Bandit Algorithms with Graphical Feedback Models and Privacy Awareness

by

Bingshan Hu
M.Sc., Beijing University of Posts and Telecommunications, 2013
B.Eng., Beijing University of Posts and Telecommunications, 2010

A Dissertation Submitted in Partial Fulfillment of the
Requirements for the Degree of

DOCTOR OF PHILOSOPHY

in the Department of Computer Science

Bandit Algorithms with Graphical Feedback Models and Privacy Awareness

by

Bingshan Hu
M.Sc., Beijing University of Posts and Telecommunications, 2013
B.Eng., Beijing University of Posts and Telecommunications, 2010

Supervisory Committee

Dr. Nishant Mehta, Supervisor
(Department of Computer Science)

Dr. Jianping Pan, Departmental Member
(Department of Computer Science)

Dr. Alex Thomo, Departmental Member
(Department of Computer Science)

Dr. Xiaodai Dong, Outside Member
(Department of Electrical and Computer Engineering)

ABSTRACT

This thesis focuses on two classes of learning problems in stochastic multi-armed bandits (MAB): graphical bandits and private bandits. Different from the basic MAB setting where the learning algorithm can only have one observation, for a bandit problem under a graphical feedback model, the learning algorithm may be able to have more than one observation every time it interacts with the environment. Meanwhile, the learning algorithm only needs to suffer a regret resulting from the pulled arm if it is not the optimal one, which is the same as the basic MAB setting. The first theme of this thesis is to derive instance-dependent regret bounds for stochastic bandits under graphical feedback models.

In a basic MAB problem, the learning algorithm can always use the learnt information to make future decisions. If each reward vector encodes information of an individual, this kind of non-private learning algorithm may "leak" sensitive information associated with individuals. In an MAB problem with privacy awareness, the learning algorithm cannot rely on the true information learnt to make future decisions in order to comply with privacy. What a private learning algorithm promises is even if an adversary sees the output of the learning algorithm, this adversary almost cannot infer any information associated with a single individual. The second theme of this thesis covers three variants of private online learning: the private bandit setting, the private full information setting, and the private graphical bandit setting.

Contents

List of Tables

List of Figures

ACKNOWLEDGEMENTS

I would like to thank Nishant, my advisor, for offering me the opportunity to do learning theory despite that my background is in electrical engineering. I also want to thank Dr. Pan for giving me the chance to study in UVic. Last but not least, many thanks for Alex and Dr. Dong for serving as committee members for my study in UVic.

DEDICATION

To my parents

Chapter 1

Introduction

This thesis focuses on two major learning problems in sequential decision-making (online learning) under uncertainty. One is the multi-armed bandit (MAB) problem under *graphical feedback models*. The other one is the multi-armed bandit problem with *privacy awareness*. Although the framework of MAB is simple, it is very powerful and meaningful, as it guides people to conquer the dilemma of *exploration-vs-exploitation* in an unknown environment.

The story of solving an MAB problem can be dated back to 1933, when William Thompson showed an idea about how to pull arms sequentially in a 2-armed bandit problem. From then, MAB problems are always drawing great attention from researchers and engineers across different fields such as maths, statistics, economics, computer science, pharmaceutical science, and electrical engineering. Actually, the idea shown by Thompson was developed into one of the most "sample-efficient" and practical learning algorithms for stochastic bandits: Thompson Sampling. The first contribution of this thesis is to provide theoretical guarantees for Thompson Sampling algorithms under the settings where feedback relationships can be represented by graphs.

In this modern information age, instead of seeing a teller in person, most of the services that our daily life needs can be fulfilled by using Apps online, for instance, banking, shopping, and filing taxes. Some interactions may involve sensitive individual information which may be collected by a third party to do data analysis in order to offer a better service. Not surprisingly, people are starting being concerned about privacy. The implementation of differential privacy mechanisms within the learning algorithm used by a third party can tackle the privacy concerns raised by the participating individuals. The remaining contribution of this thesis

is to devise online learning algorithms with privacy awareness. Through the presented private algorithms and their theoretical guarantees, we plan to answer the following fundamental questions in differentially private online learning: how to maintain the *privacy-vs-regret* trade-off and what is the price for the algorithms to protect privacy of individuals.

1.1 Multi-Armed Bandit (MAB) Problems

A multi-armed bandit (MAB) problem is a classical sequential decision-making problem [8]. In this learning problem, we have a fixed arm set $\mathcal{A}$ with size K, a Learner, and an environment that generates rewards for all arms. Learner interacts with the environment in a sequential way. In each round $t = 1, 2, \ldots, T$, the environment generates a reward vector $\boldsymbol{X}_t := (X_1(t), \ldots, X_K(t))$ that is hidden to Learner. Simultaneously, Learner pulls an arm $J_t \in \mathcal{A}$. At the end of round t, the environment reveals *only* the reward of the pulled arm $X_{J_t}(t)$ to Learner. Learner observes and obtains a reward $X_{J_t}(t)$. The goal of Learner is to pull arms sequentially to maximize its cumulative reward over T rounds. The reward vector $\boldsymbol{X}_t$ can be generated in a stochastic way or an adversarial way, depending on the learning tasks. In this thesis, all the presented learning problems are under the settings that the rewards for a specific arm $j \in \mathcal{A}$ are i.i.d. over time from a fixed but unknown probability distribution v_j with $[0, 1]$ support.

In an MAB problem, the information revealed by the environment in each round is very limited. Learner can only observe the random reward of the pulled arm. This imperfect feedback model makes Learner in an *exploration-vs-exploitation* dilemma. In every round, will Learner pull the arm with the highest empirical mean so far, or pull an arm that has not been pulled too often? The former option tends to do exploitation to gain reward while the latter option tends to do exploration to gain information about the environment. Actually, any successful learning algorithm for MAB problems cannot fail to achieve the trade-off between exploration and exploitation.

Since Learner has no idea about the distributions from which random rewards are generated, Learner can only rely on tracking the number of observations and the empirical mean among these observations to make a decision. Let μ_j be the mean of distribution v_j, i.e., the mean reward of arm j. For a K-armed stochastic bandit problem, $\mu_1, \mu_2, \ldots, \mu_K$ can fully characterize a learning problem instance.

For ease of presentation, we assume that the first arm is the unique arm with the highest mean reward, i.e., $\mu_1 > \mu_j$ for all $j \in \mathcal{A} \setminus \{1\}$.

As we will show in Section 2.1, we use (Pseudo)-Regret to measure the quality of any developed learning algorithms. The regret measures the cumulative performance loss when Learner fails to pull the arm with the highest mean reward. Under the notion of (Pseudo)-Regret, we do not make any assumption on how the problem instance is generated. If a regret bound depends on a specific problem instance $\mu_1, \mu_2, \ldots, \mu_K$, we say this regret bound is the problem-dependent one. Typically, there are two terms in a problem-dependent regret bound. We call the term depending on T the leading term and the term that does not depend on T the constant term.

In Section 2.2, we will review the existing optimal learning algorithms for stochastic bandits, i.e., the Upper Confidence Bound (UCB)-based [5, 24, 19], the Thompson Sampling-based [2], and the elimination-style learning algorithms [17, 6], and discuss the ideas how they achieve the exploration-vs-exploitation trade-off. We will also show that the problem-dependent regret bounds for these algorithms take either an $O\left(\sum_{j\in\mathcal{A}:\mu_j<\mu_1} \frac{\log(T)}{\mu_1-\mu_j}\right)$ form or an $O\left(\sum_{j\in\mathcal{A}:\mu_j<\mu_1} \frac{\log(T)(\mu_1-\mu_j)}{d_{KL}(\mu_j,\mu_1)}\right)$ form, depending on the utilized learning algorithm[1]. As implied by the problem-dependent regret bounds, the limited feedback from the environment, i.e., only obtaining one observation per round, makes the learning algorithms suffer a regret bound that is linear in the number of sub-optimal arms. In terms of the time horizon T, the problem-dependent regret bounds grow only logarithmically in T.

1.2 MAB under Graphical Feedback Models

Full information game. There is also one setting where the feedback model is perfect, i.e., the complete reward vector $\boldsymbol{X}_t = (X_1(t), \ldots, X_K(t))$ can be seen in every round regardless of which arm is pulled by Learner [18, 41, 40][2]. We call this setting the *full information setting*. A nice feature of the full information setting is that exploration is not needed. For the full information setting with stochastic

[1] $d_{KL}(x, y) := x \log\left(\frac{x}{y}\right) + (1-x)\log\left(\frac{1-x}{1-y}\right)$ indicates the Kullback–Leibler (KL) divergence between two Bernoulli distributions with parameters x and y.

[2] In this thesis, Learner plays a single action instead of playing a weight distribution over all actions.

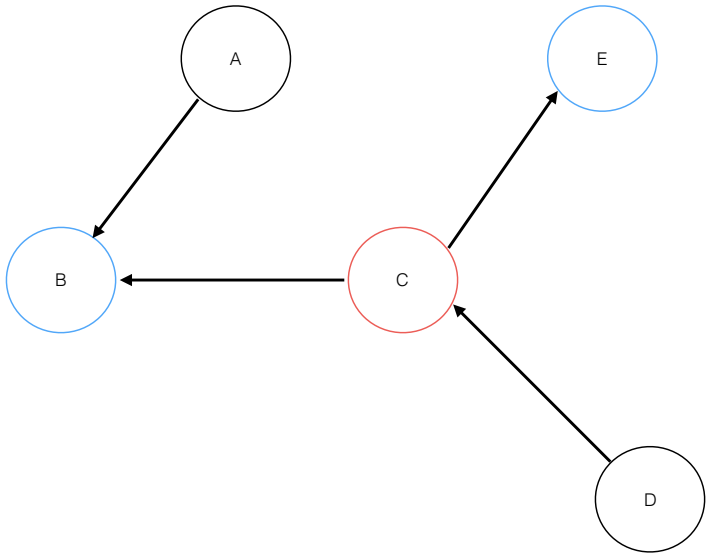

Figure 1.1: MAB under graphical feedback models

rewards, as we will show in Section 2.3, Learner will suffer only an $O\left(\frac{\log(K)}{\Delta}\right)$ regret, a constant regret that does not depend on T.

The first theme of this thesis is to study the settings that lie between the bandit setting and the full information setting: MAB problems under graphical feedback models (graphical bandits). In this thesis, Chapters 3, 6 are related to this theme. Motivated by tremendous practical applications, graphical bandits have been widely studied in [31, 10, 3, 14, 30, 22].

Formally, a graph $G := (\mathcal{A}, \mathcal{E})$ is used to represent the feedback models among all arms collected by a set $\mathcal{A}$. Each node $i \in \mathcal{A}$ in the graph corresponds to an arm and an edge $\{i, j\} \in \mathcal{E}$ indicates the feedback relationship between nodes i, j. Under a graphical feedback model, each time Learner pulls arm i, Learner can also have an observation for arm j. Figure 1.1 presents a concrete example for graphical bandits with 5 nodes. In this example, when pulling arm C, besides obtaining an observation from arm C, Learner can also have observations from arms B and E.

By leveraging the graphical feedback models, the problem-dependent regret bounds can be improved from $O\left(\frac{K\log(T)}{\Delta}\right)$ to $\widetilde{O}\left(\frac{\alpha(G)\log(T)}{\Delta}\right)$ or $\widetilde{O}\left(\frac{\beta(G)\log(T)}{\Delta}\right)$, where $\alpha(G)$ indicates the independence number and $\beta(G)$ indicates the clique covering number of graph G [10, 30, 22, 14]. If the size of the graph is very large, the improvement from linearity in K to linearity in graph-specific quantities is of great significance.

1.3 Differentially Private Online Learning

Since in an online learning setting, data points are arriving in a sequential way, we let $X_{1:T}$ be the sequence of reward vectors that are fed into a learning algorithm. Each reward vector can encode information associated with an individual. Let $X'_{1:T}$ be another sequence such that $X_{1:T}$ and $X'_{1:T}$ differ in at most one reward vector.

A good private online learning algorithm guarantees that even if we change the input from $X_{1:T}$ to $X'_{1:T}$, the output of the learning algorithm, e.g., the sequence of the pulled arms, stays almost the same. This property implies that from the output of a differentially private online learning algorithm, an external observer is very unlikely to infer any information associated with any single individual.

In differentially private online learning, typically, Learner needs to maintain a *regret-vs-privacy* trade-off. On one hand, the learning algorithm would perform better if more information is revealed in each round. On the other hand, the more information revealed, the harder for the learning algorithm to achieve differential privacy.

Formally, we say an online learning algorithm Π is ϵ-differentially private if for any two neighbouring reward sequences $X_{1:T}$ and $X'_{1:T}$, for any set $\mathcal{D}$ of decisions, it holds that

$$\mathbb{P}\left\{\Pi(X_{1:T}) \in \mathcal{D}\right\} \leq e^{\epsilon} \cdot \mathbb{P}\left\{\Pi(X'_{1:T}) \in \mathcal{D}\right\} \quad , \tag{1.1}$$

where $\Pi(X_{1:T})$ indicates the output sequence when a private algorithm Π takes $X_{1:T}$ as input and $\Pi(X'_{1:T})$ indicates the output sequence when a private algorithm Π takes $X'_{1:T}$ as input.

Let σ be an arbitrary point in the decision space. From (1.1), we know that, for a private online learning Π, we have

$$\ln\left(\frac{\mathbb{P}\left\{\Pi(X_{1:T}) = \sigma\right\}}{\mathbb{P}\left\{\Pi(X'_{1:T}) = \sigma\right\}}\right) \leq \epsilon \quad . \tag{1.2}$$

From (1.2), it is easy to see that if we set $\epsilon \to \infty$, the private algorithm $\Pi(\cdot)$ can be considered as a non-private algorithm.

The second theme of this thesis is to design ϵ-differentially private online learning algorithms, particularly, for the bandit setting, the full information setting, and the graphical bandit setting. Chapters 4, 5, and 6 are related to this theme. Differentially private online learning have been widely studied in

[23, 1, 32, 21, 33, 34, 37, 38].

1.4 Overviews of Chapters

We now present an overview of the key contributions in each chapter.

1.4.1 Overview of Chapter 3

This chapter addresses a stochastic multi-armed bandit problem with an undirected feedback graph. Just as mentioned in Section 1.2, the graphical bandit setting lies in between the bandit setting and the full information setting. By leveraging the graphical feedback structure, the problem-dependent regret bounds can be improved from being linear in the total number of arms to being linear in some graph-specific quantities such as the independence number and the clique covering number.

Motivation. Regarding the setting of stochastic graphical bandits, [10] devised the first UCB-based learning algorithm, UCB-N, for it. The key idea behind UCB-N is to construct a confidence interval for each arm. Then, Learner pulls the arm with the highest upper confidence bound, the same as in UCB1 of [5]. The leading term for UCB-N is linear with the size of an arbitrary clique covering $\mathcal{C}$. However, the constant term is still linear in the total number of arms.

Later, [28, 29] devised a Thompson Sampling-based learning algorithm, TS-N, for the setting of graphical bandits. However, they only presented problem-independent regret bounds for TS-N under the notion of Bayesian Regret instead of the commonly-used Pseudo-Regret in bandit problems. Since, empirically, Thompson Sampling-based algorithms usually perform better than the UCB-based algorithms, we are interested in deriving a more refined regret bound, the problem-dependent regret bound under the notion of Pseudo-Regret, for TS-N.

Main Results. We devise a UCB-based algorithm, UCB-NE (Algorithm 5), and derive a problem-dependent regret bound (Theorem 9) for it. The regret bound of UCB-NE improves the constant term of UCB-N while preserving the same leading term as UCB-N. The constant term of UCB-NE is linear in the size of an arbitrary clique covering $\mathcal{C}$ up to logarithmic factors.

Also, we present the first problem-dependent regret bounds for TS-N [28], where again the regret bounds are linear in the size of a clique covering $\mathcal{C}$ up to loga-

rithmic factors. We derive two problem-dependent regret bounds for TS-N. The first problem-dependent regret bound for TS-N (shown in Theorem 10) takes the $\sum_{C\in\mathcal{C}} \frac{(1+\epsilon)\log(T)\Delta_C}{d_{KL}(\mu_1-\Delta_C,\mu_1)} + O\left(\frac{\ln(|C|)+1}{\epsilon^2}\right)$ form, where Δ_C can be viewed as a clique-specific mean reward gap and $\epsilon > 0$ can be chosen as small as desired. In this regret bound, the leading term for each clique is asymptotically optimal. However, the constant term hides problem-dependent constants, which makes it difficult to tune ϵ to minimize the regret bound. The second problem-dependent regret bound for TS-N (shown in Theorem 11) takes the $\sum_{C\in\mathcal{C}} \widetilde{O}\left(\frac{\log(|C|\cdot T)}{(1-\epsilon)^2\Delta_C}\right) + O\left(\frac{\log(T)}{\epsilon^2\Delta_C}\right)$ form, where ϵ can be any value in $(0,1)$. When comparing the second bound to the first bound, there is no hidden problem-dependent constants[3] and all the terms are expressed explicitly. The exposure of all the terms is beneficial to tune parameters to minimize the regret bound.

