{M}'^{-1}_{V_t,t-1}} &= \left\|\sum_{\substack{s\in[t]\\ i_s=i}} \tilde{\boldsymbol{x}}_s\tilde{\eta}_s\right\|_{\boldsymbol{M}'^{-1}_{V_t,t-1}}\\ &\le \sqrt{2\log(\frac{u}{\delta})+\log(\frac{\det(\boldsymbol{M}'_{V_t,t-1})}{\det(\lambda\boldsymbol{I})})}\\ &\le \sqrt{2\log(\frac{u}{\delta})+d\log(1+\frac{T}{\lambda d})}\,, \quad \text{(A.190)}\end{aligned}$$

And for $\left\|\sum_{\substack{s\in[t-1]\\ i_s\in V_t}} w_{i_s,s}\boldsymbol{x}_{a_s}c_s\right\|_{\boldsymbol{M}'^{-1}_{V_t,t-1}}$, we have

$$\left\|\sum_{\substack{s\in[t-1]\\ i_s\in V_t}} w_{i_s,s}\boldsymbol{x}_{a_s}c_s\right\|_{\boldsymbol{M}'^{-1}_{V_t,t-1}} \le \sum_{\substack{s\in[t-1]\\ i_s\in V_t}} w_{i_s,s}\,|c_s|\,\|\boldsymbol{x}_{a_s}\|_{\boldsymbol{M}'^{-1}_{V_t,t-1}} \le \alpha C\,, \quad \text{(A.191)}$$

where we use $\sum_{t=1}^{T}|c_t|\le C$, $w_{i_s,s}\le \frac{\alpha}{\|\boldsymbol{x}_{a_s}\|_{\boldsymbol{M}'^{-1}_{i_s,t-1}}} \le \frac{\alpha}{\|\boldsymbol{x}_{a_s}\|_{\boldsymbol{M}'^{-1}_{V_t,t-1}}}$.

Plugging Eq.(A.191) and Eq.(A.190) into Eq.(A.189), together with Lemma A.3.2, we can complete the proof of Lemma 6.4.2.

A.4.3 Proof of Theorem 6.4.3

After T_0, we define event

$$\mathcal{E} = \{\text{the algorithm clusters all the users correctly for all } t\ge T_0\}\,. \quad \text{(A.192)}$$

Then, with Lemma A.3.2 and picking $\delta = \frac{1}{T}$, we have

$$\begin{aligned} R(T) &= \mathbb{P}(\mathcal{E})\mathbb{I}\{\mathcal{E}\}R(T) + \mathbb{P}(\overline{\mathcal{E}})\mathbb{I}\{\overline{\mathcal{E}}\}R(T) \\ &\leq \mathbb{I}\{\mathcal{E}\}R(T) + 4 \times \frac{1}{T} \times T \\ &= \mathbb{I}\{\mathcal{E}\}R(T) + 4\,. \end{aligned} \tag{A.193}$$

Then it remains to bound $\mathbb{I}\{\mathcal{E}\}R(T)$. For the first T_0 rounds, we can upper bound the regret in the first T_0 rounds by T_0. After T_0, under event $\mathcal{E}$ and by Lemma 6.4.2, we have that with probability at least $1-\delta$, for any $\boldsymbol{x}_a$:

$$\left|\boldsymbol{x}_a^{\mathrm{T}}(\hat{\boldsymbol{\theta}}_{V_t,t-1} - \boldsymbol{\theta}_{i_t})\right| \leq \beta \left\|\boldsymbol{x}_{\boldsymbol{a}}\right\|_{\boldsymbol{M}_{V_t,t-1}^{-1}} \triangleq C_{a,t}\,. \tag{A.194}$$

Therefore, for the instantaneous regret R_t at round t, with $\mathcal{E}$, with probability at least $1-\delta$, at $\forall t \geq T_0$:

$$\begin{aligned} R_t &= \boldsymbol{x}_{a_t^*}^\top \boldsymbol{\theta}_{i_t} - \boldsymbol{x}_{a_t}^\top \boldsymbol{\theta}_{i_t} \\ &= \boldsymbol{x}_{a_t^*}^\top(\boldsymbol{\theta}_{i_t} - \hat{\boldsymbol{\theta}}_{V_t,t-1}) + (\boldsymbol{x}_{a_t^*}^\top \hat{\boldsymbol{\theta}}_{V_t,t-1} + C_{a_t^*,t}) - (\boldsymbol{x}_{a_t}^\top \hat{\boldsymbol{\theta}}_{V_t,t-1} + C_{a_t,t}) \\ &\quad + \boldsymbol{x}_{a_t}^\top(\hat{\boldsymbol{\theta}}_{\overline{V}_t,t-1} - \boldsymbol{\theta}_{i_t}) + C_{a_t,t} - C_{a_t^*,t} \\ &\leq 2C_{a_t,t}\,, \end{aligned} \tag{A.195}$$

where the last inequality holds by the UCB arm selection strategy in Eq.(6.3) and Eq.(A.194).

Therefore, for $\mathbb{I}\{\mathcal{E}\}R(T)$:

$$\begin{aligned} \mathbb{I}\{\mathcal{E}\}R(T) &\leq R(T_0) + \mathbb{E}[\mathbb{I}\{\mathcal{E}\}\sum_{t=T_0+1}^{T} R_t] \\ &\leq T_0 + 2\mathbb{E}[\mathbb{I}\{\mathcal{E}\}\sum_{t=T_0+1}^{T} C_{a_t,t}]\,. \end{aligned} \tag{A.196}$$

Then it remains to bound $\mathbb{E}[\mathbb{I}\{\mathcal{E}\}\sum_{t=T_0+1}^{T} C_{a_t,t}]$. For $\sum_{t=T_0+1}^{T} C_{a_t,t}$, we can distinguish it into two cases:

$$\begin{aligned}\sum_{t=T_0+1}^{T} C_{a_t,t} &\leq \beta \sum_{t=1}^{T} \|\boldsymbol{x}_{\boldsymbol{a_t}}\|_{\boldsymbol{M}_{V_t,t-1}^{-1}} \\ &= \beta \sum_{t\in[T]:w_{i_t,t}=1} \|\boldsymbol{x}_{\boldsymbol{a_t}}\|_{\boldsymbol{M}_{V_t,t-1}^{-1}} + \beta \sum_{t\in[T]:w_{i_t,t}<1} \|\boldsymbol{x}_{\boldsymbol{a_t}}\|_{\boldsymbol{M}_{V_t,t-1}^{-1}} .\end{aligned} \tag{A.197}$$

Then, we prove the following technical lemma.

Lemma A.4.2.

$$\sum_{t=T_0+1}^{T} \min\{\mathbb{I}\{i_t \in V_j\} \|\boldsymbol{x}_{a_t}\|^2_{\boldsymbol{M}_{V_j,t-1}^{-1}}, 1\} \leq 2d\log(1+\frac{T}{\lambda d}), \forall j \in [m] . \tag{A.198}$$

Proof.

$$\begin{aligned} det(\boldsymbol{M}_{V_j,T}) &= det\bigg(\boldsymbol{M}_{V_j,T-1} + \mathbb{I}\{i_T \in V_j\}\boldsymbol{x}_{a_T}\boldsymbol{x}_{a_T}^\top\bigg) \\ &= det(\boldsymbol{M}_{V_j,T-1}) det\bigg(\boldsymbol{I} + \mathbb{I}\{i_T \in V_j\}\boldsymbol{M}_{V_j,T-1}^{-\frac{1}{2}}\boldsymbol{x}_{a_T}\boldsymbol{x}_{a_T}^\top\boldsymbol{M}_{V_j,T-1}^{-\frac{1}{2}}\bigg) \\ &= det(\boldsymbol{M}_{V_j,T-1})\bigg(1 + \mathbb{I}\{i_T \in V_j\} \|\boldsymbol{x}_{a_T}\|^2_{\boldsymbol{M}_{V_j,T-1}^{-1}}\bigg) \\ &= det(\boldsymbol{M}_{V_j,T_0}) \prod_{t=T_0+1}^{T}\bigg(1 + \mathbb{I}\{i_t \in V_j\} \|\boldsymbol{x}_{a_t}\|^2_{\boldsymbol{M}_{V_j,t-1}^{-1}}\bigg) \\ &\geq det(\lambda\boldsymbol{I}) \prod_{t=T_0+1}^{T}\bigg(1 + \mathbb{I}\{i_t \in V_j\} \|\boldsymbol{x}_{a_t}\|^2_{\boldsymbol{M}_{V_j,t-1}^{-1}}\bigg) . \end{aligned} \tag{A.199}$$

$\forall x \in [0,1]$, we have $x \leq 2\log(1+x)$. Therefore

$$\begin{aligned}
&\sum_{t=T_0+1}^{T} \min\{\mathbb{I}\{i_t \in V_j\} \left\|\boldsymbol{x}_{a_t}\right\|^2_{\boldsymbol{M}^{-1}_{V_j,t-1}}, 1\} \\
&\leq 2\sum_{t=T_0+1}^{T} \log\left(1+\mathbb{I}\{i_t \in V_j\} \left\|\boldsymbol{x}_{a_t}\right\|^2_{\boldsymbol{M}^{-1}_{V_j,t-1}}\right) \\
&= 2\log\left(\prod_{t=T_0+1}^{T}\left(1+\mathbb{I}\{i_t \in V_j\} \left\|\boldsymbol{x}_{a_t}\right\|^2_{\boldsymbol{M}^{-1}_{V_j,t-1}}\right)\right) \\
&\leq 2[\log(det(\boldsymbol{M}_{V_j,T})) - \log(det(\lambda \boldsymbol{I}))] \\
&\leq 2\log\left(\frac{trace(\lambda \boldsymbol{I} + \sum_{t=1}^{T}\mathbb{I}\{i_t \in V_j\}\boldsymbol{x}_{a_t}\boldsymbol{x}_{a_t}^\top)}{\lambda d}\right)^d \\
&\leq 2d\log(1+\frac{T}{\lambda d}). \qquad \text{(A.200)}
\end{aligned}$$

□

Denote the rounds with $w_{i_t,t} = 1$ as $\{\tilde{t}_1, \ldots, \tilde{t}_{l_1}\}$, and gram matrix $\tilde{\boldsymbol{G}}_{V_{\tilde{t}_\tau}, \tilde{t}_\tau - 1} \triangleq \lambda \boldsymbol{I} + \sum_{\substack{s\in[\tau] \\ i_s \in V_{\tilde{t}_\tau}}} \boldsymbol{x}_{a_{\tilde{t}_s}} \boldsymbol{x}_{a_{\tilde{t}_s}}^\top$; denote the rounds with $w_{i_t,t} < 1$ as $\{t'_1, \ldots, t'_{l_2}\}$, gram matrix $\boldsymbol{G}'_{V_{t'_\tau}, t'_\tau - 1} \triangleq \lambda \boldsymbol{I} + \sum_{\substack{s\in[\tau] \\ i_s \in V_{t'_\tau}}} w_{i_{t'_s}, t'_s} \boldsymbol{x}_{a_{t'_s}} \boldsymbol{x}_{a_{t'_s}}^\top$.

Then we have

$$\begin{aligned}
&\sum_{t\in[T]:w_{i_t,t}=1} \left\|\boldsymbol{x}_{\boldsymbol{a_t}}\right\|_{\boldsymbol{M}^{-1}_{V_t,t-1}} \\
&= \sum_{j=1}^{m}\sum_{\tau=1}^{l_1} \mathbb{I}\{i_{\tilde{t}_\tau} \in V_j\} \left\|\boldsymbol{x}_{\boldsymbol{a_{\tilde{t}_\tau}}}\right\|_{\boldsymbol{M}^{-1}_{V_{\tilde{t}_\tau}, \tilde{t}_\tau - 1}} \\
&\leq \sum_{j=1}^{m}\sum_{\tau=1}^{l_1} \mathbb{I}\{i_{\tilde{t}_\tau} \in V_j\} \left\|\boldsymbol{x}_{\boldsymbol{a_{\tilde{t}_\tau}}}\right\|_{\tilde{\boldsymbol{G}}^{-1}_{V_{\tilde{t}_\tau}, \tilde{t}_\tau - 1}} \qquad \text{(A.201)}
\end{aligned}$$

$$\leq \sum_{j=1}^{m} \sqrt{\sum_{\tau=1}^{l_1} \mathbb{I}\{i_{\tilde{t}_\tau} \in V_j\} \sum_{\tau=1}^{l_1} \min\{1, \mathbb{I}\{i_{\tilde{t}_\tau} \in V_j\} \left\|\boldsymbol{x}_{\boldsymbol{a}_{\tilde{\boldsymbol{t}}_\tau}}\right\|^2_{\tilde{\boldsymbol{G}}^{-1}_{V_{\tilde{t}_\tau}, \tilde{t}_\tau - 1}}\}} \tag{A.202}$$

$$\leq \sum_{j=1}^{m} \sqrt{T_{V_j,T} \times 2d \log(1 + \frac{T}{\lambda d})} \tag{A.203}$$

$$\leq \sqrt{2m \sum_{j=1}^{m} T_{V_j,T} d \log(1 + \frac{T}{\lambda d})} = \sqrt{2mdT \log(1 + \frac{T}{\lambda d})}\,, \tag{A.204}$$

where Eq.(A.201) is because $\tilde{\boldsymbol{G}}^{-1}_{V_{\tilde{t}_\tau}, \tilde{t}_\tau - 1} \succeq \boldsymbol{M}^{-1}_{V_{\tilde{t}_\tau}, \tilde{t}_\tau - 1}$ in Eq.(A.202) we use Cauchy–Schwarz inequality, in Eq.(A.203) we use Lemma A.4.2 and $\sum_{\tau=1}^{l_1} \mathbb{I}\{i_{\tilde{t}_\tau} \in V_j\} \leq T_{V_j,T}$, in Eq.(A.204) we use Cauchy–Schwarz inequality and $\sum_{j=1}^{m} T_{V_j,T} = T$.

For the second part in Eq.(A.197), Let $\boldsymbol{x}'_{a_{t'_\tau}} \triangleq \sqrt{w_{i_{t'_\tau}, t'_\tau}} \boldsymbol{x}_{a_{t'_\tau}}$, then

$$\sum_{t: w_{i_t,t} < 1} \left\|\boldsymbol{x}_{\boldsymbol{a_t}}\right\|_{\boldsymbol{M}^{-1}_{V_t, t-1}} = \sum_{t: w_{i_t,t} < 1} \frac{\left\|\boldsymbol{x}_{\boldsymbol{a_t}}\right\|^2_{\boldsymbol{M}^{-1}_{V_t,t-1}}}{\left\|\boldsymbol{x}_{\boldsymbol{a_t}}\right\|_{\boldsymbol{M}^{-1}_{V_t,t-1}}} = \sum_{t: w_{i_t,t} < 1} \frac{w_{i_t,t} \left\|\boldsymbol{x}_{\boldsymbol{a_t}}\right\|^2_{\boldsymbol{M}^{-1}_{V_t,t-1}}}{\alpha} \tag{A.205}$$

$$= \sum_{j=1}^{m} \sum_{\tau=1}^{l_2} \mathbb{I}\{i_{t'_\tau} \in V_j\} \frac{w_{i_{t'_\tau}, t'_\tau}}{\alpha} \left\|\boldsymbol{x}_{a_{t'_\tau}}\right\|^2_{\boldsymbol{M}^{-1}_{V_{t'_\tau}, t'_\tau - 1}}$$

$$\leq \sum_{j=1}^{m} \frac{\sum_{\tau=1}^{l_2} \min\{1, \mathbb{I}\{i_{t'_\tau} \in V_j\} \left\|\boldsymbol{x}'_{a_{t'_\tau}}\right\|^2_{\boldsymbol{G}'^{-1}_{V_{t'_\tau}, t'_\tau - 1}}\}}{\alpha} \tag{A.206}$$

$$\leq \sum_{j=1}^{m} \frac{2d\log(1+\frac{T}{\lambda d})}{\alpha} = \frac{2md\log(1+\frac{T}{\lambda d})}{\alpha} \tag{A.207}$$

where in Eq.(A.205) we use the definition of the weights, in Eq.(A.206) we use $\boldsymbol{G}'^{-1}_{V_{t'_\tau},t'_\tau-1} \succeq \boldsymbol{M}^{-1}_{V_{t'_\tau},t'_\tau-1}$, and Eq.(A.207) uses Lemma A.4.2.

Then, with Eq.(A.207), Eq.(A.204), Eq.(A.197), Eq.(A.193), Eq.(A.196), $\delta = \frac{1}{T}$, and $\beta = \sqrt{\lambda}+\sqrt{2\log(T)+d\log(1+\frac{T}{\lambda d})}+\alpha C$, we can get

$$\begin{aligned}
R(T) &\leq 4 + T_0 + \big(2\sqrt{\lambda} + \sqrt{2\log(T)+d\log(1+\frac{T}{\lambda d})} + \alpha C\big) \times \Bigg(\sqrt{2mdT\log(1+\frac{T}{\lambda d})} \\
&\quad + \frac{2md\log(1+\frac{T}{\lambda d})}{\alpha}\Bigg) \\
&= 4 + 16u\log(uT) + 4u\max\{\frac{288d}{\gamma^2\alpha\sqrt{\lambda}\tilde{\lambda}_x}\log(uT), \frac{16}{\tilde{\lambda}_x^2}\log(\frac{8dT}{\tilde{\lambda}_x^2}), \frac{72\sqrt{\lambda}}{\alpha\gamma^2\tilde{\lambda}_x}, \frac{72\alpha C^2}{\gamma^2\sqrt{\lambda}\tilde{\lambda}_x}\} \\
&\quad + \big(2\sqrt{\lambda} + \sqrt{2\log(T)+d\log(1+\frac{T}{\lambda d})} + \alpha C\big) \times \Bigg(\sqrt{2mdT\log(1+\frac{T}{\lambda d})} \\
&\quad + \frac{2md\log(1+\frac{T}{\lambda d})}{\alpha}\Bigg).
\end{aligned}$$

Picking $\alpha = \frac{\sqrt{\lambda}+\sqrt{d}}{C}$, we can get

$$R(T) \leq O\big((\frac{C\sqrt{d}}{\gamma^2\tilde{\lambda}_x}+\frac{1}{\tilde{\lambda}_x^2})u\log(T)\big)+O\big(d\sqrt{mT}\log(T)\big)+O\big(mCd\log^{1.5}(T)\big)\,. \tag{A.208}$$

Thus we complete the proof of Theorem 6.4.3.

A.4.4 Proof and Discussions of Theorem 6.4.4

Table 1 of the work [51] gives a lower bound for linear bandits with adversarial corruption for a single user. The lower bound of $R(T)$ is given by: $R(T) \geq \Omega(d\sqrt{T}+dC)$. Therefore, suppose

our problem with multiple users and m underlying clusters where the arrival times are T_i for each cluster, then for any algorithms, even if they know the underlying clustering structure and keep m independent linear bandit algorithms to leverage the common information of clusters, the best they can get is $R(T) \geq dC + \sum_{i\in[m]} d\sqrt{T_i}$. For a special case where $T_i = \frac{T}{m}, \forall i \in [m]$, we can get $R(T) \geq dC + \sum_{i\in[m]} d\sqrt{\frac{T}{m}} = d\sqrt{mT} + dC$, which gives a lower bound of $\Omega(d\sqrt{mT} + dC)$ for the LOCUD problem.

Recall that the regret upper bound of RCLUB-WCU shown in Theorem 6.4.3 is of $O\bigg((\frac{C\sqrt{d}}{\gamma^2\tilde{\lambda}_x} + \frac{1}{\tilde{\lambda}_x^2})u\log(T)\bigg) + O\big(d\sqrt{mT}\log(T)\big) + O\big(mCd\log^{1.5}(T)\big)$, asymptotically matching this lower bound with respect to T up to logarithmic factors and with respect to C up to $O(\sqrt{m})$ factors, showing the tightness of our theoretical results (where m are typically very small for real applications).

We conjecture that the gap for the m factor in the mC term of the lower bound is due to the strong assumption that cluster structures are known to prove our lower bound, and whether there exists a tighter lower bound will be left for future work.

A.4.5 Proof of Theorem 6.4.5

We prove the theorem using the proof by contrapositive. Specifically, in Theorem 6.4.5, we need to prove that for any $t \geq T_0$, if the detection condition in Line 7 of Algo.9 for user i, then with probability at least $1 - 5\delta$, user i is indeed a corrupted user. By the proof by contrapositive, we can prove Theorem 6.4.5 by showing that: for any $t \geq T_0$, if user i is a normal user, then with probability at least $1 - 5\delta$, the detection condition in Line 7 of

Algo.9 will not be satisfied for user i.

If the clustering structure is correct at t, then for any normal user i

$$\tilde{\boldsymbol{\theta}}_{i,t} - \hat{\boldsymbol{\theta}}_{V_{i,t},t} = \tilde{\boldsymbol{\theta}}_{i,t} - \boldsymbol{\theta}_i + \boldsymbol{\theta}_i - \hat{\boldsymbol{\theta}}_{V_{i,t},t}\,, \tag{A.209}$$

where $\tilde{\boldsymbol{\theta}}_{i,t}$ is the non-robust estimation of the ground-truth θ_i, and $\hat{\boldsymbol{\theta}}_{V_{i,t},t-1}$ is the robust estimation of the inferred cluster $V_{i,t}$ for user i at round t. Since the clustering structure is correct at t, $\hat{\boldsymbol{\theta}}_{V_{i,t},t-1}$ is the robust estimation of user i's ground-truth cluster's preference vector $\boldsymbol{\theta}^{j(i)} = \boldsymbol{\theta}_i$ at round t.

We have

$$\begin{aligned}
\tilde{\boldsymbol{\theta}}_{i,t} - \boldsymbol{\theta}_i &= (\lambda \boldsymbol{I} + \tilde{\boldsymbol{M}}_{i,t})^{-1}\tilde{\boldsymbol{b}}_{i,t} - \boldsymbol{\theta}_i \\
&= (\lambda \boldsymbol{I} + \sum_{\substack{s\in[t]\\ i_s=i}} \boldsymbol{x}_{a_s}\boldsymbol{x}_{a_s}^\top)^{-1}(\sum_{\substack{s\in[t]\\ i_s=i}} \boldsymbol{x}_{a_s} r_s) - \boldsymbol{\theta}_i \\
&= (\lambda \boldsymbol{I} + \sum_{\substack{s\in[t]\\ i_s=i}} \boldsymbol{x}_{a_s}\boldsymbol{x}_{a_s}^\top)^{-1}\big(\sum_{\substack{s\in[t]\\ i_s=i}} \boldsymbol{x}_{a_s}(\boldsymbol{x}_{a_s}^\top\boldsymbol{\theta}_i + \eta_s)\big) - \boldsymbol{\theta}_i \\
&= (\lambda \boldsymbol{I} + \sum_{\substack{s\in[t]\\ i_s=i}} \boldsymbol{x}_{a_s}\boldsymbol{x}_{a_s}^\top)^{-1}\big((\lambda \boldsymbol{I} + \sum_{\substack{s\in[t]\\ i_s=i}} \boldsymbol{x}_{a_s}\boldsymbol{x}_{a_s}^\top)\boldsymbol{\theta}_i - \lambda\boldsymbol{\theta}_i + \sum_{\substack{s\in[t]\\ i_s=i}} \boldsymbol{x}_{a_s}\eta_s\big) - \boldsymbol{\theta}_i \\
&= -\lambda \tilde{\boldsymbol{M}}'^{-1}_{i,t}\boldsymbol{\theta}_i + \tilde{\boldsymbol{M}}'^{-1}_{i,t}\sum_{\substack{s\in[t]\\ i_s=i}} \boldsymbol{x}_{a_s}\eta_s\,,
\end{aligned} \tag{A.210}$$

where we denote $\tilde{\boldsymbol{M}}'_{i,t} \triangleq \lambda \boldsymbol{I} + \sum_{\substack{s\in[t]\\ i_s=i}} \boldsymbol{x}_{a_s}\boldsymbol{x}_{a_s}^\top$, and Eq.(A.210) is because since user i is normal, we have $c_s = 0, \forall s : i_s = i$.

Then, we have

$$\left\|\tilde{\boldsymbol{\theta}}_{i,t} - \boldsymbol{\theta}_i\right\|_2 \le \left\|\lambda \tilde{\boldsymbol{M}}'^{-1}_{i,t}\boldsymbol{\theta}_i\right\|_2 + \left\|\tilde{\boldsymbol{M}}'^{-1}_{i,t}\sum_{\substack{s\in[t]\\ i_s=i}} \boldsymbol{x}_{a_s}\eta_s\right\|_2$$

$$\leq \lambda \left\| \tilde{\boldsymbol{M}}_{i,t}'^{-\frac{1}{2}} \right\|_2^2 \|\boldsymbol{\theta}_i\|_2 + \left\| \tilde{\boldsymbol{M}}_{i,t}'^{-\frac{1}{2}} \sum_{\substack{s\in[t]\\ i_s=i}} \boldsymbol{x}_{a_s}\eta_s \right\|_2 \left\| \tilde{\boldsymbol{M}}_{i,t}'^{-\frac{1}{2}} \right\|_2 \tag{A.211}$$

$$\leq \frac{\sqrt{\lambda} + \left\| \sum_{\substack{s\in[t]\\ i_s=i}} \boldsymbol{x}_{a_s}\eta_s \right\|_{\tilde{\boldsymbol{M}}_{i,t}'^{-1}}}{\sqrt{\lambda_{\min}(\tilde{\boldsymbol{M}}_{i,t}')}}, , \tag{A.212}$$

where Eq.(A.211) follows by the Cauchy–Schwarz inequality and the inequality for the operator norm of matrices, and Eq.(A.212) follows by the Courant-Fischer theorem and the fact that $\lambda_{\min}(\tilde{\boldsymbol{M}}_{i,t}') \geq \lambda$.

Following Theorem 1 in [43], for a fixed normal user i, with probability at least $1-\delta$ for some $\delta \in (0,1)$ we have:

$$\begin{aligned} \left\| \sum_{\substack{s\in[t]\\ i_s=i}} \boldsymbol{x}_{a_s}\eta_s \right\|_{\tilde{\boldsymbol{M}}_{i,t}'^{-1}} &\leq \sqrt{2\log(\frac{1}{\delta}) + \log(\frac{\det(\tilde{\boldsymbol{M}}_{i,t}')}{\det(\lambda \boldsymbol{I})})} \\ &\leq \sqrt{2\log(\frac{1}{\delta}) + d\log(1+\frac{T_{i,t}}{\lambda d})}\,, \end{aligned} \tag{A.213}$$

where Eq.(A.213) is because $\det(\tilde{\boldsymbol{M}}_{i,t}') \leq \left(\frac{\mathrm{trace}(\lambda \boldsymbol{I} + \sum_{\substack{s\in[t]\\ i_s=i}} \boldsymbol{x}_{a_s}\boldsymbol{x}_{a_s}^\top)}{d} \right)^d \leq \left(\frac{\lambda d + T_{i,t}}{d}\right)^d$, and $\det(\lambda \boldsymbol{I}) = \lambda^d$.

Plugging this into Eq.(A.212), we can get

$$\left\| \tilde{\boldsymbol{\theta}}_{i,t} - \boldsymbol{\theta}_i \right\|_2 \leq \frac{\sqrt{\lambda} + \sqrt{2\log(\frac{1}{\delta}) + d\log(1+\frac{T_{i,t}}{\lambda d})}}{\sqrt{\lambda_{\min}(\tilde{\boldsymbol{M}}_{i,t}')}}\,. \tag{A.214}$$

Then we need to bound $\left\| \boldsymbol{\theta}_i - \hat{\boldsymbol{\theta}}_{V_{i,t},t} \right\|_2$. With the correct clus-

tering, $V_{i,t} = V_{j(i)}$, we have

$$\begin{aligned}
\hat{\boldsymbol{\theta}}_{V_{i,t},t} - \boldsymbol{\theta}_i &= \boldsymbol{M}_{V_{i,t},t}^{-1} \boldsymbol{b}_{V_{j,t},t} \\
&= (\lambda \boldsymbol{I} + \sum_{\substack{s\in[t] \\ i_s \in V_{j(i)}}} w_{i_s,s} \boldsymbol{x}_{a_s} \boldsymbol{x}_{a_s}^\top)^{-1} (\sum_{\substack{s\in[t] \\ i_s \in V_{j(i)}}} w_{i_s,s} \boldsymbol{x}_{a_s} r_s) - \boldsymbol{\theta}_i \\
&= (\lambda \boldsymbol{I} + \sum_{\substack{s\in[t] \\ i_s \in V_{j(i)}}} w_{i_s,s} \boldsymbol{x}_{a_s} \boldsymbol{x}_{a_s}^\top)^{-1} (\sum_{\substack{s\in[t] \\ i_s \in V_{j(i)}}} w_{i_s,s} \boldsymbol{x}_{a_s} (\boldsymbol{x}_{a_s}^\top \boldsymbol{\theta}_i + \eta_s + c_s))) - \theta_i
\end{aligned} \tag{A.215}$$

$$\begin{aligned}
&= (\lambda \boldsymbol{I} + \sum_{\substack{s\in[t] \\ i_s \in V_{j(i)}}} w_{i_s,s} \boldsymbol{x}_{a_s} \boldsymbol{x}_{a_s}^\top)^{-1} \big((\lambda \boldsymbol{I} + \sum_{\substack{s\in[t] \\ i_s \in V_{j(i)}}} w_{i_s,s} \boldsymbol{x}_{a_s} \boldsymbol{x}_{a_s}^\top) \boldsymbol{\theta}_i - \lambda \boldsymbol{\theta}_i \\
&\quad + \sum_{\substack{s\in[t] \\ i_s \in V_{j(i)}}} w_{i_s,s} \boldsymbol{x}_{a_s} \eta_s + \sum_{\substack{s\in[t] \\ i_s \in V_{j(i)}}} w_{i_s,s} \boldsymbol{x}_{a_s} c_s) \big) - \theta_i \\
&= -\lambda \boldsymbol{M}_{V_{i,t},t}^{-1} \boldsymbol{\theta}_i + \boldsymbol{M}_{V_{i,t},t}^{-1} \sum_{\substack{s\in[t] \\ i_s \in V_{j(i)}}} w_{i_s,s} \boldsymbol{x}_{a_s} \eta_s + \boldsymbol{M}_{V_{i,t},t}^{-1} \sum_{\substack{s\in[t] \\ i_s \in V_{j(i)}}} w_{i_s,s} \boldsymbol{x}_{a_s} c_s \,.
\end{aligned} \tag{A.216}$$

Therefore, we have

$$\begin{aligned}
&\left\| \boldsymbol{\theta}_i - \hat{\boldsymbol{\theta}}_{V_{i,t},t} \right\|_2 \\
&\leq \lambda \left\| \boldsymbol{M}_{V_{i,t},t}^{-1} \boldsymbol{\theta}_i \right\|_2 + \left\| \boldsymbol{M}_{V_{i,t},t}^{-1} \sum_{\substack{s\in[t] \\ i_s \in V_{j(i)}}} w_{i_s,s} \boldsymbol{x}_{a_s} \eta_s \right\|_2 + \left\| \boldsymbol{M}_{V_{i,t},t}^{-1} \sum_{\substack{s\in[t] \\ i_s \in V_{j(i)}}} w_{i_s,s} \boldsymbol{x}_{a_s} c_s \right\|_2 \\
&\leq \lambda \left\| \boldsymbol{M}_{V_{i,t},t}^{-\frac{1}{2}} \right\|_2^2 \left\| \boldsymbol{\theta}_i \right\|_2 + \left\| \boldsymbol{M}_{V_{i,t},t}^{-\frac{1}{2}} \sum_{\substack{s\in[t] \\ i_s \in V_{j(i)}}} w_{i_s,s} \boldsymbol{x}_{a_s} \eta_s \right\|_2 \left\| \boldsymbol{M}_{V_{i,t},t}^{-\frac{1}{2}} \right\|_2
\end{aligned}$$

$$+\left\|\boldsymbol{M}_{V_{i,t},t}^{-\frac{1}{2}}\sum_{\substack{s\in[t]\\ i_s\in V_{j(i)}}} w_{i_s,s}\boldsymbol{x}_{a_s}\eta_s\right\|_2\left\|\boldsymbol{M}_{V_{i,t},t}^{-\frac{1}{2}}\right\|_2 \tag{A.217}$$

$$\leq \frac{\sqrt{\lambda}+\left\|\sum_{\substack{s\in[t]\\ i_s\in V_{j(i)}}} w_{i_s,s}\boldsymbol{x}_{a_s}\eta_s\right\|_{\boldsymbol{M}_{V_{i,t},t}^{-1}}+\left\|\sum_{\substack{s\in[t]\\ i_s\in V_{j(i)}}} w_{i_s,s}\boldsymbol{x}_{a_s}c_s\right\|_{\boldsymbol{M}_{V_{i,t},t}^{-1}}}{\sqrt{\lambda_{\min}(\boldsymbol{M}_{V_{i,t},t})}} \tag{A.218}$$

Let $\tilde{\boldsymbol{x}}_s \triangleq \sqrt{w_{i_s,s}}\boldsymbol{x}_{a_s}$, $\tilde{\eta}_s \triangleq \sqrt{w_{i_s,s}}\eta_s$, then we have: $\|\tilde{\boldsymbol{x}}_s\|_2 \leq \left\|\sqrt{w_{i_s,s}}\right\|_2\|\boldsymbol{x}_{a_s}\|_2 \leq 1$, $\tilde{\eta}_s$ is still 1-sub-gaussian (since η_s is 1-sub-gaussian and $\sqrt{w_{i_s,s}} \leq 1$), $\boldsymbol{M}_{V_{i,t},t} = \lambda\boldsymbol{I} + \sum_{\substack{s\in[t]\\ i_s\in V_{j(i)}}} \tilde{\boldsymbol{x}}_s\tilde{\boldsymbol{x}}_s^\top$, and $\left\|\sum_{\substack{s\in[t]\\ i_s\in V_{j(i)}}} w_{i_s,s}\boldsymbol{x}_{a_s}\eta_s\right\|_{\boldsymbol{M}_{V_{i,t},t}^{-1}}$ becomes $\left\|\sum_{\substack{s\in[t]\\ i_s\in V_{j(i)}}} \tilde{\boldsymbol{x}}_s\tilde{\eta}_s\right\|_{\boldsymbol{M}_{V_{i,t},t}^{-1}}$. Then, following Theorem 1 in [43], with probability at least $1-\delta$ for some $\delta \in (0,1)$, for a fixed normal user i, we have

$$\begin{aligned}\left\|\sum_{\substack{s\in[t]\\ i_s\in V_{j(i)}}} w_{i_s,s}\boldsymbol{x}_{a_s}\eta_s\right\|_{\boldsymbol{M}_{V_{i,t},t}^{-1}} &\leq \sqrt{2\log(\frac{1}{\delta})+\log(\frac{\det(\boldsymbol{M}_{V_{i,t},t})}{\det(\lambda\boldsymbol{I})})}\\ &\leq \sqrt{2\log(\frac{1}{\delta})+d\log(1+\frac{T_{V_{i,t},t}}{\lambda d})}\,,\end{aligned} \tag{A.219}$$

where Eq.(A.213) is because $\det(\boldsymbol{M}_{V_{i,t},t}) \leq \left(\frac{\mathrm{trace}(\lambda\boldsymbol{I}+\sum_{\substack{s\in[t]\\ i_s\in V_{j(i)}}}\boldsymbol{x}_{a_s}\boldsymbol{x}_{a_s}^\top)}{d}\right)^d \leq \left(\frac{\lambda d+T_{V_{i,t},t}}{d}\right)^d$, and $\det(\lambda\boldsymbol{I}) = \lambda^d$.

For $\left\|\sum_{\substack{s\in[t]\\ i_s\in V_{j(i)}}} w_{i_s,s}\boldsymbol{x}_{a_s}c_s\right\|_{\boldsymbol{M}_{V_{i,t},t}^{-1}}$, we have

$$\begin{aligned}\left\|\sum_{\substack{s\in[t]\\ i_s\in V_{j(i)}}} w_{i_s,s}\boldsymbol{x}_{a_s}c_s\right\|_{\boldsymbol{M}_{V_{i,t},t}^{-1}} &\leq \sum_{\substack{s\in[t]\\ i_s\in V_{j(i)}}} |c_s|\, w_{i_s,s}\left\|\boldsymbol{x}_{a_s}\right\|_{\boldsymbol{M}_{V_{i,t},t}^{-1}}\\ &\leq \alpha C\,, \end{aligned} \tag{A.220}$$

where Eq.(A.220) is because $w_{i_s,s} \leq \frac{\alpha}{\|\boldsymbol{x}_{a_s}\|_{\boldsymbol{M}'^{-1}_{i_s,s}}} \leq \frac{\alpha}{\|\boldsymbol{x}_{a_s}\|_{\boldsymbol{M}'^{-1}_{i_s,t}}} \leq \frac{\alpha}{\|\boldsymbol{x}_{a_s}\|_{\boldsymbol{M}_{V_{i,t},t}^{-1}}}$ (since $\boldsymbol{M}_{V_{i,t},t} \succeq \boldsymbol{M}'_{i_s,t} \succeq \boldsymbol{M}'_{i_s,s}$, $\boldsymbol{M}'^{-1}_{i_s,s} \succeq \boldsymbol{M}'^{-1}_{i_s,t} \succeq \boldsymbol{M}^{-1}_{V_{i,t},t}$, $\|\boldsymbol{x}_{a_s}\|_{\boldsymbol{M}'^{-1}_{i_s,s}} \geq \|\boldsymbol{x}_{a_s}\|_{\boldsymbol{M}'^{-1}_{i_s,t}} \geq \|\boldsymbol{x}_{a_s}\|_{\boldsymbol{M}^{-1}_{V_{i,t},t}}$), and $\sum_{s\in[t]}|c_s| \leq C$.

Therefore, we have

$$\left\|\boldsymbol{\theta}_i - \hat{\boldsymbol{\theta}}_{V_{i,t},t}\right\|_2 \leq \frac{\sqrt{\lambda}+\sqrt{2\log(\frac{1}{\delta})+d\log(1+\frac{T_{V_{i,t},t}}{\lambda d})}+\alpha C}{\sqrt{\lambda_{\min}(\boldsymbol{M}_{V_{i,t},t})}}\,. \tag{A.221}$$

With Eq.(A.221), Eq.(A.214) and Eq.(A.209), together with Lemma A.3.2, we have that for a normal user i, for any $t \geq T_0$, with probability at least $1-5\delta$ for some $\delta \in (0, \frac{1}{5})$

$$\begin{aligned}&\left\|\tilde{\boldsymbol{\theta}}_{i,t} - \hat{\boldsymbol{\theta}}_{V_{i,t},t}\right\|\\ &\leq \left\|\tilde{\boldsymbol{\theta}}_{i,t} - \boldsymbol{\theta}_i\right\|_2 + \left\|\boldsymbol{\theta}_i - \hat{\boldsymbol{\theta}}_{V_{i,t},t}\right\|_2\\ &\leq \frac{\sqrt{\lambda}+\sqrt{2\log(\frac{1}{\delta})+d\log(1+\frac{T_{i,t}}{\lambda d})}}{\sqrt{\lambda_{\min}(\tilde{\boldsymbol{M}}'_{i,t})}} + \frac{\sqrt{\lambda}+\sqrt{2\log(\frac{1}{\delta})+d\log(1+\frac{T_{V_{i,t},t}}{\lambda d})}+\alpha C}{\sqrt{\lambda_{\min}(\boldsymbol{M}_{V_{i,t},t})}}\,,\end{aligned} \tag{A.222}$$

which is exactly the detection condition in Line 7 of Algo.9.

Therefore, by the proof by contrapositive, we complete the proof of Theorem 6.4.5.

A.4.6 Description of Baselines

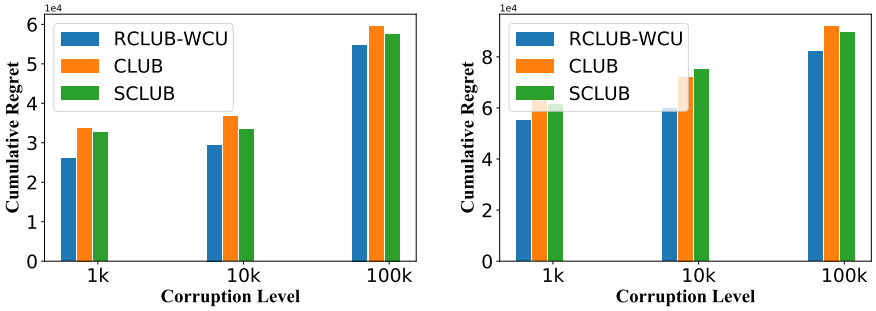
We compare RCLUB-WCU to the following five baselines for recommendations.

- LinUCB[41]: A state-of-the-art bandit approach for a single user without corruption.
- LinUCB-Ind: Use a separate LinUCB for each user.
- CW-OFUL[51]: A state-of-the-art bandit approach for single user with corruption.
- CW-OFUL-Ind: Use a separate CW-OFUL for each user.
- CLUB[46]: A graph-based clustering of bandits approach for multiple users without corruption.
- SCLUB[237]: A set-based clustering of bandits approach for multiple users without corruption.

A.4.7 More Experiments

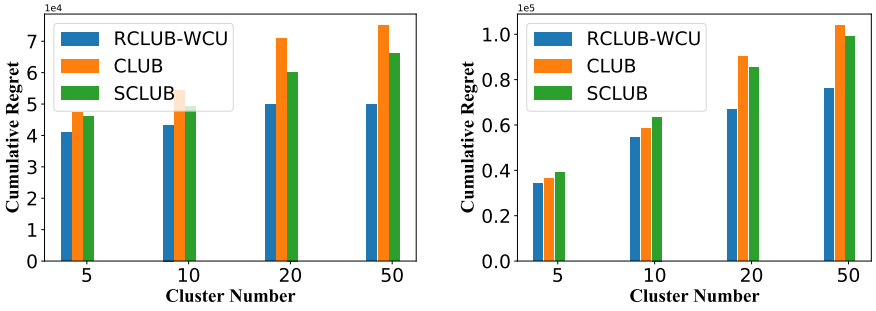
A.4.7.1 Different Corruption Levels

To see our algorithm's performance under different corruption levels, we conduct the experiments under different corruption levels for RCLUB-WCU, CLUB, and SCLUB on Amazon and Yelp datasets. Recall the corruption mechanism in Section 6.5.1, we set k as 1,000; 10,000; 100,000. The results are shown in Fig.A.2. All the algorithms' performance becomes worse when the corruption level increases. But RCLUB-WCU is much robust than the baselines.



(a) Amazon Corruption Level (b) Yelp Corruption Level

Figure A.2: Cumulative regret in different corruption levels



(a) Amazon Cluster Number (b) Yelp Cluster Number

Figure A.3: Cumulative regret with different cluster numbers

A.4.7.2 Different Cluster numbers

Following [135], we test the performances of the cluster-based algorithms (RCLUB-WCU, CLUB, SCLUB) when the underlying cluster number changes. We set m as 5, 10, 20, and 50. The results are shown in Fig.A.3. All these algorithms' performances decrease when the cluster numbers increase, matching our theoretical results. The performances of CLUB and SCLUB decrease much faster than RCLUB-WCU, indicating that RCLUB-WCU is more robust when the underlying user cluster number changes.

A.5 Appendix of Chapter 7

A.5.1 Proof of Lemma 7.3.1

Proof. According to the closed-form solution of $\boldsymbol{\theta}_t$ in Eq. (7.5) (7.6), we can calculate the estimation error as follows

$$
\begin{aligned}
\boldsymbol{\theta}_t - \boldsymbol{\theta}_* &= \boldsymbol{M}_t^{-1}\boldsymbol{b}_t - \boldsymbol{\theta}_* \\
&= \left(\sum_{\tau=1}^{t-1} \boldsymbol{x}_{a_\tau}\boldsymbol{x}_{a_\tau}^\top + \sum_{\tau=1}^{t}\sum_{k\in\mathcal{K}_\tau} \tilde{\boldsymbol{x}}_k\tilde{\boldsymbol{x}}_k^\top + \beta\boldsymbol{I}\right)^{-1}\left(\sum_{\tau=1}^{t-1}\boldsymbol{x}_{a_\tau} r_{a_\tau,\tau} + \sum_{\tau=1}^{t}\sum_{k\in\mathcal{K}_\tau}\tilde{\boldsymbol{x}}_k\tilde{r}_{k,\tau}\right) - \boldsymbol{\theta}_* \\
&= \left(\sum_{\tau=1}^{t-1} \boldsymbol{x}_{a_\tau}\boldsymbol{x}_{a_\tau}^\top + \sum_{\tau=1}^{t}\sum_{k\in\mathcal{K}_\tau} \tilde{\boldsymbol{x}}_k\tilde{\boldsymbol{x}}_k^\top + \beta\boldsymbol{I}\right)^{-1}\left(\sum_{\tau=1}^{t-1}\boldsymbol{x}_{a_\tau}\left(\boldsymbol{x}_{a_\tau}^\top\boldsymbol{\theta}_* + \epsilon_\tau\right) + \sum_{\tau=1}^{t}\sum_{k\in\mathcal{K}_\tau}\tilde{\boldsymbol{x}}_k\left(\tilde{\boldsymbol{x}}_k^\top\boldsymbol{\theta}_* + \tilde{\epsilon}_\tau\right)\right) \\
&\quad - \boldsymbol{\theta}_* \\
&= \left(\sum_{\tau=1}^{t-1} \boldsymbol{x}_{a_\tau}\boldsymbol{x}_{a_\tau}^\top + \sum_{\tau=1}^{t}\sum_{k\in\mathcal{K}_\tau} \tilde{\boldsymbol{x}}_k\tilde{\boldsymbol{x}}_k^\top + \beta\boldsymbol{I}\right)^{-1}\left(\sum_{\tau=1}^{t-1} \boldsymbol{x}_{a_\tau}\boldsymbol{x}_{a_\tau}^\top + \sum_{\tau=1}^{t}\sum_{k\in\mathcal{K}_\tau} \tilde{\boldsymbol{x}}_k\tilde{\boldsymbol{x}}_k^\top + \beta\boldsymbol{I} - \beta\boldsymbol{I}\right)\boldsymbol{\theta}_* - \boldsymbol{\theta}_* \\
&\quad + \boldsymbol{M}_t^{-1}\left(\sum_{\tau=1}^{t-1}\boldsymbol{x}_{a_\tau}\epsilon_\tau + \sum_{\tau=1}^{t}\sum_{k\in\mathcal{K}_\tau}\tilde{\boldsymbol{x}}_k\tilde{\epsilon}_\tau\right) \\
&= -\beta\boldsymbol{M}_t^{-1}\boldsymbol{\theta}_* + \boldsymbol{M}_t^{-1}\left(\sum_{\tau=1}^{t-1}\boldsymbol{x}_{a_\tau}\epsilon_\tau + \sum_{\tau=1}^{t}\sum_{k\in\mathcal{K}_\tau}\tilde{\boldsymbol{x}}_k\tilde{\epsilon}_\tau\right).
\end{aligned}
$$

We can then bound the projection of the estimation error onto the direction of the action vector $\boldsymbol{x}_a$:

$$
\begin{aligned}
&\left|\boldsymbol{x}_a^\top(\boldsymbol{\theta}_t - \boldsymbol{\theta}_*)\right| \\
&\le \beta\left|\boldsymbol{x}_a^\top\boldsymbol{M}_t^{-1}\boldsymbol{\theta}_*\right| + \left|\boldsymbol{x}_a^\top\boldsymbol{M}_t^{-1}\left(\sum_{\tau=1}^{t-1}\boldsymbol{x}_{a_\tau}\epsilon_\tau + \sum_{\tau=1}^{t}\sum_{k\in\mathcal{K}_\tau}\tilde{\boldsymbol{x}}_k\tilde{\epsilon}_\tau\right)\right| \\
&\le \beta\left\|\boldsymbol{x}_a^\top\boldsymbol{M}_t^{-\frac{1}{2}}\right\|_2\left\|\boldsymbol{M}_t^{-\frac{1}{2}}\boldsymbol{\theta}_*\right\|_2 + \left\|\boldsymbol{x}_a^\top\boldsymbol{M}_t^{-\frac{1}{2}}\right\|_2 \times \left\|\boldsymbol{M}_t^{-\frac{1}{2}}\left(\sum_{\tau=1}^{t-1}\boldsymbol{x}_{a_\tau}\epsilon_\tau + \sum_{\tau=1}^{t}\sum_{k\in\mathcal{K}_\tau}\tilde{\boldsymbol{x}}_k\tilde{\epsilon}_\tau\right)\right\|_2
\end{aligned}
\tag{A.223}
$$

$$\leq \beta \left\|\boldsymbol{x}_a\right\|_{\boldsymbol{M}_t^{-1}} \left\|\boldsymbol{M}_t^{-\frac{1}{2}}\right\|_2 \left\|\boldsymbol{\theta}_*\right\|_2 + \left\|\boldsymbol{x}_a\right\|_{\boldsymbol{M}_t^{-1}} \left\|\sum_{\tau=1}^{t-1} \boldsymbol{x}_{a_\tau,\tau}\epsilon_\tau + \sum_{\tau=1}^{t}\sum_{k\in\mathcal{K}_\tau} \tilde{\boldsymbol{x}}_k\tilde{\epsilon}_\tau\right\|_{\boldsymbol{M}_t^{-1}} \tag{A.224}$$

$$\leq \left\|\boldsymbol{x}_a\right\|_{\boldsymbol{M}_t^{-1}} \left(\sqrt{\beta}\left\|\boldsymbol{\theta}_*\right\|_2 + \left\|\sum_{\tau=1}^{t-1} \boldsymbol{x}_{a_\tau}\epsilon_\tau + \sum_{\tau=1}^{t}\sum_{k\in\mathcal{K}_\tau} \tilde{\boldsymbol{x}}_k\tilde{\epsilon}_\tau\right\|_{\boldsymbol{M}_t^{-1}}\right), \tag{A.225}$$

where Eq. (A.223) is by the Cauchy–Schwarz inequality, Eq. (A.224) is by the inequality of the matrix operator norm, and Eq. (A.225) is because $\lambda_{min}(\boldsymbol{M}_t) \geq \beta$, $\left\|\boldsymbol{M}_t^{-\frac{1}{2}}\right\|_2 = \sqrt{\lambda_{max}(\boldsymbol{M}_t^{-1})} = \sqrt{\frac{1}{\lambda_{min}(\boldsymbol{M}_t)}} \leq \sqrt{\frac{1}{\beta}}$.

Theorem 1 in [43] suggests that with probability at least $1-\delta$

$$\left\|\sum_{\tau=1}^{t-1} \boldsymbol{x}_{a_\tau}\epsilon_\tau + \sum_{\tau=1}^{t}\sum_{k\in\mathcal{K}_\tau} \tilde{\boldsymbol{x}}_k\tilde{\epsilon}_k\right\|_{\boldsymbol{M}_t^{-1}} \leq \sqrt{2\log\left(\frac{det(\boldsymbol{M}_t)^{\frac{1}{2}}det(\beta\boldsymbol{I})^{\frac{1}{2}}}{\delta}\right)}, \tag{A.226}$$

where $det(\cdot)$ denotes the determinate of the argument.

We have

$$\begin{aligned}
&det(\boldsymbol{M}_t) = \prod_{i=1}^{d}\lambda_i \\
&\leq \left(\frac{\sum_{i=1}^{d}\lambda_i}{d}\right)^d && \text{(A.227)} \\
&= \left(\frac{trace(\boldsymbol{M}_t)}{d}\right)^d && \text{(A.228)} \\
&= \left(\frac{trace(\sum_{\tau=1}^{t-1}\boldsymbol{x}_{a_\tau}\boldsymbol{x}_{a_\tau}^\top + \sum_{\tau=1}^{t}\sum_{k\in\mathcal{K}_\tau}\tilde{\boldsymbol{x}}_k\tilde{\boldsymbol{x}}_k^\top + \beta\boldsymbol{I})}{d}\right)^d \\
&\leq \left(\frac{t+b(t)+\beta d}{d}\right)^d,
\end{aligned}$$

where $\lambda_i, i = 1, 2, \ldots, d$ denotes the eigenvalues of the matrix $\boldsymbol{M}_t$, $trace(\boldsymbol{M}_t)$ denotes the trace of $\boldsymbol{M}_t$, Eq. (A.227) follows by the inequality of arithmetic and geometric means, Eq. (A.228) follows since the trace of a matrix is equal to the sum of its eigenvalues.

Plugging the above inequality and $det(\beta \boldsymbol{I}) = \beta^d$ into Eq. (A.226), we can get

$$\left\| \sum_{\tau=1}^{t-1} \boldsymbol{x}_{a_\tau} \epsilon_\tau + \sum_{\tau=1}^{t} \sum_{k \in \mathcal{K}_\tau} \tilde{\boldsymbol{x}}_k \tilde{\epsilon}_k \right\|_{\boldsymbol{M}_t^{-1}} \leq \sqrt{2 \log(\frac{1}{\delta}) + d \log(1 + \frac{b(t) + t}{\beta d})} \,. \tag{A.229}$$

The result then follows by plugging Eq. (A.229) into Eq. (A.225), and the fact that $\|\boldsymbol{\theta}^*\|_2 \leq 1$. □

A.5.2 Proof of Lemma 7.3.2

Proof. Recall that in ConLinUCB-BS, the key-terms are uniformly sampled from the pre-computed barycentric spanner $\mathcal{B}$, i.e., $k \sim$ unif($\mathcal{B}$). Therefore we have

$$\lambda_{\mathcal{B}} \coloneqq \lambda_{\min}(\boldsymbol{E}_{k \sim \text{unif}(\mathcal{B})}[\tilde{\boldsymbol{x}}_k \tilde{\boldsymbol{x}}_k^\top]) > 0 \,. \tag{A.230}$$

Using Eq. (7.11) in the Lemma 7 in [135], and the fact that $b_t = b \cdot t$, then with probability at least $1 - \delta$ for $\delta \in (0, \frac{1}{8}]$, we have

$$\lambda_{\min}(\sum_{\tau=1}^{t} \sum_{k \in \mathcal{K}_\tau} \tilde{\boldsymbol{x}}_k \tilde{\boldsymbol{x}}_k^\top) \geq \frac{\lambda_{\mathcal{B}} bt}{2} \,, \tag{A.231}$$

for all $t \geq t_0 = \frac{256}{b\lambda_{\mathcal{B}}^2} \log(\frac{128d}{\lambda_{\mathcal{B}}^2 \delta})$.

Then, by Courant–Fischer theorem [292], the fact that $\|\boldsymbol{x}_a\|_2 = 1$, together with Eq. (A.231), we have that for any $t \geq t_0$,with

probability at least $1-\delta$ for $\delta \in (0, \frac{1}{8}]$,

$$\begin{aligned}
\|\boldsymbol{x}_a\|_{\boldsymbol{M}_t^{-1}} &= \sqrt{\boldsymbol{x}_a^\top \boldsymbol{M}_t^{-1} \boldsymbol{x}_a} \\
&\leq \max_{\substack{\boldsymbol{x}\in\mathbb{R}^d \\ \|\boldsymbol{x}\|_2=1}} \sqrt{\boldsymbol{x}^\top \boldsymbol{M}_t^{-1} \boldsymbol{x}} \\
&= \sqrt{\lambda_{\max}(\boldsymbol{M}_t^{-1})} \\
&= \sqrt{\lambda_{\max}\left((\sum_{\tau=1}^{t-1} \boldsymbol{x}_{a_\tau} \boldsymbol{x}_{a_\tau}^\top + \sum_{\tau=1}^{t} \sum_{k\in\mathcal{K}_\tau} \tilde{\boldsymbol{x}}_k \tilde{\boldsymbol{x}}_k^\top + \beta \boldsymbol{I})^{-1} \right)} \\
&= \sqrt{\frac{1}{\lambda_{\min}\left(\sum_{\tau=1}^{t-1} \boldsymbol{x}_{a_\tau} \boldsymbol{x}_{a_\tau}^\top + \sum_{\tau=1}^{t} \sum_{k\in\mathcal{K}_\tau} \tilde{\boldsymbol{x}}_k \tilde{\boldsymbol{x}}_k^\top + \beta \boldsymbol{I} \right)}} \\
&\leq \sqrt{\frac{1}{\lambda_{\min}(\sum_{\tau=1}^{t} \sum_{k\in\mathcal{K}_\tau} \tilde{\boldsymbol{x}}_k \tilde{\boldsymbol{x}}_k^\top)}} \\
&\leq \sqrt{\frac{2}{\lambda_{\mathcal{B}} b t}} .
\end{aligned}$$

□

A.5.3 Proof of Theorem 7.3.3

Proof. We denote the instantaneous regret at round t as R_t. With the definition of the cumulative regret given in Eq. (7.2), the arm selection strategy shown in Eq. (7.7) and Lemma 7.3.1, we can bound the regret R_t at each round $t = 1, 2, 3, 4, ..., T$ as follows

$$\begin{aligned}
R_t &= \boldsymbol{x}_{a_t^*}^\top \boldsymbol{\theta}^* - \boldsymbol{x}_{a_t}^\top \boldsymbol{\theta}^* \\
&= \boldsymbol{x}_{a_t^*}^\top (\boldsymbol{\theta}^* - \boldsymbol{\theta}_t) + (\boldsymbol{\theta}_t^\top \boldsymbol{x}_{a_t^*} + C_{a_t^*,t}) - (\boldsymbol{\theta}_t^\top \boldsymbol{x}_{a_t} + C_{a_t,t}) \\
&\quad + \boldsymbol{x}_{a_t}^\top (\boldsymbol{\theta}_t - \boldsymbol{\theta}^*) + C_{a_t,t} - C_{a_t^*,t} \\
&\leq 2C_{a_t,t} .
\end{aligned} \tag{A.232}$$

With Lemma 7.3.2, together with the assumption that $r_t \leq 1$ for any t, with probability at least $1-\delta$ for some $\delta \in (0, \frac{1}{4}]$, we can get

$$\begin{aligned}
R(T) &= R(\lceil t_0 \rceil) + \sum_{t=\lceil t_0 \rceil + 1}^{T} R_t \\
&\leq t_0 + 1 + 2 \sum_{t=\lceil t_0 \rceil}^{T} C_{a_t,t} \\
&\leq t_0 + 1 + 2\alpha_t \sum_{t=\lceil t_0 \rceil}^{T} \|\boldsymbol{x}_{a_t}\|_{\boldsymbol{M}_t^{-1}} \\
&\leq t_0 + 1 + 2\alpha_T \sum_{t=\lceil t_0 \rceil}^{T} \|\boldsymbol{x}_{a_t}\|_{\boldsymbol{M}_t^{-1}} \qquad \text{(A.233)} \\
&\leq t_0 + 1 + 2\alpha_T \sum_{t=\lceil t_0 \rceil}^{T} \sqrt{\frac{2}{\lambda_{\mathcal{B}} b t}} \\
&\leq t_0 + 1 + 2\alpha_T \sqrt{\frac{2}{\lambda_{\mathcal{B}} b}} \int_{t_0}^{T} \sqrt{\frac{1}{t}} dt \\
&=\leq t_0 + 1 + 4\alpha_T \sqrt{\frac{2}{\lambda_{\mathcal{B}} b}} (\sqrt{T} - \sqrt{t_0}) \\
&\leq t_0 + 1 + 4\alpha_T \sqrt{\frac{2}{\lambda_{\mathcal{B}} b}} \sqrt{T}\,,
\end{aligned}$$

where Eq. (A.233) follows since α_t is non-decreasing in t.

The result follows by plugging in the definition of t_0 and α_T.

□

A.5.4 Proof of Theorem 7.3.4

Proof. We first prove the following result:
For any two positive definite matrices $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}^{d \times d}$, and any

vector $\boldsymbol{x} \in \mathbb{R}^d$, we have:

$$\|\boldsymbol{x}\|^2_{(\boldsymbol{A}+\boldsymbol{B})^{-1}} \leq \|\boldsymbol{x}\|^2_{\boldsymbol{A}^{-1}} \ . \tag{A.234}$$

This result can be proved by the following arguments:

$$\begin{aligned}
\|\boldsymbol{x}\|^2_{(\boldsymbol{A}+\boldsymbol{B})^{-1}} &= \boldsymbol{x}^\top (\boldsymbol{A}+\boldsymbol{B})^{-1}\boldsymbol{x} \\
&= \boldsymbol{x}^\top \big(\boldsymbol{A}^{-1} - \boldsymbol{A}^{-1}(\boldsymbol{B}^{-1}+\boldsymbol{A}^{-1})^{-1}\boldsymbol{A}^{-1}\big)\boldsymbol{x} \qquad \text{(A.235)} \\
&= \boldsymbol{x}^\top \boldsymbol{A}^{-1}\boldsymbol{x} - (\boldsymbol{A}^{-1}\boldsymbol{x})^\top (\boldsymbol{B}^{-1}+\boldsymbol{A}^{-1})^{-1}(\boldsymbol{A}^{-1}\boldsymbol{x}) \\
&\leq \boldsymbol{x}^\top \boldsymbol{A}^{-1}\boldsymbol{x} = \|\boldsymbol{x}\|^2_{\boldsymbol{A}^{-1}} \ , \qquad \text{(A.236)}
\end{aligned}$$

where Eq. (A.235) follows from the Woodbury matrix identity [293], and Eq. (A.236) is because $(\boldsymbol{B}^{-1}+\boldsymbol{A}^{-1})^{-1}$ is a positive definite matrix.

With the above result, then following Eq. (A.232) and the Cauchy–Schwarz inequality, we can get

$$\begin{aligned}
R(T) &\leq 2\sum_{t=1}^{T} C_{a_t,t} \\
&= 2\alpha_t \sum_{t=1}^{T} \|\boldsymbol{x}_{a_t}\|_{\boldsymbol{M}_t^{-1}} \\
&\leq 2\alpha_T \sum_{t=1}^{T} \|\boldsymbol{x}_{a_t}\|_{\boldsymbol{M}_t^{-1}} \\
&\leq 2\alpha_T \sqrt{T\sum_{t=1}^{T} \|\boldsymbol{x}_{a_t}\|^2_{\boldsymbol{M}_t^{-1}}} \\
&\leq 2\alpha_T \sqrt{T\sum_{t=1}^{T} \|\boldsymbol{x}_{a_t}\|^2_{\boldsymbol{V}_t^{-1}}} \ .
\end{aligned} \tag{A.237}$$

Using Lemma 11 in [43], with probability at least $1-\delta$, we can

get

$$\sum_{t=1}^{T} \|\boldsymbol{x}_{a,t}\|^2_{\boldsymbol{V}_t^{-1}} \leq 2\log\left(\frac{det(\boldsymbol{V}_T)}{det(\beta\boldsymbol{I})}\right). \tag{A.238}$$

Following similar steps as in Eq. (A.228), we can get that

$$\frac{det(\boldsymbol{V}_T)}{det(\beta\boldsymbol{I})} \leq \big(\frac{T+\beta d}{\beta d}\big)^d. \tag{A.239}$$

Therefore we have

$$\sum_{t=1}^{T} \|\boldsymbol{x}_{a,t}\|^2_{\boldsymbol{V}_t^{-1}} \leq 2d\log(1+\frac{T+1}{\beta d}). \tag{A.240}$$

The result then follows by plugging in the definition of α_T and Eq. (A.240) into Eq. (A.237). □

A.6 Appendix for Chapter 8

A.6.1 Restarted SAVE$^+$-BOB

In this section, we provide the details of our proposed Restarted SAVE$^+$-BOB algorithm. The Restarted SAVE$^+$-BOB algorithm is summarized in Algo.18. We divide the K rounds into $\lceil \frac{K}{H} \rceil$ blocks, with each block having H rounds (except the last one may have less than H). Within each block i, we use a fixed (α_i, w_i) pair to run the Restarted SAVE$^+$ algorithm. To adaptively learn the optimal (α, w) pair without the knowledge of V_K and B_K, we employ an adversarial bandit algorithm (Exp3 in [183]) as the meta-learner to select α_i, w_i over time for $i \in \lceil \frac{K}{H} \rceil$ blocks. Specifically, in each block, the meta learner selects a (α, w) pair from the candidate pool to feed to Restarted SAVE$^+$, and the cumulative reward received by Restarted SAVE$^+$ within the block is fed to the meta-learner as the reward feedback to select a better pair for the next block.

We set H to be $\lceil d^{\frac{2}{5}} K^{\frac{2}{5}} \rceil$, and set the candidate pool of (α, w) pairs for the Exp3 algorithm as:

$$\mathcal{P} = \{(w, \alpha) : w \in \mathcal{W}, \alpha \in \mathcal{J}\}, \tag{A.241}$$

where

$$\mathcal{W} = \{w_i = d^{\frac{1}{3}} 2^{i-1} | i \in \lceil \frac{1}{3} \log_2 K \rceil + 1\} \cup \{w_i = d^{\frac{2}{5}} 2^{i-1} | i \in \lceil \frac{2}{5} \log_2 K \rceil + 1\}, \tag{A.242}$$

and

$$\mathcal{J} = \{\alpha_i = d^{\frac{1}{3}} 2^{-i+1} | i \in \lceil \frac{1}{3} \log_2 K \rceil + 1\} \cup \{\alpha_i = d^{\frac{11}{30}} 2^{-i+1} | i \in \lceil \frac{11}{30} \log_2 K \rceil + 1\}. \tag{A.243}$$

The algorithm also labels all the $|\mathcal{P}| = \left(\lceil \frac{1}{3}\log_2 K\rceil + \lceil \frac{2}{5}\log_2 K\rceil + 2\right)\cdot\left(\lceil \frac{1}{3}\log_2 K\rceil + \lceil \frac{11}{30}\log_2 K\rceil + 2\right)$ candidate pairs of parameters in $\mathcal{P}$,*i.e.*, $\mathcal{P} = \{(w_i, \alpha_i)\}_{i=1}^{|\mathcal{P}|}$. The algorithm initializes $\{s_{j,1}\}_{j=1}^{|\mathcal{P}|}$ to be $s_{j,1} = 1, \quad \forall j = 0, 1, \ldots, |\mathcal{P}|$, which means that at the beginning, the algorithm selects a pair from $\mathcal{P}$ uniformly at random. At the beginning of each block $i \in [\lceil K/H\rceil]$, the meta-learner (Exp3) calculates the distribution $(p_{j,i})_{j=1}^{|\mathcal{P}|}$ over the candidate set $\mathcal{P}$ by

$$p_{j,i} = (1-\gamma)\frac{s_{j,i}}{\sum_{u=1}^{|\mathcal{P}|} s_{u,i}} + \frac{\gamma}{|\mathcal{P}|+1}, \quad \forall j = 1, \ldots, |\mathcal{P}|\ , \quad \text{(A.244)}$$

where γ is defined as

$$\gamma = \min\left\{1, \sqrt{\frac{(|\mathcal{P}|+1)\ln(|\mathcal{P}|+1)}{(e-1)\lceil K/H\rceil}}\right\}. \quad \text{(A.245)}$$

Then, the meta-learner draws a j_i from the distribution $(p_{j,i})_{j=1}^{|\mathcal{P}|}$, and sets the pair of parameters in block i to be (w_{j_i}, α_{j_i}), and runs the base algorithm Algo.12 from scratch in this block with (w_{j_i}, α_{j_i}), then feeds the cumulative reward in the block $\sum_{k=(i-1)H+1}^{\min\{i\cdot H, K\}} r_k$ to the meta-learner. The meta-learner rescales $\sum_{k=(i-1)H+1}^{\min\{i\cdot H,K\}} r_k$ to $\frac{\sum_{k=(i-1)H+1}^{\min\{i\cdot H,K\}} r_k}{H+R\sqrt{\frac{H}{2}\log\left(K(\frac{K}{H}+1)\right)+\frac{2}{3}\cdot R\log\left(K(\frac{K}{H}+1)\right)}}$ to make it in the range $[0,1]$ with high probability (supported by Lemma A.6.9). The meta-learner updates the parameter $s_{j_i,i+1}$ to be

$$s_{j_i,i+1} = s_{j_i,i}\cdot\exp\left(\frac{\gamma}{(|\mathcal{P}|+1)p_{j_i,i}}\left(\frac{1}{2} + \frac{\sum_{k=(i-1)H+1}^{\min\{i\cdot H,K\}} r_k}{H+R\sqrt{\frac{H}{2}\log\left(K(\frac{K}{H}+1)\right)+\frac{2}{3}\cdot R\log\left(K(\frac{K}{H}+1)\right)}}\right)\right), \quad \text{(A.246)}$$

and keep others unchanged, *i.e.*, $s_{u,i+1} = s_{u,i},\ \forall u \neq j_i$. After that, the algorithm will go to the next block, and repeat the same process in block $i+1$.

Algorithm 18 Restarted SAVE$^+$-BOB

Require: total time rounds K; problem dimension d; noise upper bound R; $\alpha > 0$; the upper bound on the ℓ_2-norm of $\mathbf{a}$ in $\mathcal{D}_k (k \geq 1)$, i.e., A; the upper bound on the ℓ_2-norm of $\boldsymbol{\theta}_k$ $(k \geq 1)$, i.e., B.

1: Initialize $H = \lceil d^{\frac{2}{5}} K^{\frac{2}{5}} \rceil$; $\mathcal{P}$ as defined in Eq.(A.241), and index the $|\mathcal{P}| = \left(\lceil \frac{1}{3} \log_2 K \rceil + \lceil \frac{2}{5} \log_2 K \rceil + 2\right) \cdot \left(\lceil \frac{1}{3} \log_2 K \rceil + \lceil \frac{11}{30} \log_2 K \rceil + 2\right)$ items in $\mathcal{P}$, *i.e.*, $\mathcal{P} = \{(w_i, \alpha_i)\}_{i=1}^{|\mathcal{P}|}$; $\gamma = \min\left\{1, \sqrt{\frac{(|\mathcal{P}|+1)\ln(|\mathcal{P}|+1)}{(e-1)\lceil K/H \rceil}}\right\}$; $\{s_{j,1}\}_{j=1}^{|\mathcal{P}|}$ is set to $s_{j,1} = 1, \quad \forall j = 0, 1, \ldots, |\mathcal{P}|$.

2: **for** $i = 1, 2, \ldots, \lceil K/H \rceil$ **do**

3: Calculate the distribution $(p_{j,i})_{j=1}^{|\mathcal{P}|}$ by $p_{j,i} = (1-\gamma)\frac{s_{j,i}}{\sum_{u=1}^{|\mathcal{P}|} s_{u,i}} + \frac{\gamma}{|\mathcal{P}|+1}, \quad \forall j = 1, \ldots, |\mathcal{P}|$.

4: Set $j_i \leftarrow j$ with probability $p_{j,i}$, and $(w_i, \alpha_i) \leftarrow (w_{i_i}, \alpha_{j_i})$.

5: Run Algo.12 from scratch in block i (*i.e.*, in rounds $k = (i-1)H + 1, \ldots, \min\{i \cdot H, K\}$) with $(w, \alpha) = (w_i, \alpha_i)$.

6: Update $s_{j_i,i+1} = s_{j_i,i} \cdot \exp\left(\frac{\gamma}{(|\mathcal{P}|+1)p_{j_i,i}}\left(\frac{1}{2} + \frac{\sum_{k=(i-1)H+1}^{\min\{i\cdot H,K\}} r_k}{H + R\sqrt{\frac{H}{2}\log\left(K(\frac{K}{H}+1)\right)} + \frac{2}{3}\cdot R\log\left(K(\frac{K}{H}+1)\right)}\right)\right)$, and keep all the others unchanged, *i.e.*, $s_{u,i+1} = s_{u,i}, \ \forall u \neq j_i$.

7: **end for**

We have the following theorem to bound the regret of Restarted SAVE$^+$-BOB.

Theorem A.6.1. *By using the BOB framework with Exp3 as the meta-algorithm and Restarted SAVE$^+$ as the base algorithm, with the candidate pool $\mathcal{P}$ for Exp3 specified as in Eq.(A.241), Eq.(A.242), Eq.(A.243), and $H = \lceil d^{\frac{2}{5}} K^{\frac{2}{5}} \rceil$, then the regret of Restarted SAVE$^+$-BOB (Algo.18) satisfies*

$$\textit{Regret}(K) = \tilde{O}(d^{4/5} V_K^{2/5} B_K^{1/5} K^{2/5} + d^{2/3} B_K^{1/3} K^{2/3} + d^{2/5} K^{7/10}). \tag{A.247}$$

Proof. See Appendix A.6.7 for the full proof. □

Remark 21. *We discuss the regret of Algo.18 in Corollary 8.2 in the following special cases. In the case where the* total variance *is small,* i.e., $V_K = \tilde{O}(1)$*, assuming* $K^2 > d$*, our result becomes* $\tilde{O}(d^{2/3}B_K^{1/3}K^{2/3} + d^{1/5}K^{7/10})$*, when* $d^{14}B_K^{10} > K$*, it becomes* $\tilde{O}(d^{2/3}B_K^{1/3}K^{2/3})$*, better than all the previous results [253, 192, 254, 194]. In the worst case where* $V_K = O(K)$*, our result becomes* $\tilde{O}(d^{4/5}B_K^{1/5}K^{4/5})$*.*

A.6.2 Additional Experiment Setup

For Restarted-WeightedOFUL$^+$, we set $\lambda = 1$, $\hat{\beta}_k = 10$, $w = 1000$, and we grid search the variance parameters α and γ, both among values [1, 1.5, 2, 2.5, 3]. Finally we set $\alpha = 1$, and $\gamma = 2$. For Restarted SAVE$^+$ we set $w = 1000$, $\hat{\beta}_{k,\ell} = 2^{-\ell+1}$, and grid search L from 1 to 10 with stepsize of 1 and finally choose $L = 6$. For SW-UCB, we set $\lambda = 1$, $w = 1000$, $\beta_k = 10$. The Modified EXP3.S requires two parameters $\bar{\alpha}$ and $\bar{\gamma}$, and we set $\bar{\gamma} = 0.01$ and $\bar{\alpha} = \frac{1}{K}$.

To test the algorithms' performance under different total time horizons, we let K vary from 3×10^4 to 2.4×10^5, with a stepsize of 3×10^4, and plot the cumulative regret Regret(K) for these different total time step K. We set $B_K = 1, 10, 20,$ and $K^{1/3}$ to observe their performance in different levels of B_K.

A.6.3 Proof of Theorem 8.3.1

We prove the lower bound in Theorem 8.3.1 here. We need the following lemma from [203].

Lemma A.6.2 (Modification from Lemma 25, [203])**.** *Fix a pos-*

itive real $0 < \delta \leq 1/3$*, and positive integers* T, d *and assume that* $T \geq d^2/(2\delta)$*. Let* $\Delta = \sqrt{d\delta/T}/(4\sqrt{2})$ *and consider the linear bandit problems* $\mathcal{L}_{\boldsymbol{\mu}}$ *parameterized with a parameter vector* $\boldsymbol{\mu} \in \{-\Delta, \Delta\}^d$ *and action set* $\mathcal{A} = \{-1/\sqrt{d}, 1/\sqrt{d}\}^d$ *so that the reward distribution for taking action* $\mathbf{a} \in \mathcal{A}$ *is a Bernoulli distribution* $B(\delta + \langle \boldsymbol{\mu}^*, \mathbf{a} \rangle)$*. Then for any bandit algorithm* $\mathcal{B}$ *such that*

$$\mathbb{E}_{\boldsymbol{\mu} \sim Unif\{-\Delta,\Delta\}^d}[Regret(T, \mathcal{L}_{\boldsymbol{\mu}})] \geq \frac{d\sqrt{T\delta}}{8\sqrt{2}}. \tag{A.248}$$

Here $Regret(T, \mathcal{L}_{\boldsymbol{\mu}})$ *represents the regret under algorithm* $\mathcal{B}$ *on the instance* $\mathcal{L}_{\boldsymbol{\mu}}$*.*

Next we prove Theorem 8.3.1.

Proof of Theorem 8.3.1. Let $T < K$ be some constant to be defined. Let δ be a constant satisfying $2\delta \leq d^2/T$. We create $w = K/T$ number of linear bandit instances with the linear parameter $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_w$, where $\boldsymbol{\mu}_i \sim \{-\Delta, \Delta\}^d, \Delta = \sqrt{d\delta/T}/4\sqrt{2}$. Our nonstationary instance $\mathcal{L}_{\boldsymbol{\mu}_1,\ldots,\boldsymbol{\mu}_w}$ consists of $\mathcal{L}_{\boldsymbol{\mu}_1}, \ldots, \mathcal{L}_{\boldsymbol{\mu}_w}$, where at the step $i \cdot T + 1, \ldots, i \cdot T + T$, $\mathcal{L}_{\boldsymbol{\mu}_1,\ldots,\boldsymbol{\mu}_w}$ follows $\mathcal{L}_{\boldsymbol{\mu}_i}$. Then by the independence of μ_i, we have

$$\mathbb{E}_{\boldsymbol{\mu}_1,\ldots,\boldsymbol{\mu}_w \sim \text{Unif}\{-\Delta,\Delta\}^d} \text{Regret}(T, \mathcal{L}_{\boldsymbol{\mu}_1,\ldots,\boldsymbol{\mu}_w}) = \sum_{i=1}^{w} \mathbb{E}_{\boldsymbol{\mu}_i \sim \text{Unif}\{-\Delta,\Delta\}^d}[\text{Regret}(T, \mathcal{L}_{\boldsymbol{\mu}_i})]$$

$$\geq \frac{d\sqrt{T\delta}}{8\sqrt{2}} \cdot \frac{K}{T}. \tag{A.249}$$

Next we calculate the total variation and total variance for instance $\mathcal{L}_{\boldsymbol{\mu}_1,\ldots,\boldsymbol{\mu}_w}$. For each step, the reward distribution is a Bernoulli distribution $B(\delta + \langle \boldsymbol{\mu}_i, \mathbf{a} \rangle)$, whose variance is

$$(\delta + \langle \boldsymbol{\mu}_i, \mathbf{a} \rangle)(1 - \delta - \langle \boldsymbol{\mu}_i, \mathbf{a} \rangle) \leq (\delta + \langle \boldsymbol{\mu}_i, \mathbf{a} \rangle) \leq 2\delta, \tag{A.250}$$

where we use the fact $\sqrt{d}\Delta \leq \delta$. Therefore, the total variance over K steps is bounded by

$$V \leq 2K\delta. \tag{A.251}$$

Next, for the total variation, we have for any $k, k+1$ belong to the same $\boldsymbol{\mu}_i$, the variation of $\boldsymbol{\mu}$ is 0. Note that for any two different $\boldsymbol{\mu}_i, \boldsymbol{\mu}_j$, their difference is at most $\|\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\| \leq 2\sqrt{d \cdot \Delta^2}$, then the total variation is bounded by

$$B \leq \frac{K}{T} \cdot 2\Delta\sqrt{d} = \sqrt{d\delta/T}/(4\sqrt{2})\frac{K}{T} \cdot 2\sqrt{d} = \frac{dK\sqrt{\delta}}{2\sqrt{2T^3}}. \tag{A.252}$$

Then we select δ and T as

$$\begin{aligned} &\delta = \frac{V_K}{2K},\ T = \max\{\left(\frac{KV_K d^2}{16B_K^2}\right)^{1/3}, d^2K/V_K\}, \\ &\text{satisfying } 2K\delta \leq V_K, \frac{dK\sqrt{\delta}}{2\sqrt{2T^3}} \leq B_K,\ T \geq \frac{d^2}{2\delta}. \end{aligned} \tag{A.253}$$

We have the lower bound as

$$\mathbb{E}_{\boldsymbol{\mu}_1,\ldots,\boldsymbol{\mu}_w \sim \text{Unif}\{-\Delta,\Delta\}^d}\text{Regret}(T, \mathcal{L}_{\boldsymbol{\mu}_1,\ldots,\boldsymbol{\mu}_w}) \geq \Omega(d^{2/3}B_K^{1/3}V_K^{1/3}K^{1/3} \wedge V_K). \tag{A.254}$$

Therefore, there must exists $\boldsymbol{\mu}_1^*, \ldots, \boldsymbol{\mu}_w^*$, satisfying

$$\text{Regret}(T, \mathcal{L}_{\boldsymbol{\mu}_1^*,\ldots,\boldsymbol{\mu}_w^*}) \geq \Omega(d^{2/3}B_K^{1/3}V_K^{1/3}K^{1/3} \wedge V_K). \tag{A.255}$$

Finally, combining (A.255) with the lower bound result in [186] concludes our proof. □

A.6.4 Proof of Lemma 8.4.1

For simplicity, we denote

$$\hat{\beta} := 12\sqrt{d\log(1+\frac{wA^2}{\alpha^2 d\lambda})\log(32(\log(\frac{\gamma^2}{\alpha}+1)\frac{w^2}{\delta})} + 30\log(32(\log(\frac{\gamma^2}{\alpha})+1)\frac{w^2}{\delta})\frac{R}{\gamma^2}+\sqrt{\lambda}B. \tag{A.256}$$

It is obvious that $\hat{\beta} \geq \hat{\beta}_k$ for all $k \in [K]$. We call the restart time rounds *grids* and denote them by $g_1, g_2, \dots g_{\lceil \frac{K}{w} \rceil - 1}$, where $g_i \% w = 0$ for all $i \in [\lceil \frac{K}{w} \rceil - 1]$. Let i_k be the grid index of time round k, *i.e.*, $g_{i_k} \leq k < g_{i_k+1}$.

For ease of exposition and without loss of generality, we prove the lemma for $k \in [1, w]$. We calculate the estimation difference $|\boldsymbol{a}^\top(\hat{\boldsymbol{\theta}}_k - \boldsymbol{\theta}_k)|$ for any $\mathbf{a} \in \mathbb{R}^d$, $\|\mathbf{a}\|_2 \leq A$, $k \in [1, w]$. By definition:

$$\hat{\boldsymbol{\theta}}_k = \hat{\boldsymbol{\Sigma}}_k^{-1}\boldsymbol{b}_k = \hat{\boldsymbol{\Sigma}}_k^{-1}(\sum_{t=1}^{k-1}\frac{r_t\mathbf{a}_t}{\bar{\sigma}_t^2}) = \hat{\boldsymbol{\Sigma}}_k^{-1}(\sum_{t=1}^{k-1}\frac{\mathbf{a}_t\mathbf{a}_t^\top\boldsymbol{\theta}_t}{\bar{\sigma}_t^2} + \sum_{t=1}^{k-1}\frac{\mathbf{a}_t\epsilon_t}{\bar{\sigma}_t^2}), \tag{A.257}$$

where $\hat{\Sigma}_k = \lambda\boldsymbol{I} + \sum_{t=g_{i_k}}^{k-1}\frac{\mathbf{a}_t\mathbf{a}_t^\top}{\bar{\sigma}_t^2}$.