Key Ideas of UCB-NE. The reason that makes UCB-NE succeed in improving the constant term for each clique from being linear in the clique size to being logarithmic in the clique size lies in our novel idea to construct the upper confidence bounds. We take the degrees of the nodes in the feedback graph into account when constructing the upper confidence bounds. We boost the exploration of an arm if it has a higher degree. The intuitive understanding of why this works is that if we pull an arm with a higher degree, we can have more observations than pulling an arm with a lower degree.

Key Ideas of TS-N. To derive a problem-dependent regret bound for TS-N, we use similar ideas that have been presented in [2] to derive problem-dependent regret bound for the standard stochastic MAB problems. We cut the mean reward gap Δ_C into three pieces by finding two clique-specific problem-dependent constants x_C and y_C such that $\mu_1 - \Delta_C < x_C < y_C < \mu_1$. By introducing x_C and y_C, the "distance" between $\mu_1 - \Delta_C$ and μ_1 is separated into three "sub-distances". Informally, the "distance" between x_C and y_C impacts the leading term while the "distance" between $\mu_1 - \Delta_C$ and x_C and the "distance" between and y_C and μ_1 both impact the constant term. The key to have problem-dependent bounds lies in the tuning of x_C and y_C properly to make sure the the probability to pull any arm in clique C is in the order of $\frac{1}{|C|}$. Then, by using a union bound, the impact of the clique size can be removed.

[3]For our presented version here, actually, the $\widetilde{O}(\cdot)$ notation hides a small problem-dependent logarithmic factor $\log^2\left(\frac{\mu_1}{\mu_1-\Delta_C}\right)$.

1.4.2 Overview of Chapter 4

In this chapter, we investigate two variants of differentially private online learning: the differentially private stochastic bandit setting and the differentially private full information setting with stochastic rewards.

Motivation. Regarding the private stochastic bandit setting, [32] devised the first private UCB algorithm for it. However, the presented regret bound is far from optimal. Later, [33] devised the Differentially Private Successive Elimination (DP-SE) algorithm, an elimination-style algorithm, which achieves the optimal $O\left(\sum_{j\in\mathcal{A}:\Delta_j>0}\frac{\log(T)}{\min\{\Delta_j,\epsilon\}}\right)$ regret bound, where Δ_j indicates the mean reward gap of a sub-optimal arm j and ϵ is the required privacy parameter. However, like other elimination style algorithms, DP-SE does not practically perform well compared to UCB-based algorithms. Also, DP-SE is not an anytime learning algorithm as the elimination rule relies on knowing T in advance. Therefore, we are motivated to have an *anytime and optimal* private UCB-based algorithm. Regarding the full information setting, to the best of our knowledge, there did not exist any learning algorithm that has a constant regret, i.e., a regret bound does not depend on T.

Main Results. Regarding differentially private stochastic bandits, we present two anytime learning algorithms: Anytime-Lazy-UCB (Algorithm 8) and Hybrid-UCB (Algorithm 9)[4]. We devise the first optimal UCB-based algorithm, Anytime-Lazy-UCB. The regret bound for Anytime-Lazy-UCB is shown in Theorem 14. Regarding differentially private full information setting with stochastic rewards, we present a novel learning algorithm, Follow-the-Noisy-Leader (FTNL, Algorithm 10). FTNL is the first learning algorithm for this setting that enjoys an $O\left(\frac{\log(K)}{\min\{\Delta_{\min},\epsilon\}}\right)$ regret bound, a constant regret (Theorem 19), where $\Delta_{\min}$ indicates the minimum mean reward gap among all sub-optimal arms. In this chapter, we also conduct experiments to see the practical performance comparison among DP-SE, Anytime-Lazy-UCB, and Hybrid-UCB. The experimental results show that Anytime-Lazy-UCB is competitive with DP-SE.

Key Ideas of Anytime-Lazy-UCB. A nice property of differential privacy is that it is immune to post-processing. From this property, we know that if Learner

[4]Previously, [37] devised the first Hybrid-UCB style algorithm. The Hybrid-UCB style algorithm can be viewed as a private version of UCB1 of [5]. However, there are some issues in their analysis. For completeness, we propose our own Hybrid-UCB learning algorithm by extending our Anytime-Lazy-UCB. Theorem 17 presents our regret bound for Hybrid-UCB.

relies on the output of an ϵ-differentially private learning algorithm to make decisions, then the learning algorithm run by Learner is also ϵ-differentially private. Keeping this property in mind, we let the algorithm that computes the empirical means be ϵ-differentially private. Then, we can claim that the learning algorithm to make decisions is also ϵ-differentially private. Actually, [32, 33] operate under this framework to design their private algorithms.

The reasons that make Anytime-Lazy-UCB succeed are the usage of *forgetfulness and laziness*. Forgetfulness comes from the idea that the differentially private empirical mean for an arm is computed only based on a certain number of newly obtained fresh observations rather than using all the observations obtained from the very beginning. Also, reusing observations is not allowed when updating the differentially private empirical mean for an arm. The idea of laziness comes from the fact that the differentially private empirical mean of the pulled arm is not updated immediately. We take a lazy way to update it. We only update the differentially private empirical mean of an arm after accumulating enough fresh observations for that arm.

Key Ideas of FTNL. Recall that in a full information game, the complete reward vector can be observed in each round, i.e., more information is revealed than in a standard bandit setting. At first glance, one may think that more noise is needed by the private learning algorithm for a full information setting. Actually, the same level of noise as for a private bandit algorithm can be maintained by taking advantage of the *Report Noisy Max* (RNM) algorithm [16]. In Section 2.5, we will provide a detailed discussion about how to use RNM in the design of a private full information learning algorithm.

The reason why FTNL succeeds is the usage of a property that full information games have — the exploration is not needed. In a full information game, all the arms always have the same amount of observations. Therefore, Learner does not need to track the empirical mean of each arm directly. Instead, Learner only needs to track the *index of the arm* that has the highest aggregated reward. From the property that differential privacy is immune to post-processing, as long as the algorithm that outputs the arm with the highest aggregated reward is ϵ-differentially private, we know that the algorithm to make decisions is also ϵ-differentially private.

1.4.3 Overview of Chapter 5

This chapter investigates a novel variant of stochastic multi-armed bandits, bi-level bandits with unknown arms, which is motivated by the following practical application.

Motivation. Consider a drug testing problem in the medical sector. We have a set of pharmaceutical companies that are testing multiple drugs. Now, a foundation (it can be viewed as Learner) plans to invest in these pharmaceutical companies to assist them to do the testing. Due to a limited budget, in each period, the foundation can only choose one company to assist it to test one particular drug. However, to lock-in continuous support from the foundation, the selected company may not want to reveal the outcomes of a specific drug, particularly, a drug that is potentially inferior, as this potential failure drug may prevent the company from receiving further support from the foundation. Therefore, the company would like the outcomes of a particular drug that is under testing to be invisible to the foundation. Since the selected company would like to use the allocated support to test a drug that will succeed with high chance, and the foundation would also like to invest in a company that will succeed with high chance, they all have the goals to accumulate as much reward as possible.

This is one of the motivating applications that can be framed as a bi-level bandit problem. In a bi-level bandit setting, there are two levels of arms, but only Level-I arms (the pharmaceutical companies) are visible to Learner. Level-II arms (drugs) remain hidden and cannot be pulled nor observed directly by Learner. By introducing an extra level (Level-I arms) between Learner and the arms associated with rewards (Level-II arms), Learner has no chance to learn the outcomes of a particular (Level-II) arm directly.

Learning Problem. Let $\mathcal{A}$ be the Level-I arm set and $\mathcal{A}_j$ be the Level-II arm set managed by Level-I arm j. Regarding the learning protocol, in each round t, the environment generates a random reward $X_{j,i}(t) \in [0,1]$ for each Level-II arm from an unknown distribution with mean $\mu_{j,i}$. Simultaneously, Learner first pulls a Level-I arm $J_t \in \mathcal{A}$. Then, the selected Level-I arm J_t pulls a Level-II arm $I_t \in \mathcal{A}_{J_t}$. Under the bi-level bandit feedback model, the environment only reveals the reward of the pulled Level-II arm $X_{J_t,I_t}(t)$ to the selected Level-I arm J_t instead of to Learner. Learner can observe a reward $Y_{J_t}(t) = X_{J_t,I_t}(t)$ reported by J_t.

Main Results. Even if the rewards associated with each Level-II arm are i.i.d.

over time, the rewards associated with a specific Level-I arm (from Learner's perspective) are non-i.i.d. over time. Note that a Level-I arm runs its own algorithm to decide which Level-II arm to pull. It is challenging to devise a learning algorithm with an $O(\log(T))$ regret bound for bi-level bandits. We devise an elimination-style learning algorithm, Two-Level Elimination Algorithm (Algorithm 12), for bi-level bandits, and derive an $O(\log(T))$ regret bound (Theorem 20).

Since drug testing usually involves the participation of volunteers, the act of revealing outcomes directly to Learner might also compromise the privacy of individuals. Therefore, we also present a differentially private learning algorithm, Algorithm 13, for bi-level bandits. The private learning algorithm for bi-level bandits guarantees that the participation of a single individual in the testing has almost no impact on the decisions of the foundation to decide which company to support or the decisions of a specific company to decide which drug to test. Theorem 22 presents a regret bound for Differentially Private Two-Level Elimination Algorithm.

Technical Difficulties and Key Ideas. Since the rewards for a specific Level-I arm are non-i.i.d, the naive algorithm where Learner runs the round-based UCB over Level-I arms and each Level-I arm runs the round-based UCB over all its managed Level-II arms may not have a provable non-trivial regret bound. The key idea behind our novel Two-Level Elimination Algorithm is the controlling of the number of pulls of a sub-optimal Level-I arm j by constructing an asymmetric confidence interval. Note that all Level-II arms managed by Level-I arm j have no chance to be pulled if j itself is not pulled by Learner.

From Learner's perspective, the empirical mean of a Level-I arm j is the empirical average of $[0, 1]$ random variables that are independently drawn from some distributions.

Although each Level-I arm is not associated with a fixed true mean reward, the expected value of the empirical mean does have an upper bound μ_{j,i^*}, the highest mean reward among all the Level-II arms managed by Level-I arm j. By letting a Level-I arm j run an elimination-style algorithm, Learner can construct an asymmetric confidence interval stating that the gap between the empirical mean of Level-I arm j is not too far from μ_{j,i^*} after accumulating enough observations.

The reason why the elimination-style learning algorithm succeeds is that the elimination-style algorithm progresses in epochs and eliminates the bad arms with the progression of epochs. The shrinking of the non-eliminated arm set implies

that all the arms kept have good mean rewards with high probability. Also, in the elimination-style algorithm, all the non-eliminated arms will be pulled equal times in each epoch. All these factors ensure that the collection of observations for a specific Level-I arm within an epoch is a mixture of random variables that are independently drawn from distributions with similar means and the number of observations drawn from each distribution is identical.

There are two key challenges to design a differentially private learning algorithm for bi-level bandits. One is still brought by the non-i.i.d. rewards for each Level-I arm. This challenge can still be tackled by constructing an asymmetric confidence interval. The other challenge is the controlling of the noise variables included in the differentially private empirical mean of each Level-I arm. Ideally, we only want to include one noise variable.

Recall that the roles of a Level-I arm can be viewed as two-fold. One is to pull Level-II arms sequentially. The other one is to report rewards to help Learner to pull Level-I arms sequentially, i.e, Learner's decision relies on the output of the algorithm run by a Level-I arm. Since the goal is to guarantee that both the algorithms of Learner and Level-I arms are differentially private, we need to be careful when designing the private learning algorithm run by each Level-I arm.

One of the naive designs is to have a *single* private algorithm (at Level-I arm's side) first and then apply the post-processing to this private algorithm twice. More specifically, we let the algorithm that computes the empirical means of the managed Level-II arms be ϵ-differentially private. From the the property that differential privacy is immune to post-processing, we know: (1) the algorithm to pull Level-II arms is ϵ-differentially private; (2) the algorithm to compute Level-I arm's empirical mean is ϵ-differentially private. However, this design will result in a sub-optimal regret bound as the differentially private empirical mean for Level-I arm j may include more than one noise variables.

By using the property that differentially private algorithms can be composed, we take the strategy of *composing two differentially private algorithms* with each having its own purpose. More specifically, for the first private algorithm, we let the algorithm that computes the empirical means of the managed Level-II arms be ϵ-differentially private. Then, from the post-processing property, we know that the algorithm to pull Level-II arms is ϵ-differentially private. For the second private algorithm, we simply let the algorithm that computes the empirical mean of Level-I arm j itself be ϵ-differentially private. By composing these two private algorithms,

the number of noise variable included in the differentially empirical mean of a Level-I arm is limited to 1.

1.4.4 Overview of Chapter 6

In this chapter, we provide more insights and a deeper understanding for differentially private online learning algorithms. The focus of this chapter can be viewed as a generalized version of the settings that have been presented in Chapter 4, or a differentially private version of the setting that has been presented in Chapter 3. We study the learning problem of differentially private graphical bandits.

Motivation. So far, there are only private online learning algorithms for private bandit settings and full information settings. Both of these two settings have the property that the number of observations revealed is fixed over time. In a private bandit setting, the number of revealed observation is 1 while in a full information setting, the number of revealed observation is K. The unchanging number of observations over time makes the amount of noise injected within the learning algorithms under control. Therefore, we are interested in *differentially private graphical bandits* where the number of observations revealed is changing over time[5].

Main Results. We devise the first learning algorithm, DP-UCB-N (Algorithm 14), for differentially private graphical bandits, and it has a regret bound that is linear in the size of the input clique covering $\mathcal{C}$ up to logarithmic factors for both the leading and constant terms. Theorem 24 presents a regret bound for DP-UCB-N. which is

$$O\left(\sum_{1\leq i\leq|\mathcal{C}|}\frac{\log\left(|C_i|\cdot T\right)}{\min\left\{\Delta_i^{\min},\epsilon\right\}}\right) \quad ,$$

where $\mathcal{C}$ is an input clique covering and $\Delta_i^{\min}$ indicates the minimum mean reward gap among all sub-optimal arms covered by the i-th clique.

The regret bound for DP-UCB-N also covers the two special cases. The first one is, if the feedback graph only contains isolated nodes, the regret bound shown in Theorem 24 will be the same as the one for Anytime-Lazy-UCB, the optimal regret bound for private stochastic bandits. The second one is, if we set $\epsilon \to \infty$ (non-private graphical bandit setting), the regret bound shown in Theorem 24 is a better bound than the one shown in Theorem 9.

[5]We still assume that the feedback graph is static.

Key Ideas of DP-UCB-N. Before discussing the key ideas behind the design of DP-UCB-N, let us present a definition of l_1-sensitivity in a brief and informal way under the settings that all the rewards are in $[0, 1]$. The l_1-sensitivity of an algorithm tells us the maximum number of impacted arms when we change a reward vector. For example, in a bandit setting, when we change a reward vector in round t, only the empirical mean of the pulled arm in round t, i.e., *only one arm*, can be impacted. In a full information setting, although the changing of a reward vector can impact the empirical mean of every arm, by using RNM, still *only the arm* with the highest empirical mean in round t can be impacted.

A nice property of the two differentlly private online learning variants that have been discussed in Chapter 4 is the number of observations obtained in each round is fixed. Consequentially, by using clever algorithms, we can make sure that the l_1-sensitivity is fixed all the time. The fixed l_1-sensitivity makes it easy to decide the needed noise to have ϵ-differentially private algorithms. Informally, the amount of noise injected is linear in the number of impacted arms when we change a complete reward vector.

However, under graphical feedback models, the number of observations obtained in each round is varying over time. It can be any value in $[1, K]$, depending on the pulled arm and its degree in the feedback graph. The changing number of observations over time makes the l_1-sensitivity also change over time. Therefore, to have a private algorithm with a good theoretical guarantee, the key lies in the controlling of the l_1-sensitivity, i.e., we would like to control the number of impacted arms when we change a single reward vector.