Then we have

$$\hat{\boldsymbol{\theta}}_k - \boldsymbol{\theta}_k = \hat{\boldsymbol{\Sigma}}_k^{-1}(\sum_{t=1}^{k-1}\frac{\mathbf{a}_t\mathbf{a}_t^\top}{\bar{\sigma}_t^2}(\boldsymbol{\theta}_t - \boldsymbol{\theta}_k) + \sum_{t=1}^{k-1}\frac{\mathbf{a}_t\epsilon_t}{\bar{\sigma}_t^2}) - \lambda\hat{\boldsymbol{\Sigma}}_k^{-1}\boldsymbol{\theta}_k\,. \tag{A.258}$$

Therefore

$$|\mathbf{a}^\top(\hat{\boldsymbol{\theta}}_k - \boldsymbol{\theta}_k)| \leq \left|\mathbf{a}^\top\hat{\boldsymbol{\Sigma}}_k^{-1}\sum_{t=1}^{k-1}\frac{\mathbf{a}_t\mathbf{a}_t^\top}{\bar{\sigma}_t^2}(\boldsymbol{\theta}_t - \boldsymbol{\theta}_k)\right| + \|\mathbf{a}\|_{\hat{\boldsymbol{\Sigma}}_k^{-1}}\|\sum_{t=1}^{k-1}\frac{\mathbf{a}_t\epsilon_t}{\bar{\sigma}_t^2}\|_{\hat{\boldsymbol{\Sigma}}_k^{-1}} + \lambda\|\mathbf{a}\|_{\hat{\boldsymbol{\Sigma}}_k^{-1}}\|\hat{\boldsymbol{\Sigma}}_k^{-\frac{1}{2}}\boldsymbol{\theta}_k\|_2\,, \tag{A.259}$$

where we use the Cauchy-Schwarz inequality.

For the first term, we have that for any $k \in [1, w]$

$$
\begin{aligned}
&\left| \mathbf{a}^\top \hat{\boldsymbol{\Sigma}}_k^{-1} \sum_{t=1}^{k} \frac{\mathbf{a}_t \mathbf{a}_t^\top}{\bar{\sigma}_t^2} (\boldsymbol{\theta}_t - \boldsymbol{\theta}_k) \right| \\
&\leq \sum_{t=1}^{k-1} |\mathbf{a}^\top \hat{\boldsymbol{\Sigma}}_k^{-1} \frac{\mathbf{a}_t}{\bar{\sigma}_t}| \cdot |\frac{\mathbf{a}_t}{\bar{\sigma}_t}^\top (\sum_{s=t}^{k-1} (\boldsymbol{\theta}_s - \boldsymbol{\theta}_{s+1}))| \qquad \text{(triangle inequality)} \\
&\leq \sum_{t=1}^{k-1} |\mathbf{a}^\top \hat{\boldsymbol{\Sigma}}_k^{-1} \frac{\mathbf{a}_t}{\bar{\sigma}_t}| \cdot \|\frac{\mathbf{a}_t}{\bar{\sigma}_t}\|_2 \cdot \| \sum_{s=t}^{k-1} (\boldsymbol{\theta}_s - \boldsymbol{\theta}_{s+1}) \|_2 \quad \text{(Cauchy-Schwarz)} \\
&\leq \frac{A}{\alpha} \sum_{t=1}^{k-1} |\mathbf{a}^\top \hat{\boldsymbol{\Sigma}}_k^{-1} \frac{\mathbf{a}_t}{\bar{\sigma}_t}| \cdot \| \sum_{s=t}^{k-1} (\boldsymbol{\theta}_s - \boldsymbol{\theta}_{s+1}) \|_2 \qquad (\|\mathbf{a}_t\| \leq A,\, \bar{\sigma}_t \geq \alpha) \\
&\leq \frac{A}{\alpha} \sum_{s=1}^{k-1} \sum_{t=1}^{s} |\mathbf{a}^\top \hat{\boldsymbol{\Sigma}}_k^{-1} \frac{\mathbf{a}_t}{\bar{\sigma}_t}| \cdot \|\boldsymbol{\theta}_s - \boldsymbol{\theta}_{s+1}\|_2 \\
&\qquad\qquad (\textstyle\sum_{t=1}^{k-1} \sum_{s=t}^{k-1} = \sum_{s=1}^{k-1} \sum_{t=1}^{s}) \\
&\leq \frac{A}{\alpha} \sum_{s=1}^{k-1} \sqrt{\left[\sum_{t=1}^{s} \mathbf{a}^\top \hat{\boldsymbol{\Sigma}}_k^{-1} \mathbf{a} \right] \cdot \left[\sum_{t=1}^{s} \frac{\mathbf{a}_t}{\bar{\sigma}_t}^\top \hat{\boldsymbol{\Sigma}}_k^{-1} \frac{\mathbf{a}_t}{\bar{\sigma}_t} \right]} \cdot \|\boldsymbol{\theta}_s - \boldsymbol{\theta}_{s+1}\|_2 \\
&\qquad\qquad \text{(Cauchy-Schwarz)} \\
&\leq \frac{A}{\alpha} \sum_{s=1}^{k-1} \sqrt{\left[\sum_{t=1}^{s} \mathbf{a}^\top \hat{\boldsymbol{\Sigma}}_k^{-1} \mathbf{a} \right] \cdot d} \cdot \|\boldsymbol{\theta}_s - \boldsymbol{\theta}_{s+1}\|_2 \qquad ((\star)) \\
&\leq \frac{A\|\mathbf{a}\|_2}{\alpha} \sqrt{d} \sum_{s=1}^{k-1} \sqrt{\frac{\sum_{t=1}^{k-1} 1}{\lambda}} \cdot \|\boldsymbol{\theta}_s - \boldsymbol{\theta}_{s+1}\|_2 \qquad (\lambda_{\max}(\hat{\boldsymbol{\Sigma}}_k^{-1}) \leq \tfrac{1}{\lambda}) \\
&\leq \frac{A^2}{\alpha} \sqrt{\frac{dw}{\lambda}} \sum_{s=1}^{k-1} \|\boldsymbol{\theta}_s - \boldsymbol{\theta}_{s+1}\|_2 \,, \qquad \text{(A.260)}
\end{aligned}
$$

where the inequality $(\star)$ follows from the fact that $\sum_{t=1}^{s} \frac{\mathbf{a}_t}{\bar{\sigma}_t}^\top \hat{\boldsymbol{\Sigma}}_k^{-1} \frac{\mathbf{a}_t}{\bar{\sigma}_t} \leq d$ that can be proved as follows. We have $\sum_{t=1}^{k-1} \frac{\mathbf{a}_t}{\bar{\sigma}_t}^\top \hat{\boldsymbol{\Sigma}}_k^{-1} \frac{\mathbf{a}_t}{\bar{\sigma}_t} =$

$\sum_{t=1}^{k-1} \mathrm{tr}\left(\frac{\mathbf{a}_t}{\bar{\sigma}_t}^\top \hat{\mathbf{\Sigma}}_k^{-1} \frac{\mathbf{a}_t}{\bar{\sigma}_t}\right) = \mathrm{tr}\left(\hat{\mathbf{\Sigma}}_k^{-1} \sum_{t=1}^{k-1} \frac{\mathbf{a}_t}{\bar{\sigma}_t}\frac{\mathbf{a}_t}{\bar{\sigma}_t}^\top\right)$. Given the eigenvalue decomposition $\sum_{t=1}^{k-1} \frac{\mathbf{a}_t}{\bar{\sigma}_t}\frac{\mathbf{a}_t}{\bar{\sigma}_t}^\top = \mathrm{diag}(\lambda_1, \ldots, \lambda_d)^\top$, we have $\hat{\mathbf{\Sigma}}_k = \mathrm{diag}(\lambda_1 + \lambda, \ldots, \lambda_d + \lambda)^\top$, and $\mathrm{tr}\left(\hat{\mathbf{\Sigma}}_k^{-1} \sum_{t=1}^{k-1} \frac{\mathbf{a}_t}{\bar{\sigma}_t}\frac{\mathbf{a}_t}{\bar{\sigma}_t}^\top\right) = \sum_{i=1}^{d} \frac{\lambda_j}{\lambda_j+\lambda} \leq d$.

For the second term, by the assumption on ϵ_k, we know that

$$
\begin{aligned}
&|\epsilon_k/\bar{\sigma}_k| \leq R/\alpha, \\
&|\epsilon_k/\bar{\sigma}_k| \cdot \min\{1, \|\mathbf{a}_k/\bar{\sigma}_k\|_{\hat{\mathbf{\Sigma}}_k^{-1}}\} \leq R\|\mathbf{a}_k\|_{\hat{\mathbf{\Sigma}}_k^{-1}}/\bar{\sigma}_k^2 \leq R/\gamma^2, \\
&\mathbb{E}[\epsilon_k|\mathbf{a}_{1:k}, \epsilon_{1:k-1}] = 0,\ \mathbb{E}[(\epsilon_k/\bar{\sigma}_k)^2|\mathbf{a}_{1:k}, \epsilon_{1:k-1}] \leq 1,\ \|\mathbf{a}_k/\bar{\sigma}_k\|_2 \leq A/\alpha,
\end{aligned}
$$

Therefore, setting $\mathcal{G}_k = \sigma(\mathbf{a}_{1:k}, \epsilon_{1:k-1})$, and using that σ_k is $\mathcal{G}_k$-measurable, applying Theorem A.6.3 to $(\boldsymbol{x}_k, \eta_k) = (\boldsymbol{a}_k/\bar{\sigma}_k, \epsilon_k/\bar{\sigma}_k)$ with $\epsilon = R/\gamma^2$, we get that with probability at least $1-\delta$, for all $k \in [1, w]$,

$$
\begin{aligned}
\|\sum_{t=1}^{k-1} \frac{\mathbf{a}_t \epsilon_t}{\bar{\sigma}_t^2}\|_{\hat{\mathbf{\Sigma}}_k^{-1}} &\leq 12\sqrt{d\log(1 + \frac{(k\%w)A^2}{\alpha^2 d\lambda})\log(32(\log(\frac{\gamma^2}{\alpha}+1)\frac{(k\%w)^2}{\delta})} \\
&+ 30\log(32(\log(\frac{\gamma^2}{\alpha})+1)\frac{(k\%w)^2}{\delta})\frac{R}{\gamma^2}. \qquad \text{(A.261)}
\end{aligned}
$$

For the last term

$$
\begin{aligned}
&\lambda\|\mathbf{a}\|_{\hat{\mathbf{\Sigma}}_k^{-1}}\|\hat{\mathbf{\Sigma}}_k^{-\frac{1}{2}}\boldsymbol{\theta}_k\|_2 \leq \lambda\|\mathbf{a}\|_{\hat{\mathbf{\Sigma}}_k^{-1}}\|\hat{\mathbf{\Sigma}}_k^{-\frac{1}{2}}\|_2\|\boldsymbol{\theta}_k\|_2 \\
&\leq \lambda\|\mathbf{a}\|_{\hat{\mathbf{\Sigma}}_k^{-1}}\frac{1}{\sqrt{\lambda_{\min}(\hat{\mathbf{\Sigma}}_k)}}\|\boldsymbol{\theta}_k\|_2 \leq \sqrt{\lambda}B\|\mathbf{a}\|_{\hat{\mathbf{\Sigma}}_k^{-1}}, \qquad \text{(A.262)}
\end{aligned}
$$

where we use the fact that $\lambda_{\min}(\hat{\mathbf{\Sigma}}_k) \geq \lambda$.

Therefore, with probabilty at least $1-\delta$, we have

$$
\begin{aligned}
&|\mathbf{a}^\top(\hat{\boldsymbol{\theta}}_k - \boldsymbol{\theta}_k)| \\
&\leq \frac{A^2}{\alpha}\sqrt{\frac{dw}{\lambda}}\sum_{t=1}^{k-1}\|\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t+1}\|_2 \\
&\quad + \|\mathbf{a}\|_{\hat{\boldsymbol{\Sigma}}_k^{-1}}\Bigg(12\sqrt{d\log(1+\frac{(k\%w)A^2}{\alpha^2 d\lambda})\log(32(\log(\frac{\gamma^2}{\alpha}+1)\frac{(k\%w)^2}{\delta})} \\
&+ 30\log(32(\log(\frac{\gamma^2}{\alpha})+1)\frac{(k\%w)^2}{\delta})\frac{R}{\gamma^2} + \sqrt{\lambda}B\Bigg) \\
&= \frac{A^2}{\alpha}\sqrt{\frac{dw}{\lambda}}\sum_{t=1}^{k-1}\|\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t+1}\|_2 + \hat{\beta}_k\|\mathbf{a}\|_{\hat{\boldsymbol{\Sigma}}_k^{-1}}, \qquad \text{(A.263)}
\end{aligned}
$$

where $\hat{\beta}_k$ is defined in Eq.(8.5).

A.6.5 Proof for Theorem 8.4.2

For simplicity of analysis, we only analyze the regret over the first grid, *i.e.*, we try to analyze $\text{Regret}(\tilde{K})$ for $\tilde{K} \in [1, w]$. Denote $\mathcal{E}_1$ as the event when Lemma 8.4.1 holds. Therefore, under event $\mathcal{E}_1$, for any $\tilde{K} \in [1, w]$, the regret can be bounded by

$$
\begin{aligned}
\text{Regret}(\tilde{K}) &= \sum_{k=1}^{\tilde{K}}\left[\langle \mathbf{a}_k^* - \mathbf{a}_k, \boldsymbol{\theta}_k\rangle\right] \\
&= \sum_{k=1}^{\tilde{K}}\Big[\langle \mathbf{a}_k^*, \boldsymbol{\theta}_k - \hat{\boldsymbol{\theta}}_k\rangle + (\langle \mathbf{a}_k^*, \hat{\boldsymbol{\theta}}_k\rangle + \hat{\beta}_k\|\mathbf{a}_k^*\|_{\hat{\boldsymbol{\Sigma}}_k^{-1}}) - (\langle \mathbf{a}_k, \hat{\boldsymbol{\theta}}_k\rangle + \hat{\beta}_k\|\mathbf{a}_k\|_{\hat{\boldsymbol{\Sigma}}_k^{-1}}) \\
&\quad + \langle \mathbf{a}_k, \hat{\boldsymbol{\theta}}_k - \boldsymbol{\theta}_k\rangle + \hat{\beta}_k\|\mathbf{a}_k\|_{\hat{\boldsymbol{\Sigma}}_k^{-1}} - \hat{\beta}_k\|\mathbf{a}_k^*\|_{\hat{\boldsymbol{\Sigma}}_k^{-1}}\Big] \\
&\leq \frac{2A^2}{\alpha}\sqrt{\frac{dw}{\lambda}}\sum_{k=1}^{\tilde{K}}\sum_{t=1}^{k-1}\|\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t+1}\|_2 + 2\sum_{k=1}^{\tilde{K}}\min\left\{1, \hat{\beta}_k\|\mathbf{a}_k\|_{\hat{\boldsymbol{\Sigma}}_k^{-1}}\right\}, \\
&\qquad \text{(A.264)}
\end{aligned}
$$

where in the last inequality we use the definition of event $\mathcal{E}_1$, the arm selection rule in Line 7 of Algo.8, and $0 \leq \langle \mathbf{a}_k^*, \boldsymbol{\theta}^* \rangle - \langle \mathbf{a}_k, \boldsymbol{\theta}^* \rangle \leq 2$.

Then we will bound the two terms in Eq.(A.264).

For the first term, we have

$$\begin{aligned}
&\frac{2A^2}{\alpha}\sqrt{\frac{dw}{\lambda}}\sum_{k=1}^{\tilde{K}}\sum_{t=1}^{k-1}\|\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t+1}\|_2 \\
&= \frac{2A^2}{\alpha}\sqrt{\frac{dw}{\lambda}}\sum_{t=1}^{\tilde{K}-1}\sum_{k=t}^{\tilde{K}}\|\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t+1}\|_2 \\
&\leq \frac{2A^2}{\alpha}\sqrt{\frac{dw}{\lambda}}w\sum_{t=1}^{\tilde{K}-1}\|\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t+1}\|_2 \,. \qquad \text{(A.265)}
\end{aligned}$$

To bound the second term in Eq.(A.264), we decompose the set $[\tilde{K}]$ into a union of two disjoint subsets $[K] = \mathcal{I}_1 \cup \mathcal{I}_2$.

$$\mathcal{I}_1 = \left\{k \in [\tilde{K}] : \|\frac{\mathbf{a}_k}{\bar{\sigma}_k}\|_{\hat{\boldsymbol{\Sigma}}_k^{-1}} \geq 1\right\},\ \mathcal{I}_2 = \left\{k \in [\tilde{K}] : \|\frac{\mathbf{a}_k}{\bar{\sigma}_k}\|_{\hat{\boldsymbol{\Sigma}}_k^{-1}} < 1\right\}. \qquad \text{(A.266)}$$

Then the following upper bound of $|\mathcal{I}_1|$ holds:

$$\begin{aligned}
|\mathcal{I}_1| &= \sum_{k\in\mathcal{I}_1}\min\left\{1, \|\frac{\mathbf{a}_k}{\bar{\sigma}_k}\|^2_{\hat{\boldsymbol{\Sigma}}_k^{-1}}\right\} \\
&\leq \sum_{k=1}^{\tilde{K}}\min\left\{1, \|\frac{\mathbf{a}_k}{\bar{\sigma}_k}\|^2_{\hat{\boldsymbol{\Sigma}}_k^{-1}}\right\} \\
&\leq 2d\iota, \qquad \text{(A.267)}
\end{aligned}$$

where $\iota = \log(1 + \frac{wA^2}{d\lambda\alpha^2})$, the first equality holds since $\|\frac{\mathbf{x}_k}{\bar{\sigma}_k}\|_{\hat{\boldsymbol{\Sigma}}_k^{-1}} \geq 1$ for $k \in \mathcal{I}_1$, the last inequality holds due to Lemma A.6.4 together with the fact $\|\frac{\mathbf{a}_k}{\bar{\sigma}_k}\|_2 \leq \frac{A}{\alpha}$ since $\bar{\sigma}_k \geq \alpha$ and $\|\mathbf{a}_k\|_2 \leq A$.

Then, we have

$$\begin{aligned}
&\sum_{k=1}^{\tilde{K}} \min\left\{1, \hat{\beta}_k \|\mathbf{a}_k\|_{\hat{\mathbf{\Sigma}}_k^{-1}}\right\} \\
&= \sum_{k\in\mathcal{I}_1} \min\left\{1, \bar{\sigma}_k\hat{\beta}_k \|\frac{\mathbf{a}_k}{\bar{\sigma}_k}\|_{\hat{\mathbf{\Sigma}}_k^{-1}}\right\} + \sum_{k\in\mathcal{I}_2} \min\left\{1, \bar{\sigma}_k\hat{\beta}_k \|\frac{\mathbf{a}_k}{\bar{\sigma}_k}\|_{\hat{\mathbf{\Sigma}}_k^{-1}}\right\} \\
&\le \left[\sum_{k\in\mathcal{I}_1} 1\right] + \sum_{k\in\mathcal{I}_2} \bar{\sigma}_k\hat{\beta}_k \|\frac{\mathbf{a}_k}{\bar{\sigma}_k}\|_{\hat{\mathbf{\Sigma}}_k^{-1}} \\
&\le 2d\iota + \hat{\beta} \sum_{k\in\mathcal{I}_2} \bar{\sigma}_k \|\frac{\mathbf{a}_k}{\bar{\sigma}_k}\|_{\hat{\mathbf{\Sigma}}_k^{-1}}, \qquad \text{(A.268)}
\end{aligned}$$

where the first inequality holds since $\min\{1, x\} \le 1$ and also $\min\{1, x\} \le x$, the second inequality holds by Eq.(A.267), and the fact the $\hat{\beta} \ge \hat{\beta}_k$ for all $k \in [K]$ ($\hat{\beta}$ is defined in Eq.(A.256)). Next we further bound the second summation term in (A.268). We decompose $\mathcal{I}_2 = \mathcal{J}_1 \cup \mathcal{J}_2$, where

$$\mathcal{J}_1 = \left\{k \in \mathcal{I}_2 : \bar{\sigma}_k = \sigma_k \cup \bar{\sigma}_k = \alpha\right\},\ \mathcal{J}_2 = \left\{k \in \mathcal{I}_2 : \bar{\sigma}_k = \gamma\sqrt{\|\mathbf{a}_k\|_{\hat{\mathbf{\Sigma}}_k^{-1}}}\right\}.$$

Then $\sum_{k\in\mathcal{I}_2} \bar{\sigma}_k \|\frac{\mathbf{a}_k}{\bar{\sigma}_k}\|_{\hat{\mathbf{\Sigma}}_k^{-1}} = \sum_{k\in\mathcal{J}_1} \bar{\sigma}_k \|\frac{\mathbf{a}_k}{\bar{\sigma}_k}\|_{\hat{\mathbf{\Sigma}}_k^{-1}} + \sum_{k\in\mathcal{J}_2} \bar{\sigma}_k \|\frac{\mathbf{a}_k}{\bar{\sigma}_k}\|_{\hat{\mathbf{\Sigma}}_k^{-1}}$. First, for $k \in \mathcal{J}_1$, we have

$$\begin{aligned}
\sum_{k\in\mathcal{J}_1} \bar{\sigma}_k \|\frac{\mathbf{a}_k}{\bar{\sigma}_k}\|_{\hat{\mathbf{\Sigma}}_k^{-1}} &\le \sum_{k\in\mathcal{J}_1} (\sigma_k + \alpha) \min\left\{1, \|\frac{\mathbf{a}_k}{\bar{\sigma}_k}\|_{\hat{\mathbf{\Sigma}}_k^{-1}}\right\} \\
&\le \sqrt{\sum_{k=1}^{\tilde{K}} (\sigma_k + \alpha)^2} \sqrt{\sum_{k=1}^{\tilde{K}} \min\left\{1, \|\frac{\mathbf{a}_k}{\bar{\sigma}_k}\|_{\hat{\mathbf{\Sigma}}_k^{-1}}\right\}^2} \\
&\le \sqrt{2\sum_{k=1}^{\tilde{K}} (\sigma_k^2 + \alpha^2)} \sqrt{\sum_{k=1}^{\tilde{K}} \min\left\{1, \|\frac{\mathbf{a}_k}{\bar{\sigma}_k}\|^2_{\hat{\mathbf{\Sigma}}_k^{-1}}\right\}} \\
&\le 2\sqrt{\sum_{k=1}^{\tilde{K}} \sigma_k^2 + \tilde{K}\alpha^2\sqrt{d\iota}}\,, \qquad \text{(A.269)}
\end{aligned}$$

where the first inequality holds since $\bar{\sigma}_k \leq \sigma_k + \alpha$ for $k \in \mathcal{J}_1$ and $\|\frac{\mathbf{a}_k}{\bar{\sigma}_k}\|_{\hat{\mathbf{\Sigma}}_k^{-1}} \leq 1$ since $k \in \mathcal{J}_1 \subseteq \mathcal{I}_2$, the second inequality holds by Cauchy-Schwarz inequality, the third inequality holds due to $(a+b)^2 \leq 2(a^2+b^2)$, and the last inequality holds due to Lemma A.6.4.

Finally we bound the summation for $k \in \mathcal{J}_2$. When $k \in \mathcal{J}_2$, we have $\bar{\sigma}_k = \gamma^2 \|\frac{\mathbf{a}_k}{\bar{\sigma}_k}\|_{\hat{\Sigma}_k^{-1}}$. Therefore we have

$$\begin{aligned}\sum_{k\in\mathcal{J}_2} \bar{\sigma}_k \|\frac{\mathbf{a}_k}{\bar{\sigma}_k}\|_{\hat{\mathbf{\Sigma}}_k^{-1}} &= \sum_{k\in\mathcal{J}_2} \gamma^2 \|\frac{\mathbf{a}_k}{\bar{\sigma}_k}\|^2_{\hat{\mathbf{\Sigma}}_k^{-1}} \\ &\leq \sum_{k=1}^{\tilde{K}} \gamma^2 \min\left\{1, \|\frac{\mathbf{a}_k}{\bar{\sigma}_k}\|^2_{\hat{\mathbf{\Sigma}}_k^{-1}}\right\} \\ &\leq 2\gamma^2 d\iota\,, \end{aligned} \tag{A.270}$$

where in the first inequality we use the fact that $\|\frac{\mathbf{a}_k}{\bar{\sigma}_k}\|_{\hat{\mathbf{\Sigma}}_k^{-1}} \leq 1$ since $k \in \mathcal{J}_2 \subseteq \mathcal{I}_2$, and in the last inequality we use Lemma A.6.4.

Therefore, with Eq.(A.264), Eq.(A.265), Eq.(A.268), Eq.(A.269), Eq.(A.270), we can get the regret upper bound for $\tilde{K} \in [1, w]$

$$\begin{aligned}\text{Regret}(\tilde{K}) \leq \frac{2A^2 w^{\frac{3}{2}}}{\alpha}\sqrt{\frac{d}{\lambda}}\sum_{k=1}^{\tilde{K}-1}\|\boldsymbol{\theta}_k - \boldsymbol{\theta}_{k+1}\|_2 + 4\hat{\beta}\sqrt{d\iota}\sqrt{\sum_{k\in[\tilde{K}]}\sigma_k^2 + w\alpha^2} \\ + 4d\iota\gamma^2\hat{\beta} + 4d\iota\,. \end{aligned} \tag{A.271}$$

Therefore, by the same deduction, we can get that

$$\begin{aligned}\text{Regret}([g_i, g_{i+1}]) \leq \frac{2A^2 w^{\frac{3}{2}}}{\alpha}\sqrt{\frac{d}{\lambda}}\sum_{k=g_i}^{g_{i+1}-1}\|\boldsymbol{\theta}_k - \boldsymbol{\theta}_{k+1}\|_2 \\ + 4\hat{\beta}\sqrt{d\iota}\sqrt{\sum_{k=g_i}^{g_{i+1}}\sigma_k^2 + w\alpha^2} + 4d\iota\gamma^2\hat{\beta} + 4d\iota\,, \end{aligned} \tag{A.272}$$

where we use $\text{Regret}([g_i, g_{i+1}])$ to denote the regret accumulated in the time period $[g_i, g_{i+1}]$.

Finally, without loss of generality, we assume $K\%w = 0$. Then we have

$$
\begin{aligned}
&\text{Regret}(\tilde{K}) \\
&= \sum_{i=0}^{\frac{K}{w}-1} \text{Regret}([g_i, g_{i+1}]) \\
&\leq \frac{2A^2 w^{\frac{3}{2}}}{\alpha}\sqrt{\frac{d}{\lambda}} \sum_{i=0}^{\frac{K}{w}-1} \sum_{k=g_i}^{g_{i+1}-1} \|\boldsymbol{\theta}_k - \boldsymbol{\theta}_{k+1}\|_2 + 4\hat{\beta}\sqrt{d\iota} \sum_{i=0}^{\frac{K}{w}-1} \sqrt{\sum_{k=g_i}^{g_{i+1}} \sigma_k^2 + w\alpha^2} \\
&+ \frac{4d\iota\gamma^2\hat{\beta}K}{w} + \frac{4dK\iota}{w} \\
&\leq \frac{2A^2 w^{\frac{3}{2}}}{\alpha}\sqrt{\frac{d}{\lambda}} \sum_{k=1}^{K-1} \|\boldsymbol{\theta}_k - \boldsymbol{\theta}_{k+1}\|_2 \\
&+ 4\hat{\beta}\sqrt{d\iota}\sqrt{\frac{K}{w}\sum_{i=0}^{\frac{K}{w}-1}(\sum_{k=g_i}^{g_{i+1}} \sigma_k^2 + w\alpha^2)} + \frac{4d\iota\gamma^2\hat{\beta}K}{w} + \frac{4dK\iota}{w} \\
&\leq \frac{2A^2 w^{\frac{3}{2}} B_K}{\alpha}\sqrt{\frac{d}{\lambda}} + 4\hat{\beta}\sqrt{\frac{Kd\iota}{w}}\sqrt{\sum_{k=1}^{K} \sigma_k^2 + K\alpha^2} + \frac{4d\iota\gamma^2\hat{\beta}K}{w} + \frac{4dK\iota}{w},
\end{aligned}
$$

where in the second inequality we use Cauchy-Schwarz inequality, and the last inequality holds due to $\sum_{k\in[K-1]} \|\boldsymbol{\theta}_k - \boldsymbol{\theta}_{k+1}\|_2 \leq B_K$.

A.6.6 Proof for Theorem 8.5.1

Recall that we call the restart time rounds *grids* and denote them by $g_1, g_2, \ldots g_{\lceil \frac{K}{w} \rceil - 1}$, where $g_i\%w = 0$ for all $i \in [\lceil \frac{K}{w} \rceil - 1]$. Let i_k be the grid index of time round k, *i.e.*, $g_{i_k} \leq k < g_{i_k+1}$. We

denote $\hat{\Psi}_{k,\ell} := \{t : t \in [g_{i_k}, k-1], \ell_t = \ell\}$.

For simplicity of analysis, we first try to bound the regret over the first grid, *i.e.*, we try to analyze Regret($\tilde{K}$) for $\tilde{K} \in [1, w]$. Note that in this case, for any $k \in [\tilde{K}]$ with $\tilde{K} \in [1, w]$, we have $g_{i_k} = 1$, so $\hat{\Psi}_{k,\ell} := \{t : t \in [1, k-1], \ell_t = \ell\}$.

First, we calculate the estimation difference $|\boldsymbol{a}^\top(\hat{\boldsymbol{\theta}}_{k,\ell} - \boldsymbol{\theta}_k)|$ for any $\mathbf{a} \in \mathbb{R}^d$, $\|\mathbf{a}\|_2 \leq A$. Recall that by definition, $\hat{\Sigma}_{k,\ell} = 2^{-2\ell}\boldsymbol{I} + \sum_{t\in\hat{\Psi}_{k,\ell}} w_t^2 \mathbf{a}_t \mathbf{a}_t^\top$, $\hat{\boldsymbol{b}}_{k,\ell} = \sum_{t\in\hat{\Psi}_{k,\ell}} w_t^2 r_t \mathbf{a}_t$, and

$$\hat{\boldsymbol{\theta}}_{k,\ell} = \hat{\Sigma}_{k,\ell}^{-1}\hat{\boldsymbol{b}}_{k,\ell} = \hat{\Sigma}_{k,\ell}^{-1}\Big(\sum_{t\in\hat{\Psi}_{k,\ell}} w_t^2 r_t \mathbf{a}_t\Big) = \hat{\Sigma}_{k,\ell}^{-1}\Big(\sum_{t\in\hat{\Psi}_{k,\ell}} w_t^2 \mathbf{a}_t\mathbf{a}_t^\top \boldsymbol{\theta}_t + \sum_{t\in\hat{\Psi}_{k,\ell}} w_t^2 \mathbf{a}_t \epsilon_t\Big)\,.$$

Then we have

$$\hat{\boldsymbol{\theta}}_{k,\ell} - \boldsymbol{\theta}_k = \hat{\Sigma}_{k,\ell}^{-1}\Big(\sum_{t\in\hat{\Psi}_{k,\ell}} w_t^2 \mathbf{a}_t\mathbf{a}_t^\top(\boldsymbol{\theta}_t - \boldsymbol{\theta}_k) + \sum_{t\in\hat{\Psi}_{k,\ell}} w_t^2 \mathbf{a}_t\epsilon_t\Big) - 2^{-2\ell}\hat{\Sigma}_{k,\ell}^{-1}\boldsymbol{\theta}_k\,. \tag{A.273}$$

Therefore, we can get

$$|\mathbf{a}^\top(\hat{\boldsymbol{\theta}}_{k,\ell} - \boldsymbol{\theta}_k)| \leq \left|\mathbf{a}^\top\hat{\boldsymbol{\Sigma}}_{k,\ell}^{-1}\sum_{t\in\hat{\Psi}_{k,\ell}} w_t^2\mathbf{a}_t\mathbf{a}_t^\top(\boldsymbol{\theta}_t - \boldsymbol{\theta}_k)\right| + \|\mathbf{a}\|_{\hat{\boldsymbol{\Sigma}}_{k,\ell}^{-1}}\Big\|\sum_{t\in\hat{\Psi}_{k,\ell}} w_t^2\mathbf{a}_t\epsilon_t\Big\|_{\hat{\boldsymbol{\Sigma}}_{k,\ell}^{-1}}$$
$$+ 2^{-2\ell}\|\mathbf{a}\|_{\hat{\boldsymbol{\Sigma}}_{k,\ell}^{-1}}\|\hat{\boldsymbol{\Sigma}}_{k,\ell}^{-\frac{1}{2}}\boldsymbol{\theta}_k\|_2\,, \tag{A.274}$$

where we use the Cauchy-Schwarz inequality.

For the first term, we have that for any $k \in [1, w]$

$$\left|\mathbf{a}^\top\hat{\boldsymbol{\Sigma}}_{k,\ell}^{-1}\sum_{t\in\hat{\Psi}_{k,\ell}} w_t^2\mathbf{a}_t\mathbf{a}_t^\top(\boldsymbol{\theta}_t - \boldsymbol{\theta}_k)\right|$$
$$\leq \sum_{t\in\hat{\Psi}_{k,\ell}} |\mathbf{a}^\top\boldsymbol{\Sigma}_{k,\ell}^{-1}w_t\mathbf{a}_t| \cdot |w_t\mathbf{a}_t^\top(\sum_{s=t}^{k-1}(\boldsymbol{\theta}_s - \boldsymbol{\theta}_{s+1}))|$$

(triangle inequality)

$$\leq \sum_{t\in\hat{\Psi}_{k,\ell}} |\mathbf{a}^\top \boldsymbol{\Sigma}_{k,\ell}^{-1} w_t \mathbf{a}_t| \cdot \|w_t \mathbf{a}_t\|_2 \cdot \|\sum_{s=t}^{k-1} (\boldsymbol{\theta}_s - \boldsymbol{\theta}_{s+1})\|_2$$

(Cauchy-Schwarz)

$$\leq A \sum_{t\in\hat{\Psi}_{k,\ell}} |\mathbf{a}^\top \hat{\boldsymbol{\Sigma}}_{k,\ell}^{-1} w_t \mathbf{a}_t| \cdot \|\sum_{s=t}^{k-1} (\boldsymbol{\theta}_s - \boldsymbol{\theta}_{s+1})\|_2$$

$(\|\mathbf{a}_t\| \leq A,\ w_t = \frac{2^{-\ell_t}}{\|\mathbf{a}_t\|_{\hat{\Sigma}_{t,\ell_t}^{-1}}} \leq 1)$

$$\leq A \sum_{s=1}^{k-1} \sum_{t\in\hat{\Psi}_{k,\ell}} |\mathbf{a}^\top \hat{\boldsymbol{\Sigma}}_{k,\ell}^{-1} w_t \mathbf{a}_t| \cdot \|\boldsymbol{\theta}_s - \boldsymbol{\theta}_{s+1}\|_2$$

$$\leq A \sum_{s=1}^{k-1} \sqrt{\left[\sum_{t\in\hat{\Psi}_{k,\ell}} \mathbf{a}^\top \hat{\boldsymbol{\Sigma}}_{k,\ell}^{-1} \mathbf{a}\right] \cdot \left[\sum_{t\in\hat{\Psi}_{k,\ell}} w_t \mathbf{a}_t{}^\top \hat{\boldsymbol{\Sigma}}_{k,\ell}^{-1} w_t \mathbf{a}_t\right]} \cdot \|\boldsymbol{\theta}_s - \boldsymbol{\theta}_{s+1}\|_2$$

(Cauchy-Schwarz)

$$\leq A \sum_{s=1}^{k-1} \sqrt{\left[\sum_{t\in\hat{\Psi}_{k,\ell}} \mathbf{a}^\top \hat{\boldsymbol{\Sigma}}_{k,\ell}^{-1} \mathbf{a}\right] \cdot d} \cdot \|\boldsymbol{\theta}_s - \boldsymbol{\theta}_{s+1}\|_2 \qquad ((\star))$$

$$\leq A\|\mathbf{a}\|_2 \sqrt{d} \sum_{s=1}^{k-1} \sqrt{2^{2\ell} \sum_{t\in\hat{\Psi}_{k,\ell}} 1} \cdot \|\boldsymbol{\theta}_s - \boldsymbol{\theta}_{s+1}\|_2$$

$(\lambda_{\max}(\hat{\boldsymbol{\Sigma}}_{k,\ell}^{-1}) \leq \frac{1}{2^{-2\ell}} = 2^{2\ell})$

$$\leq A^2 2^\ell \sqrt{dw} \sum_{s=1}^{k-1} \|\boldsymbol{\theta}_s - \boldsymbol{\theta}_{s+1}\|_2\ , \qquad \text{(A.275)}$$

where the inequality $(\star)$ follows from the fact that $\sum_{t\in\hat{\Psi}_{k,\ell}} w_t \mathbf{a}_t{}^\top \hat{\boldsymbol{\Sigma}}_{k,\ell}^{-1} w_t \mathbf{a}_t \leq d$ that can be proved as follows. We have $\sum_{t\in\hat{\Psi}_{k,\ell}} w_t \mathbf{a}_t{}^\top \hat{\boldsymbol{\Sigma}}_{k,\ell}^{-1} w_t \mathbf{a}_t = \sum_{t\in\hat{\Psi}_{k,\ell}} \mathrm{tr}\left(w_t \mathbf{a}_t{}^\top \hat{\boldsymbol{\Sigma}}_{k,\ell}^{-1} w_t \mathbf{a}_t\right) = \mathrm{tr}\left(\hat{\boldsymbol{\Sigma}}_{k,\ell}^{-1} \sum_{t\in\hat{\Psi}_{k,\ell}} w_t^2 \mathbf{a}_t \mathbf{a}_t{}^\top\right)$. Given the eigenvalue decomposition $\sum_{t\in\hat{\Psi}_{k,\ell}} w_t^2 \mathbf{a}_t \mathbf{a}_t{}^\top = \mathrm{diag}(\lambda_1, \ldots, \lambda_d)^\top$, we

have $\hat{\mathbf{\Sigma}}_{k,\ell} = \mathrm{diag}(\lambda_1+\lambda, \ldots, \lambda_d+\lambda)^\top$, and $\mathrm{tr}\left(\hat{\mathbf{\Sigma}}_{k,\ell}^{-1} \sum_{t\in\hat{\Psi}_{k,\ell}} w_t^2 \mathbf{a}_t \mathbf{a}_t^\top\right) = \sum_{i=1}^d \frac{\lambda_j}{\lambda_j+\lambda} \leq d$.

For the second term in Eq.(A.274), we can apply Theorem A.6.5 for the layer ℓ. In detail, for any $k \in [K]$, for each $t \in \hat{\Psi}_{k,\ell}$, we have

$$\|w_t \mathbf{a}_t\|_{\hat{\mathbf{\Sigma}}_{t,\ell}^{-1}} = 2^{-\ell}, \quad \mathbb{E}[w_t^2 \epsilon_t^2 | \mathcal{F}_t] \leq w_t^2 \mathbb{E}[\epsilon_t^2 | \mathcal{F}_t] \leq w_t^2 \sigma_t^2, \quad |w_t \epsilon_t| \leq |\epsilon_t| \leq R,$$

where the last inequality holds due to the fact that $w_t = \frac{2^{-\ell_t}}{\|\mathbf{a}_t\|_{\hat{\mathbf{\Sigma}}_{t,\ell_t}^{-1}}} \leq 1$. According to Theorem A.6.5, and taking a union bound, we can deduce that with probability at least $1-\delta$, for all $\ell \in [L]$, for all round $k \in \Psi_{K+1,\ell}$,

$$\| \sum_{t\in\hat{\Psi}_{k,\ell}} w_t^2 \mathbf{a}_t \epsilon_t \|_{\hat{\mathbf{\Sigma}}_{k,\ell}^{-1}} \leq 16 \cdot 2^{-\ell} \sqrt{\sum_{t\in\hat{\Psi}_{k,\ell}} w_t^2 \sigma_t^2 \log(\frac{4w^2 L}{\delta})} + 6 \cdot 2^{-\ell} R \log(\frac{4w^2 L}{\delta}) \,. \tag{A.276}$$

For simplicity, we denote $\mathcal{E}_{\mathrm{conf}}$ as the event such that Eq.(A.276) holds.

For the third term in Eq.(A.274), we have

$$\begin{aligned} &2^{-2\ell} \|\mathbf{a}\|_{\hat{\mathbf{\Sigma}}_{k,\ell}^{-1}} \|\hat{\mathbf{\Sigma}}_{k,\ell}^{-\frac{1}{2}} \boldsymbol{\theta}_k\|_2 \leq 2^{-2\ell} \|\mathbf{a}\|_{\hat{\mathbf{\Sigma}}_{k,\ell}^{-1}} \|\hat{\mathbf{\Sigma}}_k^{-\frac{1}{2}}\|_2 \|\boldsymbol{\theta}_k\|_2 \\ &\leq 2^{-2\ell} \|\mathbf{a}\|_{\hat{\mathbf{\Sigma}}_{k,\ell}^{-1}} \frac{1}{\sqrt{\lambda_{\min}(\hat{\mathbf{\Sigma}}_{k,\ell})}} \|\boldsymbol{\theta}_k\|_2 \leq 2^{-\ell} B \|\mathbf{a}\|_{\hat{\mathbf{\Sigma}}_k^{-1}} \,, \end{aligned} \tag{A.277}$$

where we use the fact that $\lambda_{\min}(\hat{\mathbf{\Sigma}}_{k,\ell}) \geq 2^{-2\ell}$.

For simplicity, we denote $\ell^* = \lceil \frac{1}{2} \log_2 \log\left(4(w+1)^2 L/\delta\right) \rceil + 8$. Then, under $\mathcal{E}_{\mathrm{conf}}$, by the definition of $\hat{\beta}_{k,\ell}$ in Eq.(8.9), Lemma A.6.6 and Lemma A.6.7, with probability at least $1-\delta$, we have

for all $\ell^* + 1 \leq \ell \leq L$,

$$\hat{\beta}_{k,\ell} \geq 16 \cdot 2^{-\ell} \sqrt{\sum_{t \in \hat{\Psi}_{k,\ell}} w_t^2 \sigma_t^2 \log(\frac{4w^2 L}{\delta})} + 6 \cdot 2^{-\ell} R \log(\frac{4w^2 L}{\delta}) + 2^{-\ell} B. \tag{A.278}$$

Therefore, with Eq.(A.274), Eq.(A.275), Eq.(A.276), Eq.(A.277), Eq.(A.278), with probability at least $1-3\delta$, for all $\ell^*+1 \leq \ell \leq L$ we have

$$|\mathbf{a}^\top(\hat{\boldsymbol{\theta}}_{k,\ell} - \boldsymbol{\theta}_k)| \leq A^2 2^\ell \sqrt{dw} \sum_{s=1}^{k-1} \|\boldsymbol{\theta}_s - \boldsymbol{\theta}_{s+1}\|_2 + \hat{\beta}_{k,\ell} \|\mathbf{a}\|_{\hat{\Sigma}_{k,\ell}^{-1}}. \tag{A.279}$$

Then for all $k \in [K]$ such that $\ell^*+1 \leq \ell_k \leq L$, with probability at least $1-3\delta$ we have

$$\begin{aligned}
\langle \mathbf{a}_k^*, \boldsymbol{\theta}_k \rangle &\leq \min_{\ell \in [L]} \langle \mathbf{a}_k^*, \hat{\boldsymbol{\theta}}_{k,\ell} \rangle + A^2 2^\ell \sqrt{dw} \sum_{s=1}^{k-1} \|\boldsymbol{\theta}_s - \boldsymbol{\theta}_{s+1}\|_2 + \hat{\beta}_{k,\ell} \|\mathbf{a}_k^*\|_{\hat{\Sigma}_{k,\ell}^{-1}} \\
&\leq A^2 2^L \sqrt{dw} \sum_{s=1}^{k-1} \|\boldsymbol{\theta}_s - \boldsymbol{\theta}_{s+1}\|_2 + \min_{\ell \in [L]} \langle \mathbf{a}_k^*, \hat{\boldsymbol{\theta}}_{k,\ell} \rangle + \hat{\beta}_{k,\ell} \|\mathbf{a}_k^*\|_{\hat{\Sigma}_{k,\ell}^{-1}} \\
&\leq A^2 2^L \sqrt{dw} \sum_{s=1}^{k-1} \|\boldsymbol{\theta}_s - \boldsymbol{\theta}_{s+1}\|_2 + \min_{\ell \in [L]} \langle \mathbf{a}_k, \hat{\boldsymbol{\theta}}_{k,\ell} \rangle + \hat{\beta}_{k,\ell} \|\mathbf{a}_k\|_{\hat{\Sigma}_{k,\ell}^{-1}} \\
&\leq A^2 2^L \sqrt{dw} \sum_{s=1}^{k-1} \|\boldsymbol{\theta}_s - \boldsymbol{\theta}_{s+1}\|_2 + \langle \mathbf{a}_k, \hat{\boldsymbol{\theta}}_{k,\ell_k-1} \rangle + \hat{\beta}_{k,\ell_k-1} \|\mathbf{a}_k\|_{\hat{\Sigma}_{k,\ell_k-1}^{-1}},
\end{aligned} \tag{A.280}$$

where the first inequality holds because of Eq.(A.279), the third inequality holds because of the arm selection rule in Line 8 of Algo.12.

We decompose the regret for $\tilde{K} \in [1, w]$ as follows

$$\begin{aligned}\text{Regret}(\tilde{K}) &= \sum_{k\in[\tilde{K}]} \left(\langle \mathbf{a}_k^*, \boldsymbol{\theta}_k\rangle - \langle \mathbf{a}_k, \boldsymbol{\theta}_k\rangle\right) \\ &= \sum_{\ell\in[\ell^*]} \sum_{k\in\hat{\Psi}_{\tilde{K}+1,\ell}} \left(\langle \mathbf{a}_k^*, \boldsymbol{\theta}_k\rangle - \langle \mathbf{a}_k, \boldsymbol{\theta}_k\rangle\right) + \sum_{\ell\in[L]\setminus[\ell^*]} \sum_{k\in\hat{\Psi}_{\tilde{K}+1,\ell}} \left(\langle \mathbf{a}_k^*, \boldsymbol{\theta}_k\rangle - \langle \mathbf{a}_k, \boldsymbol{\theta}_k\rangle\right) \\ &\quad + \sum_{k\in\hat{\Psi}_{\tilde{K}+1,L+1}} \left(\langle \mathbf{a}_k^*, \boldsymbol{\theta}_k\rangle - \langle \mathbf{a}_k, \boldsymbol{\theta}_k\rangle\right). \qquad \text{(A.281)}\end{aligned}$$

We will bound the three terms separately. For the first term, we have for layer $\ell \in [\ell^*]$ and round $k \in \hat{\Psi}_{\tilde{K}+1,\ell}$, we have

$$\begin{aligned}\sum_{k\in\hat{\Psi}_{\tilde{K}+1,\ell}} \left(\langle \mathbf{a}_k^*, \boldsymbol{\theta}^*\rangle - \langle \mathbf{a}_k, \boldsymbol{\theta}^*\rangle\right) &\le 2\left|\Psi_{K+1,\ell}\right| \\ &= 2^{2\ell+1} \sum_{k\in\hat{\Psi}_{\tilde{K}+1,\ell}} \|w_k \mathbf{a}_k\|^2_{\hat{\mathbf{\Sigma}}^{-1}_{k,\ell}} \\ &\le 2\cdot 128^2 \log(\frac{4(w+1)^2 L}{\delta}) \sum_{k\in\hat{\Psi}_{\tilde{K}+1,\ell}} \|w_k \mathbf{a}_k\|^2_{\hat{\mathbf{\Sigma}}^{-1}_{k,\ell}} \\ &\le 2\cdot 128^2 \log(\frac{4(w+1)^2 L}{\delta}) \cdot 2d \log(1 + \frac{2^{2\ell} w A^2}{d}) \\ &= \tilde{O}(d), \qquad \text{(A.282)}\end{aligned}$$

where the first inequality holds because the reward is in $[-1, 1]$, the equation follows from the fact that $\|w_k \mathbf{a}_k\|_{\hat{\mathbf{\Sigma}}^{-1}_{k,\ell}} = 2^{-\ell}$ holds for all $k \in \Psi_{K+1,\ell}$, the second inequality holds due to the fact that $2^{\ell^*} \le 128\sqrt{\log(4(w+1)^2 L/\delta)}$, and the last inequality holds due to Lemma A.6.4.

Therefore

$$\sum_{\ell\in[\ell^*]} \sum_{k\in\hat{\Psi}_{\tilde{K}+1,\ell}} \left(\langle \mathbf{a}_k^*, \boldsymbol{\theta}_k\rangle - \langle \mathbf{a}_k, \boldsymbol{\theta}_k\rangle\right) = \tilde{O}(d). \qquad \text{(A.283)}$$

For the second part in Eq.(A.281), we have

$$\begin{aligned}
&\sum_{\ell\in[L]\backslash[\ell^*]}\sum_{k\in\hat{\Psi}_{\tilde{K}+1,\ell}}\left(\langle\mathbf{a}_k^*,\boldsymbol{\theta}_k\rangle-\langle\mathbf{a}_k,\boldsymbol{\theta}_k\rangle\right)\\
&\quad\le\sum_{\ell\in[L]\backslash[\ell^*]}\sum_{k\in\hat{\Psi}_{\tilde{K}+1,\ell}}\Bigg(\langle\mathbf{a}_k,\hat{\boldsymbol{\theta}}_{k,\ell-1}\rangle+\hat{\beta}_{k,\ell-1}\|\mathbf{a}_k\|_{\hat{\Sigma}_{k,\ell-1}^{-1}}\\
&\quad+A^2 2^L\sqrt{dw}\sum_{k\in\hat{\Psi}_{\tilde{K}+1,\ell}}\|\boldsymbol{\theta}_s-\boldsymbol{\theta}_{s+1}\|_2-\langle\mathbf{a}_k,\boldsymbol{\theta}_k\rangle\Bigg)\\
&\le 2\sum_{\ell\in[L]\backslash[\ell^*]}\sum_{k\in\hat{\Psi}_{\tilde{K}+1,\ell}}\hat{\beta}_{k,\ell-1}\|\mathbf{a}_k\|_{\hat{\Sigma}_{k,\ell-1}^{-1}}+A^2\sqrt{dw}\sum_{\ell\in[L]\backslash[\ell^*]}\sum_{k\in\hat{\Psi}_{\tilde{K}+1,\ell}}2^L\sum_{s=1}^{k-1}\|\boldsymbol{\theta}_s-\boldsymbol{\theta}_{s+1}\|_2\ ,
\end{aligned}\tag{A.284}$$

where the inequality holds due to Eq.(A.280), the second inequality holds due to Eq.(A.279). We then try to bound the two terms.

For the first term in Eq.(A.284), we have

$$\begin{aligned}
\sum_{\ell\in[L]\backslash[\ell^*]}\sum_{k\in\hat{\Psi}_{\tilde{K}+1,\ell}}\hat{\beta}_{k,\ell-1}\|\mathbf{a}_k\|_{\hat{\Sigma}_{k,\ell-1}^{-1}}&\le\sum_{\ell\in[L]\backslash[\ell^*]}\sum_{k\in\hat{\Psi}_{\tilde{K}+1,\ell}}\hat{\beta}_{k,\ell-1}\cdot 2^{-\ell}\\
&\le\sum_{\ell\in[L]\backslash[\ell^*]}\hat{\beta}_{\tilde{K},\ell-1}\cdot 2^{-\ell}\left|\hat{\Psi}_{\tilde{K}+1,\ell}\right|\\
&=\sum_{\ell\in[L]\backslash[\ell^*]}\hat{\beta}_{\tilde{K},\ell-1}\cdot 2^{\ell}\sum_{k\in\hat{\Psi}_{\tilde{K}+1,\ell}}\|w_k\mathbf{a}_k\|_{\Sigma_{k,\ell}^{-1}}^2\\
&\le\sum_{\ell\in[L]\backslash[\ell^*]}\hat{\beta}_{\tilde{K},\ell-1}\cdot 2^{\ell}\cdot 2d\log(1+\frac{2^{2\ell}\tilde{K}A^2}{d})\\
&=\tilde{O}(d\cdot 2^{\ell}\cdot\hat{\beta}_{\tilde{K},\ell-1})\\
&=\tilde{O}\left(d\Big(\sqrt{\sum_{k=1}^{\tilde{K}}\sigma_k^2}+R+1\Big)\right),
\end{aligned}\tag{A.285}$$

where the first inequality holds because by the algorithm design, we have for all $k \in \hat{\Psi}_{\tilde{K}+1,\ell}$: $\|\mathbf{a}_k\|_{\hat{\Sigma}^{-1}_{k,\ell-1}} \leq 2^{-\ell}$; the second inequality holds because for all $k \in \hat{\Psi}_{\tilde{K}+1,\ell}$, $\hat{\beta}_{k,\ell-1} \leq \hat{\beta}_{\tilde{K},\ell-1}$; the first equality holds because for all $k \in \hat{\Psi}_{\tilde{K}+1,\ell}$, $\|w_k \mathbf{a}_k\|^2_{\Sigma^{-1}_{k,\ell}} = 2^{-2\ell}$; the third inequality holds by Lemma A.6.4; the last two equalities hold because by Lemma A.6.6 and Lemma A.6.7, we have $\hat{\beta}_{\tilde{K},\ell-1} = \tilde{O}\left(2^{-\ell}(\sqrt{\sum_{k=1}^{\tilde{K}} \sigma_k^2} + R + 1)\right)$.

For the second term in Eq.(A.284), we have

$$\begin{aligned}
&A^2\sqrt{dw} \sum_{\ell \in [L]\setminus[\ell^*]} \sum_{k \in \hat{\Psi}_{\tilde{K}+1,\ell}} 2^L \sum_{s=1}^{k-1} \|\boldsymbol{\theta}_s - \boldsymbol{\theta}_{s+1}\|_2 \\
&\leq A^2 2^L \sqrt{dw} \sum_{k \in [\tilde{K}-1]} \sum_{s=1}^{k-1} \|\boldsymbol{\theta}_s - \boldsymbol{\theta}_{s+1}\|_2 \\
&\leq \frac{A^2\sqrt{d} w^{\frac{3}{2}}}{\alpha} \sum_{k=1}^{\tilde{K}-1} \|\boldsymbol{\theta}_k - \boldsymbol{\theta}_{k+1}\|_2 \qquad \text{(A.286)}
\end{aligned}$$

Therefore, with this, Eq.(A.284), and Eq.(A.285), we have

$$\begin{aligned}
&\sum_{\ell \in [L]\setminus[\ell^*]} \sum_{k \in \hat{\Psi}_{\tilde{K}+1,\ell}} (\langle \mathbf{a}_k^*, \boldsymbol{\theta}_k \rangle - \langle \mathbf{a}_k, \boldsymbol{\theta}_k \rangle) \\
&\leq \frac{A^2\sqrt{d} w^{\frac{3}{2}}}{\alpha} \sum_{k=1}^{\tilde{K}-1} \|\boldsymbol{\theta}_k - \boldsymbol{\theta}_{k+1}\|_2 + \tilde{O}\left(d\Big(\sqrt{\sum_{k=1}^{\tilde{K}} \sigma_k^2} + R + 1\Big)\right). \qquad \text{(A.287)}
\end{aligned}$$

Finally, for the last term in Eq.(A.281), we have

$$\sum_{k \in \hat{\Psi}_{\tilde{K}+1,L+1}} (\langle \mathbf{a}_k^*, \boldsymbol{\theta}_k \rangle - \langle \mathbf{a}_k, \boldsymbol{\theta}_k \rangle)$$

$$
\begin{aligned}
&\leq \sum_{k\in\hat{\Psi}_{\tilde{K}+1,L+1}} \left(\langle \mathbf{a}_k, \hat{\boldsymbol{\theta}}_{k,L} \rangle + \hat{\beta}_{k,L} \|\mathbf{a}_k\|_{\hat{\Sigma}_{k,L}^{-1}} + A^2 2^L \sqrt{dw} \sum_{s=1}^{k-1} \|\boldsymbol{\theta}_s - \boldsymbol{\theta}_{s+1}\|_2 - \langle \mathbf{a}_k, \boldsymbol{\theta}_k \rangle \right) \\
&\leq \sum_{k\in\hat{\Psi}_{\tilde{K}+1,L+1}} \left(2\hat{\beta}_{k,L} \|\mathbf{a}_k\|_{\hat{\Sigma}_{k,L}^{-1}} + A^2 2^{L+1} \sqrt{dw} \sum_{s=1}^{k-1} \|\boldsymbol{\theta}_s - \boldsymbol{\theta}_{s+1}\|_2 \right) \\
&\leq \sum_{k\in\hat{\Psi}_{\tilde{K}+1,L+1}} \left(2^{-L+1} \hat{\beta}_{k,L} + A^2 2^{L+1} \sqrt{dw} \sum_{s=1}^{k-1} \|\boldsymbol{\theta}_s - \boldsymbol{\theta}_{s+1}\|_2 \right) \\
&\leq \frac{2A^2\sqrt{d}w^{\frac{3}{2}}}{\alpha} \sum_{k=1}^{\tilde{K}-1} \|\boldsymbol{\theta}_k - \boldsymbol{\theta}_{k+1}\|_2 + \sum_{k\in\hat{\Psi}_{\tilde{K}+1,L+1}} 2^{-L+1} \hat{\beta}_{\tilde{K},L} \\
&\leq \frac{2A^2\sqrt{d}w^{\frac{3}{2}}}{\alpha} \sum_{k=1}^{\tilde{K}-1} \|\boldsymbol{\theta}_k - \boldsymbol{\theta}_{k+1}\|_2 + w \cdot 2\alpha \cdot \hat{\beta}_{\tilde{K},L} \\
&= \frac{2A^2\sqrt{d}w^{\frac{3}{2}}}{\alpha} \sum_{k=1}^{\tilde{K}-1} \|\boldsymbol{\theta}_k - \boldsymbol{\theta}_{k+1}\|_2 + \tilde{O}\left(w\alpha^2 \cdot \left(\sqrt{\sum_{k=1}^{\tilde{K}} \sigma_k^2} + R + 1 \right) \right),
\end{aligned}
\tag{A.288}
$$

where the first inequality holds due to Eq.(A.280), the second inequality holds due to Eq.(A.279), the third inequality holds because by the algorithm design, we have for all $k \in \hat{\Psi}_{\tilde{K}+1,L+1}$: $\|\mathbf{a}_k\|_{\hat{\Sigma}_{k,L}^{-1}} \leq 2^{-L}$, the fourth inequality holds due to the same reasons as before, and the fact that $\hat{\beta}_{\tilde{K},L} \geq \hat{\beta}_{k,L}$ for all $k \in \hat{\beta}_{\tilde{K},L}$; the last inequality holds due to $\hat{\beta}_{\tilde{K},\ell-1} = \tilde{O}\left(\alpha(\sqrt{\sum_{k=1}^{\tilde{K}} \sigma_k^2} + R + 1) \right)$.

Plugging Eq.(A.287), Eq.(A.288), and Eq.(A.283) into Eq.(A.281), we can get that for $\tilde{K} \in [1, w]$

$$\text{Regret}(\tilde{K}) = \tilde{O}\Big(\frac{A^2\sqrt{d}w^{\frac{3}{2}}}{\alpha}\sum_{k=1}^{\tilde{K}-1}\|\boldsymbol{\theta}_k - \boldsymbol{\theta}_{k+1}\|_2 + \big(w\alpha^2 + d\big)\cdot\big(\sqrt{\sum_{k=1}^{\tilde{K}}\sigma_k^2} + R + 1\big)\Big). \tag{A.289}$$

By the same deduction we can get

$$\begin{aligned}&\text{Regret}([g_i, g_{i+1}])\\ &= \tilde{O}\Big(\frac{A^2\sqrt{d}w^{\frac{3}{2}}}{\alpha}\sum_{k=g_i}^{g_{i+1}}\|\boldsymbol{\theta}_k - \boldsymbol{\theta}_{k+1}\|_2 + \big(w\alpha^2 + d\big)\cdot\big(\sqrt{\sum_{k=g_i}^{g_{i+1}}\sigma_k^2} + R + 1\big)\Big).\end{aligned} \tag{A.290}$$

Finally, without loss of generality, we assume $K\%w = 0$. Then we have

$$\begin{aligned}&\text{Regret}(K)\\ &= \sum_{i=0}^{\frac{K}{w}-1}\text{Regret}([g_i, g_{i+1}])\\ &= \tilde{O}\Big(\frac{A^2\sqrt{d}w^{\frac{3}{2}}}{\alpha}\sum_{i=0}^{\frac{K}{w}-1}\sum_{k=g_i}^{g_{i+1}}\|\boldsymbol{\theta}_k - \boldsymbol{\theta}_{k+1}\|_2 + \big(w\alpha^2 + d\big)\cdot\sum_{i=0}^{\frac{K}{w}-1}\big(\sqrt{\sum_{k=g_i}^{g_{i+1}}\sigma_k^2} + R + 1\big)\Big)\\ &\le \tilde{O}\Big(\frac{A^2\sqrt{d}w^{\frac{3}{2}}}{\alpha}\sum_{k=1}^{K-1}\|\boldsymbol{\theta}_k - \boldsymbol{\theta}_{k+1}\|_2 + \big(w\alpha^2 + d\big)\cdot\big(\sqrt{\frac{K}{w}\sum_{i=0}^{\frac{K}{w}-1}\sum_{k=g_i}^{g_{i+1}}\sigma_k^2} + \frac{KR}{w} + \frac{K}{w}\big)\Big)\\ &\le \tilde{O}\Big(\frac{A^2\sqrt{d}w^{\frac{3}{2}}B_K}{\alpha} + \big(w\alpha^2 + d\big)\cdot\sqrt{\frac{K}{w}\sum_{k=1}^{K}\sigma_k^2} + \big(1+R\big)\cdot\big(K\alpha^2 + \frac{Kd}{w}\big)\Big),\end{aligned}$$

where the first inequality holds due to the Cauchy-Schwarz inequality, the last inequality holds because $\sum_{k=1}^{K-1}\|\boldsymbol{\theta}_k - \boldsymbol{\theta}_{k+1}\|_2 \le$

B_K.

A.6.7 Proof of Theorem A.6.1

With the candidate pool set $\mathcal{P}$ designed as in Eq.(A.241), Eq.(A.242), Eq.(A.243), and $H = \lceil d^{\frac{2}{5}} K^{\frac{2}{5}} \rceil$, we have $|\mathcal{P}| = O(\log K)$, and for any $w \in \mathcal{W}$, $w \leq H$.

We denote the optimal (w, α) with the knowledge of V_K and B_K in Corollary 8.2 as (w^*, α^*). We denote the best approximation of (w^*, α^*) in the candidate set $\mathcal{P}$ as (w^+, α^+). Then we can decompose the regret as follows

$$\begin{aligned}
\text{Regret}(K) &= \sum_{k=1}^{K} \langle \mathbf{a}_t^*, \boldsymbol{\theta}_k \rangle - \langle \mathbf{a}_t, \boldsymbol{\theta}_k \rangle \\
&= \underbrace{\sum_{k=1}^{K} \langle \mathbf{a}_t^*, \boldsymbol{\theta}_k \rangle - \sum_{i=1}^{\lceil \frac{K}{H} \rceil} \sum_{k=(i-1)H+1}^{iH} \langle \mathbf{a}_t(w^+, \alpha^+), \boldsymbol{\theta}_k \rangle}_{(1)} \\
&\quad + \underbrace{\sum_{i=1}^{\lceil \frac{K}{H} \rceil} \sum_{k=(i-1)H+1}^{iH} \langle \mathbf{a}_t(w^+, \alpha^+), \boldsymbol{\theta}_k \rangle - \langle \mathbf{a}_t(w_i, \alpha_i), \boldsymbol{\theta}_k \rangle}_{(2)} .
\end{aligned} \tag{A.291}$$

The first term (1) is the dynamic regret of Restarted SAVE$^+$ with the best parameters in the candidate pool $\mathcal{P}$. The second term (2) is the regret overhead of meta-algorithm due to adaptive exploration of unknown optimal parameters.

By the design of the candidate pool set $\mathcal{P}$ in Eq.(A.241), Eq.(A.242), Eq.(A.243), we have that there exists a pair $(w^+, \alpha^+) \in$

$\mathcal{P}$ such that $w^+ < w^* < 2w^+$, and $\alpha^+ < \alpha^* < 2\alpha^+$. Therefore, employing the regret bound in Theorem 8.5.1, we can get

$$
\begin{aligned}
(1) &\leq \sum_{i=1}^{\lceil \frac{K}{H} \rceil} \tilde{O}(\sqrt{d} w^{+1.5} B_i/\alpha^+ + \alpha^{+2}(H + \sqrt{w^+ H V_i}) + d\sqrt{H V_i/w^+} + dH/w^+) \\
&\leq \tilde{O}(\sqrt{d} w^{+1.5} B_K/\alpha^+ + \alpha^{+2}(K + \sqrt{w^+ H \frac{K}{H} \sum_{i=1}^{\lceil \frac{K}{H} \rceil} V_i}) + d\sqrt{H \frac{K}{H} \sum_{i=1}^{\lceil \frac{K}{H} \rceil} V_i/w^+} \\
&\quad + dK/w^+) \\
&= \tilde{O}(\sqrt{d} w^{+1.5} B_K/\alpha^+ + \alpha^{+2}(K + \sqrt{w^+ K V_K}) + d\sqrt{K V_K/w^+} + dK/w^+) \\
&= \tilde{O}(\sqrt{d} w^{*1.5} B_K/\alpha^* + \alpha^{*2}(K + \sqrt{w^* K V_K}) + d\sqrt{K V_K/w^*} + dK/w^*) \\
&= \tilde{O}(d^{4/5} V_K^{2/5} B_K^{1/5} K^{2/5} + d^{2/3} B_K^{1/3} K^{2/3}) \,, \qquad \text{(A.292)}
\end{aligned}
$$

where we denote B_i as the total variation budget in block i, V_i is the total variance in block i, the second inequality is by Cauchy–Schwarz inequality, the first equality holds due to $\sum_{i=1}^{\lceil \frac{K}{H} \rceil} B_i = B_K$, $\sum_{i=1}^{\lceil \frac{K}{H} \rceil} V_i = V_K$, the second equality holds due to $w^+ < w^* < 2w^+$ and $\alpha^+ < \alpha^* < 2\alpha^+$, the last equality holds by Corollary 8.2.

We then try to bound the second term (2). We denote by $\mathcal{E}$ the event such that Lemma A.6.9 holds, and denote by $R_i := \sum_{k=(i-1)H+1}^{iH} \langle \mathbf{a}_t(w^+, \alpha^+), \boldsymbol{\theta}_k \rangle - \langle \mathbf{a}_t(w_i, \alpha_i), \boldsymbol{\theta}_k \rangle$ the instantaneous regret of the meta learner in the block i. Then we have

$$
\begin{aligned}
(2) &= \mathbb{E}\Big[\sum_{i=1}^{\lceil \frac{K}{H} \rceil} R_i \Big] \\
&= \mathbb{E}\Big[\sum_{i=1}^{\lceil \frac{K}{H} \rceil} R_i | \mathcal{E} \Big] P(\mathcal{E}) + \mathbb{E}\Big[\sum_{i=1}^{\lceil \frac{K}{H} \rceil} R_i | \overline{\mathcal{E}} \Big] P(\overline{\mathcal{E}}) \\
&\leq \tilde{O}\Big(L_{\max} \sqrt{\frac{K}{H} |\mathcal{P}|} \Big) \cdot (1 - \frac{2}{K}) + \tilde{O}(K) \cdot \frac{2}{K}
\end{aligned}
$$

$$= \tilde{O}(\sqrt{H\,|\mathcal{P}|\,K}) \\ = \tilde{O}(d^{\frac{1}{5}}K^{\frac{7}{10}})\,, \tag{A.293}$$

where $L_{\max} := \max_{i\in[\lceil \frac{K}{H} \rceil]} L_i$, the first inequality holds due to the standard regret upper bound result for Exp3 [183], the third equality holds due to Lemma A.6.9, the last equality holds since $H = \lceil d^{\frac{2}{5}}K^{\frac{2}{5}} \rceil$, and $|\mathcal{P}| = O(\log K)$.

Finally, combining the above results for term (1) and term (2), we have

$$\text{Regret}(K) = \tilde{O}(d^{4/5}V_K^{2/5}B_K^{1/5}K^{2/5} + d^{2/3}B_K^{1/3}K^{2/3} + d^{\frac{1}{5}}K^{\frac{7}{10}}). \tag{A.294}$$

A.6.8 Technical Lemmas

Theorem A.6.3 (Theorem 4.3, [76]). *Let $\{\mathcal{G}_k\}_{k=1}^{\infty}$ be a filtration, and $\{\mathbf{x}_k, \eta_k\}_{k\geq 1}$ be a stochastic process such that $\mathbf{x}_k \in \mathbb{R}^d$ is $\mathcal{G}_k$-measurable and $\eta_k \in \mathbb{R}$ is $\mathcal{G}_{k+1}$-measurable. Let $L, \sigma, \lambda, \epsilon > 0$, $\boldsymbol{\mu}^* \in \mathbb{R}^d$. For $k \geq 1$, let $y_k = \langle \boldsymbol{\mu}^*, \mathbf{x}_k \rangle + \eta_k$ and suppose that $\eta_k, \mathbf{x}_k$ also satisfy*

$$\mathbb{E}[\eta_k|\mathcal{G}_k] = 0,\ \mathbb{E}[\eta_k^2|\mathcal{G}_k] \leq \sigma^2,\ |\eta_k| \leq R,\ \|\mathbf{x}_k\|_2 \leq L. \tag{A.295}$$

For $k \geq 1$, let $\mathbf{Z}_k = \lambda\mathbf{I} + \sum_{i=1}^{k} \mathbf{x}_i\mathbf{x}_i^\top$, $\mathbf{b}_k = \sum_{i=1}^{k} y_i\mathbf{x}_i$, $\boldsymbol{\mu}_k = \mathbf{Z}_k^{-1}\mathbf{b}_k$, and

$$\begin{aligned}\beta_k = {} & 12\sqrt{\sigma^2 d \log(1 + kL^2/(d\lambda)) \log(32(\log(R/\epsilon)+1)k^2/\delta)} \\ & + 24\log(32(\log(R/\epsilon)+1)k^2/\delta) \max_{1\leq i\leq k}\{|\eta_i| \min\{1, \|\mathbf{x}_i\|_{\mathbf{Z}_{i-1}^{-1}}\}\} \\ & + 6\log(32(\log(R/\epsilon)+1)k^2/\delta)\epsilon.\end{aligned}$$

Then, for any $0 < \delta < 1$*, we have with probability at least* $1 - \delta$ *that,*

$$\forall k \geq 1,\ \left\|\sum_{i=1}^{k} \mathbf{x}_i \eta_i\right\|_{\mathbf{Z}_k^{-1}} \leq \beta_k,\ \|\boldsymbol{\mu}_k - \boldsymbol{\mu}^*\|_{\mathbf{Z}_k} \leq \beta_k + \sqrt{\lambda}\|\boldsymbol{\mu}^*\|_2.$$

Lemma A.6.4 (Lemma 11, [43])**.** *For any* $\lambda > 0$ *and sequence* $\{\mathbf{x}_k\}_{k=1}^{K} \subset \mathbb{R}^d$ *for* $k \in [K]$*, define* $\mathbf{Z}_k = \lambda \mathbf{I} + \sum_{i=1}^{k-1} \mathbf{x}_i \mathbf{x}_i^\top$*. Then, provided that* $\|\mathbf{x}_k\|_2 \leq L$ *holds for all* $k \in [K]$*, we have*

$$\sum_{k=1}^{K} \min\left\{1, \|\mathbf{x}_k\|_{\mathbf{Z}_k^{-1}}^2\right\} \leq 2d \log\left(1 + KL^2/(d\lambda)\right).$$

Theorem A.6.5 (Theorem 2.1, [80])**.** *Let* $\{\mathcal{G}_k\}_{k=1}^{\infty}$ *be a filtration, and* $\{\mathbf{x}_k, \eta_k\}_{k \geq 1}$ *be a stochastic process such that* $\mathbf{x}_k \in \mathbb{R}^d$ *is* $\mathcal{G}_k$*-measurable and* $\eta_k \in \mathbb{R}$ *is* $\mathcal{G}_{k+1}$*-measurable. Let* $L, \sigma, \lambda, \epsilon > 0$*,* $\boldsymbol{\mu}^* \in \mathbb{R}^d$*. For* $k \geq 1$*, let* $y_k = \langle \boldsymbol{\mu}^*, \mathbf{x}_k \rangle + \eta_k$*, where* $\eta_k, \mathbf{x}_k$ *satisfy*

$$\mathbb{E}[\eta_k | \mathcal{G}_k] = 0,\ |\eta_k| \leq R,\ \sum_{i=1}^{k} \mathbb{E}[\eta_i^2 | \mathcal{G}_i] \leq v_k, \quad \text{for } \forall\ k \geq 1$$

For $k \geq 1$*, let* $\mathbf{Z}_k = \lambda \mathbf{I} + \sum_{i=1}^{k} \mathbf{x}_i \mathbf{x}_i^\top$*,* $\mathbf{b}_k = \sum_{i=1}^{k} y_i \mathbf{x}_i$*,* $\boldsymbol{\mu}_k = \mathbf{Z}_k^{-1} \mathbf{b}_k$*, and*

$$\beta_k = 16\rho\sqrt{v_k \log(4w^2/\delta)} + 6\rho R \log(4w^2/\delta),$$

where $\rho \geq \sup_{k \geq 1} \|\mathbf{x}_k\|_{\mathbf{Z}_{k-1}^{-1}}$*. Then, for any* $0 < \delta < 1$*, we have with probability at least* $1 - \delta$ *that,*

$$\forall k \geq 1,\ \left\|\sum_{i=1}^{k} \mathbf{x}_i \eta_i\right\|_{\mathbf{Z}_k^{-1}} \leq \beta_k, \|\boldsymbol{\mu}_k - \boldsymbol{\mu}^*\|_{\mathbf{Z}_k} \leq \beta_k + \sqrt{\lambda}\|\boldsymbol{\mu}^*\|_2.$$

Lemma A.6.6 (Adopted from Lemma B.4, [80])**.** *Let weight* w_i *be defined in Algorithm 12. With probability at least* $1 - 2\delta$*, for all*

$k \geq 1$, $\ell \in [L]$, the following two inequalities hold simultaneously:

$$\sum_{i \in \hat{\Psi}_{k+1,\ell}} w_i^2 \sigma_i^2 \leq 2 \sum_{i \in \hat{\Psi}_{k+1,\ell}} w_i^2 \epsilon_i^2 + \frac{14}{3} R^2 \log(4w^2 L/\delta),$$
$$\sum_{i \in \hat{\Psi}_{k+1,\ell}} w_i^2 \epsilon_i^2 \leq \frac{3}{2} \sum_{i \in \hat{\Psi}_{k+1,\ell}} w_i^2 \sigma_i^2 + \frac{7}{3} R^2 \log(4w^2 L/\delta).$$

For simplicity, we denote $\mathcal{E}_{\mathrm{VaR}}$ as the event such that the two inequalities in Lemma A.6.6 holds.

Lemma A.6.7 (Adopted from Lemma B.5, [80])**.** *Suppose that $\|\boldsymbol{\theta}^*\|_2 \leq B$. Let weight w_i be defined in Algorithm 12. On the event $\mathcal{E}_{\mathrm{conf}}$ and $\mathcal{E}_{\mathrm{VaR}}$ (defined in Eq.(A.276), Lemma A.6.6), for all $k \geq 1$, $\ell \in [L]$ such that $2^\ell \geq 64\sqrt{\log\left(4(w+1)^2 L/\delta\right)}$, we have the following inequalities:*

$$\sum_{i \in \Psi_{k+1,\ell}} w_i^2 \sigma_i^2 \leq 8 \sum_{i \in \Psi_{k+1,\ell}} w_i^2 \left(r_i - \langle \hat{\boldsymbol{\theta}}_{k+1,\ell}, \mathbf{a}_i \rangle\right)^2 + 6R^2 \log(4(w+1)^2 L/\delta) + 2^{-2\ell+2} B^2,$$
$$\sum_{i \in \Psi_{k+1,\ell}} w_i^2 \left(r_i - \langle \hat{\boldsymbol{\theta}}_{k+1,\ell}, \mathbf{a}_i \rangle\right)^2 \leq \frac{3}{2} \sum_{i \in \Psi_{k+1,\ell}} w_i^2 \sigma_i^2 + \frac{7}{3} R^2 \log(4w^2 L/\delta) + 2^{-2\ell} B^2.$$

Lemma A.6.8 ([294])**.** *Let $M, v > 0$ be fixed constants. Let $\{x_i\}_{i=1}^n$ be a stochastic process, $\{\mathcal{G}_i\}_i$ be a filtration so that for all $i \in [n]$, x_i is $\mathcal{G}_i$-measurable, while almost surely*

$$\mathbb{E}\left[x_i | \mathcal{G}_{i-1}\right] = 0, \quad |x_i| \leq M, \quad \sum_{i=1}^n \mathbb{E}[x_i^2 | \mathcal{G}_{i-1}] \leq v.$$

Then for any $\delta > 0$, with probability at least $1 - \delta$, we have

$$\sum_{i=1}^n x_i \leq \sqrt{2v \log(1/\delta)} + 2/3 \cdot M \log(1/\delta).$$

Lemma A.6.9. *Let $N = \lceil \frac{K}{H} \rceil$. Denote by L_i the absolute value of cumulative rewards for episode i, i.e., $L_i = \sum_{k=(i-1)H+1}^{iH} r_k$, then*

$$\mathbb{P}\left[\forall i \in [N], L_i \leq H + R\sqrt{\frac{H}{2}\log\left(K(\frac{K}{H}+1)\right)} + \frac{2}{3}\cdot R\log\left(K(\frac{K}{H}+1)\right)\right] \geq 1-\frac{1}{K}. \tag{A.296}$$

Proof. By Lemma A.6.8, we have that with probability at least $1 - 1/K$

$$\begin{aligned}\sum_{k=(i-1)\cdot H+1}^{i\cdot H} \epsilon_i &\leq \sqrt{2\sum_{k=(i-1)\cdot H+1}^{i\cdot H} \sigma_k^2 \log(NK)} + 2/3\cdot R\log(NK) \\ &\leq \sqrt{2H\frac{R^2}{4}\log(NK)} + 2/3\cdot R\log(NK) \\ &\leq R\sqrt{\frac{H}{2}\log\left(K\cdot(\frac{K}{H}+1)\right)} + \frac{2}{3}\cdot R\log\left(K\cdot(\frac{K}{H}+1)\right),\end{aligned} \tag{A.297}$$

where we use union bound, and in the second inequality we use the fact that since $|\epsilon_k| \leq R$, we have $\sigma_k^2 \leq \frac{R^2}{4}$. Finally, together with the assumption that $r_k \leq 1$ for all $k \in [K]$, we complete the proof. □

A.7 Clustering Of Neural Dueling Bandits (CONDB) Algorithm

Here we provide the complete statement of our CONDB algorithm.

A.8 Proof of Theorem 9.4.1

First, we prove the following lemma.

Lemma A.8.1. *With probability at least* $1-\delta$ *for some* $\delta \in (0,1)$*, at any* $t \in [T]$*:*

$$\left\|\hat{\boldsymbol{\theta}}_{i,t} - \boldsymbol{\theta}^{j(i)}\right\|_2 \leq \frac{\sqrt{\lambda\kappa_\mu} + \sqrt{2\log(u/\delta) + d\log(1 + T_{i,t}\kappa_\mu/d\lambda)}}{\kappa_\mu\sqrt{\lambda_{min}(\boldsymbol{V}_{i,t-1})}}, \forall i \in \mathcal{U}\,, \tag{A.301}$$

where $\boldsymbol{V}_{i,t-1} = \frac{\lambda}{\kappa_\mu}\mathbf{I} + \sum_{\substack{s\in[t-1]\\ i_s=i}} (\phi(\boldsymbol{x}_{s,1}) - \phi(\boldsymbol{x}_{s,2}))(\phi(\boldsymbol{x}_{s,1}) - \phi(\boldsymbol{x}_{s,2}))^\top$*, and* $T_{i,t}$ *denotes the number of rounds of seeing user* i *in the first* t *rounds.*

Proof. First, we prove the following result.

For a fixed user i, with probability at least $1-\delta$ for some $\delta \in (0,1)$, at any $t \in [T]$:

$$\left\|\hat{\boldsymbol{\theta}}_{i,t} - \boldsymbol{\theta}^{j(i)}\right\|_{\boldsymbol{V}_{i,t-1}} \leq \frac{\sqrt{\lambda\kappa_\mu} + \sqrt{2\log(1/\delta) + d\log(1 + 4T_{i,t}\kappa_\mu/d\lambda)}}{\kappa_\mu}\,, \tag{A.302}$$

Recall that $f_i(\boldsymbol{x}) = \boldsymbol{\theta}_i^\top \phi(\boldsymbol{x})$. In iteration s, define $\widetilde{\phi}_s = \phi(\boldsymbol{x}_{s,1}) - \phi(\boldsymbol{x}_{s,2})$. And we define $\widetilde{f}_{i,s} = f_i(\boldsymbol{x}_{s,1}) - f_i(\boldsymbol{x}_{s,2}) = \boldsymbol{\theta}_i^\top \widetilde{\phi}_s$.