Fortunately, we can use the ideas of Anytime-Lazy-UCB and FTNL to design a private learning algorithm for graphical bandits. The high-level idea behind the design is, instead of running a private UCB over the arm set directly, Learner runs Anytime-Lazy-UCB over a set of cliques (each clique can be treated as a "super arm"), and each clique runs a similar algorithm as FTNL, i.e., we only reveal the information of a single arm. By using this combination, the number of impacted arm can be limited to 1 when we change a single complete reward vector. Note that the changing of a reward vector can only impact one clique and the impacted clique only reveals the information of one particular arm. As we will show in Section 6.4.1, actually, the noise level injected for private bandit settings, private full information settings, and private graphical bandits is the same.

Chapter 2

Background Knowledge

In this chapter, we will provide some background knowledge that help readers to understand this thesis. We will first review the existing algorithms for stochastic bandits and full information game. For stochastic bandits, we will review the Upper confidence bound (UCB) algorithm [5], Thompson Sampling (TS) algorithm [2], and an elimination-style algorithm based on [17, 6]. For full information setting with stochastic rewards, we will review Follow-The-Leader (FTL) algorithm. Next, we will review the key definitions and theorems about differential privacy. Then, we will present an algorithm called Report Noisy Max, which can be viewed as a private algorithm of FTL. At the end of this chapter, we list all the useful inequalities and probability distributions related to this thesis.

2.1 (Pseudo)-Regret

Before we present the learning algorithms in detail, let us define a performance metric to measure the quality of the developed online learning algorithm. Through this chapter, let $\mu_j := \mathbb{E}\left[X_j(t)\right]$ be the mean reward of arm $j \in \mathcal{A}$. We assume that the first arm is the unique best arm, the arm with the highest mean reward. Let $\Delta_j := \mu_1 - \mu_j$ be the mean reward gap between the best arm and a sub-optimal arm j. Intuitively, Δ_j states the performance loss per round when Learner pulls a sub-optimal j instead of pulling the best arm 1.

We use regret $\mathcal{R}(T)$ to measure the quality of the learning algorithms.

Definition 1. *((Pseudo)-Regret). The regret $\mathcal{R}(T)$ is defined as the expected performance loss caused by the fact that Learner may fail to pull the arm with the highest mean reward.*

Mathematically, it is defined as

$$\begin{aligned} \mathcal{R}(T) &= \max_{j\in\mathcal{A}} \mathbb{E}\left[\sum_{t=1}^{T} X_j(t)\right] - \mathbb{E}\left[\sum_{t=1}^{T} X_{J_t}(t)\right] \\ &= T\cdot\mu_1 - \mathbb{E}\left[\sum_{t=1}^{T} X_{J_t}(t)\right] \quad . \end{aligned} \tag{2.1}$$

There are two types of regret bounds. One type of regret bound does not depend on a specific problem instance $(\mu_1, \ldots, \mu_K)$. We call this type of regret bound the *problem-independent* regret bound. From [4], we know that the best problem-independent bound is $O\left(\sqrt{KT}\right)$. Compared to the $O\left(\sqrt{KT}\right)$ regret bound, a more refined regret bound would be the *problem-dependent* regret bound, which depends on a specific problem instance. As we will show in Theorems 1, 2, and 3, the problem-dependent regret bounds for stochastic bandits take the form either $\sum_{j\in\mathcal{A}:\Delta_j>0} O\left(\frac{\log(T)}{\Delta_j}\right)$ or $\sum_{j\in\mathcal{A}:\Delta_j>0} O\left(\frac{\log(T)\Delta_j}{d_{KL}(\mu_1-\Delta_j,\mu_1)}\right)$, where $d_{KL}(x,y)$ indicates the Kullback–Leibler (KL) divergence between Bernoulli distributions with parameters x and y. As we will show in Theorem 4, the regret bound for the full information setting with stochastic rewards takes the $O\left(\frac{\log(K)}{\Delta}\right)$ form, a constant regret that does not grow with T.

2.2 Stochastic MAB Algorithms

We now discuss each learning algorithm one by one.

2.2.1 Upper Confidence Bound (UCB)

Algorithm 1 UCB1 [5]

1: **Input:** Arm set $\mathcal{A}$;
2: **Initialization:** $O_j \leftarrow 0, \quad \widehat{\mu}_{j,O_j} \leftarrow 0$ for all $j \in \mathcal{A}$;
3: **for** $t = 1, 2, \ldots$ **do**
4: Construct $\overline{\mu}_j(t) = \widehat{\mu}_{j,O_j} + \sqrt{\frac{2\ln(t)}{O_j}}$ for all $j \in \mathcal{A}$;
Pull $J_t \leftarrow \arg\max_{j\in\mathcal{A}} \overline{\mu}_j(t)$;
Set $\widehat{\mu}_{J_t,O_{J_t}} \leftarrow \frac{\widehat{\mu}_{J_t,O_{J_t}}\cdot O_{J_t} + X_{J_t}(t)}{O_{J_t}+1}, \quad O_{J_t} \leftarrow O_{J_t} + 1$.
5: **end for**

There are several variants of UCB-based learning algorithms [5, 24, 19], depending on how to construct the confidence intervals. This thesis is based on UCB1 [5].

Let $O_j(t-1) := \sum_{s=1}^{t-1} \mathbf{1}\{J_s = j\}$ be the number of pulls of arm j by the end of round $t-1$. Let $\widehat{\mu}_{j,O_j(t-1)}$ be the empirical mean of these $O_j(t-1)$ observations. The idea behind UCB is to construct a confidence interval around the empirical mean. From Hoeffding's inequality (Inequality 1), we know that $\widehat{\mu}_{j,O_j(t-1)}$ is not too far from μ_j with high probabolity.

In each round t, Learner constructs the upper confidence bound $\overline{\mu}_j(t)$ as

$$\overline{\mu}_j(t) = \widehat{\mu}_{j,O_j(t-1)} + \sqrt{\frac{2\ln(t)}{O_j(t-1)}} \tag{2.2}$$

and pulls the arm with the highest upper confidence bound, i.e., $J_t \leftarrow \arg\max_{j \in \mathcal{A}} \overline{\mu}_j(t)$. After seeing $X_{J_t}(t)$, the number of pulls of arm J_t, and the empirical mean of arm J_t will be updated.

The first term in the RHS of (2.2) contributes to the exploitation while the second term in the RHS of (2.2) contributes to the exploration. By constructing the upper confidence bound in this way, even if an arm has a small empirical mean, it still has the chance to be pulled, as the second term may boost the upper confidence bound of this specific arm. Algorithm 1 presents the UCB learning algorithm in detail.

We now present a problem-dependent regret bound for Algorithm 1.

Theorem 1 (Theorem 1 [6]). *The regret of Algorithm 1 is at most*

$$\mathcal{R}_{UCB1}(T) \leq \sum_{j \in \mathcal{A}: \Delta_j > 0} \frac{8\ln(T)}{\Delta_j} + \left(1 + \frac{\pi^2}{3}\Delta_j\right) \quad .$$

Remark. For the problem-dependent regret bound, there are two terms with one term depending on T. We call the term depending on T the leading term. For the term that does not depend on T, we call it the constant term. In a standard stochastic bandit problem, both the leading term and the constant term are linear in the number of sub-optimal arms.

2.2.2 Thompson Sampling

Algorithm 2 Thompson Sampling over Bernoulli Bandits [2]

1: **Input:** Arm set $\mathcal{A}$;
2: **Initialization:** $O_j \leftarrow 0, \quad Q_j \leftarrow 0$ for all $j \in \mathcal{A}$;
3: **for** $t = 1, 2, \ldots$ **do**
4: Sample $\theta_j(t) \sim \text{Beta}\left(Q_j + 1, O_j - Q_j + 1\right)$;
5: Pull $J_t \leftarrow \arg\max_{j \in \mathcal{A}} \theta_j(t)$;
6: Set $O_{J_t} \leftarrow O_{J_t} + 1, \quad Q_{J_t} \leftarrow Q_{J_t} + X_{J_t}(t)$.
7: **end for**

Thompson Sampling was first introduced in [36] for a 2-armed bandit problem but only very recently, theoretical regret bounds were provided [2, 25]. It has been shown in [12] that Thompson Sampling practically performs extremely well for stochastic bandit problems.

Thompson Sampling is a Bayesian-style heuristic algorithm. The key idea is we maintain a belief about how the problem instance is generated. More specifically, we maintain a belief for the mean reward μ_j of each arm j. For example, in a Bernoulli bandit problem, we believe that each μ_j is generated from a distribution with $[0, 1]$ support. After seeing an observation from arm j, we correct our belief in a Bayesian manner.

The probability density function $\text{Beta}(\alpha, \beta)$ of Beta distribution with parameters α, β is shown in (2.15). Note that $\text{Beta}(1, 1)$ is a uniform distribution with $[0, 1]$ support. As we have no information about how the mean reward is generated before the learning task starts, it is natural to use Beta(1,1) as a prior distribution for the mean reward of arm j. In Thompson Sampling, Learner draws an arm from the posterior probability of being the best arm.

Algorithm 2 presents the Thompson Sampling over Bernoulli rewards in detail. Let $O_j(t-1)$ be the number of pulls of an arm j and $Q_j(t-1)$ be the number of Bernoulli trials that have succeeded by the end of round $t-1$. In each round t, Learner draws a random posterior sample $\theta_j(t)$ from the posterior distribution $\text{Beta}(Q_j(t-1)+1, O_j(t-1) - Q_j(t-1)+1)$, and pulls the arm with the highest $\theta_j(t)$, i.e., $J_t \leftarrow \arg\max_{j \in \mathcal{A}} \theta_j(t)$. After obtaining an observation $X_{J_t}(t)$, we update the posterior distribution of arm J_t based on Bayes' theorem, i.e., if $X_{J_t}(t) = 1$, we increment the first parameter in the Beta distribution of arm J_t by one; if $X_{J_t}(t) = 0$,

we increment the second parameter by one. The trade-off between exploration and exploitation is achieved by these random posterior samples.

We now present a problem-dependent regret bound for Algorithm 2.

Theorem 2 (Theorem 1 [2]). *The regret of Algorithm 2 is at most*

$$\mathcal{R}_{TS}(T) \leq \sum_{j \in \mathcal{A}:\Delta_j > 0} \left(\frac{(1+\epsilon)\ln(T)\Delta_j}{d_{KL}\left(\mu_j, \mu_1\right)} + O\left(\frac{1}{\epsilon^2}\right) \right) \quad ,$$

where ϵ can be any value in $(0, 1]$.

Several remarks are in order for Theorem 2. In the above mentioned regret bound, the constant term hides problem-dependent constants. From Pinsker's inequality, we know $\frac{1}{d_{KL}(\mu_j,\mu_1)} \leq \frac{1}{2\Delta_j^2}$. Therefore, regarding the leading term, Thompson Sampling is better than UCB1, as

$$\begin{aligned} \mathcal{R}_{\mathrm{TS}}(T) &\leq \sum_{j \in \mathcal{A}:\Delta_j > 0} \left(\frac{(1+\epsilon)\ln(T)\Delta_j}{d_{\mathrm{KL}}\left(\mu_j, \mu_1\right)} + O\left(\frac{1}{\epsilon^2}\right) \right) \\ &\leq \sum_{j \in \mathcal{A}:\Delta_j > 0} \left(\frac{(1+\epsilon)\ln(T)}{2\Delta_j} + O\left(\frac{1}{\epsilon^2}\right) \right) \quad . \end{aligned}$$

Thompson Sampling is asymptotically optimal as we have the following regret lower bound: [26]

$$\liminf_{T \to \infty} \frac{\mathcal{R}(T)}{\ln(T)} \geq \sum_{j \in \mathcal{A}:\Delta_j > 0} \frac{\Delta_j}{d_{KL}(\mu_j, \mu_1)} \quad . \tag{2.3}$$

2.2.3 Elimination-Style Algorithm

Recall that both UCB and Thompson Sampling are round-based, i.e., Learner pulls an arm and updates the statistics of the pulled arm in each round. Arm elimination-style learning algorithms are usually epoch-based. Instead of making a decision to pull an arm in each round, Learner maintains an active arm set $V_r \subseteq \mathcal{A}$ in each epoch r and pulls every arm in the active arm set V_r for a certain number of times. At the end of epoch r, Learner eliminates the arms that have performed poorly from the active arm set. We say an arm performs poorly if its upper confidence bound is still smaller than the maximum lower confidence bound. By doing the

elimination, the size of the active arm set shrinks with the progress of the epoch, and eventually, only the best arm will be kept with high probability.

In this section, we present a modified version of the elimination-style algorithms that have been shown in [6, 17]. The key modification is, instead of using all the observations that are collected from the very beginning, in each epoch, we do not reuse any observation obtained from the previous epochs. As we will show in Chapters 4, 5, and 6, the dropping of observations is extremely important to devise differentially private online learning algorithms with good regret bounds.

Algorithm 3 presents the modified version of the elimination-style algorithm in detail. Initially, we set $V_1 \leftarrow \mathcal{A}$. With the progression of epochs, the size of the active arm set shrinks until either only one arm is left in the active arm set or all T rounds have been consumed.

Algorithm 3 Elimination Algorithm

1: **Input:** Arm set $\mathcal{A}$ and T ;
2: **Initialization:** Set $r \leftarrow 1$, active arm set $V \leftarrow \mathcal{A}$;
3: **while** Still have rounds left and $|V| > 1$ **do**
4: Set $L_r := 2\log(KT) \cdot 2^{2r}$;
Pull each arm $j \in V$ for L_r times ;
Eliminate an arm $j \in V$ if rule (2.4) is satisfied ;
Set $r \leftarrow r + 1$;
5: **end while**
6: Pull the single arm left in V until all T rounds are consumed .

In epoch r, we pull each arm in V_r for $L_r := 2\log(KT)2^{2r}$ times. Let $\widehat{\mu}_{j,r}$ be the empirical mean of arm j among these L_r observations in epoch r. We will eliminate an arm $j \in V_r$ at the end of epoch r if arm j's upper confidence bound is smaller than the maximum lower confidence bound among all arms in V_r, i.e., we will eliminate arm j if the following rule is satisfied:

$$\widehat{\mu}_{j,r} + \sqrt{\frac{2\log(KT)}{L_r}} < \max_{i \in V_r} \left(\widehat{\mu}_{i,r} - \sqrt{\frac{2\log(KT)}{L_r}} \right) \quad . \tag{2.4}$$

We now present a problem-dependent regret bound for Algorithm 3.

Theorem 3. *The regret for Algorithm 3 is at most*

$$\mathcal{R}_{AE}(T) \leq \sum_{j \in \mathcal{A}: \Delta_j > 0} O\left(\frac{\log(KT)}{\Delta_j}\right) \quad .$$

Proof sketch of Theorem 3: By using rule (2.4) to decide whether a sub-optimal arm $j \in V_r$ will be eliminated or not, we know that with high probability Learner will eliminate an arm $j \in V_r$ if its mean reward gap Δ_j satisfies $\Delta_j \geq 4 \cdot 0.5^r$ by the end of epoch r. That is also to say, an arm with mean reward Δ_j can be in the active arm set for at most $\log\left(\frac{4}{\Delta_j}\right)$ epochs with high probability. Also, if arm $j \in V_r$, it implies its mean reward gap $\Delta_j < 4 \cdot 0.5^{r-1}$ with high probability (otherwise, it cannot be in V_r). Therefore, the regret caused by pulling a sub-optimal arm j is at most

$$\begin{aligned} & \sum_{r=1}^{\log\left(\frac{4}{\Delta_j}\right)} L_r \cdot \Delta_j \\ \leq & \sum_{r=1}^{\log\left(\frac{4}{\Delta_j}\right)} 2\log(KT) 2^{2r} \cdot 4 \cdot 0.5^{r-1} \\ \leq & \; O\left(\frac{\log(KT)}{\Delta_j}\right) \quad . \end{aligned} \tag{2.5}$$

□

Remark. The elimination-style algorithms that will be presented in Chapter 5 are built on top of Algorithm 3. Typically, the elimination style learning algorithm relies on T as input, i.e., it is not an *anytime* learning algorithm, which is different from UCB and Thompson Sampling-based learning algorithms.

2.3 Full Information Game Algorithm

In this section, we review a simple but optimal learning algorithm, Follow-the-Leader (FTL), for full information setting with stochastic rewards. A nice feature of the full information game is that exploration is not needed. Therefore, we only need to do pure exploitation. We can behave in a greedy way.

Let $\widehat{\mu}_j(t-1)$ be the empirical mean of arm j by the end of round $t-1$. Since in a full information game, the reward vector can be seen in each round, $\widehat{\mu}_j(t-1)$ is

the empirical mean of $t-1$ observations of arm j. FTL simply pulls the arm with the highest empirical mean in each round, i.e., $J_t \leftarrow \arg\max_{j \in \mathcal{A}} \widehat{\mu}_j(t-1)$. At the end of round t, the empirical means of all the arms will be updated.

Algorithm 4 FTL [40]

1: **Input:** Arm set $\mathcal{A}$;
2: **Initialization:** $\widehat{\mu}_j \leftarrow 0$ for all $j \in \mathcal{A}$;
3: **for** $t = 1, 2, \ldots$ **do**
4: Pull $J_t \leftarrow \arg\max_{j \in \mathcal{A}} \widehat{\mu}_j$;
5: Set $\widehat{\mu}_j \leftarrow \frac{\widehat{\mu}_j \cdot (t-1) + X_j(t)}{t}$ for all $j \in \mathcal{A}$.
6: **end for**

We now present a problem-dependent regret bound for Algorithm 4.

Theorem 4. *The regret of Algorithm 4 is at most*

$$\mathcal{R}_{FTL}(T) \leq O\left(\frac{\log(K)}{\Delta_{\min}}\right) \quad ,$$

where $\Delta_{\min} = \min_{j \in \mathcal{A}: \Delta_j > 0} \Delta_j$ *indicates the minimum gap among all sub-optimal arms.*

Remark. The regret bound for a full information setting with stochastic rewards does not depend on T. It only depends on the number of arms. Some differentially private learning algorithms in Sections 4.5.1 and 6.4.1 are built on top of FTL.

2.4 Differential Privacy

We now review the key technical definitions and theorems about differential privacy. As differential privacy was originally used do private data analysis over a database, during the review, we will discuss how to link the ideas for databases to diffferentially private online learning.

Let a database D hold data from individuals. Each row in D contains information associated with a single individual. Let $\mathcal{X}$ be the set of all rows in a database D. Differentially private data analysis ensures that a statistical analysis over a database can be done in an accurate way, and, simultaneously, the privacy of each individual is protected.

The very first definition is the l_1-sensitivity of a function $f : \mathbb{N}^{|\mathcal{X}|} \to \mathbb{R}^k$. A function f can be viewed as a query on a database D.

Definition 2. *(Definition 3.1, l_1-sensitivity [16]). The l_1-sensitivity of a function $f : \mathbb{N}^{|\mathcal{X}|} \to \mathbb{R}^k$ is*

$$\Delta_f = \max_{x,y \in \mathbb{N}^{|\mathcal{X}|}: \|x-y\|_1 = 1} \|f(x) - f(y)\|_1 \quad , \tag{2.6}$$

where $\|x - y\|_1$ measures the number of rows on which $x, y \in \mathbb{N}^{|\mathcal{X}|}$ differ.

For two databases $x, y \in \mathbb{N}^{|\mathcal{X}|}$ such that $\|x - y\|_1 = 1$, we say that databases x, y are neighbouring each other. The l_1-sensitivity of a function f measures the maximum output gap when f works over any two neighbouring databases.

Remark. When linking to differentially private online learning, the information associated with an individual (stored in a row) can be viewed as a reward vector with k entries with each entry in $\{0, 1\}$. For a function $f : \{0, 1\}^k \to \mathbb{R}$ that computes the sum of a vector with k entries, the l_1-sensitivity is 1, as changing the value of a single entry in the vector can impact the sum at most 1.

Definition 3. *(Definition 3.3, Laplace mechanism [16]). Given any function $f : \mathbb{N}^{|\mathcal{X}|} \to \mathbb{R}^k$, the Laplace mechanism is defined as*

$$\mathcal{M}_L(x, f(\cdot), \epsilon) = f(x) + (Y_1, Y_2, \ldots, Y_k) \quad , \tag{2.7}$$

where Y_i are i.i.d. random variables drawn from Lap $\left(\frac{\Delta_f}{\epsilon}\right)$.

In this thesis, we use $X \sim \text{Lap}(b)$ to denote a random variable drawn from Laplace distribution centered at 0 with scale b. The probability density function of a Laplace distribution centered at 0 with scale b is shown in (2.16). All the presented differentially private online learning algorithms in this thesis use the Laplace mechanism to inject noise except the one called Hybrid-UCB that will be presented in Section 4.4.2.

With these preparations, we now present privacy and the accuracy guarantees for the Laplace mechanism.

Theorem 5. *(Theorem 3.6 in [16]). The Laplace mechanism preserves ϵ-differential privacy.*

Theorem 6. *(Theorem 3.8 in [16], accuracy of Laplace mechanism). For any $\delta > 0$, we have*

$$\mathbb{P}\left\{\|\mathcal{M}_L(x, f(\cdot), \epsilon) - f(x)\|_\infty \geq \ln\left(\frac{k}{\delta}\right) \cdot \frac{\Delta_f}{\epsilon}\right\} \leq \delta \quad . \tag{2.8}$$

We now present a nice property for differentially privacy, which is *differential privacy is immune to post-processing*.

Proposition 1. *(Proposition 2.1, Post-Processing [16]). Let $\mathcal{M} : \mathbb{N}^{|\mathcal{X}|} \to R$ be a randomized algorithm that is ϵ-differentially private. Let $f : R \to R'$ be an arbitrary randomized mapping. Then $f \circ \mathcal{M} : \mathbb{N}^{|\mathcal{X}|} \to R'$ is ϵ-differentially private.*

In this thesis, we will use Proposition 1 repeatedly. This proposition motivates the design of differentially private online learning algorithms. Since Learner relies on the differentially private empirical means to make decisions, as long as we ensure that the algorithm that computes the empirical mean is ϵ-differentially private, from Proposition 1, we know that the online learning algorithm is also ϵ-differentially private.

Theorem 7. *(Theorem 3.16, Composition theorem [16]). Let $\mathcal{M}_i : \mathbb{N}^{|\mathcal{X}|} \to \mathcal{R}_i$ be an ϵ_i-differentially private algorithm for $i \in [k]$. Then if $\mathcal{M}_{[k]} : \mathbb{N}^{|\mathcal{X}|} \to \Pi_{i=1}^k \mathcal{R}_i$ is defined to be $\mathcal{M}_{[k]}(x) = (\mathcal{M}_1(x), \ldots, \mathcal{M}_k(x))$, then $\mathcal{M}_{[k]}$ is $\sum_{i=1}^k \epsilon_i$-differentially private.*

To design differentially private online learning algorithms with good theoretical guarantees, we repeatedly use the Post-Processing Proposition, the Composition Theorem, and the Laplace mechanism.

2.5 Report Noisy Max

For Chapters 4 and 6, we also use an algorithm called Report Noisy Max (RNM) [16], which is very related to Follow-the-Leader (FTL) that has been shown in Algorithm 4.

Here, to link with differentially private online learning, we use a different way to explain the learning problem that RNM can solve. Let $M_{m\times n}$ be a matrix with m rows and n columns with the value of each entry $X_{i,j} \in \{0, 1\}$ for all $1 \leq i \leq m, 1 \leq j \leq n$. Matrix $M_{m\times n}$ can be considered as a simple database. We now have a data

analyst who wants to find *which column has the highest sum over all rows, i.e., finding* $\arg\max_{1\leq j\leq n} \sum_{i=1}^{m} X_{i,j}$*, in a differentially private way*. How can the data analyst design a private algorithm to do this?

One of the straightforward algorithms is to query the noisy sum of each column first. Then, the data analyst find the column that has the highest noisy sum. Since a row can affect the sum of all the columns, the l_1-sensitivity of the function $f : \{0,1\}^{m\times n} \to \mathbb{R}^n$ is $\Delta_f = n$. Therefore, from Definition 3, we know that a noise variable drawn from Lap $\left(\frac{n}{\epsilon}\right)$ is needed to be injected to the sum of each column if we use the Laplace mechanism.

Practically, we can have a better private algorithm for this problem, which is the Report Noisy Max (RNM). Just as the name of the algorithm implied, in RNM, only the *index* of the column with the highest noisy sum is returned. The noisy sum of each column remains hidden to the public. In RNM, a noise variable drawn from Lap $\left(\frac{1}{\epsilon}\right)$ is needed.

We now present a privacy guarantee for RNM.

Theorem 8. *(Claim 3.9). RNM is ϵ-differentially private.*

Remark. Several remarks are in order for RNM. If a private online learning would like to use RNM more than once, each time fresh observations (reward vectors) must be fed into RNM. Reusing any observation is not allowed.

If we want to let RNM return both the index of the column with the highest noisy sum and the highest noisy sum itself, we need to inject a noise variable drawn from Lap $\left(\frac{1}{0.5\epsilon}\right)$. One of the intuitive ways to understand why Lap $\left(\frac{1}{0.5\epsilon}\right)$ is needed is, the data analyst can first query the database *which column has the highest noisy sum?* If we inject Lap $\left(\frac{1}{0.5\epsilon}\right)$ noise to each column, the private algorithm answering this query, RNM, is 0.5ϵ-differentially private (from Theorem 8). Then, the data analyst can do a follow-up query, *what is the noisy sum of Column X?* Note that X is the returned column index from the first query. The private algorithm answering the second query is also 0.5ϵ-differentially private, as even if we change a row, the l_1-sensitivity of the algorithm that computes the sum of column X is 1. Therefore, from Definition 3 and Theorem 5, we know that the algorithm answering the second query is 0.5ϵ-differentially private. From Theorem 7, the composition theorem, we know that this modified version of RNM is ϵ-differentially private.

2.6 Useful Facts

In this section, we list all the useful inequalities that are used in this thesis.

Inequality 1. *(Hoeffding's inequality). Let $X_1, X_2, \dots X_n$ be independent random variables with each $X_i \in [0,1]$. Let $\widehat{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$ be the empirical mean of these random variables. For any $\epsilon > 0$, we have*

$$P\left\{\left|\widehat{X} - \mathbb{E}\left[\widehat{X}\right]\right| \geq \epsilon\right\} \leq 2e^{-2n\epsilon^2} \quad . \tag{2.9}$$

Inequality 2. *(Chernoff-Hoeffding theorem). Let $X_1, X_2, \dots X_n$ be independent random variables with each $X_i \in \{0,1\}$. Let $\widehat{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$ be the empirical mean of these random variables. Let $\mu := \mathbb{E}\left[\widehat{X}\right]$. Then, for any $0 < \lambda < 1 - \mu$, we have*

$$\mathbb{P}\left\{\widehat{X} \geq \mu + \lambda\right\} \leq e^{-n \cdot d_{KL}(\mu+\lambda, \mu)} \quad , \tag{2.10}$$

and, for any $0 < \lambda < \mu$, we have

$$\mathbb{P}\left\{\widehat{X} \leq \mu - \lambda\right\} \leq e^{-n \cdot d_{KL}(\mu-\lambda, \mu)} \quad . \tag{2.11}$$

Inequality 3. *Let $Y \sim Lap\,(b)$. For any $\delta > 0$, we have*

$$\mathbb{P}\left\{|Y| \geq \ln\left(\frac{1}{\delta}\right) \cdot b\right\} \leq \delta \quad . \tag{2.12}$$

Inequality 4. *(Lemma 2.8 [11]). Let $Y_1, Y_2, \dots Y_N$ be i.i.d. random variables that are drawn from distribution $Lap\,(b)$. Suppose $Y := \sum_{i=1}^{N} Y_i$. Let $v \geq b\sqrt{N}$ and $0 < \lambda < \frac{2\sqrt{2}v^2}{b}$. Then, we have*

$$\mathbb{P}\left\{Y > \lambda\right\} \leq e^{-\frac{\lambda^2}{8v^2}} \quad . \tag{2.13}$$

Inequality 5. *(Corollary 2.9 in [11]). Let Y_i, v, and b be defined as in Inequality 4. Suppose $0 < \delta < 1$ and $v > b \cdot \max\left\{\sqrt{N}, \sqrt{\ln\left(\frac{2}{\delta}\right)}\right\}$. Then, we have*

$$\mathbb{P}\left\{|Y| > v\sqrt{8\ln\left(\frac{2}{\delta}\right)}\right\} \leq \delta \quad . \tag{2.14}$$

Distribution 1. *(Beta distribution). The probability density function of a Beta distribution with parameters* $\alpha, \beta > 0$ *is*

$$f(x; \alpha, \beta) = \frac{x^{\alpha-1}(1-x)^{\beta-1}}{\int_0^1 y^{\alpha-1}(1-y)^{\beta-1}\, dy} \quad , \tag{2.15}$$

where $x \in [0, 1]$.

Distribution 2. *(Laplace distribution). The Laplace distribution (centered at* 0*) with scale* b *is the distribution with probability density function*

$$Lap(x \mid b) = \frac{1}{2b} e^{-\frac{|x|}{b}} \quad . \tag{2.16}$$

Chapter 3

Problem-dependent Regret Bounds for Online Learning with Feedback Graphs

This chapter addresses the stochastic multi-armed bandit problem with an undirected feedback graph. We devise a UCB-based algorithm UCB-NE and derive a problem-dependent regret bound for UCB-NE that is linear in the size of a clique covering up to logarithmic factors. Also, we provide problem-dependent regret bounds for a Thompson Sampling-based algorithm, TS-N, where again the regret bounds are linear in the size of a clique covering. Finally, we conduct experiments to see how UCB-NE, TS-N, and a few related algorithms perform practically.

3.1 Introduction

In the stochastic multi-armed bandit problem, a learning agent sequentially decides to pull an arm in each of T rounds in order to maximize its cumulative reward. Each arm emits rewards that are i.i.d. according to a fixed but unknown distribution specific to that arm, and in a given round the agent only observes the reward of the arm it pulled in that round. Naturally, the limited feedback aspect of this game creates a tension between exploration — acquiring information to better estimate the mean reward of an arm — and exploitation — pulling the arm that empirically looks the best so far.

The standard notion of regret in this setting is the *(pseudo)-regret* (hereafter re-

ferred to simply as "regret"), which measures the difference between the agent's expected cumulative reward and the expected cumulative reward of the arm with the highest mean reward. For simplicity of this initial exposition, we consider the case of K arms where one arm has a mean reward of μ and all other arms have a mean reward of $\mu - \Delta$ for some $\Delta > 0$. While it is known that a *problem-independent* regret bound of order $O(\sqrt{TK})$ is possible [4], more refined, *problem-dependent* regret bounds that take into account the distributions from which the rewards are generated also exist [5, 19, 2]. These problem-dependent regret bounds grow only logarithmically in T and take the form $O\left(\frac{K\log(T)}{\Delta}\right)$ or $O\left(\frac{K\log(T)\Delta}{d_{KL}(\mu-\Delta,\mu)}\right)$.[1]

A number of recent works have considered the setting of online learning with feedback graphs. This setting can be viewed as an extension of the multi-armed bandit setting where additional *side observations* are available when pulling an arm, as specified by a feedback graph G. When pulling an arm, one receives observations from that arm and all of its neighbors in the feedback graph. A concrete application is an online advertising/promotion system in a social network. A merchant may give a special discount to some selected users to promote their items. The merchant can then observe whether the selected users like the advertised items or not. Meanwhile, the selected users are likely to recommend the advertised items to their friends via social networks. Therefore, the merchant may also get additional observations from the friends of the selected users.

Whereas the regret bounds in the standard multi-armed bandit problem are inherently linear in the number of arms, under the graphical bandit setting it is possible to break this dependence, replacing K by certain graph-theoretic properties. For instance, in the case of undirected feedback graphs, [10] developed an UCB-based algorithm, UCB-N, that replaces K by the clique covering number in the *leading term* of the regret bound (the term depending on T); however, their regret bound still has a constant term (the term not depending on T) that is linear in K. For directed feedback graphs, [14] developed an arm elimination-style algorithm which, remarkably, replaces K by $\alpha(G)\log K$ for the leading term (and also the constant term) in a problem-dependent bound; here, $\alpha(G)$ is the independence number of feedback graph G (where directed edges are counted as undirected edges). However, as we explain in Section 3.5, the additional $\log K$ factor is sometimes unnecessary and the algorithm does not perform well in practice.