For any $\boldsymbol{\theta}_{f'} \in \mathbb{R}^d$, define

$$G_{i,t}(\boldsymbol{\theta}_{f'}) = \sum_{\substack{s\in[t-1]:\\ i_s=i}} \left(\mu(\boldsymbol{\theta}_{f'}^\top \widetilde{\phi}_s) - \mu(\boldsymbol{\theta}_i^\top \widetilde{\phi}_s)\right)\widetilde{\phi}_s + \lambda\boldsymbol{\theta}_{f'}.$$

For $\lambda' \in (0,1)$, setting $\boldsymbol{\theta}_{\bar{f}} = \lambda' \boldsymbol{\theta}_{f_1'} + (1-\lambda')\boldsymbol{\theta}_{f_2'}$. and using the mean-value theorem, we get:

$$G_{i,t}(\boldsymbol{\theta}_{f_1'}) - G_{i,t}(\boldsymbol{\theta}_{f_2'}) = \left[\sum_{\substack{s \in [t-1]: \\ i_s = i}} \nabla\mu(\boldsymbol{\theta}_{\bar{f}}^\top \widetilde{\boldsymbol{\phi}}_s) \widetilde{\boldsymbol{\phi}}_s \widetilde{\boldsymbol{\phi}}_s^\top + \lambda \mathbf{I} \right] (\boldsymbol{\theta}_{f_1'} - \boldsymbol{\theta}_{f_2'}) \quad (\boldsymbol{\theta}_i \text{ is constant}) \tag{A.303}$$

Define $\boldsymbol{M}_{i,t-1} = \left[\sum_{\substack{s \in [t-1]: \\ i_s = i}} \nabla\mu(\boldsymbol{\theta}_{\bar{f}}^\top \widetilde{\boldsymbol{\phi}}_s) \widetilde{\boldsymbol{\phi}}_s \widetilde{\boldsymbol{\phi}}_s^\top + \lambda \mathbf{I} \right]$, and recall that $\boldsymbol{V}_{i,t-1} = \sum_{\substack{s \in [t-1]: \\ i_s = i}} \widetilde{\boldsymbol{\phi}}_s \widetilde{\boldsymbol{\phi}}_s^\top + \frac{\lambda}{\kappa_\mu} \mathbf{I}$. Then we have that $\boldsymbol{M}_{i,t-1} \succeq \kappa_\mu \boldsymbol{V}_{i,t-1}$ and that $\boldsymbol{V}_{i,t-1}^{-1} \succeq \kappa_\mu \boldsymbol{M}_{i,t-1}^{-1}$, where we use the notation $\boldsymbol{M} \succeq \boldsymbol{V}$ to denote that $\boldsymbol{M} - \boldsymbol{V}$ is a positive semi-definite matrix. Then we have

$$\begin{aligned}
&\left\| G_{i,t}(\hat{\boldsymbol{\theta}}_{i,t}) - \lambda \boldsymbol{\theta}_i \right\|_{\boldsymbol{V}_{i,t-1}^{-1}}^2 \\
&= \left\| G_{i,t}(\boldsymbol{\theta}_i) - G_t(\hat{\boldsymbol{\theta}}_{i,t}) \right\|_{\boldsymbol{V}_{i,t-1}^{-1}}^2 \\
&= \left\| \boldsymbol{M}_{i,t-1}(\boldsymbol{\theta}_i - \hat{\boldsymbol{\theta}}_{i,t}) \right\|_{\boldsymbol{V}_{i,t-1}^{-1}}^2 && (G_{i,t}(\boldsymbol{\theta}_i) = \lambda \boldsymbol{\theta}_i \text{ by definition}) \\
&= (\boldsymbol{\theta}_i - \hat{\boldsymbol{\theta}}_{i,t})^\top \boldsymbol{M}_{i,t-1} \boldsymbol{V}_{i,t-1}^{-1} \boldsymbol{M}_{i,t-1} (\boldsymbol{\theta}_i - \hat{\boldsymbol{\theta}}_{i,t}) \\
&\geq (\boldsymbol{\theta}_i - \hat{\boldsymbol{\theta}}_{i,t})^\top \boldsymbol{M}_{i,t-1} \kappa_\mu \boldsymbol{M}_{i,t-1}^{-1} \boldsymbol{M}_{i,t-1} (\boldsymbol{\theta}_i - \hat{\boldsymbol{\theta}}_{i,t}) \\
&= \kappa_\mu (\boldsymbol{\theta}_i - \hat{\boldsymbol{\theta}}_{i,t})^\top \boldsymbol{M}_{i,t-1} (\boldsymbol{\theta}_i - \hat{\boldsymbol{\theta}}_{i,t}) \\
&\geq \kappa_\mu (\boldsymbol{\theta}_i - \hat{\boldsymbol{\theta}}_{i,t})^\top \kappa_\mu \boldsymbol{V}_{i,t-1} (\boldsymbol{\theta}_i - \hat{\boldsymbol{\theta}}_{i,t}) \\
&= \kappa_\mu^2 (\boldsymbol{\theta}_i - \hat{\boldsymbol{\theta}}_{i,t})^\top \boldsymbol{V}_{i,t-1} (\boldsymbol{\theta}_i - \hat{\boldsymbol{\theta}}_{i,t}) \\
&= \kappa_\mu^2 \left\| \boldsymbol{\theta}_i - \hat{\boldsymbol{\theta}}_{i,t} \right\|_{\boldsymbol{V}_{i,t-1}}^2 && \left(\text{as } ||\boldsymbol{x}||_{\boldsymbol{A}}^2 = \boldsymbol{x}^\top \boldsymbol{A} \boldsymbol{x}\right)
\end{aligned}$$

The first inequality is because $\boldsymbol{V}_{i,t-1}^{-1} \succeq \kappa_\mu \boldsymbol{M}_{i,t-1}^{-1}$, and the second inequality follows from $\boldsymbol{M}_{i,t-1} \succeq \kappa_\mu \boldsymbol{V}_{i,t-1}$.

Note that $\frac{\kappa_\mu}{\lambda}\mathbf{I} \succeq \boldsymbol{V}_{i,t-1}$, which allows us to show that

$$\|\lambda\boldsymbol{\theta}_i\|_{\boldsymbol{V}_{i,t-1}^{-1}} = \lambda\sqrt{\boldsymbol{\theta}_i^\top \boldsymbol{V}_{i,t-1}^{-1}\boldsymbol{\theta}_i} \leq \lambda\sqrt{\boldsymbol{\theta}_i^\top \frac{\kappa_\mu}{\lambda}\boldsymbol{\theta}_i} \leq \sqrt{\lambda\kappa_\mu}\,\|\boldsymbol{\theta}_i\|_2 \leq \sqrt{\lambda\kappa_\mu}. \tag{A.304}$$

Using the two equations above, we have that

$$\begin{aligned}\left\|\boldsymbol{\theta}_i - \hat{\boldsymbol{\theta}}_{i,t}\right\|_{\boldsymbol{V}_{i,t-1}} \leq \frac{1}{\kappa_\mu}\left\|G_{i,t}(\hat{\boldsymbol{\theta}}_{i,t}) - \lambda\boldsymbol{\theta}_i\right\|_{\boldsymbol{V}_{i,t-1}^{-1}} &\leq \frac{1}{\kappa_\mu}\left\|G_{i,t}(\hat{\boldsymbol{\theta}}_{i,t})\right\|_{\boldsymbol{V}_{i,t-1}^{-1}} + \frac{1}{\kappa_\mu}\|\lambda\boldsymbol{\theta}_i\|_{\boldsymbol{V}_{i,t-1}^{-1}} \\ &\leq \frac{1}{\kappa_\mu}\left\|G_{i,t}(\hat{\boldsymbol{\theta}}_{i,t})\right\|_{\boldsymbol{V}_{i,t-1}^{-1}} + \sqrt{\frac{\lambda}{\kappa_\mu}}\end{aligned} \tag{A.305}$$

Then, let $f_{t,s}^i = \hat{\boldsymbol{\theta}}_{i,t}^\top \widetilde{\phi}_s$, we have:

$$\begin{aligned}\frac{1}{\kappa_\mu^2}\left\|G_{i,t}(\hat{\boldsymbol{\theta}}_{i,t})\right\|_{\boldsymbol{V}_{i,t-1}^{-1}}^2 &= \frac{1}{\kappa_\mu^2}\left\|\sum_{\substack{s\in[t-1]:\\ i_s=i}}(\mu(\hat{\boldsymbol{\theta}}_{i,t}^\top\widetilde{\phi}_s) - \mu(\boldsymbol{\theta}_i^\top\widetilde{\phi}_s))\widetilde{\phi}_s + \lambda\hat{\boldsymbol{\theta}}_{i,t}\right\|_{\boldsymbol{V}_{i,t-1}^{-1}}^2) \\ &= \frac{1}{\kappa_\mu^2}\left\|\sum_{\substack{s\in[t-1]:\\ i_s=i}}(\mu(f_{t,s}^i) - \mu(\widetilde{f}_{i,s}))\widetilde{\phi}_s + \lambda\hat{\boldsymbol{\theta}}_{i,t}\right\|_{\boldsymbol{V}_{i,t-1}^{-1}}^2 \\ &= \frac{1}{\kappa_\mu^2}\left\|\sum_{\substack{s\in[t-1]:\\ i_s=i}}(\mu(f_{t,s}^i) - (y_s - \epsilon_s))\widetilde{\phi}_s + \lambda\hat{\boldsymbol{\theta}}_{i,t}\right\|_{\boldsymbol{V}_{i,t-1}^{-1}}^2 \\ &= \frac{1}{\kappa_\mu^2}\left\|\sum_{\substack{s\in[t-1]:\\ i_s=i}}\left(\mu(f_{t,s}^i) - y_s\right)\widetilde{\phi}_s + \sum_{\substack{s\in[t-1]:\\ i_s=i}}\epsilon_s\widetilde{\phi}_s + \lambda\hat{\boldsymbol{\theta}}_{i,t}\right\|_{\boldsymbol{V}_{i,t-1}^{-1}}^2 \\ &\leq \frac{1}{\kappa_\mu^2}\left\|\sum_{\substack{s\in[t-1]:\\ i_s=i}}\epsilon_s\widetilde{\phi}_s\right\|_{\boldsymbol{V}_{i,t-1}^{-1}}^2.\end{aligned}$$

The last step holds due to the following reasoning. Recall that $\hat{\boldsymbol{\theta}}_{i,t}$ is computed using MLE by solving the following equation:

$$\hat{\boldsymbol{\theta}}_{i_t,t} = \arg\min_{\boldsymbol{\theta}} \Bigg[- \sum_{\substack{s\in[t-1]\\ i_s=i_t}} \Big(y_s \log \mu\big(\boldsymbol{\theta}^\top[\phi(\boldsymbol{x}_{s,1}) - \phi(\boldsymbol{x}_{s,2})]\big) + (1-y_s)\log\mu\big(\boldsymbol{\theta}^\top[\phi(\boldsymbol{x}_{s,2}) - \phi(\boldsymbol{x}_{s,1})]\big)\Big) + \frac{\lambda}{2}\|\boldsymbol{\theta}\|_2^2 \Bigg]. \tag{A.306}$$

Setting its gradient to 0, the following is satisfied:

$$\sum_{\substack{s\in[t-1]:\\ i_s=i}} \left(\mu\left(\hat{\boldsymbol{\theta}}_{i,t}^\top \widetilde{\phi}_s\right) - y_s\right)\widetilde{\phi}_s + \lambda\hat{\boldsymbol{\theta}}_{i,t} = 0, \tag{A.307}$$

which is used in the last step.

Now we have

$$\frac{1}{\kappa_\mu^2}\left\|G_{i,t}(\hat{\boldsymbol{\theta}}_{i,t})\right\|_{\boldsymbol{V}_{i,t-1}^{-1}}^2 \le \frac{1}{\kappa_\mu^2}\left\|\sum_{\substack{s\in[t-1]:\\ i_s=i}} \epsilon_s\widetilde{\phi}_s\right\|_{\boldsymbol{V}_{i,t-1}^{-1}}^2. \tag{A.308}$$

Denote $\boldsymbol{V} \triangleq \frac{\lambda}{\kappa_\mu}\mathbf{I}$. Note that the sequence of observation noises $\{\epsilon_s\}$ is 1-sub-Gaussian.

Next, we can apply Theorem 1 from [43], to obtain

$$\left\|\sum_{\substack{s\in[t-1]:\\ i_s=i}} \epsilon_s\widetilde{\phi}_s\right\|_{\boldsymbol{V}_{i,t-1}^{-1}}^2 \le 2\log\left(\frac{\det(\boldsymbol{V}_{i,t-1})^{1/2}}{\delta\det(\boldsymbol{V})^{1/2}}\right), \tag{A.309}$$

which holds with probability of at least $1-\delta$.

Next, based on our assumption that $\left\|\widetilde{\phi}_s\right\|_2 \le 2$, according to Lemma 10 from [43], we have that

$$\det(\boldsymbol{V}_{i,t-1}) \le \left(\lambda/\kappa_\mu + 4T_{i,t}/d\right)^d, \tag{A.310}$$

where $T_{i,t}$ denotes the number of rounds of serving user i in the first t rounds. Therefore,

$$\sqrt{\frac{\det \boldsymbol{V}_{i,t-1}}{\det(V)}} \leq \sqrt{\frac{\left(\lambda/\kappa_\mu + 4T_{i,t}/d\right)^d}{(\lambda/\kappa_\mu)^d}} = \left(1 + 4T_{i,t}\kappa_\mu/(d\lambda)\right)^{\frac{d}{2}} \tag{A.311}$$

This gives us

$$\left\| \sum_{\substack{s\in[t-1]:\\ i_s=i}} \epsilon_s \widetilde{\phi}_s \right\|^2_{\boldsymbol{V}^{-1}_{i,t-1}} \leq 2\log\left(\frac{\det(\boldsymbol{V}_{i,t-1})^{1/2}}{\delta \det(V)^{1/2}}\right) \leq 2\log(1/\delta) + d\log\left(1 + 4T_{i,t}\kappa_\mu/(d\lambda)\right) \tag{A.312}$$

Then, with the above reasoning, we have that with probability at least $1-\delta$ for some $\delta \in (0,1)$, at any $t \in [T]$:

$$\left\|\hat{\boldsymbol{\theta}}_{i,t} - \boldsymbol{\theta}^{j(i)}\right\|_{\boldsymbol{V}_{i,t-1}} \leq \frac{\sqrt{\lambda\kappa_\mu} + \sqrt{2\log(1/\delta) + d\log(1 + 4T_{i,t}\kappa_\mu/d\lambda)}}{\kappa_\mu}, \tag{A.313}$$

Taking a union bound over u users, we have that with probability at least $1-\delta$ for some $\delta \in (0,1)$, at any $t \in [T]$:

$$\left\|\hat{\boldsymbol{\theta}}_{i,t} - \boldsymbol{\theta}^{j(i)}\right\|_{\boldsymbol{V}_{i,t-1}} \leq \frac{\sqrt{\lambda\kappa_\mu} + \sqrt{2\log(u/\delta) + d\log(1 + 4T_{i,t}\kappa_\mu/d\lambda)}}{\kappa_\mu}, \forall i \in \mathcal{U}. \tag{A.314}$$

Then we have that with probability at least $1-\delta$ for all $t \in [T]$ and all $i \in \mathcal{U}$

$$\begin{aligned}\left\|\hat{\boldsymbol{\theta}}_{i,t} - \boldsymbol{\theta}^{j(i)}\right\| &\leq \frac{\left\|\hat{\boldsymbol{\theta}}_{i,t} - \boldsymbol{\theta}^{j(i)}\right\|_{\boldsymbol{V}_{i,t-1}}}{\sqrt{\lambda_{\min(\boldsymbol{V}_{i,t-1})}}} \\ &\leq \frac{\sqrt{\lambda\kappa_\mu} + \sqrt{2\log(u/\delta) + d\log(1 + 4T_{i,t}\kappa_\mu/d\lambda)}}{\kappa_\mu\sqrt{\lambda_{\min(\boldsymbol{V}_{i,t-1})}}}.\end{aligned} \tag{A.315}$$

□

Then, we prove the following lemma, which gives a sufficient time T_0 for the COLDB algorithm to cluster all the users correctly with high probability.

Lemma A.8.2. *With the carefully designed edge deletion rule, after*

$$\begin{aligned} T_0 &\triangleq 16u\log(\frac{u}{\delta}) + 4u\max\{\frac{128d}{\kappa_\mu^2\tilde{\lambda}_x\gamma^2}\log(\frac{u}{\delta}), \frac{16}{\tilde{\lambda}_x^2}\log(\frac{8ud}{\tilde{\lambda}_x^2\delta})\} \\ &= O\left(u\left(\frac{d}{\kappa_\mu^2\tilde{\lambda}_x\gamma^2} + \frac{1}{\tilde{\lambda}_x^2}\right)\log\frac{1}{\delta}\right) \end{aligned}$$

rounds, with probability at least $1-3\delta$ *for some* $\delta \in (0, \frac{1}{3})$*, COLDB can cluster all the users correctly.*

Proof. Then, with the item regularity assumption stated in Assumption 9.4, Lemma J.1 in [4], together with Lemma 7 in [135], and applying a union bound, with probability at least $1-\delta$, for all $i \in \mathcal{U}$, at any t such that $T_{i,t} \geq \frac{16}{\tilde{\lambda}_x^2}\log(\frac{8ud}{\tilde{\lambda}_x^2\delta})$, we have:

$$\lambda_{\min}(\boldsymbol{V}_{i,t}) \geq 2\tilde{\lambda}_x T_{i,t}\,. \tag{A.316}$$

Then, together with Lemma A.8.1, we have: if $T_{i,t} \geq \frac{16}{\tilde{\lambda}_x^2}\log(\frac{8ud}{\tilde{\lambda}_x^2\delta})$, then with probability $\geq 1-2\delta$, we have:

$$\begin{aligned} \left\|\hat{\boldsymbol{\theta}}_{i,t} - \boldsymbol{\theta}^{j(i)}\right\| &\leq \frac{\sqrt{\lambda\kappa_\mu} + \sqrt{2\log(u/\delta) + d\log(1+4T_{i,t}\kappa_\mu/d\lambda)}}{\kappa_\mu\sqrt{\lambda_{\min(\boldsymbol{V}_{i,t-1})}}} \\ &\leq \frac{\sqrt{\lambda\kappa_\mu} + \sqrt{2\log(u/\delta) + d\log(1+4T_{i,t}\kappa_\mu/d\lambda)}}{\kappa_\mu\sqrt{2\tilde{\lambda}_x T_{i,t}}}\,. \end{aligned}$$

Now, let

$$\frac{\sqrt{\lambda\kappa_\mu}+\sqrt{2\log(u/\delta)+d\log(1+4T_{i,t}\kappa_\mu/d\lambda)}}{\kappa_\mu\sqrt{2\tilde{\lambda}_x T_{i,t}}}<\frac{\gamma}{4}\,, \tag{A.317}$$

Let $\lambda\kappa_\mu \leq 2\log(u/\delta)+d\log(1+4T_{i,t}\kappa_\mu/d\lambda)$, which typically holds (κ_μ is typically very small), we can get

$$\frac{2\log(u/\delta)+d\log(1+4T_{i,t}\kappa_\mu/d\lambda)}{2\kappa_\mu^2\tilde{\lambda}_x T_{i,t}}<\frac{\gamma^2}{64}\,, \tag{A.318}$$

and a sufficient condition for it to hold is

$$\frac{2\log(u/\delta)}{2\kappa_\mu^2\tilde{\lambda}_x T_{i,t}}<\frac{\gamma^2}{128} \tag{A.319}$$

and

$$\frac{d\log(1+4T_{i,t}\kappa_\mu/d\lambda)}{2\kappa_\mu^2\tilde{\lambda}_x T_{i,t}}<\frac{\gamma^2}{128}\,. \tag{A.320}$$

Solving Eq.(A.319), we can get

$$T_{i,t}\geq\frac{128\log(u/\delta)}{\kappa_\mu^2\tilde{\lambda}_x\gamma^2}\,. \tag{A.321}$$

Following Lemma 9 in [135], we can get the following sufficient condition for Eq.(A.320):

$$T_{i,t}\geq\frac{128d}{\kappa_\mu^2\tilde{\lambda}_x\gamma^2}\log(\frac{512}{\lambda\kappa_\mu\tilde{\lambda}_x\gamma^2})\,. \tag{A.322}$$

Let $u/\delta\geq 512/\lambda\kappa_\mu\tilde{\lambda}_x\gamma^2$, which is typically held. Then, combining all together, we have that if

$$T_{i,t}\geq\max\{\frac{128d}{\kappa_\mu^2\tilde{\lambda}_x\gamma^2}\log(\frac{u}{\delta}),\frac{16}{\tilde{\lambda}_x^2}\log(\frac{8ud}{\tilde{\lambda}_x^2\delta})\},\forall i\in\mathcal{U}\,, \tag{A.323}$$

then with probability at least $1-2\delta$, we have

$$\left\|\hat{\boldsymbol{\theta}}_{i,t}-\boldsymbol{\theta}^{j(i)}\right\|<\gamma/4,\forall i\in\mathcal{U}\,. \tag{A.324}$$

By Lemma 8 in [135], and Assumption 9.3 of user arrival uniformness, we have that for all

$$\begin{aligned}T_0 &\triangleq 16u\log(\frac{u}{\delta})+4u\max\{\frac{128d}{\kappa_\mu^2\tilde{\lambda}_x\gamma^2}\log(\frac{u}{\delta}),\frac{16}{\tilde{\lambda}_x^2}\log(\frac{8ud}{\tilde{\lambda}_x^2\delta})\}\\ &=O\left(u\left(\frac{d}{\kappa_\mu^2\tilde{\lambda}_x\gamma^2}+\frac{1}{\tilde{\lambda}_x^2}\right)\log\frac{1}{\delta}\right),\end{aligned}$$

the condition in Eq.(A.323) is satisfied with probability at least $1-\delta$.

Therefore we have that for all $t\geq T_0$, with probability $\geq 1-3\delta$:

$$\left\|\hat{\boldsymbol{\theta}}_{i,t}-\boldsymbol{\theta}^{j(i)}\right\|_2<\frac{\gamma}{4}\,,\forall i\in\mathcal{U}\,. \tag{A.325}$$

Finally, we only need to show that with $\left\|\hat{\boldsymbol{\theta}}_{i,t}-\boldsymbol{\theta}^{j(i)}\right\|_2<\frac{\gamma}{4}\,,\forall i\in\mathcal{U}$, the algorithm can cluster all the users correctly. First, when the edge (i,l) is deleted, user i and user j must belong to different *ground-truth clusters*, i.e., $\|\boldsymbol{\theta}_i-\boldsymbol{\theta}_l\|_2>0$. This is because by the deletion rule of the algorithm, the concentration bound, and triangle inequality

$$\begin{aligned}&\|\boldsymbol{\theta}_i-\boldsymbol{\theta}_l\|_2=\left\|\boldsymbol{\theta}^{j(i)}-\boldsymbol{\theta}^{j(l)}\right\|_2\\ &\geq\left\|\hat{\boldsymbol{\theta}}_{i,t}-\hat{\boldsymbol{\theta}}_{l,t}\right\|_2-\left\|\boldsymbol{\theta}^{j(l)}-\hat{\boldsymbol{\theta}}_{l,t}\right\|_2-\left\|\boldsymbol{\theta}^{j(i)}-\hat{\boldsymbol{\theta}}_{i,t}\right\|_2\\ &\geq\left\|\hat{\boldsymbol{\theta}}_{i,t}-\hat{\boldsymbol{\theta}}_{l,t}\right\|_2-f(T_{i,t})-f(T_{l,t})>0\,.\end{aligned} \tag{A.326}$$

Second, we can show that if $\|\boldsymbol{\theta}_i-\boldsymbol{\theta}_l\|>\gamma$, meaning that user i and user l are not in the same *ground-truth cluster*, COLDB will

delete the edge (i,l) after T_0. This is because

$$\begin{aligned}\left\|\hat{\boldsymbol{\theta}}_{i,t}-\hat{\boldsymbol{\theta}}_{l,t}\right\| &\geq \|\boldsymbol{\theta}_i-\boldsymbol{\theta}_l\| - \left\|\hat{\boldsymbol{\theta}}_{i,t}-\boldsymbol{\theta}^{j(i)}\right\|_2 - \left\|\hat{\boldsymbol{\theta}}_{l,t}-\boldsymbol{\theta}^{j(l)}\right\|_2 \\ &> \gamma - \frac{\gamma}{4} - \frac{\gamma}{4} \\ &= \frac{\gamma}{2} > f(T_{i,t}) + f(T_{l,t})\,, \end{aligned} \tag{A.327}$$

which will trigger the edge deletion rule to delete edge (i,l). Combining all the reasoning above, we can finish the proof. □

Then, we prove the following lemmas for the cluster-based statistics.

Lemma A.8.3. *With probability at least* $1-4\delta$ *for some* $\delta \in (0,1/4)$*, at any* $t \geq T_0$*:*

$$\left\|\overline{\boldsymbol{\theta}}_t - \boldsymbol{\theta}_{i_t}\right\|_{\boldsymbol{V}_{t-1}} \leq \frac{\sqrt{\lambda\kappa_\mu}+\sqrt{2\log(u/\delta)+d\log(1+4T\kappa_\mu/d\lambda)}}{\kappa_\mu}\,. \tag{A.328}$$

Proof. First, by Lemma A.8.2, we have that with probability at least $1-3\delta$, all the users are clustered correctly, i.e., $\overline{C}_t = C_{j(i_t)}, \forall t \geq T_0$. Recall that $f_i(\boldsymbol{x}) = \boldsymbol{\theta}_i^\top \phi(\boldsymbol{x})$. In iteration s, define $\widetilde{\phi}_s = \phi(\boldsymbol{x}_{s,1}) - \phi(\boldsymbol{x}_{s,2})$. And we define $\widetilde{f}_{i,s} = f_i(\boldsymbol{x}_{s,1}) - f_i(\boldsymbol{x}_{s,2}) = \boldsymbol{\theta}_i^\top \widetilde{\phi}_s$.

For any $\boldsymbol{\theta}_{f'} \in \mathbb{R}^d$, define

$$G_t(\boldsymbol{\theta}_{f'}) = \sum_{\substack{s\in[t-1]:\\ i_s\in\overline{C}_t}} \left(\mu(\boldsymbol{\theta}_{f'}^\top\widetilde{\phi}_s) - \mu(\boldsymbol{\theta}_{i_t}^\top\widetilde{\phi}_s)\right)\widetilde{\phi}_s + \lambda\boldsymbol{\theta}_{f'}.$$

For $\lambda' \in (0,1)$, setting $\boldsymbol{\theta}_{\bar{f}} = \lambda'\boldsymbol{\theta}_{f_1'} + (1-\lambda')\boldsymbol{\theta}_{f_2'}$. and using the mean-value theorem, we get:

$$G_t(\boldsymbol{\theta}_{f_1'}) - G_t(\boldsymbol{\theta}_{f_2'}) = \left[\sum_{\substack{s\in[t-1]:\\ i_s\in\overline{C}_t}} \nabla\mu(\boldsymbol{\theta}_{\bar{f}}^\top\widetilde{\phi}_s)\widetilde{\phi}_s\widetilde{\phi}_s^\top + \lambda\mathbf{I}\right](\boldsymbol{\theta}_{f_1'} - \boldsymbol{\theta}_{f_2'})$$

(A.329)

Define $\boldsymbol{M}_{t-1} = \left[\sum_{\substack{s\in[t-1]:\\ i_s\in\overline{C}_t}} \nabla\mu(\boldsymbol{\theta}_{\bar{f}}^\top\widetilde{\phi}_s)\widetilde{\phi}_s\widetilde{\phi}_s^\top + \lambda\mathbf{I}\right]$, and recall that $\boldsymbol{V}_{t-1} = \sum_{\substack{s\in[t-1]:\\ i_s\in\overline{C}_t}} \widetilde{\phi}_s\widetilde{\phi}_s^\top + \frac{\lambda}{\kappa_\mu}\mathbf{I}$. Then we have that $\boldsymbol{M}_{t-1} \succeq \kappa_\mu \boldsymbol{V}_{t-1}$ and that $\boldsymbol{V}_{t-1}^{-1} \succeq \kappa_\mu \boldsymbol{M}_{t-1}^{-1}$. Then we have

$$
\begin{aligned}
&\left\|G_t(\overline{\boldsymbol{\theta}}_t) - \lambda\boldsymbol{\theta}_{i_t}\right\|_{\boldsymbol{V}_{t-1}^{-1}}^2 \\
&= \left\|G_t(\boldsymbol{\theta}_{i_t}) - G_t(\overline{\boldsymbol{\theta}}_t)\right\|_{\boldsymbol{V}_{t-1}^{-1}}^2 \\
&= \left\|\boldsymbol{M}_{t-1}(\boldsymbol{\theta}_{i_t} - \overline{\boldsymbol{\theta}}_t)\right\|_{\boldsymbol{V}_{t-1}^{-1}}^2 && (G_t(\boldsymbol{\theta}_{i_t}) = \lambda\boldsymbol{\theta}_{i_t} \text{ by definition}) \\
&= (\boldsymbol{\theta}_{i_t} - \overline{\boldsymbol{\theta}}_t)^\top \boldsymbol{M}_{t-1}\boldsymbol{V}_{t-1}^{-1}\boldsymbol{M}_{t-1}(\boldsymbol{\theta}_{i_t} - \overline{\boldsymbol{\theta}}_t) \\
&\geq (\boldsymbol{\theta}_{i_t} - \overline{\boldsymbol{\theta}}_t)^\top \boldsymbol{M}_{t-1}\kappa_\mu\boldsymbol{M}_{t-1}^{-1}\boldsymbol{M}_{t-1}(\boldsymbol{\theta}_{i_t} - \overline{\boldsymbol{\theta}}_t) \\
&= \kappa_\mu(\boldsymbol{\theta}_{i_t} - \overline{\boldsymbol{\theta}}_t)^\top \boldsymbol{M}_{t-1}(\boldsymbol{\theta}_{i_t} - \overline{\boldsymbol{\theta}}_t) \\
&\geq \kappa_\mu(\boldsymbol{\theta}_{i_t} - \overline{\boldsymbol{\theta}}_t)^\top \kappa_\mu\boldsymbol{V}_{t-1}(\boldsymbol{\theta}_{i_t} - \overline{\boldsymbol{\theta}}_t) \\
&= \kappa_\mu^2(\boldsymbol{\theta}_{i_t} - \overline{\boldsymbol{\theta}}_t)^\top \boldsymbol{V}_{t-1}(\boldsymbol{\theta}_{i_t} - \overline{\boldsymbol{\theta}}_t) \\
&= \kappa_\mu^2\left\|\boldsymbol{\theta}_{i_t} - \overline{\boldsymbol{\theta}}_t\right\|_{\boldsymbol{V}_{t-1}}^2 && \left(\text{as } ||\boldsymbol{x}||_{\boldsymbol{A}}^2 = \boldsymbol{x}^\top\boldsymbol{A}\boldsymbol{x}\right)
\end{aligned}
$$

The first inequality is because $\boldsymbol{V}_{t-1}^{-1} \succeq \kappa_\mu\boldsymbol{M}_{t-1}^{-1}$, and the second inequality follows from $\boldsymbol{M}_{t-1} \succeq \kappa_\mu\boldsymbol{V}_{t-1}$.

Note that $\frac{\kappa_\mu}{\lambda}\mathbf{I} \succeq \boldsymbol{V}_{t-1}$, which allows us to show that

$$
\|\lambda\boldsymbol{\theta}_{i_t}\|_{\boldsymbol{V}_{t-1}^{-1}} = \lambda\sqrt{\boldsymbol{\theta}_{i_t}^\top\boldsymbol{V}_{t-1}^{-1}\boldsymbol{\theta}_{i_t}} \leq \lambda\sqrt{\boldsymbol{\theta}_{i_t}^\top\frac{\kappa_\mu}{\lambda}\boldsymbol{\theta}_{i_t}} \leq \sqrt{\lambda\kappa_\mu}\,\|\boldsymbol{\theta}_{i_t}\|_2 \leq \sqrt{\lambda\kappa_\mu}.
$$

(A.330)

Using the two equations above, we have that

$$\left\|\boldsymbol{\theta}_{i_t}-\overline{\boldsymbol{\theta}}_t\right\|_{\boldsymbol{V}_{t-1}} \leq \frac{1}{\kappa_\mu}\left\|G_t(\overline{\boldsymbol{\theta}}_t)-\lambda\boldsymbol{\theta}_{i_t}\right\|_{\boldsymbol{V}_{t-1}^{-1}} \leq \frac{1}{\kappa_\mu}\left\|G_t(\overline{\boldsymbol{\theta}}_t)\right\|_{\boldsymbol{V}_{t-1}^{-1}} + \frac{1}{\kappa_\mu}\left\|\lambda\boldsymbol{\theta}_{i_t}\right\|_{\boldsymbol{V}_{t-1}^{-1}}$$
$$\leq \frac{1}{\kappa_\mu}\left\|G_t(\overline{\boldsymbol{\theta}}_t)\right\|_{\boldsymbol{V}_{t-1}^{-1}} + \sqrt{\frac{\lambda}{\kappa_\mu}} \tag{A.331}$$

Then, let $\overline{f}_{t,s} = \overline{\boldsymbol{\theta}}_t^\top \widetilde{\phi}_s$, we have:

$$\begin{aligned}
\frac{1}{\kappa_\mu^2}\left\|G_t(\overline{\boldsymbol{\theta}}_t)\right\|_{\boldsymbol{V}_{t-1}^{-1}}^2 &\leq \frac{1}{\kappa_\mu^2}\left\|\sum_{\substack{s\in[t-1]:\\ i_s\in\overline{C}_t}}(\mu(\overline{\boldsymbol{\theta}}_t^\top\widetilde{\phi}_s)-\mu(\boldsymbol{\theta}_{i_t}^\top\widetilde{\phi}_s))\widetilde{\phi}_s+\lambda\overline{\boldsymbol{\theta}}_t\right\|_{\boldsymbol{V}_{t-1}^{-1}}^2 \\
&= \frac{1}{\kappa_\mu^2}\left\|\sum_{\substack{s\in[t-1]:\\ i_s\in\overline{C}_t}}(\mu(\overline{f}_{t,s})-\mu(\widetilde{f}_{i_t,s}))\widetilde{\phi}_s+\lambda\overline{\boldsymbol{\theta}}_t\right\|_{\boldsymbol{V}_{t-1}^{-1}}^2 \\
&= \frac{1}{\kappa_\mu^2}\left\|\sum_{\substack{s\in[t-1]:\\ i_s\in\overline{C}_t}}(\mu(\overline{f}_{t,s})-(y_s-\epsilon_s))\widetilde{\phi}_s+\lambda\overline{\boldsymbol{\theta}}_t\right\|_{\boldsymbol{V}_{t-1}^{-1}}^2 \\
&\left(y_s=\mu(\widetilde{f}_{i_t,s})+\epsilon_s \text{if } i_s=i_t, \text{and} i_s=i_t, \forall i_s\in\overline{C}_t, \forall t\geq T_0\right) \\
&= \frac{1}{\kappa_\mu^2}\left\|\sum_{\substack{s\in[t-1]:\\ i_s\in\overline{C}_t}}\left(\mu(\overline{f}_{t,s})-y_s\right)\widetilde{\phi}_s+\sum_{\substack{s\in[t-1]:\\ i_s\in\overline{C}_t}}\epsilon_s\widetilde{\phi}_s+\lambda\overline{\boldsymbol{\theta}}_t\right\|_{\boldsymbol{V}_{t-1}^{-1}}^2 \\
&\leq \frac{1}{\kappa_\mu^2}\left\|\sum_{\substack{s\in[t-1]:\\ i_s\in\overline{C}_t}}\epsilon_s\widetilde{\phi}_s\right\|_{\boldsymbol{V}_{t-1}^{-1}}^2 .
\end{aligned}$$

The last step holds due to the following reasoning. Recall that

$\overline{\boldsymbol{\theta}}_t$ is computed using MLE by solving the following equation:

$$\overline{\boldsymbol{\theta}}_t = \arg\min_{\boldsymbol{\theta}} - \sum_{\substack{s\in[t-1]\\ i_s\in\overline{C}_t}} \Big(y_s \log\mu\Big(\boldsymbol{\theta}^\top\left[\phi(\boldsymbol{x}_{s,1}) - \phi(\boldsymbol{x}_{s,2})\right]\Big) + (1-y_s)\log\mu\left(\boldsymbol{\theta}^\top\left[\phi(\boldsymbol{x}_{s,2}) - \phi(\boldsymbol{x}_{s,1})\right]\right)\Big) + \frac{1}{2}\lambda\left\|\boldsymbol{\theta}\right\|_2^2. \quad \text{(A.332)}$$

Setting its gradient to 0, the following is satisfied:

$$\sum_{\substack{s\in[t-1]:\\ i_s\in\overline{C}_t}} \left(\mu\left(\overline{\boldsymbol{\theta}}_t^\top \widetilde{\phi}_s\right) - y_s\right)\widetilde{\phi}_s + \lambda\overline{\boldsymbol{\theta}}_t = 0, \quad \text{(A.333)}$$

which is used in the last step.

Now we have

$$\left\|\boldsymbol{\theta}_{i_t} - \overline{\boldsymbol{\theta}}_t\right\|_{\boldsymbol{V}_{t-1}} \le \frac{1}{\kappa_\mu}\left\|\sum_{\substack{s\in[t-1]:\\ i_s\in\overline{C}_t}} \epsilon_s\widetilde{\phi}_s\right\|_{\boldsymbol{V}_{t-1}^{-1}} + \sqrt{\frac{\lambda}{\kappa_\mu}}. \quad \text{(A.334)}$$

Denote $\boldsymbol{V} \triangleq \frac{\lambda}{\kappa_\mu}\mathbf{I}$. Note that the sequence of observation noises $\{\epsilon_s\}$ is 1-sub-Gaussian.

Next, we can apply Theorem 1 from [43], to obtain

$$\left\|\sum_{\substack{s\in[t-1]:\\ i_s\in\overline{C}_t}} \epsilon_s\widetilde{\phi}_s\right\|^2_{\boldsymbol{V}_{t-1}^{-1}} \le 2\log\left(\frac{\det(\boldsymbol{V}_{t-1})^{1/2}}{\delta\det(\boldsymbol{V})^{1/2}}\right), \quad \text{(A.335)}$$

which holds with probability of at least $1-\delta$.

Next, based on our assumption that $\left\|\widetilde{\phi}_s\right\|_2 \le 2$, according to Lemma 10 from [43], we have that

$$\det(\boldsymbol{V}_{t-1}) \le \left(\lambda/\kappa_\mu + 4T/d\right)^d. \quad \text{(A.336)}$$

Therefore,

$$\sqrt{\frac{\det\boldsymbol{V}_{t-1}}{\det(V)}} \le \sqrt{\frac{\left(\lambda/\kappa_\mu + 4T/d\right)^d}{\left(\lambda/\kappa_\mu\right)^d}} = \left(1 + 4T\kappa_\mu/(d\lambda)\right)^{\frac{d}{2}} \quad \text{(A.337)}$$

This gives us

$$\left\| \sum_{\substack{s\in[t-1]:\\ i_s\in\overline{C}_t}} \epsilon_s \widetilde{\phi}_s \right\|^2_{\boldsymbol{V}_{t-1}^{-1}} \leq 2\log\left(\frac{\det(\boldsymbol{V}_{t-1})^{1/2}}{\delta\det(V)^{1/2}}\right) \leq 2\log(1/\delta)+d\log\left(1+4T\kappa_\mu/(d\lambda)\right) \tag{A.338}$$

Combining all together, we have with probability at least $1-4\delta$ for some $\delta\in(0,1/4)$, at any $t\geq T_0$:

$$\left\|\overline{\boldsymbol{\theta}}_t-\boldsymbol{\theta}_{i_t}\right\|_{\boldsymbol{V}_{t-1}} \leq \frac{\sqrt{\lambda\kappa_\mu}+\sqrt{2\log(u/\delta)+d\log(1+4T\kappa_\mu/d\lambda)}}{\kappa_\mu}. \tag{A.339}$$

□

Then, we prove the following lemma with the help of Lemma A.8.3.

Lemma A.8.4. *For any iteration $t\geq T_0$, for all $\boldsymbol{x},\boldsymbol{x}'\in\mathcal{X}_t$, with probability of at least $1-4\delta$, we have*

$$\left|\left(f_{i_t}(\boldsymbol{x})-f_{i_t}(\boldsymbol{x}')\right)-\overline{\boldsymbol{\theta}}_t^\top\left(\phi(\boldsymbol{x})-\phi(\boldsymbol{x}')\right)\right| \leq \frac{\beta_T}{\kappa_\mu}\left\|\phi(\boldsymbol{x})-\phi(\boldsymbol{x}')\right\|_{\boldsymbol{V}_{t-1}^{-1}},$$

where $\beta_T=\sqrt{\lambda\kappa_\mu}+\sqrt{2\log(u/\delta)+d\log(1+4T\kappa_\mu/d\lambda)}$.

Proof.

$$\begin{aligned}
&\left|\left(f_{i_t}(\boldsymbol{x})-f_{i_t}(\boldsymbol{x}')\right)-\overline{\boldsymbol{\theta}}_t^\top\left(\phi(\boldsymbol{x})-\phi(\boldsymbol{x}')\right)\right| \\
&=\left|\boldsymbol{\theta}_{i_t}^\top\left[\left(\phi(\boldsymbol{x})-\phi(\boldsymbol{x}')\right]-\overline{\boldsymbol{\theta}}_t^\top\left[\phi(\boldsymbol{x})-\phi(\boldsymbol{x}')\right]\right| \\
&=\left|\left(\boldsymbol{\theta}_{i_t}-\overline{\boldsymbol{\theta}}_t\right)^\top\left[\phi(\boldsymbol{x})-\phi(\boldsymbol{x}')\right]\right| \\
&\leq \left\|\boldsymbol{\theta}_{i_t}-\overline{\boldsymbol{\theta}}_t\right\|_{\boldsymbol{V}_{t-1}}\left\|\phi(\boldsymbol{x})-\phi(\boldsymbol{x}')\right\|_{\boldsymbol{V}_{t-1}^{-1}} \\
&\leq \frac{\beta_T}{\kappa_\mu}\left\|\phi(\boldsymbol{x})-\phi(\boldsymbol{x}')\right\|_{\boldsymbol{V}_{t-1}^{-1}},
\end{aligned}$$

in which the last inequality follows from Lemma A.8.3. □

We also prove the following lemma to upper bound the summation of squared norms which will be used in proving the final regret bound.

Lemma A.8.5. *With probability at least* $1-4\delta$*, we have*

$$\sum_{t=T_0}^{T} \mathbb{I}\{i_t \in C_j\} \left\|\phi(\boldsymbol{x}_{t,1}) - \phi(\boldsymbol{x}_{t,2})\right\|^2_{\boldsymbol{V}_{t-1}^{-1}} \leq 2d \log\left(1 + 4T\kappa_\mu/(d\lambda)\right), \forall j \in [m],$$

where $\mathbb{I}$ denotes the indicator function.

Proof. We denote $\widetilde{\phi}_t = \phi(\boldsymbol{x}_{t,1}) - \phi(\boldsymbol{x}_{t,2})$. Recall that we have assumed that $\left\|\phi(\boldsymbol{x}_{t,1}) - \phi(\boldsymbol{x}_{t,2})\right\|_2 \leq 2$. It is easy to verify that $\boldsymbol{V}_{t-1} \succeq \frac{\lambda}{\kappa_\mu} I$ and hence $\boldsymbol{V}_{t-1}^{-1} \preceq \frac{\kappa_\mu}{\lambda} I$. Therefore, we have that $\left\|\widetilde{\phi}_t\right\|^2_{\boldsymbol{V}_{t-1}^{-1}} \leq \frac{\kappa_\mu}{\lambda} \left\|\widetilde{\phi}_t\right\|_2^2 \leq \frac{4\kappa_\mu}{\lambda}$. We choose λ such that $\frac{4\kappa_\mu}{\lambda} \leq 1$, which ensures that $\left\|\widetilde{\phi}_t\right\|^2_{\boldsymbol{V}_{t-1}^{-1}} \leq 1$. Our proof here mostly follows from Lemma 11 of [43] and Lemma J.2 of [4]. To begin with, note that $x \leq 2\log(1+x)$ for $x \in [0,1]$. Denote $\boldsymbol{V}_{t,j} = \sum_{\substack{s\in[t-1]:\\ is\in C_j}} \widetilde{\phi}_s \widetilde{\phi}_s^\top + \frac{\lambda}{\kappa_\mu}\mathbf{I}$. Then we have that

$$\begin{aligned}
\sum_{t=T_0}^{T} \mathbb{I}\{i_t \in C_j\} \left\|\widetilde{\phi}_t\right\|^2_{\boldsymbol{V}_{t-1}^{-1}} &\leq \sum_{t=T_0}^{T} 2\log\left(1 + \mathbb{I}\{i_t \in C_j\} \left\|\widetilde{\phi}_t\right\|^2_{\boldsymbol{V}_{t-1}^{-1}}\right) \\
&= 2\left(\log\det V_{T,j} - \log\det V\right) \\
&= 2\log\frac{\det V_{T,j}}{\det V} \\
&\leq 2\log\left(\left(1 + 4T\kappa_\mu/(d\lambda)\right)^d\right) \\
&= 2d\log\left(1 + 4T\kappa_\mu/(d\lambda)\right).
\end{aligned} \tag{A.340}$$

The second inequality follows the same reasoning as (A.337). This completes the proof. □

Now we are ready to prove Theorem 9.4.1. First, we have

$$R_T = \sum_{t=1}^{T} r_t \leq T_0 + \sum_{t=T_0}^{T} r_t \,, \tag{A.341}$$

where we use that the reward at each round is bounded by 1.

Then, we only need to upper bound the regret after T_0. By Lemma A.8.2, we know that with probability at least $1 - 4\delta$, the algorithm can cluster all the users correctly, $\overline{C}_t = C_{j(i_t)}$, and the statements of all the above lemmas hold. We have that for any

$t \geq T_0$:

$$
\begin{aligned}
r_t &= f_{i_t}(\boldsymbol{x}_t^*) - f_{i_t}(\boldsymbol{x}_{t,1}) + f_{i_t}(\boldsymbol{x}_t^*) - f_{i_t}(x_{t,2}) \\
&\overset{(a)}{\leq} \overline{\boldsymbol{\theta}}_t^\top \left(\phi(\boldsymbol{x}_t^*) - \phi(\boldsymbol{x}_{t,1})\right) + \frac{\beta_T}{\kappa_\mu} \left\|\phi(\boldsymbol{x}_t^*) - \phi(\boldsymbol{x}_{t,1})\right\|_{\boldsymbol{V}_{t-1}^{-1}} + \overline{\boldsymbol{\theta}}_t^\top \left(\phi(\boldsymbol{x}_t^*) - \phi(\boldsymbol{x}_{t,2})\right) \\
&+ \frac{\beta_T}{\kappa_\mu} \left\|\phi(\boldsymbol{x}_t^*) - \phi(\boldsymbol{x}_{t,2})\right\|_{\boldsymbol{V}_{t-1}^{-1}} \\
&= \overline{\boldsymbol{\theta}}_t^\top \left(\phi(\boldsymbol{x}_t^*) - \phi(\boldsymbol{x}_{t,1})\right) + \frac{\beta_T}{\kappa_\mu} \left\|\phi(\boldsymbol{x}_t^*) - \phi(\boldsymbol{x}_{t,1})\right\|_{\boldsymbol{V}_{t-1}^{-1}} + \\
&\qquad \overline{\boldsymbol{\theta}}_t^\top \left(\phi(\boldsymbol{x}_t^*) - \phi(\boldsymbol{x}_{t,1})\right) + \overline{\boldsymbol{\theta}}_t^\top \left(\phi(\boldsymbol{x}_{t,1}) - \phi(\boldsymbol{x}_{t,2})\right) + \\
&\frac{\beta_T}{\kappa_\mu} \left\|\phi(\boldsymbol{x}_t^*) - \phi(\boldsymbol{x}_{t,1}) + \phi(\boldsymbol{x}_{t,1}) - \phi(\boldsymbol{x}_{t,2})\right\|_{\boldsymbol{V}_{t-1}^{-1}} \\
&\overset{(b)}{\leq} 2\overline{\boldsymbol{\theta}}_t^\top \left(\phi(x^*) - \phi(\boldsymbol{x}_{t,1})\right) + 2\frac{\beta_T}{\kappa_\mu} \left\|\phi(x^*) - \phi(\boldsymbol{x}_{t,1})\right\|_{\boldsymbol{V}_{t-1}^{-1}} + \\
&\qquad \overline{\boldsymbol{\theta}}_t^\top \left(\phi(\boldsymbol{x}_{t,1}) - \phi(\boldsymbol{x}_{t,2})\right) + \frac{\beta_T}{\kappa_\mu} \left\|\phi(\boldsymbol{x}_{t,1}) - \phi(\boldsymbol{x}_{t,2})\right\|_{\boldsymbol{V}_{t-1}^{-1}} \\
&\overset{(c)}{\leq} 2\overline{\boldsymbol{\theta}}_t^\top \left(\phi(\boldsymbol{x}_{t,2}) - \phi(\boldsymbol{x}_{t,1})\right) + 2\frac{\beta_T}{\kappa_\mu} \left\|\phi(\boldsymbol{x}_{t,2}) - \phi(\boldsymbol{x}_{t,1})\right\|_{\boldsymbol{V}_{t-1}^{-1}} + \\
&\qquad \overline{\boldsymbol{\theta}}_t^\top \left(\phi(\boldsymbol{x}_{t,1}) - \phi(\boldsymbol{x}_{t,2})\right) + \frac{\beta_T}{\kappa_\mu} \left\|\phi(\boldsymbol{x}_{t,1}) - \phi(\boldsymbol{x}_{t,2})\right\|_{\boldsymbol{V}_{t-1}^{-1}} \\
&\leq \overline{\boldsymbol{\theta}}_t^\top \left(\phi(\boldsymbol{x}_{t,2}) - \phi(\boldsymbol{x}_{t,1})\right) + 3\frac{\beta_T}{\kappa_\mu} \left\|\phi(\boldsymbol{x}_{t,2}) - \phi(\boldsymbol{x}_{t,1})\right\|_{\boldsymbol{V}_{t-1}^{-1}} \\
&\overset{(d)}{\leq} 3\frac{\beta_T}{\kappa_\mu} \left\|\phi(\boldsymbol{x}_{t,1}) - \phi(\boldsymbol{x}_{t,2})\right\|_{\boldsymbol{V}_{t-1}^{-1}}
\end{aligned}
$$

Step (a) follows from Lemma A.8.4. Step (b) makes use of the triangle inequality. Step (c) follows from the way in which we choose the second arm $\boldsymbol{x}_{t,2}$: $\boldsymbol{x}_{t,2} = \arg\max_{x \in \mathcal{X}_t} \overline{\boldsymbol{\theta}}_t^\top \left(\phi(x) - \phi(\boldsymbol{x}_{t,1})\right) + \frac{\beta_T}{\kappa_\mu} \left\|\phi(x) - \phi(\boldsymbol{x}_{t,1})\right\|_{\boldsymbol{V}_{t-1}^{-1}}$. Step (d) results from the way in which we select the first arm: $\boldsymbol{x}_{t,1} = \arg\max_{x \in \mathcal{X}_t} \overline{\boldsymbol{\theta}}_t^\top \phi(x)$.

Then we have

$$\begin{aligned}\sum_{t=T_0}^{T} r_t &\leq 3\frac{\beta_T}{\kappa_\mu}\sum_{t=T_0}^{T}\|\phi(\boldsymbol{x}_{t,1})-\phi(\boldsymbol{x}_{t,2})\|_{\boldsymbol{V}_{t-1}^{-1}}\\ &= 3\frac{\beta_T}{\kappa_\mu}\sum_{t=T_0}^{T}\sum_{j\in[m]}\mathbb{I}\{i_t\in C_j\}\,\|\phi(\boldsymbol{x}_{t,1})-\phi(\boldsymbol{x}_{t,2})\|_{\boldsymbol{V}_{t-1}^{-1}}\\ &\leq 3\frac{\beta_T}{\kappa_\mu}\sqrt{\sum_{t=T_0}^{T}\sum_{j\in[m]}\mathbb{I}\{i_t\in C_j\}\sum_{t=T_0}^{T}\sum_{j\in[m]}\mathbb{I}\{i_t\in C_j\}\,\|\phi(\boldsymbol{x}_{t,1})-\phi(\boldsymbol{x}_{t,2})\|^2_{\boldsymbol{V}_{t-1}^{-1}}}\\ &\leq 3\frac{\beta_T}{\kappa_\mu}\sqrt{T\cdot m\cdot 2d\log\left(1+4T\kappa_\mu/(d\lambda)\right)}\,, \end{aligned}\tag{A.342}$$

where in the second inequality we use the Cauchy-Swarchz inequality, and in the last step we use $\sum_{t=T_0}^{T}\sum_{j\in[m]}\mathbb{I}\{i_t\in C_j\}\leq T$ and Lemma A.8.5.

Therefore, finally, we have with probability at least $1-4\delta$

$$\begin{aligned}R_T &\leq T_0+3\frac{\beta_T}{\kappa_\mu}\sqrt{T\cdot m\cdot 2d\log\left(1+4T\kappa_\mu/(d\lambda)\right)}\\ &\leq O(u(\frac{d}{\kappa_\mu^2\tilde{\lambda}_x\gamma^2}+\frac{1}{\tilde{\lambda}_x^2})\log T+\frac{1}{\kappa_\mu}d\sqrt{mT})\\ &= O(\frac{1}{\kappa_\mu}d\sqrt{mT})\,, \end{aligned}\tag{A.343}$$

A.9 Proof of Theorem 9.4.2

A.9.1 Auxiliary Definitions and Explanations

Denifition of the NTK matrix $\mathbf{H}_j$ for cluster j. Recall that we use T_j to denote the total number of iterations in which the users in cluster j are served. For cluster j, let $\{x_{(i)}\}_{i=1}^{T_jK}$ be a set of all $T_j\times K$ possible arm feature vectors: $\{x_{t,a}\}_{1\leq t\leq T_j,1\leq a\leq K}$, where

$i = K(t-1)+a$. Firstly, we define $\mathbf{h}_t = [f^j(x_{(i)})]_{i=1,\ldots,T_jK}^\top$, i.e., $\mathbf{h}_t$ is the T_jK-dimensional vector containing the reward function values of the arms corresponding to cluster j. Next, define

$$\widetilde{\mathbf{H}}^{(1)}_{p,q} = \overset{(1)}{}_{p,q} = \langle x_{(p)}, x_{(q)} \rangle, \mathbf{A}^{(l)}_{p,q} = \begin{pmatrix} \overset{(l)}{}_{p,q} & \overset{(l)}{}_{p,q} \\ \overset{(l)}{}_{p,q} & \overset{(l)}{}_{q,q} \end{pmatrix},$$

$$\overset{(l+1)}{}_{p,q} = 2\mathbb{E}_{(u,v)\sim\mathcal{N}(0,\mathbf{A}^{(l)}_{p,q})}[\max\{u,0\}\max\{v,0\}],$$

$$\widetilde{\mathbf{H}}^{(l+1)}_{p,q} = 2\widetilde{\mathbf{H}}^{(l)}_{p,q}\mathbb{E}_{(u,v)\sim\mathcal{N}(0,\mathbf{A}^{(l)}_{p,q})}[\mathbb{1}(u\geq 0)\mathbb{1}(v\geq 0)] + \overset{(l+1)}{}_{p,q}.$$

With these definitions, the NTK matrix for cluster j is then defined as $\mathbf{H}_j = (\widetilde{\mathbf{H}}^{(L)} + \overset{(L)}{})/2$.

The Initial Parameters $\boldsymbol{\theta}_0$. Next, we discuss how the initial parameters $\boldsymbol{\theta}_0$ are obtained. We adopt the same initialization method from [161, 160]. Specifically, for each $l = 1, \ldots, L-1$, let $\mathbf{W}_l = \begin{pmatrix} \mathbf{W} & \mathbf{0} \\ \mathbf{0} & \mathbf{W} \end{pmatrix}$ in which every entry of $\mathbf{W}$ is independently and randomly sampled from $\mathcal{N}(0, 4/m_{\text{NN}})$, and choose $\mathbf{W}_L = (\mathbf{w}^\top, -\mathbf{w}^\top)$ in which every entry of $\mathbf{w}$ is independently and randomly sampled from $\mathcal{N}(0, 2/m_{\text{NN}})$.

Justifications for Assumption 9.5. The last assumption in Assumption 9.5, together with the way we initialize θ_0 as discussed above, ensures that the initial output of the NN is 0: $h(x;\theta_0) = 0, \forall x \in \mathcal{X}$. The assumption of $x_j = x_{j+d/2}$ from Assumption 9.5 is a mild assumption which is commonly adopted by previous works on neural bandits [160, 161]. To ensure that this assumption holds, for any arm x, we can always firstly normalize it such that

$||x|| = 1$, and then construct a new context $x' = (x^\top, x^\top)^\top/\sqrt{2}$ to satisfy this assumption [160].

A.9.2 Proof

To begin with, we first list the specific conditions we need for the width m_{NN} of the NN:

$$\begin{aligned} &m_{\text{NN}} \geq CT^4K^4L^6\log(T^2K^2L/\delta)/\lambda_0^4, \\ &m_{\text{NN}}(\log m)^{-3} \geq C\kappa_\mu^{-3}T^8L^{21}\lambda^{-5}, \\ &m_{\text{NN}}(\log m_{\text{NN}})^{-3} \geq C\kappa_\mu^{-3}T^{14}L^{21}\lambda^{-11}L_\mu^6, \\ &m_{\text{NN}}(\log m_{\text{NN}})^{-3} \geq CT^{14}L^{18}\lambda^{-8}, \end{aligned} \tag{A.344}$$

for some absolute constant $C > 0$. To ease exposition, we express these conditions above as $m_{\text{NN}} \geq \text{poly}(T, L, K, 1/\kappa_\mu, L_\mu, 1/\lambda_0, 1/\lambda, \log(1/\delta))$.

In our proof here, we use the gradient of the NN at $\boldsymbol{\theta}_0$ to derive the feature mapping for the arms, i.e., we let $\phi(\boldsymbol{x}) = g(\boldsymbol{x};\boldsymbol{\theta}_0)/\sqrt{m_{\text{NN}}}$. We use $\hat{\boldsymbol{\theta}}_{i,t}$ to denote the paramters of the NN after training in iteration t (see Algorithm 19).

We use the following lemma to show that for every cluster $j \in \mathcal{C}$, its reward function f^j can be expressed as a linear function with respect to the initial gradient $g(\boldsymbol{x};\boldsymbol{\theta}_0)$.

Lemma A.9.1 (Lemma B.3 of [161])**.** *As long as the width m of the NN is large enough:*

$$m_{NN} \geq C_0T^4K^4L^6\log(T^2K^2L/\delta)/\lambda_0^4,$$

then for all clusters $j \in [m]$, with probability of at least $1-\delta$, there exits a $\boldsymbol{\theta}_f^j$ such that

$$f^j(\boldsymbol{x}) = \langle g(\boldsymbol{x};\boldsymbol{\theta}_0), \boldsymbol{\theta}_f^j - \boldsymbol{\theta}_0\rangle, \qquad \sqrt{m_{NN}}\left\|\boldsymbol{\theta}_f^j - \boldsymbol{\theta}_0\right\|_2 \leq \sqrt{2\mathbf{h}_j^\top\mathbf{H}_j^{-1}\mathbf{h}_j} \leq B.$$

for all $\boldsymbol{x} \in \mathcal{X}_t$, $t \in [T]$ *with* $i_t \in C_j$.

Lemma A.9.1 is the formal statement of Lemma 9.2.1 from Sec. 9.2.2. Note that the constant B is applicable to all m clusters.

The following lemma converts our assumption about cluster separation (Assumption 9.6) into the difference between the linearized parameters for different clusters.

Lemma A.9.2. *If users* i *and* l *belong to different clusters, then we have that*

$$\sqrt{m_{NN}}\,\|\boldsymbol{\theta}_{f,i} - \boldsymbol{\theta}_{f,l}\| \geq \gamma'.$$

Proof. To begin with, Lemma A.9.1 tells us that

$$|f_i(\boldsymbol{x}) - f_l(\boldsymbol{x})| = |\langle g(\boldsymbol{x};\boldsymbol{\theta}_0), \boldsymbol{\theta}_{f,i} - \boldsymbol{\theta}_{f,l}\rangle| \leq \|g(\boldsymbol{x};\boldsymbol{\theta}_0)\|\,\|\boldsymbol{\theta}_{f,i} - \boldsymbol{\theta}_{f,l}\|\,. \tag{A.345}$$

This leads to

$$\|\boldsymbol{\theta}_{f,i} - \boldsymbol{\theta}_{f,l}\| \geq \frac{|f_i(\boldsymbol{x}) - f_l(\boldsymbol{x})|}{\|g(\boldsymbol{x};\boldsymbol{\theta}_0)\|} \geq \frac{\gamma'}{\sqrt{m_{\mathrm{NN}}}}, \tag{A.346}$$

in which we have made use of Assumption 9.6 and our assumption that $\frac{1}{m_{\mathrm{NN}}}\langle g(\boldsymbol{x};\boldsymbol{\theta}_0), g(\boldsymbol{x};\boldsymbol{\theta}_0)\rangle \leq 1$ in the last inequality. This completes the proof. □

The following lemma shows that for every user, the output of the NN trained using its own local data can be approximated by a linear function.

Lemma A.9.3. *Let* $\varepsilon'_{m_{NN},t} \triangleq C_2 m_{NN}^{-1/6}\sqrt{\log m_{NN}}L^3\left(\frac{t}{\lambda}\right)^{4/3}$ *where* $C_2 > 0$ *is an absolute constant. Then*

$$|\langle g(\boldsymbol{x};\boldsymbol{\theta}_0), \hat{\boldsymbol{\theta}}_{i,t} - \boldsymbol{\theta}_0\rangle - h(\boldsymbol{x};\hat{\boldsymbol{\theta}}_{i,t})| \leq \varepsilon'_{m_{NN},t},\ \ \forall t \in [T], \boldsymbol{x}, \boldsymbol{x}' \in \mathcal{X}_t.$$

Proof. This lemma can be proved following a similar line of proof as Lemma 1 from [60]. Here the t in $\varepsilon'_{m_{\text{NN}},t}$ can in fact be replaced by $T_{i,t} \leq t$, however, we have simply used its upper bound t for simplicity. □

Lemma A.9.4. *Let* $\beta_T \triangleq \frac{1}{\kappa_\mu}\sqrt{\widetilde{d} + 2\log(u/\delta)}$. *Assuming that the conditions on* m_{NN} *from eq.* (A.344) *are satisfied. With probability of at least* $1-\delta$, *we have that*

$$\sqrt{m_{NN}}\left\|\boldsymbol{\theta}_{f,i} - \hat{\boldsymbol{\theta}}_{i,t}\right\|_2 \leq \frac{\beta_T + B\sqrt{\frac{\lambda}{\kappa_\mu}} + 1}{\sqrt{\lambda_{\min}(\boldsymbol{V}_{i,t-1})}}, \qquad \forall t \in [T].$$

where $\boldsymbol{V}_{i,t-1} = \frac{\lambda}{\kappa_\mu}\mathbf{I} + \sum_{\substack{s\in[t-1]\\ i_s=i}} (\phi(\boldsymbol{x}_{s,1}) - \phi(\boldsymbol{x}_{s,2}))(\phi(\boldsymbol{x}_{s,1}) - \phi(\boldsymbol{x}_{s,2}))^\top$, $\phi(\boldsymbol{x}) = \frac{1}{\sqrt{m_{NN}}} g(\boldsymbol{x};\boldsymbol{\theta}_0)$, *and* $T_{i,t}$ *denotes the number of rounds of seeing user* i *in the first* t *rounds.*

Proof. In iteration t, for any user $i \in \mathcal{U}$, the user leverages its current history of observations $\{(\boldsymbol{x}_{s,1}, \boldsymbol{x}_{s,2}, y_s)\}_{s\in[t-1], i_s=i}$ to train the NN by minimizing the loss function ((A.299)), to obtain the NN parameters $\hat{\boldsymbol{\theta}}_{i,t}$. Note that the NN has been trained when the most recent observation in $\{(\boldsymbol{x}_{s,1}, \boldsymbol{x}_{s,2}, y_s)\}_{s\in[t-1], i_s=i}$ was collected, i.e., the last time when user i was encountered. Of note, according to Lemma A.9.1, the latent reward function of user i can be expressed as $f_i(\boldsymbol{x}) = \langle g(\boldsymbol{x};\boldsymbol{\theta}_0), \boldsymbol{\theta}_{f,i} - \boldsymbol{\theta}_0\rangle$. Therefore, from the perspective of each individual user i, the user is faced with a *neural dueling bandit* problem instance. As a result, we can modifying the proof of Lemma 6 from [60] to show that with probability of at least $1-\delta$,

$$\sqrt{m_{\text{NN}}}\left\|\boldsymbol{\theta}_{f,i} - \hat{\boldsymbol{\theta}}_{i,t}\right\|_{\boldsymbol{V}_{i,t-1}} \leq \beta_T + B\sqrt{\frac{\lambda}{\kappa_\mu}} + 1, \qquad \forall t \in [T], i \in \mathcal{U}.$$

Here in our definition of $\beta_T \triangleq \frac{1}{\kappa_\mu}\sqrt{\widetilde{d} + 2\log(u/\delta)}$, we have replaced the error probability δ (from [60]) by δ/u to account for the use of an extra union bound over all u users.

This allows us to show that

$$\begin{aligned}\sqrt{m_{\mathrm{NN}}}\left\|\boldsymbol{\theta}_{f,i} - \hat{\boldsymbol{\theta}}_{i,t}\right\|_2 &\leq \frac{\sqrt{m_{\mathrm{NN}}}\left\|\boldsymbol{\theta}_{f,i} - \hat{\boldsymbol{\theta}}_{i,t}\right\|_{\boldsymbol{V}_{i,t-1}}}{\sqrt{\lambda_{\min}(\boldsymbol{V}_{i,t-1})}} \\ &\leq \frac{\beta_T + B\sqrt{\frac{\lambda}{\kappa_\mu}} + 1}{\sqrt{\lambda_{\min}(\boldsymbol{V}_{i,t-1})}}\end{aligned} \tag{A.347}$$

This completes the proof. □

Lemma A.9.5. *With the carefully designed edge deletion rule in Algorithm 19, after*

$$\begin{aligned}T_0 &\triangleq 16u\log(\frac{u}{\delta}) + 4u\max\left\{\frac{32\left(\widetilde{d} + 2\log(u/\delta)\right)}{\tilde{\lambda}_x\gamma^2\kappa_\mu^2}, \frac{16}{\tilde{\lambda}_x^2}\log(\frac{24udm^2(L-1)}{\tilde{\lambda}_x^2\delta})\right\} \\ &= O\left(u\left(\frac{\widetilde{d}}{\kappa_\mu^2\tilde{\lambda}_x\gamma^2} + \frac{1}{\tilde{\lambda}_x^2}\right)\log(\frac{1}{\delta})\right),\end{aligned}$$

rounds, with probability at least $1-3\delta$ *for some* $\delta \in (0, \frac{1}{3})$*, CONDB can cluster all the users correctly.*

Proof. Recall that we use $p = dm_{\mathrm{NN}} + m_{\mathrm{NN}}^2(L-1) + m_{\mathrm{NN}}$ to denote the total number of parameters of the NN. Similar to the proof of Lemma A.8.2, with the item regularity assumption stated in Assumption 9.4, Lemma J.1 in [4], together with Lemma 7 in [135] (note that when using these technical results, we use $g(\boldsymbol{x};\boldsymbol{\theta})/\sqrt{m_{\mathrm{NN}}}$ as the feature vector to replace the original feature vector of $\boldsymbol{x}$), and applying a union bound, with probability at

least $1-\delta$, for all $i \in \mathcal{U}$, at any t such that $T_{i,t} \geq \frac{16}{\tilde{\lambda}_x^2} \log(\frac{8up}{\tilde{\lambda}_x^2 \delta})$, we have:

$$\lambda_{\min}(\boldsymbol{V}_{i,t}) \geq 2\tilde{\lambda}_x T_{i,t} \,. \tag{A.348}$$

Note that compared with the proof of A.8.2, in the lower bound on $T_{i,t}$ here, we have replaced the dimension d by p. This has led to a logarithmic dependence on the width m_{NN} of the NN. To simplify the exposition, using the fact that $p \geq 3dm_{\text{NN}}^2(L-1)$, we replace this condition on $T_{i,t}$ by a slightly stricter condition: $T_{i,t} \geq \frac{16}{\tilde{\lambda}_x^2} \log(\frac{8u \times 3dm_{\text{NN}}^2(L-1)}{\tilde{\lambda}_x^2 \delta}) = \frac{16}{\tilde{\lambda}_x^2} \log(\frac{24udm_{\text{NN}}^2(L-1)}{\tilde{\lambda}_x^2 \delta})$.

Then, together with Lemma A.9.4, we have: if $T_{i,t} \geq \frac{16}{\tilde{\lambda}_x^2} \log(\frac{8u \times 3dm_{\text{NN}}^2(L-1)}{\tilde{\lambda}_x^2 \delta})$, then with probability $\geq 1-2\delta$, we have:

$$\sqrt{m_{\text{NN}}} \left\| \hat{\boldsymbol{\theta}}_{i,t} - \boldsymbol{\theta}^{j(i)} \right\| \leq \frac{\beta_T + B\sqrt{\frac{\lambda}{\kappa_\mu}} + 1}{\sqrt{\lambda_{\min}(\boldsymbol{V}_{i,t-1})}} \leq \frac{\beta_T + B\sqrt{\frac{\lambda}{\kappa_\mu}} + 1}{\sqrt{2\tilde{\lambda}_x T_{i,t}}} \,.$$

Now, let

$$\frac{\beta_T + B\sqrt{\frac{\lambda}{\kappa_\mu}} + 1}{\sqrt{2\tilde{\lambda}_x T_{i,t}}} < \frac{\gamma}{4} \,, \tag{A.349}$$

Note that in Algorithm 19, we have defined the funciton f as

$$f(T_{i,t}) \triangleq \frac{\beta_T + B\sqrt{\frac{\lambda}{\kappa_\mu}} + 1}{\sqrt{2\tilde{\lambda}_x T_{i,t}}} \tag{A.350}$$

This immediately leads to

$$\sqrt{m_{\text{NN}}} \left\| \hat{\boldsymbol{\theta}}_{i,t} - \boldsymbol{\theta}^{j(i)} \right\| \leq f(T_{i,t}) < \frac{\gamma}{4}. \tag{A.351}$$

For simplicity, now let $B\sqrt{\frac{\lambda}{\kappa_\mu}} + 1 \leq \beta_T$ which is typically

satisfied. This allows us to show that

$$T_{i,t} > \frac{32\beta_T^2}{\tilde{\lambda}_x\gamma^2} = \frac{32\left(\frac{1}{\kappa_\mu}\sqrt{\widetilde{d}+2\log(u/\delta)}\right)^2}{\tilde{\lambda}_x\gamma^2} = \frac{32\left(\widetilde{d}+2\log(u/\delta)\right)}{\tilde{\lambda}_x\gamma^2\kappa_\mu^2}. \tag{A.352}$$

Combining both conditions on $T_{i,t}$ together, we have that

$$T_{i,t} \geq \max\left\{\frac{32\left(\widetilde{d}+2\log(u/\delta)\right)}{\tilde{\lambda}_x\gamma^2\kappa_\mu^2}, \frac{16}{\tilde{\lambda}_x^2}\log(\frac{24udm_{\text{NN}}^2(L-1)}{\tilde{\lambda}_x^2\delta})\right\} \tag{A.353}$$

By Lemma 8 in [135] and Assumption 9.3 of user arrival uniformness, we have that for all

$$\begin{aligned} T_0 &\triangleq 16u\log(\frac{u}{\delta}) + 4u\max\left\{\frac{32\left(\widetilde{d}+2\log(u/\delta)\right)}{\tilde{\lambda}_x\gamma^2\kappa_\mu^2}, \frac{16}{\tilde{\lambda}_x^2}\log(\frac{24udm_{\text{NN}}^2(L-1)}{\tilde{\lambda}_x^2\delta})\right\} \\ &= O\left(u\left(\frac{\widetilde{d}}{\kappa_\mu^2\tilde{\lambda}_x\gamma^2}+\frac{1}{\tilde{\lambda}_x^2}\right)\log(\frac{1}{\delta})\right), \end{aligned}$$

the condition in Eq.(A.352) is satisfied with probability at least $1-\delta$.

Therefore we have that for all $t \geq T_0$, with probability $\geq 1-3\delta$:

$$\sqrt{m_{\text{NN}}}\left\|\hat{\boldsymbol{\theta}}_{i,t} - \boldsymbol{\theta}^{j(i)}\right\|_2 < \frac{\gamma}{4}, \forall i \in \mathcal{U}. \tag{A.354}$$

Finally, we show that as long as the condition $\sqrt{m_{\text{NN}}}\left\|\hat{\boldsymbol{\theta}}_{i,t} - \boldsymbol{\theta}^{j(i)}\right\|_2 < \frac{\gamma}{4}, \forall i \in \mathcal{U}$, our algorithm can cluster all the users correctly.

First, we show that when the edge (i,l) is deleted, user i and user j must belong to different *ground-truth clusters*, i.e., $\|\boldsymbol{\theta}_{f,i} - \boldsymbol{\theta}_{f,l}\|_2 > 0$. This is because by the deletion rule of the

algorithm, the concentration bound, and triangle inequality

$$\begin{aligned}
&\sqrt{m_{\mathrm{NN}}}\left\|\boldsymbol{\theta}_{f,i}-\boldsymbol{\theta}_{f,l}\right\|_2=\sqrt{m_{\mathrm{NN}}}\left\|\boldsymbol{\theta}^{j(i)}-\boldsymbol{\theta}^{j(l)}\right\|_2\\
&\geq\sqrt{m_{\mathrm{NN}}}\left\|\hat{\boldsymbol{\theta}}_{i,t}-\hat{\boldsymbol{\theta}}_{l,t}\right\|_2-\sqrt{m_{\mathrm{NN}}}\left\|\boldsymbol{\theta}^{j(l)}-\hat{\boldsymbol{\theta}}_{l,t}\right\|_2-\sqrt{m_{\mathrm{NN}}}\left\|\boldsymbol{\theta}^{j(i)}-\hat{\boldsymbol{\theta}}_{i,t}\right\|_2\\
&\geq\sqrt{m_{\mathrm{NN}}}\left\|\hat{\boldsymbol{\theta}}_{i,t}-\hat{\boldsymbol{\theta}}_{l,t}\right\|_2-f(T_{i,t})-f(T_{l,t})>0\,. \qquad \text{(A.355)}
\end{aligned}$$

Second, we can show that if $|f_i(\boldsymbol{x})-f_l(\boldsymbol{x})|\geq\gamma',\forall\boldsymbol{x}\in\mathcal{X}$, meaning that user i and user l are not in the same *ground-truth cluster*, CONDB will delete the edge (i,l) after T_0. Note that when user i and user l are not in the same *ground-truth cluster*, Lemma A.9.2 tells us that $\sqrt{m_{\mathrm{NN}}}\left\|\boldsymbol{\theta}_{f,i}-\boldsymbol{\theta}_{f,l}\right\|\geq\gamma'$. Then we have that

$$\begin{aligned}
&\sqrt{m_{\mathrm{NN}}}\left\|\hat{\boldsymbol{\theta}}_{i,t}-\hat{\boldsymbol{\theta}}_{l,t}\right\|\\
&\geq\sqrt{m_{\mathrm{NN}}}\left\|\boldsymbol{\theta}_{f,i}-\boldsymbol{\theta}_{f,l}\right\|-\sqrt{m_{\mathrm{NN}}}\left\|\hat{\boldsymbol{\theta}}_{i,t}-\boldsymbol{\theta}^{j(i)}\right\|_2-\sqrt{m_{\mathrm{NN}}}\left\|\hat{\boldsymbol{\theta}}_{l,t}-\boldsymbol{\theta}^{j(l)}\right\|_2\\
&>\gamma-\frac{\gamma}{4}-\frac{\gamma}{4}\\
&=\frac{\gamma}{2}>f(T_{i,t})+f(T_{l,t})\,, \qquad \text{(A.356)}
\end{aligned}$$

which will trigger the edge deletion rule to delete edge (i,l). This completes the proof. □

Then, we prove the following lemmas for the cluster-based statistics.

Lemma A.9.6. *Assuming that the conditions on m from eq.* (A.344) *are satisfied. With probability at least* $1-4\delta$ *for some* $\delta\in(0,1/4)$*, at any* $t\geq T_0$*:*

$$\sqrt{m_{NN}}\left\|\boldsymbol{\theta}_{f,i_t}-\overline{\boldsymbol{\theta}}_t\right\|_{\boldsymbol{V}_{t-1}}\leq\beta_T+B\sqrt{\frac{\lambda}{\kappa_\mu}}+1,\qquad\forall t\in[T].$$

Proof. To begin with, note that by Lemma A.9.5, we have that with probability of at least $1-3\delta$, all users are clustered correctly, i.e., $\overline{C}_t = C_{j(i_t)}, \forall t \geq T_0$. Note that according to our Algorithm 19, in iteration t, we select the pair of arms using all the data collected by all users in cluster $\overline{C}_t$. That is, $\overline{\boldsymbol{\theta}}_t$ represents the NN parameters trained using the data from all users in the cluster $\overline{C}_t$ (i.e., $\{(\boldsymbol{x}_{s,1}, \boldsymbol{x}_{s,2}, y_s)\}_{s\in[t-1], i_s\in\overline{C}_t}$), and $\boldsymbol{V}_t$ also contains the data from all users in this cluster $\overline{C}_t$. Therefore, in iteration t, we are effectively following a neural dueling bandit algorithm using $\{(\boldsymbol{x}_{s,1}, \boldsymbol{x}_{s,2}, y_s)\}_{s\in[t-1], i_s\in\overline{C}_t}$ as the current observation history. This allows us to leverage the proof of Lemma 6 from [60] to complete the proof. □

Lemma A.9.7. *Let* $\varepsilon'_{m_{NN},t} \triangleq C_2 m_{NN}^{-1/6}\sqrt{\log m_{NN}}L^3\left(\frac{t}{\lambda}\right)^{4/3}$ *where* $C_2 > 0$ *is an absolute constant. Then*

$$|\langle g(\boldsymbol{x};\boldsymbol{\theta}_0)-g(\boldsymbol{x}';\boldsymbol{\theta}_0), \overline{\boldsymbol{\theta}}_t-\boldsymbol{\theta}_0\rangle-(h(\boldsymbol{x};\overline{\boldsymbol{\theta}}_t)-h(\boldsymbol{x}';\overline{\boldsymbol{\theta}}_t))| \leq 2\varepsilon'_{m_{NN},t}, \quad \forall t\in[T], \boldsymbol{x}, \boldsymbol{x}'\in\mathcal{X}_t.$$

Proof. This lemma can be proved following a similar line of proof as Lemma 1 from [60]. □

Lemma A.9.8. *Let* $\delta \in (0,1)$, $\varepsilon'_{m_{NN},t} \doteq C_2 m_{NN}^{-1/6}\sqrt{\log m_{NN}}L^3\left(\frac{t}{\lambda}\right)^{4/3}$ *for some* $C_2 > 0$. *As long as* $m_{NN} \geq poly(T, L, K, u, 1/\kappa_\mu, L_\mu, 1/\lambda_0, 1/\lambda, \log(1/\delta))$, *then with probability of at least* $1-\delta$, *at any* $t \geq T_0$,

$$|\,[f_{i_t}(\boldsymbol{x}) - f_{i_t}(\boldsymbol{x}')]-\left[h(\boldsymbol{x};\overline{\boldsymbol{\theta}}_t) - h(\boldsymbol{x}';\overline{\boldsymbol{\theta}}_t)\right]\,| \leq \nu_T\sigma_{t-1}(\boldsymbol{x},\boldsymbol{x}')+2\varepsilon'_{m_{NN},t},$$

for all $\boldsymbol{x}, \boldsymbol{x}' \in \mathcal{X}_t, t \in [T]$.

Proof. Denote $\phi(\boldsymbol{x}) = \frac{1}{\sqrt{m_{\text{NN}}}}g(\boldsymbol{x};\boldsymbol{\theta}_0)$. Recall that theorem A.9.1 tells us that $f_{i_t}(\boldsymbol{x}) = \langle g(\boldsymbol{x};\boldsymbol{\theta}_0), \boldsymbol{\theta}_{f,i_t} - \boldsymbol{\theta}_0\rangle = \langle\phi(\boldsymbol{x}), \boldsymbol{\theta}_{f,i_t} - \boldsymbol{\theta}_0\rangle$ for

all $\boldsymbol{x} \in \mathcal{X}_t, t \in [T]$. To begin with, for all $\boldsymbol{x}, \boldsymbol{x}' \in \mathcal{X}_t, t \in [T]$ we have that

$$
\begin{aligned}
&|f_{i_t}(\boldsymbol{x}) - f_{i_t}(\boldsymbol{x}') - \langle g(\boldsymbol{x}; \boldsymbol{\theta}_0) - g(\boldsymbol{x}'; \boldsymbol{\theta}_0), \overline{\boldsymbol{\theta}}_t - \boldsymbol{\theta}_0 \rangle| \\
&= |\langle g(\boldsymbol{x}; \boldsymbol{\theta}_0) - g(\boldsymbol{x}'; \boldsymbol{\theta}_0), \boldsymbol{\theta}_{f,i_t} - \theta_0 \rangle - \langle g(\boldsymbol{x}; \boldsymbol{\theta}_0) - g(\boldsymbol{x}'; \boldsymbol{\theta}_0), \overline{\boldsymbol{\theta}}_t - \boldsymbol{\theta}_0 \rangle| \\
&= |\langle g(\boldsymbol{x}; \boldsymbol{\theta}_0) - g(\boldsymbol{x}'; \boldsymbol{\theta}_0), \boldsymbol{\theta}_{f,i_t} - \overline{\boldsymbol{\theta}}_t \rangle\rangle| \\
&= |\langle \phi(\boldsymbol{x}) - \phi(\boldsymbol{x}'), \sqrt{m_{\mathrm{NN}}} \left(\boldsymbol{\theta}_{f,i_t} - \overline{\boldsymbol{\theta}}_t \right) \rangle| \\
&\leq \|(\phi(\boldsymbol{x}) - \phi(\boldsymbol{x}'))\|_{\boldsymbol{V}_{t-1}^{-1}} \sqrt{m_{\mathrm{NN}}} \left\| \boldsymbol{\theta}_{f,i_t} - \overline{\boldsymbol{\theta}}_t \right\|_{\boldsymbol{V}_{t-1}} \\
&\leq \|(\phi(\boldsymbol{x}) - \phi(\boldsymbol{x}'))\|_{\boldsymbol{V}_{t-1}^{-1}} \left(\beta_T + B \sqrt{\frac{\lambda}{\kappa_\mu}} + 1 \right),
\end{aligned}
\tag{A.357}
$$

in which we have used Lemma A.9.6 in the last inequality. Now making use of the equation above and theorem A.9.7, we have that

$$
\begin{aligned}
&|f_{i_t}(\boldsymbol{x}) - f_{i_t}(\boldsymbol{x}') - (h(\boldsymbol{x}; \boldsymbol{\theta}_t) - h(\boldsymbol{x}'; \boldsymbol{\theta}_t))| \\
&= |f_{i_t}(\boldsymbol{x}) - f_{i_t}(\boldsymbol{x}') - \langle g(\boldsymbol{x}; \boldsymbol{\theta}_0) - g(\boldsymbol{x}'; \boldsymbol{\theta}_0), \overline{\boldsymbol{\theta}}_t - \boldsymbol{\theta}_0 \rangle \\
&\qquad\qquad + \langle g(\boldsymbol{x}; \boldsymbol{\theta}_0) - g(\boldsymbol{x}'; \boldsymbol{\theta}_0), \overline{\boldsymbol{\theta}}_t - \boldsymbol{\theta}_0 \rangle - (h(\boldsymbol{x}; \overline{\boldsymbol{\theta}}_t) - h(\boldsymbol{x}'; \overline{\boldsymbol{\theta}}_t))| \\
&\leq |f_{i_t}(\boldsymbol{x}) - f_{i_t}(\boldsymbol{x}') - \langle g(\boldsymbol{x}; \boldsymbol{\theta}_0) - g(\boldsymbol{x}'; \boldsymbol{\theta}_0), \overline{\boldsymbol{\theta}}_t - \boldsymbol{\theta}_0 \rangle| \\
&\qquad\qquad + |\langle g(\boldsymbol{x}; \boldsymbol{\theta}_0) - g(\boldsymbol{x}'; \boldsymbol{\theta}_0), \overline{\boldsymbol{\theta}}_t - \boldsymbol{\theta}_0 \rangle - (h(\boldsymbol{x}; \overline{\boldsymbol{\theta}}_t) - h(\boldsymbol{x}'; \overline{\boldsymbol{\theta}}_t))| \\
&\leq \left\| \frac{1}{\sqrt{m_{\mathrm{NN}}}} (\phi(\boldsymbol{x}) - \phi(\boldsymbol{x}')) \right\|_{\boldsymbol{V}_{t-1}^{-1}} \left(\beta_T + B \sqrt{\frac{\lambda}{\kappa_\mu}} + 1 \right) + 2\varepsilon'_{m_{\mathrm{NN}},t}.
\end{aligned}
\tag{A.358}
$$

This completes the proof. □

We also prove the following lemma to upper bound the summation of squared norms which will be used in proving the final regret bound.

Lemma A.9.9. *With probability at least* $1-4\delta$*, we have*

$$\sum_{t=T_0}^{T} \mathbb{I}\{i_t \in C_j\} \left\|\phi(\boldsymbol{x}_{t,1}) - \phi(\boldsymbol{x}_{t,2})\right\|^2_{\boldsymbol{V}_{t-1}^{-1}} \leq 16\widetilde{d}\,, \forall j \in [m]\,,$$

where $\mathbb{I}$ denotes the indicator function.