[1] $d_{KL}(p,q) = p\log\frac{p}{q} + (1-p)\log\frac{1-p}{1-q}$ is the KL divergence of a Bernoulli distribution with success probability p from a Bernoulli distribution with success probability q.

Thompson Sampling-based algorithms typically perform the best, and this is also the case for the setting of online learning with feedback graphs. Indeed, an algorithm called TS-N (due to [28]) exhibits excellent empirical performance in the case of feedback graphs. However, whereas there are problem-dependent regret bounds for Thompson Sampling in the case of standard bandit feedback [2, 25], no problem-dependent regret bounds have been shown for TS-N in the case of feedback graphs. Existing bounds, due to [28, 29], do depend on the clique covering number or $\alpha(G)$ but are only on the Bayesian regret.

Our core contributions, all for undirected feedback graphs, are as follows:

1. We devise a new UCB-based algorithm, UCB-NE, for the stochastic $\mathcal{N}$-armed bandit problem with an undirected feedback graph. We prove a problem-dependent regret bound for this algorithm which, for any clique covering, is linear in the size of the clique covering and logarithmic in the size of the cliques, both with respect to the leading and constant terms; the precise result can be found in Theorem 9. A nice feature of UCB-NE is that it does not depend on a clique covering as input. Instead, only the degrees of the nodes of the graph are used to construct the upper confidence bounds.[2]

2. For the TS-N algorithm of [28], we give two problem-dependent regret bounds that, similar to UCB-NE, depend only linearly on the size of a clique covering and logarithmically on the size of each clique. These are the first problem-dependent regret bounds for any Thompson Sampling algorithm that improve with properties of feedback graphs. Both bounds involve a free parameter ϵ which allows a trade-off between the leading and constant terms, similar to the previous bounds by [2, 25]. The first bound, Theorem 10, tends to optimize the leading term and hides problem-dependent constants, again similar to the previous regret bounds by [2, 25] in the standard bandit setting. This makes it difficult to assess the trade-off between the leading and constant terms, as is needed to tune ϵ. We therefore present our second regret bound, Theorem 11, that gives an explicit form for the constant term, thereby enabling to suitably tune ϵ. We note that our bounds also hold for the special case of standard bandit feedback, in which case our bounds represent

[2]We note in passing that [10] introduced an algorithm called UCB-MaxN that also attempted to improve the constant term. However, as we explain in Section 3.3, the regret analysis of this algorithm may not always realize such an improvement.

the first fully explicit bounds for Thompson Sampling; previous bounds did not explicitly control the constant term, which in some cases may actually be larger than the leading term.

3. We present experimental results to practically study how the regret grows for UCB-NE, TS-N, UCB-N, the arm elimination-style algorithm of [14], and another algorithm called TS-MaxN [38].

3.2 Stochastic Graphical Bandits

We consider a stochastic $\mathcal{N}$-armed bandit problem with an undirected feedback graph. The learner plays this game for T rounds. At the beginning of round t, the environment generates random rewards in $[0,1]$ for all arms independently[3] from fixed but unknown distributions. Let $X_i(t) \in [0,1]$ be the random reward for arm i at round t.

Graph $\mathcal{G} := (\mathcal{N}, \mathcal{E})$ denotes an undirected feedback graph that captures all the feedback relationships over arm set $\mathcal{N}$. An edge $\{i,j\} \in \mathcal{E}$ means that Learner can get a side observation of arm j when pulling arm i, and vice versa. Note that pulling arm i always lets Learner observe the reward of arm i itself, i.e., $\mathcal{E}$ includes self-loops. We assume that graph $\mathcal{G}$ does not vary over time. For each $i \in \mathcal{N}$, let set $\mathcal{N}_i$ collect arm i and all its neighbors in $\mathcal{G}$. In each round t, the learner pulls an arm $I_t \in \mathcal{N}$. Then, Learner obtains the reward of the pulled arm $X_{I_t}(t)$ and observes the reward of each arm in $\mathcal{N}_{I_t}$. The goal of the learner is to pull arms sequentially to maximize its expected cumulative reward over T rounds.

Let μ_i denote the true mean of arm i's reward. We assume that the first arm is the unique best arm, i.e., $\mu_1 > \mu_i, \forall i \neq 1$. It is possible to modify the analysis if there are multiple best arms. Let $\Delta_i := \mu_1 - \mu_i$ be the mean reward gap between arm i and the best arm. Note that $\Delta_1 = 0$. To measure the quality of our learning algorithms, we use the (pseudo-)regret $\mathcal{R}(T)$, which is defined as

$$\mathcal{R}(T) = \mathbb{E}\left[\sum_{t=1}^{T} \mu_1 - \mu_{I_t}\right] \quad . \tag{3.1}$$

[3] Actually, for UCB-NE, it is not required that the random rewards of all arms be generated independently, i.e., they can be generated from a joint distribution.

In this work, an arbitrary clique covering $\mathcal{C}$ is used to derive our regret bound[4]. $\mathcal{C}$ is a set of cliques such that $\bigcup_{C \in \mathcal{C}} C = \mathcal{N}$ where $C \in \mathcal{C}$ is a clique. A clique in $\mathcal{G}$ is a subset of $\mathcal{N}$ such that all nodes are neighbors with each other. Then, the regret $\mathcal{R}(T)$ can be further expressed as

$$\begin{aligned} \mathcal{R}(T) &= \sum_{i \in \mathcal{N}} \sum_{t=1}^{T} \mathbb{E}\left[\mathbf{1}\{I_t = i\}\right] \cdot \Delta_i \\ &\leq \sum_{C \in \mathcal{C}} \mathbb{E}\left[\underbrace{\sum_{t=1}^{T} \sum_{i \in C} \mathbf{1}\{I_t = i\} \cdot \Delta_i}_{R_C(T)} \right] , \end{aligned} \tag{3.2}$$

where $R_C(T) := \sum_{t=1}^{T} \sum_{i \in C} \mathbf{1}\{I_t = i\} \cdot \Delta_i$ denotes the *intra-clique regret*, i.e., the regret of pulling any sub-optimal arm in clique C. Note that we only need to analyze the cliques that are not equal to $\{1\}$. For any $C \neq \{1\}$, let $\mu_C^{\max} := \max_{i \in C \setminus \{1\}} \mu_i$, $\Delta_C^{\max} := \max_{i \in C \setminus \{1\}} \Delta_i$, and $\Delta_C^{\min} := \min_{i \in C \setminus \{1\}} \Delta_i$.

3.3 Literature

To fully exploit the feedback structure, previous works have used either a *clique covering* $\mathcal{C}$ over all the nodes in $\mathcal{G}$ or the *independence number* $\alpha(\mathcal{G})$ to derive regret bounds. The independence number of a graph is defined as the cardinality of the maximum independent set. The first regret bound of a stochastic $\mathcal{N}$-armed bandit problem with an undirected feedback graph was provided by [10]. The authors devised two UCB-based algorithms: UCB-N and UCB-MaxN. In UCB-N, just like the standard UCB in previous work [5], Learner pulls the arm with the highest upper confidence bound in each round while in UCB-MaxN, Learner first locates the arm with the highest upper confidence bound but actually pulls the arm with the highest empirical mean among the neighbors of the arm with the highest confidence bound. [10] exploited properties of clique coverings to derive problem-dependent regret bounds, i.e., pulling any arm within a clique C allows Learner to obtain an observation of all the arms within C. The leading term for UCB-N is

[4]None of the presented algorithms relies on $\mathcal{C}$ as input. Only the regret analysis needs to use $\mathcal{C}$.

$O\left(\sum_{C\in\mathcal{C}}\frac{\Delta_C^{\max}\ln(T)}{(\Delta_C^{\min})^2}\right)$ while the constant term is $O\left(\sum_{C\in\mathcal{C}}|C|\right) = O(|\mathcal{N}|)$. Note that the algorithm does not need to know the feedback graph in advance for UCB-N. Regarding UCB-MaxN, it seems to be possible to improve the problem-dependent constant term to $O(|\mathcal{C}|)$ asymptotically under an assumption, i.e., that the best sub-optimal arm within each clique is unique and the gap δ between this best sub-optimal arm and the second best sub-optimal arm (within the same clique) is not arbitrarily small. However, as we explain in Appendix 3.7.3, there appears to be a subtle issue with the proof of the regret bound for UCB-MaxN. Our algorithm UCB-NE improves the constant term in their regret bounds by avoiding dependence on δ and provides a regret bound that holds for an arbitrary clique covering. Note that in UCB-NE, the learning algorithm only needs to know the feedback graph instead of the knowledge of clique coverings. Later, [30] exploited the properties of an independent set I to derive a problem-dependent regret bound for UCB-N. Their regret bound is $O\left(\log(T)\log(KT)\max_{I\in\mathcal{I}(\mathcal{G})}\sum_{i\in I}\frac{1}{\Delta_i}\right)$, where $\mathcal{I}(\mathcal{G})$ collects all the independent sets of graph $\mathcal{G}$. Although their regret depends on the size of the independent sets, the leading term has an extra $\log(T)$ factor, which may be sub-optimal.

[14] devised an elimination-based algorithm[5] to exploit a directed feedback graph. Note that an undirected feedback graph can treated as a special directed feedback graph. They gave a problem-dependent regret bound that scales with the independence number $\alpha(\mathcal{G})$. Their regret bound is $O\left(\sum_{v\in V'}\frac{\ln(T)}{\Delta_v}\right)$, where V' is the set of $O\left(\alpha(\mathcal{G})\ln(|\mathcal{N}|)\right)$ arms with the smallest gaps. Although the independence number $\alpha(\mathcal{G})$ is always no greater than the clique covering number, due to the multiplicative interaction with $\ln(|\mathcal{N}|)$, their regret bound may not be always better than one which scales with the clique covering number. Also, although this elimination-based algorithm has a good theoretical guarantee, it does not work well practically as shown by [29] and further confirmed by our experiments in Section 3.5. Additionally, the learning algorithm needs to know the time horizon T in advance. Otherwise, a "doubling trick" shown in [6] may be needed to have an anytime learning algorithm.

[28] and [29] devised a Thompson Sampling-based algorithm, TS-N, to exploit an undirected feedback graph. They gave regret bounds scaling with the

[5] Their algorithm admits regret bounds even if $\mathcal{G}$ varies over time.

clique covering number (an $O\left(\sqrt{|\mathcal{C}|T\ln(|\mathcal{N}|)}\right)$ regret bound) and the independence number (an $O\left(\sqrt{\alpha(\mathcal{G})T\ln(|\mathcal{N}|)}\right)$ regret bound). However, they used the *Bayesian regret* instead of the *pseudo-regret* to measure the quality of the learning algorithms, and their regret bounds are problem-independent. We derive problem-dependent regret bounds for TS-N that depend on a clique covering. Later, [30] derived a problem-dependent bound for TS-N. However, their regret bound is still $O\left(\log(T)\log(KT)\max_{I\in\mathcal{I}(\mathcal{G})}\sum_{i\in I}\frac{1}{\Delta_i}\right)$, which may be sub-optimal in terms of T.

3.4 UCB-NE and TS-N

3.4.1 UCB-NE and Regret Analysis

Algorithm 5 presents the UCB-NE ('E' stands for extra exploration). Let $O_i(t)$ be the number of observations of arm i until the end of round t and $\hat{\mu}_{i,O_i(t)}$ be the empirical mean of arm i until the end of round t.

Let $\bar{\mu}_i(t) := \hat{\mu}_{i,O_i(t-1)} + \sqrt{\frac{2\ln\left(|\mathcal{N}_i|^{\frac{1}{4}}\cdot t\right)}{O_i(t-1)}}$ be the upper confidence bound of arm i at round t. Note that the second term in the upper confidence bound is enlarged as compared to the standard value of $\sqrt{\frac{2\ln(t)}{O_i(t-1)}}$. This enlargement makes the algorithm explore more and, in the regret analysis, enables us to get rid of the factor that makes the constant term scale linearly in the size of the clique. More specifically, the extra exploration allows the constant term from each clique to be divided by something no smaller than the clique size.

In every round t, Learner pulls the arm with the highest upper confidence bound, i.e., $I_t \leftarrow \arg\max_{i\in\mathcal{N}} \bar{\mu}_i(t)$. Then, at the end of round t, all the neighboring arms of the pulled arm including itself, i.e., all $i \in \mathcal{N}_{I_t}$, will be observed and the corresponding $O_i(t)$ and $\hat{\mu}_{i,O_i(t)}$ will be updated.

Although UCB-NE does not depend on a clique covering as input, the algorithm needs the knowledge of graph structure as the degree information for each arm is used to construct the upper confidence bound. Let $N_C := \max_{i\in C}\left\{|\mathcal{N}_i|^{\frac{1}{4}}\right\}$.

Algorithm 5 UCB-NE

Set $O_i \leftarrow 0, \quad \hat{\mu}_{i,O_i} \leftarrow 0, \forall i \in \mathcal{N}$;
for $t = 1, 2, \ldots,$ **do**
  Set $\bar{\mu}_i(t) = \hat{\mu}_{i,O_i} + \sqrt{\frac{2\ln\left(|\mathcal{N}_i|^{\frac{1}{4}} \cdot t\right)}{O_i}}, \forall i \in \mathcal{N}$;
  Pull arm $I_t \leftarrow \arg\max_{i \in \mathcal{N}} \bar{\mu}_i(t)$;
  for $i \in \mathcal{N}_{I_t}$ **do**
    Set $O_i \leftarrow O_i + 1$;
    Observe $X_i(t)$;
    Set $\hat{\mu}_{i,O_i} \leftarrow \frac{\hat{\mu}_{i,O_i} \cdot (O_i - 1) + X_i(t)}{O_i}$.
  end for
end for

Theorem 9. *The regret $\mathcal{R}(T)$ of UCB-NE is at most*

$$\inf_{\mathcal{C}} \left\{ \sum_{C \in \mathcal{C}, C \neq \{1\}} \mathbb{E}\left[R_C(T)\right] \right\} \leq \inf_{\mathcal{C}} \sum_{\substack{C \in \mathcal{C} \\ C \neq \{1\}}} \left(\frac{8\Delta_C^{\max} \ln(N_C \cdot T)}{(\Delta_C^{\min})^2} + \left(1 + \frac{\pi^2}{3}\right) \Delta_C^{\max} \right).$$

Several remarks are in order. First, we discuss the case where no side observations are available, i.e., a standard stochastic multi-armed bandit problem. We can take a trivial clique covering $\mathcal{C} = \{\{i\}, \forall i \in \mathcal{N}\}$ to recover the regret bound of this classic setting. From $\mathcal{C} = \{\{i\}, \forall i \in \mathcal{N}\}$ we have $\Delta_C^{\min} = \Delta_C^{\max} = \Delta_i$ and $N_C = |\mathcal{N}_i| = 1$ for all $C \neq \{1\}$. Then, our regret bound is the same as the one for UCB1 in [5]. Next, we discuss the difference between UCB-N in [5] and UCB-NE if side observations are available. Given the same feedback graph, the leading term of UCB-NE and UCB-N is the same. With respect to the constant term, for each clique C, UCB-N is $O(|C|)$ while UCB-NE improves to $O\left(\frac{\ln(N_C)}{\Delta}\right)$ when the clique size is large. However, when taking the trivial clique covering $\mathcal{C} = \{\{i\}, \forall i \in \mathcal{N}\}$, UCB-N boils down to the same regret bound as UCB1 while UCB-NE needs to pay an additional price of $\frac{2\ln(|\mathcal{N}_i|)}{\Delta_i}$ for each sub-optimal arm i.

Similar to the analysis of UCB-N, to obtain our regret bound, we also bound the total number of times that Learner pulls any sub-optimal arm within each

clique. For each clique C, the regret can be decomposed into two regimes, the under-sampled regime and the sufficiently sampled regime. Specifically, we say that a clique C is in the under-sampled regime if the total number of times that Learner has pulled any arm in C is less than a threshold $L_C := \left\lceil \frac{8 \ln(N_C \cdot T)}{(\Delta_C^{\min})^2} \right\rceil + 1$, where we recall that $N_C = \max_{i \in C} \left\{ |\mathcal{N}_i|^{\frac{1}{4}} \right\}$. For the rounds when clique C is in the under-sampled regime, the total regret is at most $L_C \cdot \Delta_C^{\max}$ while for the rounds when clique C is in the sufficiently sampled regime, we use a concentration inequality to bound the total regret from this regime by a constant not depending on the clique size. Note that the term N_C appearing in L_C typically would not be present in a standard UCB analysis or the analysis of UCB-N. We use this term because, as explained earlier, UCB-NE's upper confidence bounds have an extra exploration term $|\mathcal{N}_i|^{1/4}$ that is upper bounded by N_C.

3.4.2 TS-N and Regret Analysis

Algorithm 6 presents TS-N in detail. Unlike the previous section, $O_i(t)$ denotes the number of times that arm i has been observed until the end of round $t-1$. $Q_i(t)$ denotes the number of times that Learner gets reward equal to 1 among these $O_i(t)$ observations, i.e., the number of times that the Bernoulli trial succeeds until the end of round $t-1$. For each arm $i \in \mathcal{N}$, let $\theta_i(t)$ denote a random value independently generated from posterior distribution $\text{Beta}(Q_i(t)+1, O_i(t)-Q_i(t)+1)$ at round t, where $\text{Beta}(\alpha, \beta)$ denotes a beta distribution with parameter α, β. At the end of round t, all the neighboring arms of the pulled arm including itself will be observed and the parameters of the corresponding beta distributions will be updated. Let $X_i(t) \in \{0,1\}$ be the random reward for arm i at round t. Note that TS-N does not depend on a clique covering as input nor needing the knowledge of the feedback graph.

Let $N_C := \max_{i \in C} |\mathcal{N}_i|$ and, for $a, b \in [0,1]$, let $d(a,b) := a \ln(\frac{a}{b}) + (1-a) \ln(\frac{1-a}{1-b})$ be the Kullback-Leibler (KL) divergence of a Bernoulli distribution with success probability a from a Bernoulli distribution with success probability b.

Algorithm 6 TS-N [28]

Set $O_i \leftarrow 0, Q_i \leftarrow 0, \forall i \in \mathcal{N}$;
for $t = 1, 2, \ldots$ **do**
  Sample $\theta_i(t)$ from Beta$(Q_i + 1, O_i - Q_i + 1), \forall i \in \mathcal{N}$;
  Pull arm $I_t \leftarrow \arg\max_{i \in \mathcal{N}} \theta_i(t)$;
  for $i \in \mathcal{N}_{I_t}$ **do**
    Set $O_i \leftarrow O_i + 1$;
    Observe $X_i(t)$;
    Set $Q_i \leftarrow Q_i + X_i(t)$.
  end for
end for

Theorem 10. *The regret $\mathcal{R}(T)$ of TS-N is at most*

$$\inf_{\mathcal{C}} \left\{ \sum_{C \in \mathcal{C}, C \neq \{1\}} \mathbb{E}\left[R_C(T)\right] \right\}$$
$$\leq \inf_{\mathcal{C}} \left\{ \sum_{C \in \mathcal{C}, C \neq \{1\}} \frac{\Delta_C^{\max}(1+\epsilon_C)\ln(T)}{d(\mu_C^{\max}, \mu_1)} + O\left(\frac{\ln(N_C)+1}{(\epsilon_C)^2}\right) \right\} \quad ,$$

where ϵ_C can be any value in $\left(0, \min\left\{\frac{d(\mu_C^{\max}, \mu_1)}{d(m_C, \mu_1)} - 1, 1\right\}\right)$ and $m_C \in (\mu_C^{\max}, \mu_1)$ is a unique clique-specific problem-dependent constant. The Big-Oh notation in the constant term hides problem-dependent constants.

Let us make a few remarks about this theorem. First, we discuss the case where there is no feedback graph, i.e., a standard stochastic multi-armed bandit problem. We compare our regret bound with Theorem 1 in [2]. We can take a trivial clique covering $\mathcal{C} = \{\{i\}, \forall i \in \mathcal{N}\}$ to represent the case where there is no feedback graph. Then, we have $\mu_C^{\max} = \mu_i$ and $N_C = |\mathcal{N}_i| = 1$ for all $C \neq \{1\}$, and our regret bound boils down to Theorem 1 in [2] with the only difference of the choice of ϵ_C. In [2], they have freedom to choose any $\epsilon_C \in (0, 1)$ while we may not have that freedom. During the proof of our Theorem 10, more precisely, in Lemma 1, we present the range of ϵ_C in our regret bound. We use ϵ_C to control the problem-dependent constant term to make it scale logarithmically with the clique size. It is important to note that ϵ_C does not depend on the number of arms within clique C. Instead, ϵ_C only depends on μ_1 and $\mu_C^{\max}$ (the mean reward of the best sub-optimal arm in clique C). Next, we discuss the difference between TS-N and UCB-NE. With

respect to the leading term, TS-N is better than UCB-NE while for the constant term, TS-N may be worse than UCB-NE. However, the constant terms for TS-N and UCB-NE both scale logarithmically with the clique size instead of linearly.

With respect to the leading term, Theorem 10 provides a good theoretical guarantee while for the constant term, it hides many problem-dependent constants. The hidden terms can be found in the proof. Also, there is a limitation of the choice of ϵ_C for each clique C. Therefore, we provide another theorem for which any $\epsilon \in (0,1)$ is allowed and the constant terms can be expressed explicitly. The exposure of the previously-hidden constant term enables us to achieve a good trade-off between the leading and constant terms by tuning ϵ properly.

Theorem 11. *For any $\epsilon \in (0,1)$, the regret $\mathcal{R}(T)$ of TS-N is at most*

$$\inf_{\mathcal{C}} \left\{ \sum_{C\in\mathcal{C}, C\neq\{1\}} \mathbb{E}\left[R_C(T)\right] \right\}$$
$$\leq \inf_{\mathcal{C}} \left\{ \begin{array}{l} \sum\limits_{C\in\mathcal{C}, C\neq\{1\}} \frac{(3+\lambda_C)^2 \Delta_C^{\max} \ln(T)}{2(1-\epsilon)^2 (\Delta_C^{\min})^2} \\ + \frac{(3+\lambda_C)^2 \Delta_C^{\max} (\ln(N_C)+1)}{2(1-\epsilon)^2 (\Delta_C^{\min})^2} + O\left(\frac{\Delta_C^{\max}}{\epsilon^4 (\Delta_C^{\min})^4}\right) \end{array} \right\} \quad ,$$

where $\lambda_C := \log\left(\frac{\mu_1 - \epsilon\Delta_C^{\min}}{\mu_C^{\max}}\right)$.

In Appendix 3.7.2, we show that instead of paying $O\left(\frac{\Delta_C^{\max}}{\epsilon^4 (\Delta_C^{\min})^4}\right)$, an alternative is to pay $O\left(\frac{\Delta_C^{\max} \ln(T\cdot\epsilon\Delta_C^{\min})}{\epsilon^2 (\Delta_C^{\min})^2}\right)$.

Notation and definitions: Before presenting the analysis, we first introduce some important notation and definitions. Let $T_C(t)$ be the total number of times that Learner pulls any arm in clique C until the end of round $t-1$, i.e., $T_C(t) := \sum_{s=1}^{t-1} \mathbf{1}\{\exists j \in C \text{ s.t. } I_s = j\}$.

Different from UCB-NE, in TS-N, $\hat{\mu}_i(t) = \frac{Q_i(t)}{O_i(t)+1}$ is defined as the empirical mean of arm i at round t. $\mathcal{F}_t$ collects all the history information until the end of round t sequentially, which is $\mathcal{F}_t = \{I_s, X_i(s), \forall i \in \mathcal{N}_{I_s}, s = 1, 2, \cdots, t\}$. Define $\mathcal{F}_0 = \{\}$, and note that $\mathcal{F}_0 \subseteq \mathcal{F}_1 \subseteq \cdots \subseteq \mathcal{F}_{T-1}$ always holds. For each arm i,

note that $O_i(t)$, $Q_i(t)$, and $\hat{\mu}_i(t)$ are determined by $\mathcal{F}_{t-1}$. Also, the distribution that generates $\theta_i(t)$ is determined by $\mathcal{F}_{t-1}$.

To prove Theorem 10, we first do a regret decomposition. L_C is a clique-specific positive integer that will be chosen later, and tuning L_C needs some novel techniques.

$$\begin{aligned} R_C(T) &= \sum_{t=1}^{T} \sum_{i \in C} \mathbf{1}\{I_t = i\} \cdot \Delta_i \\ &= \underbrace{\sum_{t=1}^{T} \sum_{i \in C} \mathbf{1}\{I_t = i, T_C(t) < L_C\} \cdot \Delta_i}_{\leq L_C \cdot \Delta_C^{\max}} \\ &\quad + \underbrace{\sum_{t=1}^{T} \sum_{i \in C} \mathbf{1}\{I_t = i, T_C(t) \geq L_C\} \cdot \Delta_i}_{\Psi} \end{aligned} \tag{3.3}$$

The first term in (3.3) is upper bounded by $L_C \cdot \Delta_C^{\max}$ by bounding the indicator function directly. We show how to choose L_C properly via Lemma 1 and the discussions following it.

Lemma 1. *For clique C, we can always find $x_C \in (\mu_C^{\max}, \mu_1)$, $y_C \in (\mu_C^{\max}, \mu_1)$, and a sufficiently small $0 < \epsilon_C < 1$ such that the following hold simultaneously:*

(i) $\mu_C^{\max} < x_C < y_C < \mu_1$;
(ii) $d(x_C, \mu_1) = \frac{1}{1+\epsilon_C} \cdot d(\mu_C^{\max}, \mu_1)$;
(iii) $d(x_C, y_C) = \frac{1}{1+\epsilon_C} \cdot d(x_C, \mu_1)$;
(iv) $d(x_C, y_C) \geq d(x_C, \mu_C^{\max})$.

After fixing x_C, y_C, and ϵ_C that satisfy all the conditions in Lemma 1, set $L_C := \frac{\ln((N_C)^{\eta_C} \cdot T)}{d(x_C, y_C)} + 2$, where $N_C = \max_{i \in C} |\mathcal{N}_i|$ and $\eta_C := \frac{d(x_C, y_C)}{d(x_C, \mu_C^{\max})} \geq 1$ (condition (iv) in Lemma 1).

Several remarks are in order for Lemma 1 and the choice of L_C. Regarding the choice of x_i, y_i, and ϵ in the standard Thompson Sampling analysis in [2], ϵ can be any value in $(0, 1)$. They chose to fix $\epsilon \in (0, 1)$ first, and then chose $x_i \in (\mu_i, \mu_1)$ such that $d(x_i, \mu_1) = \frac{d(\mu_i, \mu_1)}{1+\epsilon}$ and $y_i \in (x_i, \mu_1)$ such that $d(x_i, y_i) = \frac{d(x_i, \mu_1)}{1+\epsilon}$. However, in this paper, if we exactly reuse the ideas in [2] to choose x_C and y_C, i.e., fixing $\epsilon_C \in (0, 1)$ first and then choosing x_C and y_C only satisfying conditions (i), (ii), and (iii) in Lemma 1, and then set $L_C = \frac{\ln(T)}{d(x_C, y_C)} + 2$, to the best of our knowledge, for each clique C, we can only derive a problem-dependent regret bound

for which the constant term scales with the clique size instead of logarithmically scaling with the clique size. To have a regret bound for which the constant term scales logarithmically with the clique size in a finite time horizon setting, we may sacrifice some freedom of the choice of ϵ_C. However, ϵ_C can always be chosen as small as desired.

The second term Ψ in (3.3) can be further decomposed into Ψ_1, Ψ_2, and Ψ_3 by introducing events $E_C^{\mu}(t) := \left\{ \max_{i \in C \setminus \{1\}} \hat{\mu}_i(t) \leq x_C \right\}$ and $E_C^{\theta}(t) := \left\{ \max_{i \in C \setminus \{1\}} \theta_i(t) \leq y_C \right\}$, which is shown in (3.4).

$$\begin{aligned} \Psi = & \underbrace{\sum_{t=1}^{T} \sum_{i \in C} \mathbf{1}\{I_t = i, \overline{E_C^{\mu}(t)}, T_C(t) \geq L_C\} \cdot \Delta_i}_{\Psi_1} \\ & + \underbrace{\sum_{t=1}^{T} \sum_{i \in C} \mathbf{1}\{I_t = i, E_C^{\mu}(t), \overline{E_C^{\theta}(t)}, T_C(t) \geq L_C\} \cdot \Delta_i}_{\Psi_2} \\ & + \underbrace{\sum_{t=1}^{T} \sum_{i \in C} \mathbf{1}\{I_t = i, E_C^{\mu}(t), E_C^{\theta}(t), T_C(t) \geq L_C\} \cdot \Delta_i}_{\Psi_3} \end{aligned} \tag{3.4}$$

After the aforementioned further regret decomposition, we show that $\mathbb{E}[\Psi_1] \leq \frac{\Delta_C^{\max}}{d(x_C, \mu_C^{\max})}$ (see Lemma 4)[6]. The key step in proving this result (see Lemma 3) is to show that after a fixed arm $i \in C \setminus \{1\}$ has been observed enough times, i.e., $O_i(t) \geq L_C$, it is a rare event that its empirical mean $\hat{\mu}_i(t)$ is greater than x_C. Next, we upper bound $\mathbb{E}[\Psi_2]$ by $\Delta_C^{\max}$, which is accomplished by Lemma 6. This lemma relies on a result, Lemma 5, which states that after a fixed arm $i \in C \setminus \{1\}$ has been observed enough times, i.e., $O_i(t) \geq L_C$, and its empirical mean $\hat{\mu}_i(t)$ is close enough to its true mean, i.e., $\hat{\mu}_i(t) \leq x_C$, it is a rare event that its posterior sampling value $\theta_i(t)$ is greater than y_C. Lemma 5 crucially relies on condition (iv) of Lemma 1, i.e. that $d(x_C, y_C) \geq d(x_C, \mu_C^{\max})$, without which we do not know if it is possible to obtain our desired bound in Lemma 5. This control is important, as Lemma 6 is proved roughly by taking a union bound over all the arms in C, of which there are at most $|C| \leq N_C$. Finally, we show that $\mathbb{E}[\Psi_3]$ is $O(1)$ in the sense that it does not grow with T; here, the Big-Oh notation hides problem-dependent constants. We do this via Lemma 8, which is roughly analogous to Lemmas 2.9 and

[6]Lemmas 3 through 8 are in Appendix 3.7.2.

2.10 of [2]. We mention in passing that Lemma 8 relies on another result, Lemma 7, which is analogous to Lemma 2.8 of [2].

Proof sketch of Theorem 10. As we are analyzing the regret for clique C, for ease of presentation, we drop the subscript C in ϵ_C. Let $\phi_C := \ln\left(\frac{\mu_1(1-\mu_C^{\max})}{\mu_C^{\max}(1-\mu_1)}\right) > 0$, $\Delta_C' := \mu_1 - y_C$, and $D_C := d(y_C, \mu_1)$. Recall conditions (i) to (iv) in Lemma 1 when choosing x_C, y_C, and ϵ. From condition (ii), $d(x_C, \mu_1) = \frac{d(\mu_C^{\max}, \mu_1)}{(1+\epsilon)}$, we have $x_C - \mu_C^{\max} \geq \frac{\epsilon}{\epsilon+1}\frac{d(\mu_C^{\max}, \mu_1)}{\phi_C}$ due to the convexity of function $x \mapsto d(x, \mu_1)$ when $x \in [\mu_C^{\max}, \mu_1]$. Then from Pinsker's inequality we have $\frac{1}{d(x_C, \mu_C^{\max})} \leq \frac{1}{2(x_C - \mu_C^{\max})^2} \leq \frac{(1+\epsilon)^2\phi_C^2}{2\epsilon^2(d(\mu_C^{\max}, \mu_1))^2}$. Putting together condition (ii) and condition (iii), i.e., $d(x_C, y_C) = \frac{d(x_C, \mu_1)}{1+\epsilon}$ and $d(x_C, \mu_1) = \frac{d(\mu_C^{\max}, \mu_1)}{1+\epsilon}$, we have $d(x_C, y_C) = \frac{d(\mu_C^{\max}, \mu_1)}{(1+\epsilon)^2}$.

Now, rewriting $L_C = \frac{\ln(T)}{d(x_C, y_C)} + \frac{\ln(N_C)}{d(x_C, \mu_C^{\max})} + 2$ and by applying $\frac{1}{d(x_C, y_C)} = \frac{(1+\epsilon)^2}{d(\mu_C^{\max}, \mu_1)}$ and $\frac{1}{d(x_C, \mu_C^{\max})} \leq \frac{(1+\epsilon)^2\phi_C^2}{2\epsilon^2(d(\mu_C^{\max}, \mu_1))^2}$ to L_C, we have $L_C \leq \frac{(1+\epsilon)^2\ln(T)}{d(\mu_C^{\max}, \mu_1)} + \frac{(1+\epsilon)^2\phi_C^2\ln(N_C)}{2\epsilon^2(d(\mu_C^{\max}, \mu_1))^2} + 2$.