Proof. We denote $\widetilde{\phi}_t = \phi(\boldsymbol{x}_{t,1}) - \phi(\boldsymbol{x}_{t,2})$. Note that we have defined $\phi(\boldsymbol{x}) = \frac{1}{\sqrt{m_{\text{NN}}}} g(\boldsymbol{x}; \boldsymbol{\theta}_0)$. Here we assume that $\left\|\phi(\boldsymbol{x}_{t,1}) - \phi(\boldsymbol{x}_{t,2})\right\|_2 = \frac{1}{\sqrt{m_{\text{NN}}}} \left\|g(\boldsymbol{x}_{t,1}; \boldsymbol{\theta}_0) - g(\boldsymbol{x}_{t,2}; \boldsymbol{\theta}_0)\right\|_2 \leq 2$. Replacing 2 by an absolute constant c_0 would only change the final regret bound by a constant factor, so we omit it for simplicity.

It is easy to verify that $\boldsymbol{V}_{t-1} \succeq \frac{\lambda}{\kappa_\mu} I$ and hence $\boldsymbol{V}_{t-1}^{-1} \preceq \frac{\kappa_\mu}{\lambda} I$. Therefore, we have that $\left\|\widetilde{\phi}_t\right\|^2_{\boldsymbol{V}_{t-1}^{-1}} \leq \frac{\kappa_\mu}{\lambda} \left\|\widetilde{\phi}_t\right\|_2^2 \leq \frac{4\kappa_\mu}{\lambda}$. We choose λ such that $\frac{4\kappa_\mu}{\lambda} \leq 1$, which ensures that $\left\|\widetilde{\phi}_t\right\|^2_{\boldsymbol{V}_{t-1}^{-1}} \leq 1$. Our proof here mostly follows from Lemma 11 of [43] and Lemma J.2 of [4]. To begin with, note that $x \leq 2\log(1+x)$ for $x \in [0,1]$. Denote $\boldsymbol{V}_{t,j} = \sum_{\substack{s\in[t-1]:\\ is\in C_j}} \widetilde{\phi}_s \widetilde{\phi}_s^\top + \frac{\lambda}{\kappa_\mu}\mathbf{I}$. Then we have that

$$\begin{aligned}
\sum_{t=T_0}^{T} \mathbb{I}\{i_t \in C_j\} \left\|\widetilde{\phi}_t\right\|^2_{\boldsymbol{V}_{t-1}^{-1}} &\leq \sum_{t=T_0}^{T} 2\log\left(1 + \mathbb{I}\{i_t \in C_j\} \left\|\widetilde{\phi}_t\right\|^2_{\boldsymbol{V}_{t-1}^{-1}}\right) \\
&\leq 16 \log\det\left(\frac{\kappa_\mu}{\lambda}\mathbf{H}' + \mathbf{I}\right) \\
&\triangleq 16\widetilde{d}.
\end{aligned} \tag{A.359}$$

The second inequality follows from the proof in Section A.3 from [60]. This completes the proof. □

Now we are ready to prove Theorem 9.4.2. To begin with, we have that $R_T = \sum_{t=1}^{T} r_t \leq T_0 + \sum_{t=T_0}^{T} r_t$.

Then, we only need to upper-bound the regret after T_0. By Lemma A.9.5, we know that with probability at least $1-4\delta$, the algorithm can cluster all the users correctly, $\overline{C}_t = C_{j(i_t)}$, and the statements of all the above lemmas hold. We have that for any $t \geq T_0$:

To simplify exposion here, we denote $\beta'_T \triangleq \beta_T + B\sqrt{\lambda/\kappa_\mu} + 1$.

$$
\begin{aligned}
r_t &= f_{i_t}(\boldsymbol{x}_t^*) - f_{i_t}(\boldsymbol{x}_{t,1}) + f_{i_t}(\boldsymbol{x}_t^*) - f_{i_t}(x_{t,2}) \\
&\overset{(a)}{\leq} \langle g(\boldsymbol{x}_t^*;\boldsymbol{\theta}_0) - g(\boldsymbol{x}_{t,1};\boldsymbol{\theta}_0), \overline{\boldsymbol{\theta}}_t - \boldsymbol{\theta}_0\rangle + \beta'_T \left\|\phi(\boldsymbol{x}_t^*) - \phi(\boldsymbol{x}_{t,1})\right\|_{\boldsymbol{V}_{t-1}^{-1}} + \\
&\qquad \langle g(\boldsymbol{x}_t^*;\boldsymbol{\theta}_0) - g(\boldsymbol{x}_{t,2};\boldsymbol{\theta}_0), \overline{\boldsymbol{\theta}}_t - \boldsymbol{\theta}_0\rangle + \beta'_T \left\|\phi(\boldsymbol{x}_t^*) - \phi(\boldsymbol{x}_{t,2})\right\|_{\boldsymbol{V}_{t-1}^{-1}} \\
&= \langle g(\boldsymbol{x}_t^*;\boldsymbol{\theta}_0) - g(\boldsymbol{x}_{t,1};\boldsymbol{\theta}_0), \overline{\boldsymbol{\theta}}_t - \boldsymbol{\theta}_0\rangle + \beta'_T \left\|\phi(\boldsymbol{x}_t^*) - \phi(\boldsymbol{x}_{t,1})\right\|_{\boldsymbol{V}_{t-1}^{-1}} + \\
&\qquad \langle g(\boldsymbol{x}_t^*;\boldsymbol{\theta}_0) - g(\boldsymbol{x}_{t,1};\boldsymbol{\theta}_0), \overline{\boldsymbol{\theta}}_t - \boldsymbol{\theta}_0\rangle + \langle g(\boldsymbol{x}_{t,1};\boldsymbol{\theta}_0) - g(\boldsymbol{x}_{t,2};\boldsymbol{\theta}_0), \overline{\boldsymbol{\theta}}_t - \boldsymbol{\theta}_0\rangle + \\
&\qquad \beta'_T \left\|\phi(\boldsymbol{x}_t^*) - \phi(\boldsymbol{x}_{t,1}) + \phi(\boldsymbol{x}_{t,1}) - \phi(\boldsymbol{x}_{t,2})\right\|_{\boldsymbol{V}_{t-1}^{-1}} \\
&\overset{(b)}{\leq} 2\langle g(\boldsymbol{x}_t^*;\boldsymbol{\theta}_0) - g(\boldsymbol{x}_{t,1};\boldsymbol{\theta}_0), \overline{\boldsymbol{\theta}}_t - \boldsymbol{\theta}_0\rangle + 2\beta'_T \left\|\phi(\boldsymbol{x}_t^*) - \phi(\boldsymbol{x}_{t,1})\right\|_{\boldsymbol{V}_{t-1}^{-1}} + \\
&\qquad \langle g(\boldsymbol{x}_{t,1};\boldsymbol{\theta}_0) - g(\boldsymbol{x}_{t,2};\boldsymbol{\theta}_0), \overline{\boldsymbol{\theta}}_t - \boldsymbol{\theta}_0\rangle + \beta'_T \left\|\phi(\boldsymbol{x}_{t,1}) - \phi(\boldsymbol{x}_{t,2})\right\|_{\boldsymbol{V}_{t-1}^{-1}} \\
&\overset{(c)}{\leq} 2h(\boldsymbol{x}_t^*;\overline{\boldsymbol{\theta}}_t) - 2h(\boldsymbol{x}_{t,1};\overline{\boldsymbol{\theta}}_t) + 4\varepsilon'_{m_{\mathrm{NN}},t} + 2\beta'_T \left\|\phi(\boldsymbol{x}_t^*) - \phi(\boldsymbol{x}_{t,1})\right\|_{\boldsymbol{V}_{t-1}^{-1}} + \\
&\qquad h(\boldsymbol{x}_{t,1};\overline{\boldsymbol{\theta}}_t) - h(\boldsymbol{x}_{t,2};\overline{\boldsymbol{\theta}}_t) + 2\varepsilon'_{m_{\mathrm{NN}},t} + \beta'_T \left\|\phi(\boldsymbol{x}_{t,1}) - \phi(\boldsymbol{x}_{t,2})\right\|_{\boldsymbol{V}_{t-1}^{-1}} \\
&\overset{(d)}{\leq} 2h(\boldsymbol{x}_{t,2};\overline{\boldsymbol{\theta}}_t) - 2h(\boldsymbol{x}_{t,1};\overline{\boldsymbol{\theta}}_t) + 2\beta'_T \left\|\phi(\boldsymbol{x}_{t,2}) - \phi(\boldsymbol{x}_{t,1})\right\|_{\boldsymbol{V}_{t-1}^{-1}} + \\
&\qquad h(\boldsymbol{x}_{t,1};\overline{\boldsymbol{\theta}}_t) - h(\boldsymbol{x}_{t,2};\overline{\boldsymbol{\theta}}_t) + 6\varepsilon'_{m_{\mathrm{NN}},t} + \beta'_T \left\|\phi(\boldsymbol{x}_{t,1}) - \phi(\boldsymbol{x}_{t,2})\right\|_{\boldsymbol{V}_{t-1}^{-1}} \\
&= h(\boldsymbol{x}_{t,2};\overline{\boldsymbol{\theta}}_t) - h(\boldsymbol{x}_{t,1};\overline{\boldsymbol{\theta}}_t) + 3\beta'_T \left\|\phi(\boldsymbol{x}_{t,1}) - \phi(\boldsymbol{x}_{t,2})\right\|_{\boldsymbol{V}_{t-1}^{-1}} + 6\varepsilon'_{m_{\mathrm{NN}},t} \\
&\overset{(e)}{\leq} 3\beta'_T \left\|\phi(\boldsymbol{x}_{t,1}) - \phi(\boldsymbol{x}_{t,2})\right\|_{\boldsymbol{V}_{t-1}^{-1}} + 6\varepsilon'_{m_{\mathrm{NN}},t}
\end{aligned}
\tag{A.360}
$$

Step (a) follows from Equation A.357, step (b) results from the

triangle inequality, step (c) has made use of Lemma A.9.7. Step (d) follows from the way in which we choose the second arm $\boldsymbol{x}_{t,2}$: $\boldsymbol{x}_{t,2} = \arg\max_{\boldsymbol{x}\in\mathcal{X}_t} h(\boldsymbol{x}; \overline{\boldsymbol{\theta}}_t) + \left(\beta_T + B\sqrt{\frac{\lambda}{\kappa_\mu}} + 1\right) \|(\phi(\boldsymbol{x}) - \phi(\boldsymbol{x}_{t,1}))\|_{\boldsymbol{V}_{t-1}^{-1}}$. Step (e) results from the way in which we select the first arm: $\boldsymbol{x}_{t,1} = \arg\max_{\boldsymbol{x}\in\mathcal{X}_t} h(\boldsymbol{x}; \overline{\boldsymbol{\theta}}_t)$.

Then we have

$$
\begin{aligned}
\sum_{t=T_0}^{T} r_t &\le 3\beta_T' \sum_{t=T_0}^{T} \|\phi(\boldsymbol{x}_{t,1}) - \phi(\boldsymbol{x}_{t,2})\|_{\boldsymbol{V}_{t-1}^{-1}} + 6T\varepsilon'_{m_{\mathrm{NN}},T} \\
&= 3\beta_T' \sum_{t=T_0}^{T} \sum_{j\in[m]} \mathbb{I}\{i_t \in C_j\} \|\phi(\boldsymbol{x}_{t,1}) - \phi(\boldsymbol{x}_{t,2})\|_{\boldsymbol{V}_{t-1}^{-1}} + 6T\varepsilon'_{m_{\mathrm{NN}},T} \\
&\le 3\beta_T' \sqrt{\sum_{t=T_0}^{T} \sum_{j\in[m]} \mathbb{I}\{i_t \in C_j\} \sum_{t=T_0}^{T} \sum_{j\in[m]} \mathbb{I}\{i_t \in C_j\} \|\phi(\boldsymbol{x}_{t,1}) - \phi(\boldsymbol{x}_{t,2})\|^2_{\boldsymbol{V}_{t-1}^{-1}}} \\
&\quad + 6T\varepsilon'_{m_{\mathrm{NN}},T} \\
&\le 3\beta_T' \sqrt{T \cdot m \cdot 16\widetilde{d}} + 6T\varepsilon'_{m_{\mathrm{NN}},T} \qquad \text{(A.361)} \\
&\le 12\beta_T' \sqrt{T \cdot m \cdot \widetilde{d}} + 6T\varepsilon'_{m_{\mathrm{NN}},T}\,, \qquad \text{(A.362)}
\end{aligned}
$$

where in the second inequality we use the Cauchy-Swarchz inequality, and in the last step we use $\sum_{t=T_0}^{T} \sum_{j\in[m]} \mathbb{I}\{i_t \in C_j\} \le T$ and Lemma A.9.9. It can be easily verified that as long as the conditions on m specified in eq. (A.344) are satisfied (i.e., as long as the NN is wide enough), we have that $6T\varepsilon'_{m_{\mathrm{NN}},T} \le 1$.

Recall that $\beta_T' \triangleq \beta_T + B\sqrt{\lambda/\kappa_\mu} + 1$ and $\beta_T \triangleq \frac{1}{\kappa_\mu}\sqrt{\widetilde{d} + 2\log(u/\delta)}$. Therefore, finally, we have with probability at least $1 - 4\delta$

$$
\begin{aligned}
R_T &\le T_0 + 12(\beta_T + B\sqrt{\lambda/\kappa_\mu} + 1)\sqrt{T \cdot m \cdot \widetilde{d}} + 1 \\
&\le O\left(u(\frac{\widetilde{d}}{\kappa_\mu^2 \tilde{\lambda}_x \gamma^2} + \frac{1}{\tilde{\lambda}_x^2}) \log T + \left(\frac{\sqrt{\widetilde{d}}}{\kappa_\mu} + B\sqrt{\frac{\lambda}{\kappa_\mu}}\right) \sqrt{\widetilde{d} m T}\right)
\end{aligned}
$$

$$= O\left(\left(\frac{\sqrt{\widetilde{d}}}{\kappa_\mu} + B\sqrt{\frac{\lambda}{\kappa_\mu}}\right)\sqrt{\widetilde{d}mT}\right). \tag{A.363}$$

Algorithm 19 Clustering Of Neural Dueling Bandits (CONDB)

1: **Input:** $f(T_{i,t}) \triangleq \frac{\beta_T + B\sqrt{\frac{\lambda}{\kappa_\mu}}+1}{\sqrt{2\tilde{\lambda}_x T_{i,t}}}$, regularization parameter $\lambda > 0$, confidence parameter $\beta_T \triangleq \frac{1}{\kappa_\mu}\sqrt{\widetilde{d} + 2\log(u/\delta)}$. $\phi(\boldsymbol{x}) = \frac{1}{\sqrt{m_{\text{NN}}}} g(\boldsymbol{x}; \boldsymbol{\theta}_0)$ where $\boldsymbol{\theta}_0$ denotes the NN parameters at initialization.

2: **Initialization:** $\boldsymbol{V}_0 = \boldsymbol{V}_{i,0} = \frac{\lambda}{\kappa_\mu}\mathbf{I}$, $\hat{\boldsymbol{\theta}}_{i,0} = \mathbf{0}$, $\forall i \in \mathcal{U}$, a complete Graph $G_0 = (\mathcal{U}, E_0)$ over $\mathcal{U}$.

3: **for** $t = 1, \ldots, T$ **do**

4: Receive user $i_t \in \mathcal{U}$, and feasible arm set $\mathcal{X}_t$;

5: Find the connected component $\overline{C}_t$ for user i_t in the current graph G_{t-1} as the current cluster;

6: Train the neural network using $\{(\boldsymbol{x}_{s,1}, \boldsymbol{x}_{s,2}, y_s)\}_{s\in[t-1], i_s \in \overline{C}_t}$ by minimizing the following loss function:

$$\overline{\boldsymbol{\theta}}_t = \arg\min_{\boldsymbol{\theta}} -\frac{1}{m} \sum_{\substack{s\in[t-1]\\ i_s\in\overline{C}_t}} \left(y_s \log \mu\left(h(\boldsymbol{x}_{s,1};\boldsymbol{\theta}) - h(\boldsymbol{x}_{s,2};\boldsymbol{\theta})\right) + (1-y_s)\log\mu\left(h(\boldsymbol{x}_{s,2};\boldsymbol{\theta}) - h(\boldsymbol{x}_{s,1};\boldsymbol{\theta})\right)\right)$$
$$+ \frac{\lambda}{2}\left\|\boldsymbol{\theta} - \boldsymbol{\theta}_0\right\|_2^2; \tag{A.298}$$

7: Calculate the aggregated information matrix for cluster $\overline{C}_t$: $\boldsymbol{V}_{t-1} = \boldsymbol{V}_0 + \sum_{\substack{s\in[t-1]\\ i_s\in\overline{C}_t}} (\phi(\boldsymbol{x}_{s,1}) - \phi(\boldsymbol{x}_{s,2}))(\phi(\boldsymbol{x}_{s,1}) - \phi(\boldsymbol{x}_{s,2}))^\top$.

8: Choose the first arm $\boldsymbol{x}_{t,1} = \arg\max_{\boldsymbol{x}\in\mathcal{X}_t} h(\boldsymbol{x}; \overline{\boldsymbol{\theta}}_t)$;

9: Choose the second arm $\boldsymbol{x}_{t,2} = \arg\max_{\boldsymbol{x}\in\mathcal{X}_t} h(\boldsymbol{x}; \overline{\boldsymbol{\theta}}_t) + \left(\beta_T + B\sqrt{\frac{\lambda}{\kappa_\mu}} + 1\right)\left\|(\phi(\boldsymbol{x}) - \phi(\boldsymbol{x}_{t,1}))\right\|_{\boldsymbol{V}_{t-1}^{-1}}$;

10: Observe the preference feedback: $y_t = \mathbb{1}(\boldsymbol{x}_{t,1} \succ \boldsymbol{x}_{t,2})$, and update history: $\mathcal{D}_t = \{i_s, \boldsymbol{x}_{s,1}, \boldsymbol{x}_{s,2}, y_s\}_{s=1,\ldots,t}$;

11: Train the neural network using all data for user i_t: $\{(\boldsymbol{x}_{s,1}, \boldsymbol{x}_{s,2}, y_s)\}_{s\in[t], i_s = i_t}$ by minimizing the following loss function:

$$\hat{\boldsymbol{\theta}}_{i_t,t} = \text{argmin}_{\boldsymbol{\theta}} -\frac{1}{m_{\text{NN}}} \sum_{\substack{s\in[t-1]\\ i_s = i_t}} \Big(y_s \log\mu\left(h(\boldsymbol{x}_{s,1};\boldsymbol{\theta}) - h(\boldsymbol{x}_{s,2};\boldsymbol{\theta})\right)$$
$$+ (1-y_s)\log\mu\left(h(\boldsymbol{x}_{s,2};\boldsymbol{\theta}) - h(\boldsymbol{x}_{s,1};\boldsymbol{\theta})\right)\Big) + \frac{\lambda}{2}\left\|\boldsymbol{\theta} - \boldsymbol{\theta}_0\right\|_2^2; \tag{A.299}$$

keep the estimations of other users unchanged;

12: Delete the edge $(i_t, \ell) \in E_{t-1}$ if

$$\sqrt{m_{\text{NN}}}\left\|\hat{\boldsymbol{\theta}}_{i_t,t} - \hat{\boldsymbol{\theta}}_{\ell,t}\right\|_2 > f(T_{i_t,t}) + f(T_{\ell,t}) \tag{A.300}$$

13: **end for**

Bibliography

[1] Zhiyong Wang, Dongruo Zhou, John Lui, and Wen Sun. "Model-based rl as a minimalist approach to horizon-free and second-order bounds". In: *arXiv preprint arXiv:2408.08994* (2024).

[2] Zhiyong Wang, Chen Yang, John CS Lui, and Dongruo Zhou. "Towards Zero-Shot Generalization in Offline Reinforcement Learning". In: *ICML 2024 Workshop: Aligning Reinforcement Learning Experimentalists and Theorists*.

[3] Ishita Mediratta, Qingfei You, Minqi Jiang, and Roberta Raileanu. "The Generalization Gap in Offline Reinforcement Learning". In: *arXiv preprint arXiv:2312.05742* (2023).

[4] Zhiyong Wang, Jize Xie, Xutong Liu, Shuai Li, and John Lui. "Online clustering of bandits with misspecified user models". In: *Advances in Neural Information Processing Systems* 36 (2024).

[5] Zhiyong Wang, Jize Xie, Tong Yu, Shuai Li, and John Lui. "Online corrupted user detection and regret minimization". In: *Advances in Neural Information Processing Systems* 36 (2024).

[6] Zhiyong Wang, Jiahang Sun, Mingze Kong, Jize Xie, Qinghua Hu, John Lui, and Zhongxiang Dai. "Online Clustering of Dueling Bandits". In: *arXiv preprint arXiv:2502.02079* (2025).

[7] Zhiyong Wang, Xutong Liu, Shuai Li, and John CS Lui. "Efficient explorative key-term selection strategies for conversational contextual bandits". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 37. 8. 2023, pp. 10288–10295.

[8] Xiangxiang Dai*, Zhiyong Wang*, Jize Xie, Xutong Liu, and John CS Lui. "Conversational Recommendation with Online Learning and Clustering on Misspecified Users". In: *IEEE Transactions on Knowledge and Data Engineering* (2024).

[9] Xiangxiang Dai*, Zhiyong Wang*, Jize Xie, Tong Yu, and John CS Lui. "Online Learning and Detecting Corrupted Users for Conversational Recommendation Systems". In: *IEEE Transactions on Knowledge and Data Engineering* (2024).

[10] Zhiyong Wang, Jize Xie, Yi Chen, John Lui, and Dongruo Zhou. "Variance-Dependent Regret Bounds for Non-stationary Linear Bandits". In: *arXiv preprint arXiv:2403.10732* (2024).

[11] Xiangxiang Dai*, Zhiyong Wang*, Jiancheng Ye, and John CS Lui. "Quantifying the Merits of Network-Assist Online Learning in Optimizing Network Protocols". In: *2024 IEEE/ACM 32nd International Symposium on Quality of Service (IWQoS)*. IEEE. 2024, pp. 1–10.

[12] Weibo Chu, Xiaoyan Zhang, Xinming Jia, John CS Lui, and Zhiyong Wang. "Online optimal service caching for multi-access edge computing: A constrained Multi-Armed Bandit optimization approach". In: *Computer Networks* 246 (2024), p. 110395.

[13] Xutong Liu, Siwei Wang, Jinhang Zuo, Han Zhong, Xuchuang Wang, Zhiyong Wang, Shuai Li, Mohammad Hajiesmaili, John Lui, and Wei Chen. "Combinatorial Multivariant Multi-Armed Bandits with Applications to Episodic Reinforcement Learning and Beyond". In: *arXiv preprint arXiv:2406.01386* (2024).

[14] Hantao Yang, Xutong Liu, Zhiyong Wang, Hong Xie, John CS Lui, Defu Lian, and Enhong Chen. "Federated Contextual Cascading Bandits with Asynchronous Communication and Heterogeneous Users". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 38. 18. 2024, pp. 20596–20603.

[15] Jinhang Zuo, Zhiyao Zhang, Zhiyong Wang, Shuai Li, Mohammad Hajiesmaili, and Adam Wierman. "Adversarial Attacks on Online Learning to Rank with Click Feedback". In: *arXiv preprint arXiv:2305.17071* (2023).

[16] Eric W Aboaf, Steven Mark Drucker, and Christopher G Atkeson. "Task-level robot learning: Juggling a tennis ball more accurately". In: *Proceedings, 1989 International Conference on Robotics and Automation*. IEEE. 1989, pp. 1290–1295.

[17] Marc Deisenroth, Carl Rasmussen, and Dieter Fox. "Learning to Control a Low-Cost Manipulator using Data-Efficient Reinforcement Learning". In: *Robotics: Science and Systems VII* (2011).

[18] Arun Venkatraman, Roberto Capobianco, Lerrel Pinto, Martial Hebert, Daniele Nardi, and J Andrew Bagnell. "Improved learning of dynamics models for control". In: *2016 International Symposium on Experimental Robotics*. Springer. 2017, pp. 703–713.

[19] Grady Williams, Nolan Wagener, Brian Goldfain, Paul Drews, James M Rehg, Byron Boots, and Evangelos A Theodorou. "Information theoretic mpc for model-based reinforcement learning". In: *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE. 2017, pp. 1714–1721.

[20] Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. "Deep reinforcement learning in a handful of trials using probabilistic dynamics models". In: *Advances in neural information processing systems* 31 (2018).

[21] Lukasz Kaiser, Mohammad Babaeizadeh, Piotr Milos, Blazej Osinski, Roy H Campbell, Konrad Czechowski, Dumitru Erhan, Chelsea Finn, Piotr Kozakowski, Sergey Levine, et al. "Model-based reinforcement learning for atari". In: *arXiv preprint arXiv:1903.00374* (2019).

[22] Mengjiao Yang, Yilun Du, Kamyar Ghasemipour, Jonathan Tompson, Dale Schuurmans, and Pieter Abbeel. "Learning Interactive Real-World Simulators". In: *arXiv preprint arXiv:2310.06114* (2023).

[23] Wen Sun, Nan Jiang, Akshay Krishnamurthy, Alekh Agarwal, and John Langford. "Model-based rl in contextual decision processes: Pac bounds and exponential improvements over model-free approaches". In: *Conference on learning theory*. PMLR. 2019, pp. 2898–2933.

[24] Masatoshi Uehara and Wen Sun. "Pessimistic model-based offline reinforcement learning under partial coverage". In: *arXiv preprint arXiv:2107.06226* (2021).

[25] Horia Mania, Stephen Tu, and Benjamin Recht. "Certainty equivalence is efficient for linear quadratic control". In: *Advances in Neural Information Processing Systems* 32 (2019).

[26] Qinghua Liu, Praneeth Netrapalli, Csaba Szepesvari, and Chi Jin. "Optimistic mle: A generic model-based algorithm for partially observable sequential decision making". In: *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*. 2023, pp. 363–376.

[27] Stephane Ross and J Andrew Bagnell. "Agnostic system identification for model-based reinforcement learning". In: *arXiv preprint arXiv:1203.1007* (2012).

[28] Aravind Rajeswaran, Kendall Lowrey, Emanuel Todorov, and Sham M. Kakade. "Towards Generalization and Simplicity in Continuous Control". In: *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*. 2017, pp. 6550–6561.

[29] Marlos C. Machado, Marc G. Bellemare, Erik Talvitie, Joel Veness, Matthew J. Hausknecht, and Michael H. Bowling. "Revisiting the Arcade Learning Environment: Evaluation Protocols and Open Problems for General Agents". In: *IJCAI*. 2018.

[30] Niels Justesen, Ruben Rodriguez Torrado, Philip Bontrager, Ahmed Khalifa, Julian Togelius, and Sebastian Risi. "Illuminating Generalization in Deep Reinforcement Learning through Procedural Level Generation". In: *arXiv: Learning* (2018).

[31] Charles Packer, Katelyn Gao, Jernej Kos, Philipp Krähenbühl, Vladlen Koltun, and Dawn Song. "Assessing Generalization in Deep Reinforcement Learning". In: *ICLR* (2019).

[32] Amy Zhang, Nicolas Ballas, and Joelle Pineau. "A Dissection of Overfitting and Generalization in Continuous Reinforcement Learning". In: *ArXiv* abs/1806.07937 (2018).

[33] Chiyuan Zhang, Oriol Vinyals, Rémi Munos, and Samy Bengio. "A Study on Overfitting in Deep Reinforcement Learning". In: *ArXiv* abs/1804.06893 (2018).

[34] Rui Yang, Lin Yong, Xiaoteng Ma, Hao Hu, Chongjie Zhang, and Tong Zhang. "What is essential for unseen goal generalization of offline goal-conditioned RL?" In: *International Conference on Machine Learning*. PMLR. 2023, pp. 39543–39571.

[35] Bogdan Mazoure, Ilya Kostrikov, Ofir Nachum, and Jonathan J Tompson. "Improving zero-shot generalization in offline reinforcement learning using generalized similarity functions". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 25088–25101.

[36] Avinandan Bose, Simon Shaolei Du, and Maryam Fazel. "Offline Multi-task Transfer RL with Representational Penalization". In: *arXiv preprint arXiv:2402.12570* (2024).

[37] Haque Ishfaq, Thanh Nguyen-Tang, Songtao Feng, Raman Arora, Mengdi Wang, Ming Yin, and Doina Precup. "Offline multitask representation learning for reinforcement learning". In: *arXiv preprint arXiv:2403.11574* (2024).

[38] Pushmeet Kohli, Mahyar Salek, and Greg Stoddard. "A fast bandit algorithm for recommendation to users with heterogenous tastes". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 27. 1. 2013, pp. 1135–1141.

[39] Xutong Liu, Jinhang Zuo, Hong Xie, Carlee Joe-Wong, and John CS Lui. "Variance-Adaptive Algorithm for Probabilistic Maximum Coverage Bandits with General Feedback". In: *IEEE INFOCOM 2023-IEEE Conference on Computer Communications*. IEEE. 2023, pp. 1–10.

[40] Kechao Cai, Xutong Liu, Yu-Zhen Janice Chen, and John CS Lui. "An online learning approach to network application optimization with guarantee". In: *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*. IEEE. 2018, pp. 2006–2014.

[41] Lihong Li, Wei Chu, John Langford, and Robert E Schapire. "A contextual-bandit approach to personalized news article recommendation". In: *Proceedings of the 19th international conference on World wide web*. 2010, pp. 661–670.

[42] Wei Chu, Lihong Li, Lev Reyzin, and Robert Schapire. "Contextual bandits with linear payoff functions". In: *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*. JMLR Workshop and Conference Proceedings. 2011, pp. 208–214.

[43] Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. "Improved algorithms for linear stochastic bandits". In: *Advances in neural information processing systems* 24 (2011).

[44] Xutong Liu, Jinhang Zuo, Siwei Wang, John CS Lui, Mohammad Hajiesmaili, Adam Wierman, and Wei Chen. "Contextual Combinatorial Bandits with Probabilistically Triggered Arms". In: *International Conference on Machine Learning*. PMLR. 2023, pp. 22559–22593.

[45] Fang Kong, Canzhe Zhao, and Shuai Li. "Best-of-three-worlds Analysis for Linear Bandits with Follow-the-regularized-leader Algorithm". In: *arXiv preprint arXiv:2303.06825* (2023).

[46] Claudio Gentile, Shuai Li, and Giovanni Zappella. "Online clustering of bandits". In: *International Conference on Machine Learning*. PMLR. 2014, pp. 757–765.

[47] Jens Hainmueller and Chad Hazlett. "Kernel regularized least squares: Reducing misspecification bias with a flexible and interpretable machine learning approach". In: *Political Analysis* 22.2 (2014), pp. 143–168.

[48] Avishek Ghosh, Sayak Ray Chowdhury, and Aditya Gopalan. "Misspecified linear bandits". In: *Thirty-First AAAI Conference on Artificial Intelligence*. 2017.

[49] Thodoris Lykouris, Vahab Mirrokni, and Renato Paes Leme. "Stochastic bandits robust to adversarial corruptions". In: *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*. 2018, pp. 114–122.

[50] Yuzhe Ma, Kwang-Sung Jun, Lihong Li, and Xiaojin Zhu. "Data poisoning attacks in contextual bandits". In: *International Conference on Decision and Game Theory for Security*. Springer. 2018, pp. 186–204.

[51] Jiafan He, Dongruo Zhou, Tong Zhang, and Quanquan Gu. "Nearly Optimal Algorithms for Linear Contextual Bandits with Adversarial Corruptions". In: *Advances in Neural Information Processing Systems (2022)*. 2022.

[52] Mohammad Hajiesmaili, Mohammad Sadegh Talebi, John Lui, Wing Shing Wong, et al. "Adversarial bandits with corruptions: Regret lower bound and no-regret algorithm". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 19943–19952.

[53] Anupam Gupta, Tomer Koren, and Kunal Talwar. "Better algorithms for stochastic bandits with adversarial corruptions". In: *Conference on Learning Theory*. PMLR. 2019, pp. 1562–1578.

[54] Yingkai Li, Edmund Y Lou, and Liren Shan. "Stochastic linear optimization with adversarial corruption". In: *arXiv preprint arXiv:1909.02109* (2019).

[55] Daixin Wang, Jianbin Lin, Peng Cui, Quanhui Jia, Zhen Wang, Yanming Fang, Quan Yu, Jun Zhou, Shuang Yang, and Yuan Qi. "A semi-supervised graph attentive network for financial fraud detection". In: *2019 IEEE International Conference on Data Mining (ICDM)*. IEEE. 2019, pp. 598–607.

[56] Yingtong Dou, Zhiwei Liu, Li Sun, Yutong Deng, Hao Peng, and Philip S Yu. "Enhancing graph neural network-based fraud detectors against camouflaged fraudsters". In: *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 2020, pp. 315–324.

[57] Ge Zhang, Jia Wu, Jian Yang, Amin Beheshti, Shan Xue, Chuan Zhou, and Quan Z Sheng. "FRAUDRE: fraud detection dual-resistant to graph inconsistency and imbalance". In: *2021 IEEE International Conference on Data Mining (ICDM)*. IEEE. 2021, pp. 867–876.

[58] Yisong Yue, Josef Broder, Robert Kleinberg, and Thorsten Joachims. "The k-armed dueling bandits problem". In: *Journal of Computer and System Sciences* (2012), pp. 1538–1556.

[59] Xiaoqiang Lin, Zhongxiang Dai, Arun Verma, See-Kiong Ng, Patrick Jaillet, and Bryan Kian Hsiang Low. "Prompt Optimization with Human Feedback". In: *arXiv preprint arXiv:2405.17346* (2024).

[60] Arun Verma, Zhongxiang Dai, Xiaoqiang Lin, Patrick Jaillet, and Bryan Kian Hsiang Low. "Neural dueling bandits". In: *arXiv preprint arXiv:2407.17112* (2024).

[61] Konstantina Christakopoulou, Filip Radlinski, and Katja Hofmann. "Towards conversational recommender systems". In: *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 2016, pp. 815–824.

[62] Xiaoying Zhang, Hong Xie, Hang Li, and John CS Lui. "Conversational contextual bandit: Algorithm and application". In: *Proceedings of The Web Conference 2020*. 2020, pp. 662–672.

[63] Dylan J Foster, Sham M Kakade, Jian Qian, and Alexander Rakhlin. "The statistical complexity of interactive decision making". In: *arXiv preprint arXiv:2112.13487* (2021).

[64] Yuda Song and Wen Sun. "Pc-mlp: Model-based reinforcement learning with policy cover guided exploration". In: *International Conference on Machine Learning*. PMLR. 2021, pp. 9801–9811.

[65] Wenhao Zhan, Masatoshi Uehara, Wen Sun, and Jason D Lee. "Pac reinforcement learning for predictive state representations". In: *arXiv preprint arXiv:2207.05738* (2022).

[66] Qinghua Liu, Alan Chung, Csaba Szepesvári, and Chi Jin. "When is partially observable reinforcement learning not scary?" In: *Conference on Learning Theory*. PMLR. 2022, pp. 5175–5220.

[67] Han Zhong, Wei Xiong, Sirui Zheng, Liwei Wang, Zhaoran Wang, Zhuoran Yang, and Tong Zhang. "Gec: A unified framework for interactive decision making in mdp, pomdp, and beyond". In: *arXiv preprint arXiv:2211.01962* (2022).

[68] Alekh Agarwal, Sham Kakade, Akshay Krishnamurthy, and Wen Sun. "Flambe: Structural complexity and representation learning of low rank mdps". In: *Advances in neural information processing systems* 33 (2020), pp. 20095–20107.

[69] Masatoshi Uehara, Xuezhou Zhang, and Wen Sun. "Representation learning for online and offline rl in low-rank mdps". In: *arXiv preprint arXiv:2110.04652* (2021).

[70] Ruosong Wang, Simon S Du, Lin F Yang, and Sham M Kakade. "Is long horizon reinforcement learning more difficult than short horizon reinforcement learning?" In: *arXiv preprint arXiv:2005.00527* (2020).

[71] Zihan Zhang, Xiangyang Ji, and Simon Du. "Is reinforcement learning more difficult than bandits? a near-optimal algorithm escaping the curse of horizon". In: *Conference on Learning Theory*. PMLR. 2021, pp. 4528–4531.

[72] Tongzheng Ren, Jialian Li, Bo Dai, Simon S Du, and Sujay Sanghavi. "Nearly horizon-free offline reinforcement learning". In: *Advances in neural information processing systems* 34 (2021), pp. 15621–15634.

[73] Jean Tarbouriech, Runlong Zhou, Simon S Du, Matteo Pirotta, Michal Valko, and Alessandro Lazaric. "Stochastic shortest path: Minimax, parameter-free and towards horizon-free regret". In: *Advances in neural information processing systems* 34 (2021), pp. 6843–6855.

[74] Yeoneung Kim, Insoon Yang, and Kwang-Sung Jun. "Improved regret analysis for variance-adaptive linear bandits and horizon-free linear mixture mdps". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 1060–1072.

[75] Zihan Zhang, Jiaqi Yang, Xiangyang Ji, and Simon S Du. "Improved variance-aware confidence sets for linear bandits and linear mixture mdp". In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 4342–4355.

[76] Dongruo Zhou and Quanquan Gu. "Computationally efficient horizon-free reinforcement learning for linear mixture mdps". In: *Advances in neural information processing systems* 35 (2022), pp. 36337–36349.

[77] Qiwei Di, Jiafan He, Dongruo Zhou, and Quanquan Gu. "Nearly minimax optimal regret for learning linear mixture stochastic shortest path". In: *International Conference on Machine Learning*. PMLR. 2023, pp. 7837–7864.

[78] Zihan Zhang, Jason D Lee, Yuxin Chen, and Simon S Du. "Horizon-Free Regret for Linear Markov Decision Processes". In: *arXiv preprint arXiv:2403.10738* (2024).

[79] Junkai Zhang, Weitong Zhang, and Quanquan Gu. "Optimal horizon-free reward-free exploration for linear mixture mdps". In: *International Conference on Machine Learning*. PMLR. 2023, pp. 41902–41930.

[80] Heyang Zhao, Jiafan He, Dongruo Zhou, Tong Zhang, and Quanquan Gu. "Variance-dependent regret bounds for linear bandits and reinforcement learning: Adaptivity and computational efficiency". In: *The Thirty Sixth Annual Conference on Learning Theory*. PMLR. 2023, pp. 4977–5020.

[81] Yuanzhi Li, Ruosong Wang, and Lin F Yang. "Settling the horizon-dependence of sample complexity in reinforcement learning". In: *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE. 2022, pp. 965–976.

[82] Zihan Zhang, Xiangyang Ji, and Simon Du. "Horizon-free reinforcement learning in polynomial time: the power of stationary policies". In: *Conference on Learning Theory*. PMLR. 2022, pp. 3858–3904.

[83] Jiayi Huang, Han Zhong, Liwei Wang, and Lin Yang. "Horizon-free and instance-dependent regret bounds for reinforcement learning with general function approximation". In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2024, pp. 3673–3681.

[84] Dylan J Foster, Adam Block, and Dipendra Misra. "Is Behavior Cloning All You Need? Understanding Horizon in Imitation Learning". In: *arXiv preprint arXiv:2407.15007* (2024).

[85] Damien Ernst, Pierre Geurts, and Louis Wehenkel. "Tree-based batch mode reinforcement learning". In: *Journal of Machine Learning Research* 6 (2005).

[86] Martin Riedmiller. "Neural fitted Q iteration–first experiences with a data efficient neural reinforcement learning method". In: *Machine Learning: ECML 2005: 16th European Conference on Machine Learning, Porto, Portugal, October 3-7, 2005. Proceedings 16*. Springer. 2005, pp. 317–328.

[87] Sascha Lange, Thomas Gabel, and Martin Riedmiller. "Batch reinforcement learning". In: *Reinforcement learning: State-of-the-art*. Springer, 2012, pp. 45–73.

[88] Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. "Offline reinforcement learning: Tutorial, review, and perspectives on open problems". In: *arXiv preprint arXiv:2005.01643* (2020).

[89] Yao Liu, Adith Swaminathan, Alekh Agarwal, and Emma Brunskill. "Provably good batch off-policy reinforcement learning without great exploration". In: *Advances in neural information processing systems* 33 (2020), pp. 1264–1274.

[90] Paria Rashidinejad, Banghua Zhu, Cong Ma, Jiantao Jiao, and Stuart Russell. "Bridging offline reinforcement learning and imitation learning: A tale of pessimism". In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 11702–11716.

[91] Ying Jin, Zhuoran Yang, and Zhaoran Wang. "Is pessimism provably efficient for offline rl?" In: *International Conference on Machine Learning*. PMLR. 2021, pp. 5084–5096.

[92] Tianhe Yu, Garrett Thomas, Lantao Yu, Stefano Ermon, James Y Zou, Sergey Levine, Chelsea Finn, and Tengyu Ma. "Mopo: Model-based offline policy optimization". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 14129–14142.

[93] Tengyang Xie, Nan Jiang, Huan Wang, Caiming Xiong, and Yu Bai. "Policy finetuning: Bridging sample-efficient offline and online reinforcement learning". In: *Advances in neural information processing systems* 34 (2021), pp. 27395–27407.

[94] Ming Yin, Yaqi Duan, Mengdi Wang, and Yu-Xiang Wang. "Near-optimal offline reinforcement learning with linear representation: Leveraging variance information with pessimism". In: *arXiv preprint arXiv:2203.05804* (2022).

[95] Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. "Conservative q-learning for offline reinforcement learning". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 1179–1191.

[96] Yue Wu, Shuangfei Zhai, Nitish Srivastava, Joshua Susskind, Jian Zhang, Ruslan Salakhutdinov, and Hanlin Goh. "Uncertainty weighted actor-critic for offline reinforcement learning". In: *arXiv preprint arXiv:2105.08140* (2021).