From (3.3) we have $\mathbb{E}[R_C(T)] \leq L_C \cdot \Delta_C^{\max} + \mathbb{E}[\Psi]$ and by applying Lemmas 4, 6, and 8, and using the above rewrite of L_C,

we further have that $\mathbb{E}[R_C(T)]$ is at most

$$
\begin{aligned}
& L_C \cdot \Delta_C^{\max} + \underbrace{\frac{\Delta_C^{\max}}{d(x_C, \mu_C^{\max})}}_{\text{Lemma 4}} + \underbrace{\Delta_C^{\max}}_{\text{Lemma 6}} \\
& \underbrace{+ \frac{24\Delta_C^{\max}}{\Delta_C'^2} + O\left(\frac{\Delta_C^{\max}}{\Delta_C'^2} + \frac{\Delta_C^{\max}}{\Delta_C' D_C} + \frac{\Delta_C^{\max}}{\Delta_C'^4}\right)}_{\text{Lemma 8}} \\
& \leq \underbrace{\frac{\Delta_C^{\max}(1+\epsilon)^2\ln(T)}{d(\mu_C^{\max}, \mu_1)} + \frac{\Delta_C^{\max}(1+\epsilon)^2\phi_C^2(\ln(N_C)+1)}{2\epsilon^2(d(\mu_C^{\max}, \mu_1))^2}}_{(L_C-2)\cdot\Delta_C^{\max}} \\
& \quad \underbrace{+ 3\Delta_C^{\max} + \frac{24\Delta_C^{\max}}{\Delta_C'^2} + O\left(\frac{\Delta_C^{\max}}{\Delta_C'^2} + \frac{\Delta_C^{\max}}{\Delta_C' D_C} + \frac{\Delta_C^{\max}}{\Delta_C'^4}\right)}_{O(1)} \\
& \leq \frac{\Delta_C^{\max}(1+\epsilon')\ln(T)}{d(\mu_C^{\max}, \mu_1)} + O\left(\frac{\ln(N_C)+1}{\epsilon'^2}\right) + O(1) \quad ,
\end{aligned}
\tag{3.5}
$$

where $\epsilon' = 3\epsilon$ and the Big-Oh notations in the last inequality hide problem-dependent constants. □

Before presenting the proof of Theorem 11, we present a new lemma that gives a novel way to choose x_C and y_C. After fixing x_C and y_C, we prove Theorem 11 by exploiting the properties of the squared Hellinger distance [39] and its link to the KL divergence $d(a,b)$. The squared Hellinger distance between two Bernoulli distributions with success probabilities a and b is defined as $d_H^2(a,b) := (\sqrt{a} - \sqrt{b})^2 + (\sqrt{1-a} - \sqrt{1-b})^2$.

Lemma 2. *For clique C and any $\epsilon \in (0,1)$, we can always find $x_C \in (\mu_C^{\max}, \mu_1)$ and $y_C \in (\mu_C^{\max}, \mu_1)$ such that $\mu_C^{\max} < x_C < y_C < \mu_1$ and $d(x_C, y_C) = d(x_C, \mu_C^{\max})$ hold simultaneously.*

Proof of Lemma 2. Fix $\epsilon \in (0,1)$ and then set $y_C = \mu_1 - \epsilon\Delta_C^{\min}$. Clearly, $y_C \in (\mu_C^{\max}, \mu_1)$ as $\epsilon \in (0,1)$. Then we construct a monotonic function $h(b) = d(b, y_C) - d(b, \mu_C^{\max})$ where $b \in [\mu_C^{\max}, y_C]$. Note that $h(b)$ is strictly decreasing when $b \in [\mu_C^{\max}, y_C]$ since $h'(b) = \ln\left(\frac{\mu_C^{\max}}{y_C}\frac{1-y_C}{1-\mu_C^{\max}}\right) < 0$. Also, we know that $h(\mu_C^{\max}) = d(\mu_C^{\max}, y_C) > 0$ and $h(y_C) = -d(y_C, \mu_C^{\max}) < 0$. Therefore, there exists a unique $m' \in (\mu_C^{\max}, y_C)$ such that $h(m') = d(m', y_C) - d(m', \mu_C^{\max}) = 0$ and $m' = \mu_C^{\max} + \frac{d(\mu_C^{\max}, y_C)(1-\epsilon)\Delta_C^{\min}}{d(\mu_C^{\max}, y_C) + d(y_C, \mu_C^{\max})}$ by using the linearity of the function h. Now, set $x_C = m'$. Note that setting $x_C = m'$ guarantees $\eta_C = \frac{d(x_C, y_C)}{d(x_C, \mu_C^{\max})} = 1$, concluding the proof. □

Proof sketch of Theorem 11: After fixing x_C and y_C that satisfy the conditions shown in Lemma 2, all the proofs of Lemmas 3 through 8 still hold as only Lemma 5 needs to use the condition $\eta_C \geq 1$. Just as when proving Theorem 10, let $L_C = \frac{\ln((N_C)^{\eta_C} \cdot T)}{d(x_C, y_C)} + 2$, where $\eta_C = \frac{d(x_C, y_C)}{d(x_C, \mu_C^{\max})} = 1$. Then we have $\frac{1}{d(x_C, y_C)} = \frac{1}{d(x_C, \mu_C^{\max})} \leq \frac{\left(1 + \frac{d(y_C, \mu_C^{\max})}{d(\mu_C^{\max}, y_C)}\right)^2}{2(1-\epsilon)^2(\Delta_C^{\min})^2} = \frac{\left(1 + \frac{d(\mu_1 - \epsilon\Delta_C^{\min}, \mu_C^{\max})}{d(\mu_C^{\max}, \mu_1 - \epsilon\Delta_C^{\min})}\right)^2}{2(1-\epsilon)^2(\Delta_C^{\min})^2}$ by using Pinsker's inequality.

Let $\zeta_C := \frac{\left(1 + \frac{d(\mu_1 - \epsilon\Delta_C^{\min}, \mu_C^{\max})}{d(\mu_C^{\max}, \mu_1 - \epsilon\Delta_C^{\min})}\right)^2}{2(1-\epsilon)^2}$. Now we upper bound ζ_C. Let $V_C := \frac{\mu_1 - \epsilon\Delta_C^{\min}}{\mu_C^{\max}} > 1$. From Lemma 4 in [42] and the symmetric property of the squared Hellinger distance, we have $d(\mu_1 - \epsilon\Delta_C^{\min}, \mu_C^{\max}) \leq (2 + \log(V_C)) \cdot d_H^2(\mu_1 - \epsilon\Delta_C^{\min}, \mu_C^{\max}) = (2 + \log(V_C)) \cdot d_H^2(\mu_C^{\max}, \mu_1 - \epsilon\Delta_C^{\min}) \leq (2 + \log(V_C)) \cdot d(\mu_C^{\max}, \mu_1 - \epsilon\Delta_C^{\min})$. Then we have $\zeta_C \leq \frac{(3 + \log(V_C))^2}{2(1-\epsilon)^2}$.

Recall that $\Delta_C^{'} = \mu_1 - y_C$ and $D_C = d(y_C, \mu_1)$. By applying $y_C = \mu_1 - \epsilon\Delta_C^{\min}$ to $\Delta_C^{'}$ and D_C, we have $\Delta_C^{'} = \epsilon\Delta_C^{\min}$ and $D_C = d(\mu_1 - \epsilon\Delta_C^{\min}, \mu_1) \leq \epsilon^2(\Delta_C^{\min})^2$. Now, applying L_C, Lemma 4, Lemma 6, and Lemma 8 to (3.5), we have that $\mathbb{E}[R_C(T)]$ is

at most

$$
\begin{aligned}
& L_C \cdot \Delta_C^{\max} + \underbrace{\frac{\Delta_C^{\max}}{d(x_C, \mu_C^{\max})}}_{\textit{Lemma 4}} + \underbrace{\Delta_C^{\max}}_{\textit{Lemma 6}} \\
& + \underbrace{\frac{24\Delta_C^{\max}}{\epsilon^2(\Delta_C^{\min})^2} + O\left(\frac{\Delta_C^{\max}}{\epsilon^4(\Delta_C^{\min})^4}\right)}_{\textit{Lemma 8}} \\
& \leq \frac{(3+\log(V_C))^2\Delta_C^{\max}\ln(N_C \cdot T)}{2(1-\epsilon)^2(\Delta_C^{\min})^2} \\
& \quad + \frac{(3+\log(V_C))^2\Delta_C^{\max}}{2(1-\epsilon)^2(\Delta_C^{\min})^2} + O\left(\frac{\Delta_C^{\max}}{\epsilon^4(\Delta_C^{\min})^4}\right) \quad ,
\end{aligned}
$$

where $V_C = \frac{\mu_1 - \epsilon\Delta_C^{\min}}{\mu_C^{\max}}$. As we explain at the end of the proof of Lemma 8, instead of paying $O\left(\frac{\Delta_C^{\max}}{\epsilon^4(\Delta_C^{\min})^4}\right)$, an alternative is to pay $O\left(\frac{\Delta_C^{\max}\ln(T\cdot\epsilon\Delta_C^{\min})}{(\epsilon)^2(\Delta_C^{\min})^2}\right)$. □

3.5 Experimental Results

We conducted experiments with fixed (i.e. not time-varying) undirected feedback graphs with two equally-sized cliques. The reward for each arm is generated i.i.d. according to a Bernoulli distribution and the rewards of the arms in a given round are independently generated. In the experiment, there is only one optimal arm, which means one clique can include the unique optimal arm while the other clique only contains sub-optimal arms. Also, we set all the sub-optimal arms with the same mean reward (and hence the same gap). We set the gaps for the sub-optimal arms to be the same, i.e., letting $\Delta_C^{\max} = \Delta_C^{\min} =: \Delta$. Therefore, all the other factors that may impact the regret have been removed except for the size of the clique. We vary the size of the cliques. In our experiments, we double the number of arms in each clique to study the effect of clique size on the regret, starting at 2 arms per clique (hence 4 arms total) until we hit 1024 arms per clique (2056 arms total). Each experiment is run for $T = 35,000$ rounds for each run, and we take the average of 100 independent runs.

We compare the performance of UCB-N, UCB-NE, TS-N, the elimination-based algorithm of [14], and an algorithm called TS-MaxN devised by [38]. The rea-

son why we do not compare to UCB-MaxN is that it becomes equivalent to UCB-N for our choice of feedback graphs. Algorithm 7 presents the TS-MaxN algorithm in detail. Compared to TS-N, instead of pulling the arm with the highest posterior sampling value, i.e., $J_t \leftarrow \arg\max_{i \in \mathcal{N}} \theta_i(t)$, in TS-MaxN Learner pulls the arm with the highest empirical mean among all the neighboring arms of J_t, i.e., $I_t \leftarrow \arg\max_{i \in \mathcal{N}_{J_t}} \hat{\mu}_i(t)$. Note that TS-MaxN needs the knowledge of the feedback graph.

As can be seen from our experimental results (Figure 3.2), the elimination-based algorithm does not perform well practically. Also, the regret bound of the elimination-based algorithm is $O\left(|\mathcal{C}|\frac{\ln(|\mathcal{N}|)\cdot\ln(T)}{\Delta}\right)$ while UCB-NE's regret bound is $O\left(|\mathcal{C}|\frac{\ln(T)}{\Delta} + |\mathcal{C}|\frac{\ln(|\mathcal{N}|)}{\Delta}\right)$. Hence, our selected problem instances are the ones for which UCB-NE's theoretical guarantee is better than that of the elimination-based algorithm. Figure 3.1 shows the regret of all the remaining algorithms except for the elimination algorithm. We can see that although the number of arms per clique increases exponentially, the regret grows almost linearly with respect to $\ln(|C|)$ for UCB-NE and TS-N. Also, UCB-N always performs better than UCB-NE, TS-N always performs better than UCB-N and UCB-NE, and TS-MaxN performs better than TS-N.

Algorithm 7 TS-MaxN [38]

Set $O_i \leftarrow 0, Q_i \leftarrow 0, \forall i \in \mathcal{N}$;
for $t = 1, 2, \ldots$ **do**
  Sample $\theta_i(t)$ from Beta$(Q_i + 1, O_i - Q_i + 1)$ distribution for all $i \in \mathcal{N}$;
  Locate arm $J_t \leftarrow \arg\max_{i \in \mathcal{N}} \theta_i(t)$;
  Pull arm $I_t \leftarrow \arg\max_{i \in \mathcal{N}_{J_t}} \hat{\mu}_i(t)$;
  for $i \in \mathcal{N}_{I_t}$ **do**
    Set $O_i \leftarrow O_i + 1$;
    Observe $X_i(t)$;
    Set $Q_i \leftarrow Q_i + X_i(t)$.
  end for
end for

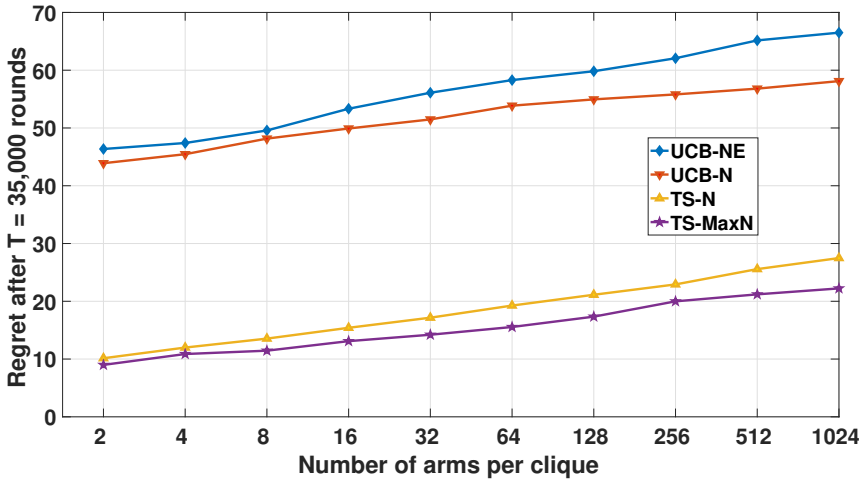

Figure 3.1: Regret for UCB-N, UCB-NE, TS-N, and TS-MaxN with different number of arms per clique

3.6 Conclusion

In this work, we have shown new problem-dependent regret bounds for the stochastic multi-armed bandit problem with feedback graphs. Our UCB-style algorithm, UCB-NE, is the first algorithm of this type that provably obtains regret that is linear in the size of a clique covering rather than linear in the total number of arms. Our regret bounds for the Thompson Sampling-style algorithm TS-N are the first problem-dependent regret bounds for Thompson Sampling that improve with side observations. To ensure that the regret bound is linear in the size of a clique covering rather than linear in the total number of arms, we required important innovations to the previous analysis of [2].

While UCB-NE achieves this by improving the constant term (relative to UCB-N) to $O\left(\sum_{C\in\mathcal{C}} \frac{\Delta_C^{\max} \ln\left(\max_{i\in C} N_i\right)}{(\Delta_C^{\min})^2}\right)$, we still believe that the same constant term can be achieved for UCB-N, i.e., without modifying the way of constructing upper confidence bounds. In Appendix 3.7.4, we provide more discussions about UCB-N.