[97] Chenjia Bai, Lingxiao Wang, Zhuoran Yang, Zhihong Deng, Animesh Garg, Peng Liu, and Zhaoran Wang. "Pessimistic bootstrapping for uncertainty-driven offline reinforcement learning". In: *arXiv preprint arXiv:2202.11566* (2022).

[98] Kamyar Ghasemipour, Shixiang Shane Gu, and Ofir Nachum. "Why so pessimistic? estimating uncertainties for offline rl through ensembles, and why their independence matters". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 18267–18281.

[99] Yuling Yan, Gen Li, Yuxin Chen, and Jianqing Fan. "The efficacy of pessimism in asynchronous Q-learning". In: *IEEE Transactions on Information Theory* (2023).

[100] Shideh Rezaeifar, Robert Dadashi, Nino Vieillard, Léonard Hussenot, Olivier Bachem, Olivier Pietquin, and Matthieu Geist. "Offline reinforcement learning as anti-exploration". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 36. 7. 2022, pp. 8106–8114.

[101] Tengyang Xie, Ching-An Cheng, Nan Jiang, Paul Mineiro, and Alekh Agarwal. "Bellman-consistent pessimism for offline reinforcement learning". In: *Advances in neural information processing systems* 34 (2021), pp. 6683–6694.

[102] Andrea Zanette, Martin J Wainwright, and Emma Brunskill. "Provable benefits of actor-critic methods for offline reinforcement learning". In: *Advances in neural information processing systems* 34 (2021), pp. 13626–13640.

[103] Thanh Nguyen-Tang and Raman Arora. "On Sample-Efficient Offline Reinforcement Learning: Data Diversity, Posterior Sampling and Beyond". In: *Advances in Neural Information Processing Systems* 36 (2024).

[104] Denis Yarats, David Brandfonbrener, Hao Liu, Michael Laskin, Pieter Abbeel, Alessandro Lazaric, and Lerrel Pinto. "Don't change the algorithm, change the data: Exploratory data for offline reinforcement learning". In: *arXiv preprint arXiv:2201.13425* (2022).

[105] Alex Nichol, V. Pfau, Christopher Hesse, O. Klimov, and John Schulman. "Gotta Learn Fast: A New Benchmark for Generalization in RL". In: *ArXiv* abs/1804.03720 (2018).

[106] Karl Cobbe, Oleg Klimov, Christopher Hesse, Taehoon Kim, and J. Schulman. "Quantifying Generalization in Reinforcement Learning". In: *International Conference On Machine Learning* (2018).

[107] Heinrich Küttler, Nantas Nardelli, Alexander H. Miller, Roberta Raileanu, Marco Selvatici, Edward Grefenstette, and Tim Rocktäschel. "The NetHack Learning Environment". In: *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*. 2020.

[108] Emmanuel Bengio, Joelle Pineau, and Doina Precup. "Interference and Generalization in Temporal Difference Learning". In: *International Conference On Machine Learning* (2020).

[109] Martin Bertran, Natalia Martinez, Mariano Phielipp, and Guillermo Sapiro. "Instance-based generalization in reinforcement learning". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 11333–11344.

[110] Dibya Ghosh, Jad Rahme, Aviral Kumar, Amy Zhang, Ryan P Adams, and Sergey Levine. "Why generalization in rl is difficult: Epistemic pomdps and implicit partial observability". In: *Advances in neural information processing systems* 34 (2021), pp. 25502–25515.

[111] Robert Kirk, Amy Zhang, Edward Grefenstette, and Tim Rocktäschel. "A survey of zero-shot generalisation in deep reinforcement learning". In: *Journal of Artificial Intelligence Research* 76 (2023), pp. 201–264.

[112] Arthur Juliani, Ahmed Khalifa, Vincent-Pierre Berges, Jonathan Harper, Ervin Teng, Hunter Henry, Adam Crespi, Julian Togelius, and Danny Lange. "Obstacle Tower: A Generalization Challenge in Vision, Control, and Planning". In: *IJCAI*. 2019.

[113] Anurag Ajay, Ge Yang, Ofir Nachum, and Pulkit Agrawal. "Understanding the Generalization Gap in Visual Reinforcement Learning". In: (2021).

[114] Mikayel Samvelyan, Robert Kirk, Vitaly Kurin, Jack Parker-Holder, Minqi Jiang, Eric Hambro, Fabio Petroni, Heinrich Küttler, Edward Grefenstette, and Tim Rocktäschel. "Minihack the planet: A sandbox for open-ended reinforcement learning research". In: *arXiv preprint arXiv:2109.13202* (2021).

[115] Kevin Frans and Phillip Isola. "Powderworld: A Platform for Understanding Generalization via Rich Task Distributions". In: *arXiv preprint arXiv:2211.13051* (2022).

[116] Joshua Albrecht, Abraham Fetterman, Bryden Fogelman, Ellie Kitanidis, Bartosz Wróblewski, Nicole Seo, Michael Rosenthal, Maksis Knutins, Zack Polizzi, James Simon, et al. "Avalon: A Benchmark for RL Generalization Using Procedurally Generated Worlds". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 12813–12825.

[117] Andy Ehrenberg, Robert Kirk, Minqi Jiang, Edward Grefenstette, and Tim Rocktäschel. "A Study of Off-Policy Learning in Environments with Procedural Content Generation". In: *ICLR Workshop on Agent Learning in Open-Endedness*. 2022.

[118] Xingyou Song, Yiding Jiang, Stephen Tu, Yilun Du, and Behnam Neyshabur. "Observational Overfitting in Reinforcement Learning". In: *International Conference on Learning Representations*. 2020.

[119] Clare Lyle, Mark Rowland, Will Dabney, Marta Kwiatkowska, and Yarin Gal. "Learning dynamics and generalization in deep reinforcement learning". In: *International Conference on Machine Learning*. PMLR. 2022, pp. 14560–14581.

[120] Chang Ye, Ahmed Khalifa, Philip Bontrager, and Julian Togelius. "Rotation, translation, and cropping for zero-shot generalization". In: *2020 IEEE Conference on Games (CoG)*. IEEE. 2020, pp. 57–64.

[121] Kimin Lee, Kibok Lee, Jinwoo Shin, and Honglak Lee. "Network randomization: A simple technique for generalization in deep reinforcement learning". In: *International Conference on Learning Representations. https://openreview. net/forum*. 2020.

[122] Yiding Jiang, J Zico Kolter, and Roberta Raileanu. "Uncertainty-Driven Exploration for Generalization in Reinforcement Learning". In: *Deep Reinforcement Learning Workshop NeurIPS 2022*.

[123] Ahmed Touati, Jérémy Rapin, and Yann Ollivier. "Does zero-shot reinforcement learning exist?" In: *ICLR*. 2023.

[124] Huan Wang, Stephan Zheng, Caiming Xiong, and Richard Socher. "On the generalization gap in reparameterizable reinforcement learning". In: *International Conference on Machine Learning*. PMLR. 2019, pp. 6648–6658.

[125] Dhruv Malik, Yuanzhi Li, and Pradeep Ravikumar. "When Is Generalizable Reinforcement Learning Tractable?" In: *Advances in Neural Information Processing Systems* 34 (2021).

[126] Haotian Ye, Xiaoyu Chen, Liwei Wang, and Simon Shaolei Du. "On the power of pre-training for generalization in RL: provable benefits and hardness". In: *International Conference on Machine Learning*. PMLR. 2023, pp. 39770–39800.

[127] Emma Brunskill and Lihong Li. "Sample complexity of multi-task reinforcement learning". In: *arXiv preprint arXiv:1309.6821* (2013).

[128] Andrea Tirinzoni, Riccardo Poiani, and Marcello Restelli. "Sequential transfer in reinforcement learning with a generative model". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 9481–9492.

[129] Jiachen Hu, Xiaoyu Chen, Chi Jin, Lihong Li, and Liwei Wang. "Near-optimal representation learning for linear bandits and linear rl". In: *International Conference on Machine Learning*. PMLR. 2021, pp. 4349–4358.

[130] Chicheng Zhang and Zhi Wang. "Provably efficient multi-task reinforcement learning with model transfer". In: *Advances in Neural Information Processing Systems* 34 (2021).

[131] Rui Lu, Gao Huang, and Simon S Du. "On the power of multitask representation learning in linear mdp". In: *arXiv preprint arXiv:2106.08053* (2021).

[132] Weitong Zhang, Jiafan He, Dongruo Zhou, Amy Zhang, and Quanquan Gu. "Provably efficient representation selection in low-rank Markov decision processes: from online to offline RL". In: *Proceedings of the Thirty-Ninth Conference on Uncertainty in Artificial Intelligence*. 2023, pp. 2488–2497.

[133] Rui Lu, Yang Yue, Andrew Zhao, Simon Du, and Gao Huang. "Towards Understanding the Benefit of Multitask Representation Learning in Decision Process". In: *arXiv preprint arXiv:2503.00345* (2025).

[134] Shuai Li, Alexandros Karatzoglou, and Claudio Gentile. "Collaborative filtering bandits". In: *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. 2016, pp. 539–548.

[135] Shuai Li and Shengyu Zhang. "Online clustering of contextual cascading bandits". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 32. 1. 2018.

[136] Shuai Li, Wei Chen, Shuai Li, and Kwong-Sak Leung. "Improved Algorithm on Online Clustering of Bandits". In: *Proceedings of the 28th International Joint Conference on Artificial Intelligence*. IJCAI'19. Macao, China: AAAI Press, 2019, pp. 2923–2929.

[137] Xutong Liu, Haoru Zhao, Tong Yu, Shuai Li, and John Lui. "Federated Online Clustering of Bandits". In: *The 38th Conference on Uncertainty in Artificial Intelligence*. 2022.

[138] Tor Lattimore, Csaba Szepesvari, and Gellert Weisz. "Learning with good feature representations in bandits and in rl with a generative model". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 5662–5670.

[139] Tor Lattimore and Csaba Szepesvári. *Bandit algorithms*. Cambridge University Press, 2020.

[140] Aldo Pacchiano, My Phan, Yasin Abbasi Yadkori, Anup Rao, Julian Zimmert, Tor Lattimore, and Csaba Szepesvari. "Model selection in contextual stochastic bandit problems". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 10328–10337.

[141] Dylan J Foster, Claudio Gentile, Mehryar Mohri, and Julian Zimmert. "Adapting to misspecification in contextual bandits". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 11478–11489.

[142] Qin Ding, Cho-Jui Hsieh, and James Sharpnack. "Robust stochastic linear contextual bandits under adversarial attacks". In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2022, pp. 7111–7123.

[143] Yisong Yue and Thorsten Joachims. "Interactively optimizing information retrieval systems as a dueling bandits problem". In: *Proc. ICML*. 2009, pp. 1201–1208.

[144] Yisong Yue and Thorsten Joachims. "Beat the mean bandit". In: *Proc. ICML*. 2011, pp. 241–248.

[145] Masrour Zoghi, Shimon A Whiteson, Maarten De Rijke, and Remi Munos. "Relative confidence sampling for efficient on-line ranker evaluation". In: *Proc. WSDM*. 2014, pp. 73–82.

[146] Nir Ailon, Zohar Karnin, and Thorsten Joachims. "Reducing dueling bandits to cardinal bandits". In: *Proc. ICML*. 2014, pp. 856–864.

[147] Masrour Zoghi, Shimon Whiteson, Remi Munos, and Maarten Rijke. "Relative upper confidence bound for the k-armed dueling bandit problem". In: *Proc. ICML*. 2014, pp. 10–18.

[148] Junpei Komiyama, Junya Honda, Hisashi Kashima, and Hiroshi Nakagawa. "Regret lower bound and optimal algorithm in dueling bandit problem". In: *Proc. COLT*. 2015, pp. 1141–1154.

[149] Pratik Gajane, Tanguy Urvoy, and Fabrice Clérot. "A relative exponential weighing algorithm for adversarial utility-based dueling bandits". In: *Proc. ICML*. 2015, pp. 218–227.

[150] Aadirupa Saha and Aditya Gopalan. "Battle of Bandits". In: *Proc. UAI*. 2018, pp. 805–814.

[151] Aadirupa Saha and Aditya Gopalan. "Active ranking with subset-wise preferences". In: *Proc. AISTATS*. 2019, pp. 3312–3321.

[152] Aadirupa Saha and Aditya Gopalan. "PAC battling bandits in the plackett-luce model". In: *Proc. ALT*. 2019, pp. 700–737.

[153] Aadirupa Saha and Suprovat Ghoshal. "Exploiting correlation to achieve faster learning rates in low-rank preference bandits". In: *Proc. AISTATS*. 2022, pp. 456–482.

[154] Banghua Zhu, Michael Jordan, and Jiantao Jiao. "Principled reinforcement learning with human feedback from pairwise or k-wise comparisons". In: *Proc. ICML*. 2023, pp. 43037–43067.

[155] Aadirupa Saha. "Optimal Algorithms for Stochastic Contextual Preference Bandits". In: *Proc. NeurIPS*. 2021, pp. 30050–30062.

[156] Aadirupa Saha and Akshay Krishnamurthy. "Efficient and optimal algorithms for contextual dueling bandits under realizability". In: *Proc. ALT*. 2022, pp. 968–994.

[157] Viktor Bengs, Aadirupa Saha, and Eyke Hüllermeier. "Stochastic Contextual Dueling Bandits under Linear Stochastic Transitivity Models". In: *Proc. ICML*. 2022, pp. 1764–1786.

[158] Qiwei Di, Tao Jin, Yue Wu, Heyang Zhao, Farzad Farnoud, and Quanquan Gu. "Variance-Aware Regret Bounds for Stochastic Contextual Dueling Bandits". In: *arXiv:2310.00968* (2023).

[159] Xuheng Li, Heyang Zhao, and Quanquan Gu. "Feel-Good Thompson Sampling for Contextual Dueling Bandits". In: *arXiv:2404.06013* (2024).

[160] Dongruo Zhou, Lihong Li, and Quanquan Gu. "Neural contextual bandits with UCB-based exploration". In: *Proc. ICML*. 2020, pp. 11492–11502.

[161] Weitong Zhang, Dongruo Zhou, Lihong Li, and Quanquan Gu. "Neural Thompson sampling". In: *Proc. ICLR*. 2021.

[162] Pan Xu, Zheng Wen, Handong Zhao, and Quanquan Gu. *Neural contextual bandits with deep representation and shallow exploration*. arXiv:2012.01780. 2020.

[163] Parnian Kassraie and Andreas Krause. "Neural Contextual Bandits without Regret". In: *Proc. AISTATS*. 2022, pp. 240–278.

[164] Quanquan Gu, Amin Karbasi, Khashayar Khosravi, Vahab Mirrokni, and Dongruo Zhou. *Batched Neural Bandits*. arXiv:2102.13028. 2021.

[165] Ofir Nabati, Tom Zahavy, and Shie Mannor. "Online Limited Memory Neural-Linear Bandits with Likelihood Matching". In: *Proc. ICML*. 2021.

[166] Michal Lisicki, Arash Afkanpour, and Graham W Taylor. "An Empirical Study of Neural Kernel Bandits". In: *NeurIPS Workshop on Bayesian Deep Learning*. 2021.

[167] Yikun Ban, Yuchen Yan, Arindam Banerjee, and Jingrui He. "EE-Net: Exploitation-Exploration Neural Networks in Contextual Bandits". In: *Proc. ICLR*. 2022.

[168] Yikun Ban and Jingrui He. *Convolutional neural bandit: Provable algorithm for visual-aware advertising*. arXiv:2107.07438. 2021.

[169] Yiling Jia, Weitong Zhang, Dongruo Zhou, Quanquan Gu, and Hongning Wang. "Learning Neural Contextual Bandits through Perturbed Rewards". In: *Proc. ICLR*. 2021.

[170] Thanh Nguyen-Tang, Sunil Gupta, A Tuan Nguyen, and Svetha Venkatesh. "Offline Neural Contextual Bandits: Pessimism, Optimization and Generalization". In: *Proc. ICLR*. 2022.

[171] Yinglun Zhu, Dongruo Zhou, Ruoxi Jiang, Quanquan Gu, Rebecca Willett, and Robert Nowak. "Pure exploration in kernel and neural bandits". In: *Proc. NeurIPS*. Vol. 34. 2021, pp. 11618–11630.

[172] Parnian Kassraie, Andreas Krause, and Ilija Bogunovic. "Graph Neural Network Bandits". In: *Proc. NeurIPS*. 2022.

[173] Sudeep Salgia, Sattar Vakili, and Qing Zhao. *Provably and Practically Efficient Neural Contextual Bandits*. arXiv:2206.00099. 2022.

[174] Zhongxiang Dai, Yao Shu, Bryan Kian Hsiang Low, and Patrick Jaillet. "Sample-then-optimize batch neural Thompson sampling". In: *Proc. NeurIPS*. 2022.

[175] Taehyun Hwang, Kyuwook Chai, and Min-hwan Oh. "Combinatorial neural bandits". In: *International Conference on Machine Learning*. PMLR. 2023, pp. 14203–14236.

[176] Yunzhe Qi, Yikun Ban, and Jingrui He. "Graph neural bandits". In: *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2023, pp. 1920–1931.

[177] Yunzhe Qi, Yikun Ban, Tianxin Wei, Jiaru Zou, Huaxiu Yao, and Jingrui He. "Meta-learning with neural bandit scheduler". In: *Advances in Neural Information Processing Systems* 36 (2024).

[178] Yikun Ban, Yunzhe Qi, Tianxin Wei, Lihui Liu, and Jingrui He. "Meta clustering of neural bandits". In: *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2024, pp. 95–106.

[179] Qingyun Wu, Huazheng Wang, Quanquan Gu, and Hongning Wang. "Contextual bandits in a collaborative environment". In: *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. 2016, pp. 529–538.

[180] Junda Wu, Canzhe Zhao, Tong Yu, Jingyang Li, and Shuai Li. "Clustering of Conversational Bandits for User Preference Learning and Elicitation". In: *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 2021, pp. 2129–2139.

[181] Zhihui Xie, Tong Yu, Canzhe Zhao, and Shuai Li. "Comparison-based Conversational Recommender System with Relative Bandit Feedback". In: *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2021, pp. 1400–1409.

[182] Canzhe Zhao, Tong Yu, Zhihui Xie, and Shuai Li. "Knowledge-aware Conversational Preference Elicitation with Bandit Feedback". In: *Proceedings of the ACM Web Conference 2022*. 2022, pp. 483–492.

[183] Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E Schapire. "The nonstochastic multiarmed bandit problem". In: *SIAM journal on computing* 32.1 (2002), pp. 48–77.

[184] Aurélien Garivier and Eric Moulines. "On Upper-Confidence Bound Policies for Switching Bandit Problems". In: *Algorithmic Learning Theory*. Ed. by Jyrki Kivinen, Csaba Szepesvári, Esko Ukkonen, and Thomas Zeugmann. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 174–188.

[185] Omar Besbes, Yonatan Gur, and Assaf Zeevi. "Stochastic multi-armed-bandit problem with non-stationary rewards". In: *Advances in neural information processing systems* 27 (2014).

[186] Chen-Yu Wei, Yi-Te Hong, and Chi-Jen Lu. "Tracking the best expert in non-stationary stochastic environments". In: *Advances in neural information processing systems* 29 (2016).

[187] Wang Chi Cheung, David Simchi-Levi, and Ruihao Zhu. "Learning to optimize under non-stationarity". In: *The 22nd International Conference on Artificial Intelligence and Statistics*. PMLR. 2019, pp. 1079–1087.

[188] Yoan Russac, Claire Vernade, and Olivier Cappé. "Weighted linear bandits for non-stationary environments". In: *Advances in Neural Information Processing Systems* (2019).

[189] Peter Auer, Pratik Gajane, and Ronald Ortner. "Adaptively Tracking the Best Bandit Arm with an Unknown Number of Distribution Changes". In: *Proceedings of the Thirty-Second Conference on Learning Theory*. Ed. by Alina Beygelzimer and Daniel Hsu. Vol. 99. Proceedings of Machine Learning Research. PMLR, 25–28 Jun 2019, pp. 138–158.

[190] Yifang Chen, Chung-Wei Lee, Haipeng Luo, and Chen-Yu Wei. "A New Algorithm for Non-stationary Contextual Bandits: Efficient, Optimal and Parameter-free". In: *Proceedings of the Thirty-Second Conference on Learning Theory*. Ed. by Alina Beygelzimer and Daniel Hsu. Vol. 99. Proceedings of Machine Learning Research. PMLR, 25–28 Jun 2019, pp. 696–726.

[191] Yoan Russac, Olivier Cappé, and Aurélien Garivier. "Algorithms for non-stationary generalized linear bandits". In: *arXiv preprint arXiv:2003.10113* (2020).

[192] Peng Zhao, Lijun Zhang, Yuan Jiang, and Zhi-Hua Zhou. "A simple approach for non-stationary linear bandits". In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2020, pp. 746–755.

[193] Baekjin Kim and Ambuj Tewari. "Randomized Exploration for Non-Stationary Stochastic Linear Bandits". In: *Conference on Uncertainty in Artificial Intelligence*. PMLR. 2020, pp. 71–80.

[194] Chen-Yu Wei and Haipeng Luo. "Non-stationary reinforcement learning without prior knowledge: An optimal black-box approach". In: *Conference on learning theory*. PMLR. 2021, pp. 4300–4354.

[195] Yoan Russac, Louis Faury, Olivier Cappé, and Aurélien Garivier. "Self-concordant analysis of generalized linear bandits with forgetting". In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2021, pp. 658–666.

[196] Wei Chen, Liwei Wang, Haoyu Zhao, and Kai Zheng. "Combinatorial semi-bandit in the non-stationary environment". In: *Uncertainty in Artificial Intelligence*. PMLR. 2021, pp. 865–875.

[197] Yuntian Deng, Xingyu Zhou, Baekjin Kim, Ambuj Tewari, Abhishek Gupta, and Ness Shroff. "Weighted gaussian process bandits for non-stationary environments". In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2022, pp. 6909–6932.

[198] Joe Suk and Samory Kpotufe. "Tracking most significant arm switches in bandits". In: *Conference on Learning Theory*. PMLR. 2022, pp. 2160–2182.

[199] Yueyang Liu, Benjamin Van Roy, and Kuang Xu. "A Definition of Non-Stationary Bandits". In: *arXiv preprint arXiv:2302.12202* (2023).

[200] Yasin Abbasi-Yadkori, András György, and Nevena Lazić. "A new look at dynamic regret for non-stationary stochastic bandits". In: *Journal of Machine Learning Research* 24.288 (2023), pp. 1–37.

[201] Giulia Clerici, Pierre Laforgue, and Nicolò Cesa-Bianchi. *Linear Bandits with Memory: from Rotting to Rising*. 2023. arXiv: 2302.08345 [cs.LG].

[202] Johannes Kirschner and Andreas Krause. "Information directed sampling and bandits with heteroscedastic noise". In: *Conference On Learning Theory*. PMLR. 2018, pp. 358–384.

[203] Dongruo Zhou, Quanquan Gu, and Csaba Szepesvari. "Nearly minimax optimal reinforcement learning for linear mixture markov decision processes". In: *Conference on Learning Theory*. PMLR. 2021, pp. 4532–4576.

[204] Yan Dai, Ruosong Wang, and Simon S Du. "Variance-aware sparse linear bandits". In: *arXiv preprint arXiv:2205.13450* (2022).

[205] Kaiwen Wang, Owen Oertell, Alekh Agarwal, Nathan Kallus, and Wen Sun. "More Benefits of Being Distributional: Second-Order Bounds for Reinforcement Learning". In: *arXiv preprint arXiv:2402.07198* (2024).

[206] Chenlu Ye, Rui Yang, Quanquan Gu, and Tong Zhang. "Corruption-robust offline reinforcement learning with general function approximation". In: *Advances in Neural Information Processing Systems* 36 (2024).

[207] Dylan J Foster and Akshay Krishnamurthy. "Efficient first-order contextual bandits: Prediction, allocation, and triangular discrimination". In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 18907–18919.

[208] Kaiwen Wang, Kevin Zhou, Runzhe Wu, Nathan Kallus, and Wen Sun. "The benefits of being distributional: Small-loss bounds for reinforcement learning". In: *Advances in Neural Information Processing Systems* 36 (2023).

[209] Daniel Russo and Benjamin Van Roy. "Eluder dimension and the sample complexity of optimistic exploration". In: *Advances in Neural Information Processing Systems* 26 (2013).

[210] Sara A Geer. *Empirical Processes in M-estimation*. Vol. 6. Cambridge university press, 2000.

[211] Lerrel Pinto, James Davidson, Rahul Sukthankar, and Abhinav Gupta. "Robust adversarial reinforcement learning". In: *International conference on machine learning*. PMLR. 2017, pp. 2817–2826.

[212] Jacob Beck, Risto Vuorio, Evan Zheran Liu, Zheng Xiong, Luisa Zintgraf, Chelsea Finn, and Shimon Whiteson. "A survey of meta-reinforcement learning". In: *arXiv preprint arXiv:2301.08028* (2023).

[213] Chelsea Finn, Pieter Abbeel, and Sergey Levine. "Model-agnostic meta-learning for fast adaptation of deep networks". In: *International conference on machine learning*. PMLR. 2017, pp. 1126–1135.

[214] Yan Duan, John Schulman, Xi Chen, Peter L Bartlett, Ilya Sutskever, and Pieter Abbeel. "Rl2: Fast reinforcement learning via slow reinforcement learning". In: *arXiv preprint arXiv:1611.02779* (2016).

[215] Anthony R Cassandra, Leslie Pack Kaelbling, and Michael L Littman. "Acting optimally in partially observable stochastic domains". In: *Aaai*. Vol. 94. 1994, pp. 1023–1028.

[216] Ian Osband, Charles Blundell, Alexander Pritzel, and Benjamin Van Roy. "Deep exploration via bootstrapped DQN". In: *Advances in neural information processing systems* 29 (2016).

[217] Qi Cai, Zhuoran Yang, Chi Jin, and Zhaoran Wang. "Provably efficient exploration in policy optimization". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 1283–1294.

[218] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. "Proximal policy optimization algorithms". In: *arXiv preprint arXiv:1707.06347* (2017).

[219] Lin Yang and Mengdi Wang. "Sample-optimal parametric q-learning using linearly additive features". In: *International Conference on Machine Learning*. 2019, pp. 6995–7004.

[220] Chi Jin, Zhuoran Yang, Zhaoran Wang, and Michael I Jordan. "Provably efficient reinforcement learning with linear function approximation". In: *arXiv preprint arXiv:1907.05388* (2019).

[221] Yaqi Duan, Zeyu Jia, and Mengdi Wang. "Minimax-optimal off-policy evaluation with linear function approximation". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 2701–2709.

[222] Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. "Finite-time analysis of the multiarmed bandit problem". In: *Machine learning* 47.2 (2002), pp. 235–256.

[223] Sébastien Bubeck, Nicolo Cesa-Bianchi, et al. "Regret analysis of stochastic and nonstochastic multi-armed bandit problems". In: *Foundations and Trends® in Machine Learning* 5.1 (2012), pp. 1–122.

[224] Negar Hariri, Bamshad Mobasher, and Robin Burke. "Context adaptation in interactive recommender systems". In: *Proceedings of the 8th ACM Conference on Recommender Systems*. 2014, pp. 41–48.

[225] Zhiyong Wang, Jize Xie, Tong Yu, Shuai Li, and John Lui. "Online Corrupted User Detection and Regret Minimization". In: *arXiv preprint arXiv:2310.04768* (2023).

[226] Claudio Gentile, Shuai Li, Purushottam Kar, Alexandros Karatzoglou, Giovanni Zappella, and Evans Etrue. "On context-dependent clustering of bandits". In: *International Conference on machine learning*. PMLR. 2017, pp. 1253–1262.

[227] Yikun Ban and Jingrui He. "Local clustering in contextual multi-armed bandits". In: *Proceedings of the Web Conference 2021*. 2021, pp. 2335–2346.

[228] F Maxwell Harper and Joseph A Konstan. "The movielens datasets: History and context". In: *Acm transactions on interactive intelligent systems (tiis)* 5.4 (2015), pp. 1–19.

[229] Shi Zong, Hao Ni, Kenny Sung, Nan Rosemary Ke, Zheng Wen, and Branislav Kveton. "Cascading bandits for large-scale recommendation problems". In: *arXiv preprint arXiv:1603.05359* (2016).

[230] Charu C Aggarwal et al. *Recommender systems*. Vol. 1. Springer, 2016.

[231] Evrard Garcelon, Baptiste Roziere, Laurent Meunier, Jean Tarbouriech, Olivier Teytaud, Alessandro Lazaric, and Matteo Pirotta. "Adversarial attacks on linear contextual bandits". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 14362–14373.

[232] Xutong Liu, Jinhang Zuo, Siwei Wang, Carlee Joe-Wong, John Lui, and Wei Chen. "Batch-size independent regret bounds for combinatorial semi-bandits with probabilistically triggered arms or independent arms". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 14904–14916.

[233] Kwang-Sung Jun, Lihong Li, Yuzhe Ma, and Jerry Zhu. "Adversarial attacks on stochastic bandits". In: *Advances in neural information processing systems* 31 (2018).

[234] Fang Liu and Ness Shroff. "Data poisoning attacks on stochastic bandits". In: *International Conference on Machine Learning*. PMLR. 2019, pp. 4042–4050.

[235] Yang Liu, Xiang Ao, Zidi Qin, Jianfeng Chi, Jinghua Feng, Hao Yang, and Qing He. "Pick and choose: a GNN-based imbalanced learning approach for fraud detection". In: *Proceedings of the Web Conference 2021*. 2021, pp. 3168–3177.

[236] Mengda Huang, Yang Liu, Xiang Ao, Kuan Li, Jianfeng Chi, Jinghua Feng, Hao Yang, and Qing He. "AUC-oriented Graph Neural Network for Fraud Detection". In: *Proceedings of the ACM Web Conference 2022*. 2022, pp. 1311–1321.

[237] Shuai Li, Wei Chen, and Kwong-Sak Leung. "Improved algorithm on online clustering of bandits". In: *arXiv preprint arXiv:1902.09162* (2019).

[238] Qiutong Li, Yanshen He, Cong Xu, Feng Wu, Jianliang Gao, and Zhao Li. "Dual-Augment Graph Neural Network for Fraud Detection". In: *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 2022, pp. 4188–4192.

[239] Zidi Qin, Yang Liu, Qing He, and Xiang Ao. "Explainable Graph-based Fraud Detection via Neural Meta-graph Search". In: *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 2022, pp. 4414–4418.

[240] Heyang Zhao, Dongruo Zhou, and Quanquan Gu. "Linear contextual bandits with adversarial corruptions". In: *arXiv preprint arXiv:2110.12615* (2021).

[241] Ilija Bogunovic, Arpan Losalka, Andreas Krause, and Jonathan Scarlett. "Stochastic linear bandits robust to adversarial attacks". In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2021, pp. 991–999.

[242] Julian John McAuley and Jure Leskovec. "From amateurs to connoisseurs: modeling the evolution of user expertise through online reviews". In: *Proceedings of the 22nd international conference on World Wide Web*. 2013, pp. 897–908.

[243] Shebuti Rayana and Leman Akoglu. "Collective opinion spam detection: Bridging review networks and metadata". In: *Proceedings of the 21th acm sigkdd international conference on knowledge discovery and data mining*. 2015, pp. 985–994.

[244] Shenghua Liu, Bryan Hooi, and Christos Faloutsos. "Holoscope: Topology-and-spike aware fraud detection". In: *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. 2017, pp. 1539–1548.

[245] Konstantina Christakopoulou, Alex Beutel, Rui Li, Sagar Jain, and Ed H Chi. "Q&R: A two-stage approach toward interactive recommendation". In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2018, pp. 139–148.

[246] Yueming Sun and Yi Zhang. "Conversational recommender system". In: *The 41st international acm sigir conference on research & development in information retrieval*. 2018, pp. 235–244.

[247] Yongfeng Zhang, Xu Chen, Qingyao Ai, Liu Yang, and W Bruce Croft. "Towards conversational search and recommendation: System ask, user respond". In: *Proceedings of the 27th acm international conference on information and knowledge management*. 2018, pp. 177–186.

[248] Shijun Li, Wenqiang Lei, Qingyun Wu, Xiangnan He, Peng Jiang, and Tat-Seng Chua. "Seamlessly unifying attributes and items: Conversational recommendation for cold-start users". In: *ACM Transactions on Information Systems (TOIS)* 39.4 (2021), pp. 1–29.

[249] Chongming Gao, Wenqiang Lei, Xiangnan He, Maarten de Rijke, and Tat-Seng Chua. "Advances and challenges in conversational recommender systems: A survey". In: *AI Open* 2 (2021), pp. 100–126.

[250] Chaitanya Amballa, Manu K Gupta, and Sanjay P Bhat. "Computing an Efficient Exploration Basis for Learning with Univariate Polynomial Features". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35. 8. 2021, pp. 6636–6643.

[251] Baruch Awerbuch and Robert Kleinberg. "Online linear optimization and adaptive routing". In: *Journal of Computer and System Sciences* 74.1 (2008), pp. 97–114.

[252] Iván Cantador, Peter Brusilovsky, and Tsvi Kuflik. "2nd Workshop on Information Heterogeneity and Fusion in Recommender Systems (HetRec 2011)". In: *Proceedings of the 5th ACM conference on Recommender systems*. RecSys 2011. Chicago, IL, USA: ACM, 2011.

[253] Wang Chi Cheung, David Simchi-Levi, and Ruihao Zhu. "Hedging the drift: Learning to optimize under non-stationarity". In: *Available at SSRN 3261050* (2018).

[254] Jing Wang, Peng Zhao, and Zhi-Hua Zhou. "Revisiting Weighted Strategy for Non-stationary Parametric Bandits". In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2023, pp. 7913–7942.

[255] Louis Faury, Yoan Russac, Marc Abeille, and Clément Calauzènes. "Regret bounds for generalized linear bandits under parameter drift". In: *arXiv preprint arXiv:2103.05750* (2021).

[256] Weichao Mao, Kaiqing Zhang, Ruihao Zhu, David Simchi-Levi, and Tamer Basar. "Near-Optimal Model-Free Reinforcement Learning in Non-Stationary Episodic MDPs". In: *Proceedings of the 38th International Conference on Machine Learning*. Ed. by Marina Meila and Tong Zhang. Vol. 139. Proceedings of Machine Learning Research. PMLR, 18–24 Jul 2021, pp. 7447–7458.

[257] Ahmed Touati and Pascal Vincent. "Efficient Learning in Non-Stationary Linear Markov Decision Processes". In: *arXiv preprint arXiv:2010.12870* (2020).

[258] Pratik Gajane, Ronald Ortner, and Peter Auer. "A sliding-window algorithm for markov decision processes with arbitrarily changing rewards and transitions". In: *arXiv preprint arXiv:1805.10066* (2018).

[259] Wang Chi Cheung, David Simchi-Levi, and Ruihao Zhu. "Reinforcement learning for non-stationary markov decision processes: The blessing of (more) optimism". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 1843–1854.

[260] Baekjin Kim and Ambuj Tewari. "Randomized Exploration for Non-Stationary Stochastic Linear Bandits". In: *Uncertainty in Artificial Intelligence*. 2020.

[261] Peng Zhao, Lijun Zhang, Yuan Jiang, and Zhi-Hua Zhou. "A Simple Approach for Non-stationary Linear Bandits". In: *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*. Ed. by Silvia Chiappa and Roberto Calandra. Vol. 108. Proceedings of Machine Learning Research. PMLR, 26–28 Aug 2020, pp. 746–755.

[262] Jiafan He, Dongruo Zhou, and Quanquan Gu. "Uniform-pac bounds for reinforcement learning with linear function approximation". In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 14188–14199.

[263] Omar Besbes, Yonatan Gur, and Assaf Zeevi. "Optimal Exploration-Exploitation in a Multi-Armed-Bandit Problem with Non-Stationary Rewards". In: *SSRN Electronic Journal* (2014).

[264] Xiaoqiang Lin, Zhaoxuan Wu, Zhongxiang Dai, Wenyang Hu, Yao Shu, See-Kiong Ng, Patrick Jaillet, and Bryan Kian Hsiang Low. "Use Your INSTINCT: INSTruction optimization usIng Neural bandits Coupled with Transformers". In: *Proc. ICML*. 2024.

[265] David R Hunter. "MM Algorithms for Generalized Bradley-Terry Models". In: *Annals of Statistics* (2004), pp. 384–406.

[266] R Duncan Luce. *Individual choice behavior: A theoretical analysis*. Courier Corporation, 2005.

[267] Lihong Li, Yu Lu, and Dengyong Zhou. "Provably Optimal Algorithms for Generalized Linear Contextual Bandits". In: *Proc. ICML*. 2017, pp. 2071–2080.

[268] Arthur Jacot, Franck Gabriel, and Clément Hongler. "Neural tangent kernel: Convergence and generalization in neural networks". In: *Proc. NeurIPS*. 2018.

[269] Zhongxiang Dai, Yao Shu, Arun Verma, Flint Xiaofeng Fan, Bryan Kian Hsiang Low, and Patrick Jaillet. "Federated Neural Bandits". In: *Proc. ICLR*. 2023.

[270] Richard M Dudley. "Central limit theorems for empirical measures". In: *The Annals of Probability* (1978), pp. 899–929.

[271] Tong Zhang. "From ε-entropy to KL-entropy: Analysis of minimum information complexity density estimation". In: *The Annals of Statistics* (2006), pp. 2180–2210.

[272] Chi Jin, Zhuoran Yang, Zhaoran Wang, and Michael I Jordan. "Provably efficient reinforcement learning with linear function approximation". In: *Conference on learning theory*. PMLR. 2020, pp. 2137–2143.

[273] Yu Song, Shuai Sun, Jianxun Lian, Hong Huang, Yu Li, Hai Jin, and Xing Xie. "Show Me the Whole World: Towards Entire Item Space Exploration for Interactive Personalized Recommendations". In: *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*. 2022, pp. 947–956.

[274] Alexandre Letard, Tassadit Amghar, Olivier Camp, and Nicolas Gutowski. "Partial Bandit and Semi-Bandit: Making the Most Out of Scarce Users' Feedback". In: *2020 IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI)*. IEEE. 2020, pp. 1073–1078.

[275] Alexandre Letard, Tassadit Amghar, Olivier Camp, and Nicolas Gutowski. "COM-MABs: From Users' Feedback to Recommendation". In: *The International FLAIRS Conference Proceedings*. Vol. 35. 2022.

[276] Heyang Zhao, Jiafan He, and Quanquan Gu. "A nearly optimal and low-switching algorithm for reinforcement learning with general function approximation". In: *arXiv preprint arXiv:2311.15238* (2023).

[277] Alekh Agarwal, Nan Jiang, Sham M Kakade, and Wen Sun. "Reinforcement learning: Theory and algorithms". In: (2019).

[278] Chi Jin, Zeyuan Allen-Zhu, Sebastien Bubeck, and Michael I Jordan. "Is Q-learning provably efficient?" In: *Advances in neural information processing systems* 31 (2018).

[279] Runlong Zhou, Zhang Zihan, and Simon Shaolei Du. "Sharp variance-dependent bounds in reinforcement learning: Best of both worlds in stochastic and deterministic environments". In: *International Conference on Machine Learning*. PMLR. 2023, pp. 42878–42914.

[280] Leonardo Cella and Massimiliano Pontil. "Multi-task and meta-learning with sparse linear bandits". In: *Uncertainty in Artificial Intelligence*. PMLR. 2021, pp. 1692–1702.

[281] Aniket Anand Deshmukh, Urun Dogan, and Clay Scott. "Multi-task learning for contextual bandits". In: *Advances in neural information processing systems* 30 (2017).

[282] Marta Soare, Ouais Alsharif, Alessandro Lazaric, and Joelle Pineau. "Multi-task linear bandits". In: *NIPS2014 workshop on transfer and multi-task learning: theory meets practice*. 2014.

[283] Leonardo Cella, Karim Lounici, Grégoire Pacreau, and Massimiliano Pontil. "Multi-task representation learning with stochastic linear bandits". In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2023, pp. 4822–4847.

[284] Zhi Wang, Chicheng Zhang, and Kamalika Chaudhuri. "Thompson Sampling for Robust Transfer in Multi-Task Bandits". In: *arXiv preprint arXiv:2206.08556* (2022).

[285] Zhi Wang, Chicheng Zhang, Manish Kumar Singh, Laurel Riek, and Kamalika Chaudhuri. "Multitask bandit learning through heterogeneous feedback aggregation". In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2021, pp. 1531–1539.

[286] Runzhe Wan, Lin Ge, and Rui Song. "Towards scalable and robust structured bandits: A meta-learning framework". In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2023, pp. 1144–1173.

[287] Joey Hong, Branislav Kveton, Manzil Zaheer, and Mohammad Ghavamzadeh. "Hierarchical bayesian bandits". In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2022, pp. 7724–7741.

[288] Leonardo Cella, Alessandro Lazaric, and Massimiliano Pontil. "Meta-learning with stochastic linear bandits". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 1360–1370.

[289] Chengshuai Shi and Cong Shen. "Federated multi-armed bandits". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35. 11. 2021, pp. 9603–9611.

[290] Ruiquan Huang, Weiqiang Wu, Jing Yang, and Cong Shen. "Federated linear contextual bandits". In: *Advances in neural information processing systems* 34 (2021), pp. 27057–27068.

[291] Runzhe Wan, Lin Ge, and Rui Song. "Metadata-based multi-task bandits with bayesian hierarchical models". In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 29655–29668.

[292] Yasuhiko Ikebe, Toshiyuki Inagaki, and Sadaaki Miyamoto. "The monotonicity theorem, Cauchy's interlace theorem, and the Courant-Fischer theorem". In: *The American Mathematical Monthly* 94.4 (1987), pp. 352–354.

[293] Max A Woodbury. *Inverting modified matrices*. Statistical Research Group, 1950.

[294] David A Freedman. "On tail probabilities for martingales". In: *the Annals of Probability* (1975), pp. 100–118.