Regarding the elimination-based algorithm in [14], although the independence number is always no greater than the clique covering number, their regret bound's leading term scales with the worst $O(\alpha(\mathcal{G}) \cdot \ln(|\mathcal{N}|))$ arms. Instead, for regret bounds that depend on clique coverings, for each clique we pay for its worst arm only once. If an undirected feedback graph satisfies $\alpha(\mathcal{G}) = |\mathcal{C}|$, the leading term of

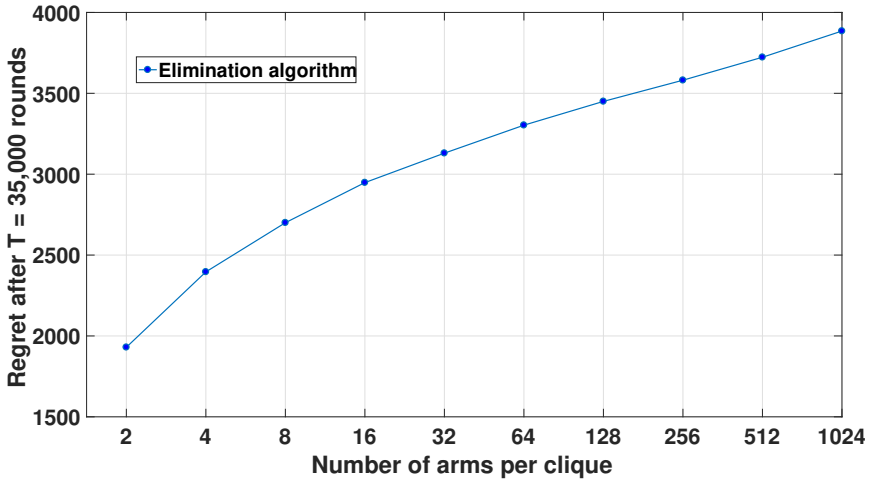

Figure 3.2: Regret for elimination algorithm with different number of arms per clique

the elimination-based algorithm is $O\left(|\mathcal{C}|\frac{\ln(|\mathcal{N}|)\cdot\ln(T)}{\Delta}\right)$ while UCB-NE can achieve $O\left(|\mathcal{C}|\frac{\ln(T)}{\Delta}\right)$.

The same as [38], our experimental results in Figure 3.1 also confirm that TS-MaxN outperforms TS-N practically. Therefore, it is desirable to have a problem-dependent regret bound for TS-MaxN and we also believe that the constant term also scales logarithmically with the clique size.

3.7 Appendix of this Chapter

The organization of this appendix is as follows:

3.7.1 - Proofs of Theorem 9 ;

3.7.2 - Proofs of Theorem 10 ;

3.7.3 - Discussion of UCB-MaxN [10] ;

3.7.4 - Refined Regret Bound of UCB-N ;

3.7.5 - Constant term in Theorem 1.1 of [2] .

3.7.1 Proofs of Theorem 9

Proof of Theorem 9: To derive a regret bound of UCB-NE, let $T_i(t) := \sum_{s=1}^{t} \mathbf{1}\{I_s = i\}$ be the total number of times that arm i has been pulled until the end of round t and $T_C(t)$ be the total number of times that Learner pulls any arm in clique C until the end of round t, i.e., $T_C(t) := \sum_{s=1}^{t} \mathbf{1}\{\exists j \in C \; s.t. \; I_s = j\}$.

Recall that $N_C = \max_{i \in C} \left\{ |\mathcal{N}_i|^{\frac{1}{4}} \right\}$ and $L_C = \left\lceil \frac{8 \ln(N_C \cdot T)}{(\Delta_C^{\min})^2} \right\rceil$.

Then, we have

$$
\begin{aligned}
& \mathbb{E}\left[R_C(T)\right] \\
= & \sum_{i \in C} \sum_{t=1}^{T} \mathbb{E}\left[\mathbf{1}\{I_t = i\}\right] \cdot \Delta_i \\
= & \sum_{i \in C \setminus \{1\}} \sum_{t=1}^{T} \mathbb{E}\left[\mathbf{1}\{I_t = i\}\right] \cdot \Delta_i \\
\leq & \sum_{t=1}^{T} \mathbb{E}\left[\mathbf{1}\{\exists i \in C \setminus \{1\} \; s.t. \; \bar{\mu}_i(t) \geq \bar{\mu}_1(t), T_i(t) > T_i(t-1), T_C(t-1) \leq L_C\}\right] \Delta_C^{\max} \\
+ & \sum_{t=1}^{T} \mathbb{E}\left[\mathbf{1}\{\exists i \in C \setminus \{1\} \; s.t. \; \bar{\mu}_i(t) \geq \bar{\mu}_1(t), T_i(t) > T_i(t-1), T_C(t-1) > L_C\}\right] \Delta_C^{\max} \\
\leq & \; L_C \cdot \Delta_C^{\max} \\
+ & \sum_{t=1}^{T} \underbrace{\mathbb{E}\left[\mathbf{1}\{\exists i \in C \setminus \{1\} \; s.t. \; \bar{\mu}_i(t) \geq \bar{\mu}_1(t), T_i(t) > T_i(t-1), T_C(t-1) > L_C\}\right]}_{(\alpha)} \Delta_C^{\max}.
\end{aligned}
\tag{3.6}
$$

Now, we analyze term (α). We have

$$
\begin{aligned}
(\alpha) &= \mathbb{E}\left[\mathbf{1}\{\exists i \in C\backslash\{1\}\ s.t.\ \bar{\mu}_i(t) \geq \bar{\mu}_1(t), T_i(t) > T_i(t-1), T_C(t-1) > L_C\}\right] \\
&\leq \mathbb{E}\left[\mathbf{1}\left\{\max_{i\in C\backslash\{1\}} \bar{\mu}_i(t) \geq \bar{\mu}_1(t), T_i(t) > T_i(t-1), T_C(t-1) > L_C\right\}\right] \\
&\leq \mathbb{E}\left[\mathbf{1}\left\{\max_{i\in C\backslash\{1\}} \left\{\hat{\mu}_{i,O_i(t-1)} + \sqrt{\frac{2\ln(|\mathcal{N}_i|^{\frac{1}{4}}\cdot t)}{O_i(t-1)}}\right\} \geq \hat{\mu}_{1,O_1(t-1)} + \sqrt{\frac{2\ln(|\mathcal{N}_1|^{\frac{1}{4}}\cdot t)}{O_1(t-1)}},\right.\right. \\
&\qquad \left.\left. T_i(t) > T_i(t-1), T_C(t-1) > L_C\right\}\right] \\
&\leq \sum_{O_C=L_C}^{t-1}\sum_{O_1=1}^{t-1} \mathbb{E}\left[\mathbf{1}\left\{\max_{i\in C\backslash\{1\}} \left\{\hat{\mu}_{i,O_C} + \sqrt{\frac{2\ln(|\mathcal{N}_i|^{\frac{1}{4}}\cdot t)}{O_C}}\right\} \geq \hat{\mu}_{1,O_1} + \sqrt{\frac{2\ln(|\mathcal{N}_1|^{\frac{1}{4}}\cdot t)}{O_1}},\right.\right. \\
&\qquad \left.\left. T_i(t) > T_i(t-1)\right\}\right] \\
&\leq \sum_{O_C=L_C}^{t-1}\sum_{O_1=1}^{t-1} \\
&\qquad \mathbb{E}\left[\mathbf{1}\left\{\underbrace{\max_{i\in C\backslash\{1\}} \left\{\hat{\mu}_{i,O_C} + \sqrt{\frac{2\ln(N_C\cdot t)}{O_C}}\right\} \geq \hat{\mu}_{1,O_1} + \sqrt{\frac{2\ln(|\mathcal{N}_1|^{\frac{1}{4}}\cdot t)}{O_1}}}_{(\beta)}\right\}\right].
\end{aligned}
\tag{3.7}
$$

If (β) holds, it means at least one of the following three inequalities must hold:

$$
\left\{\hat{\mu}_{1,O_1} \leq \mu_1 - \sqrt{\frac{2\ln(|\mathcal{N}_1|^{\frac{1}{4}}\cdot t)}{O_1}}\right\} \quad , \tag{3.8}
$$

$$
\left\{\max_{i\in C\backslash\{1\}} \hat{\mu}_{i,O_C} \geq \mu_C^{\max} + \sqrt{\frac{2\ln(N_C\cdot t)}{O_C}}\right\} \quad , \tag{3.9}
$$

$$
\left\{\mu_1 < \mu_C^{\max} + 2\sqrt{\frac{2\ln(N_C\cdot t)}{O_C}}\right\} \quad . \tag{3.10}
$$

Note that when $O_C \geq \frac{8\ln(N_C\cdot T)}{(\Delta_C^{\min})^2}$, inequality $\left\{\mu_1 < \mu_C^{\max} + 2\sqrt{\frac{2\ln(N_C\cdot t)}{O_C}}\right\}$ cannot

be true. Then, we have

$$
\begin{aligned}
(\alpha) \quad &\leq \sum_{O_C=L_C}^{t-1} \sum_{O_1=1}^{t-1} \\
&\mathbb{E}\left[\mathbf{1}\left\{\max_{i\in C\setminus\{1\}} \hat{\mu}_{i,O_C} \geq \mu_C^{\max} + \sqrt{\frac{2\ln(N_C\cdot t)}{O_C}}\right\} + \mathbf{1}\left\{\hat{\mu}_{1,O_1} \leq \mu_1 - \sqrt{\frac{2\ln(|\mathcal{N}_1|^{\frac{1}{4}}\cdot t)}{O_1}}\right\}\right] \\
= \quad &\sum_{O_C=L_C}^{t-1} \sum_{O_1=1}^{t-1} \\
&\left(\mathbb{P}\left\{\max_{i\in C\setminus\{1\}} \hat{\mu}_{i,O_C} \geq \mu_C^{\max} + \sqrt{\frac{2\ln(N_C\cdot t)}{O_C}}\right\} + \mathbb{P}\left\{\hat{\mu}_{1,O_1} \leq \mu_1 - \sqrt{\frac{2\ln(|\mathcal{N}_1|^{\frac{1}{4}}\cdot t)}{O_1}}\right\}\right).
\end{aligned} \tag{3.11}
$$

By applying Hoeffding's inequality, we have

$$
\begin{aligned}
&\mathbb{P}\left\{\hat{\mu}_{1,O_1} \leq \mu_1 - \sqrt{\frac{2\ln(|\mathcal{N}_1|^{\frac{1}{4}}\cdot t)}{O_1}}\right\} \\
&\leq \tfrac{1}{t^4} \cdot \tfrac{1}{|\mathcal{N}_1|} \leq \tfrac{1}{t^4}
\end{aligned} \tag{3.12}
$$

and

$$
\begin{aligned}
&\mathbb{P}\left\{\max_{i\in C\setminus\{1\}} \hat{\mu}_{i,O_C} \geq \mu_C^{\max} + \sqrt{\frac{2\ln(N_C\cdot t)}{O_C}}\right\} \\
&\leq \sum_{i\in C\setminus\{1\}} \mathbb{P}\left\{\hat{\mu}_{i,O_C} \geq \mu_C^{\max} + \sqrt{\frac{2\ln(N_C\cdot t)}{O_C}}\right\} \\
&\leq \sum_{i\in C\setminus\{1\}} \mathbb{P}\left\{\hat{\mu}_{i,O_C} \geq \mu_i + \sqrt{\frac{2\ln(N_C\cdot t)}{O_C}}\right\} \\
&\leq \tfrac{1}{t^4} \cdot \tfrac{|C|}{(N_C)^4} \leq \tfrac{1}{t^4} \quad .
\end{aligned} \tag{3.13}
$$

By plugging (3.12) and (3.13) into (3.11), we have that $(\alpha) \leq \frac{2}{t^2}$. Then, by plugging the upper bound of term (α) and L_C into $\mathbb{E}\left[R_C(T)\right]$, we have

$$
\mathbb{E}[R_C(T)] \quad \leq \quad \frac{8\ln(N_C\cdot T)}{(\Delta_C^{\min})^2}\Delta_C^{\max} + \left(1+\frac{\pi^2}{3}\right)\Delta_C^{\max} \quad . \tag{3.14}
$$

Then, the regret of UCB-NE is at most

$$
\mathcal{R}(T) \leq \inf_{\mathcal{C}} \left\{\sum_{C\in\mathcal{C}, C\neq\{1\}} \frac{8\Delta_C^{\max}\ln(N_C\cdot T)}{(\Delta_C^{\min})^2} + \left(1+\frac{\pi^2}{3}\right)\Delta_C^{\max}\right\} \quad ,
$$

where $N_C = \max_{i\in C}\left\{|\mathcal{N}_i|^{\frac{1}{4}}\right\}$. □

3.7.2 Proofs of Theorem 10

Proof of Lemma 1:

First, we construct a monotonic function $g(a) = d(a, \mu_1) - 2d(a, \mu_C^{\max})$ where $a \in [\mu_C^{\max}, \mu_1]$. We now claim that $g(a)$ is a strictly decreasing function when $a \in [\mu_C^{\max}, \mu_1]$. It is trivial to prove this claim as $g'(a) = \ln\left(\frac{(\mu_C^{\max})^2}{a \cdot \mu_1} \cdot \frac{(1-a)(1-\mu_1)}{(1-\mu_C^{\max})^2}\right) < 0$ when $a \in [\mu_C^{\max}, \mu_1]$. Also, we know that $g(\mu_C^{\max}) = d(\mu_C^{\max}, \mu_1) > 0$ and $g(\mu_1) = -2d(\mu_1, \mu_C^{\max}) < 0$. There thus exists a unique $m_C \in (\mu_C^{\max}, \mu_1)$ such that $g(m_C) = 0$. Therefore, we have $g(a) \geq 0$ when $a \in (\mu_C^{\max}, m_C]$ while $g(a) < 0$ when $a \in (m_C, \mu_1)$.

Now, we choose x_C such that $x_C \in (\mu_C^{\max}, m_C]$ and $d(x_C, \mu_1) > \frac{1}{2}d(\mu_C^{\max}, \mu_1)$ hold simultaneously. The fact that $x_C \in (\mu_C^{\max}, m_C]$ and $d(x_C, \mu_1) > \frac{1}{2}d(\mu_C^{\max}, \mu_1)$ hold simultaneously implies ϵ can be any value in $\left(0, \min\left\{\frac{d(\mu_C^{\max}, \mu_1)}{d(m_C, \mu_1)} - 1, 1\right\}\right)$. As $x_C \in (\mu_C^{\max}, m_C]$, it means $g(x_C) = d(x_C, \mu_1) - 2d(x_C, \mu_C^{\max}) \geq g(m_C) = 0$, which guarantees $d(x_C, \mu_1) \geq 2d(x_C, \mu_C^{\max})$. Then, we can always find $y_C \in (x_C, \mu_1)$ such that $d(x_C, y_C) = \frac{d(x_C, \mu_1)}{1+\epsilon}$. After fixing x_C and y_C, it is trivial to prove condition (iv) since $d(x_C, y_C) - d(x_C, \mu_C^{\max}) = \frac{d(x_C, \mu_1)}{1+\epsilon} - d(x_C, \mu_C^{\max}) \geq \frac{d(x_C, \mu_C^{\max})}{1+\epsilon} - d(x_C, \mu_C^{\max}) \geq 0$. □

As there is no closed-form expression of m_C, we give a lower bound of m_C. From $g'(a) = \ln\left(\frac{(\mu_C^{\max})^2}{a \cdot \mu_1} \cdot \frac{(1-a)(1-\mu_1)}{(1-\mu_C^{\max})^2}\right)$ we know $g''(a) < 0$. Then, from the concavity of $g(a)$ we have $m_C \geq \mu_C^{\max} + \frac{d(\mu_C^{\max}, \mu_1) \cdot \Delta_C^{\min}}{2d(\mu_1, \mu_C^{\max}) + d(\mu_C^{\max}, \mu_1)}$.

To analyze the first term in (3.4), i.e., term Ψ_1, we prepare two lemmas. Lemma 3 claims that after an arm $i \in C \backslash \{1\}$ has been observed enough times, i.e., $O_i(t) \geq L_C$, it is a rare event that its empirical mean $\hat{\mu}_i(t)$ is greater than x_C. Lemma 4 uses a union bound over all the arms in clique C based on Lemma 3.

Lemma 3. *For $i \in C \backslash \{1\}$, we have*

$$\sum_{t=1}^{T} \mathbb{P}\{\hat{\mu}_i(t) > x_C, O_i(t+1) > O_i(t), O_i(t) \geq L_C\} \leq \frac{1}{N_C} \cdot \frac{1}{d(x_C, \mu_C^{\max})} \quad .$$

Proof of Lemma 3: The proof uses the fact that in each round, we can get at most one observation for any arm. Let τ_k denote the time stamp when the k-th observation of arm i happens. Note that $\mathbf{1}\{O_i(t+1) > O_i(t)\}$ cannot be true during the rounds

when $t \in \{\tau_k + 1, \cdots, \tau_{k+1} - 1\}$ since no new update can be conducted at the end of these rounds.

We have

$$\begin{aligned}
&\sum_{t=1}^{T} \mathbb{P}\{\hat{\mu}_i(t) > x_C, O_i(t+1) > O_i(t), O_i(t) \geq L_C\} \\
&\leq \mathbb{E}\left[\sum_{k=L_C}^{T} \sum_{t=\tau_k}^{\tau_{k+1}-1} \mathbf{1}\{\hat{\mu}_i(t) > x_C, O_i(t+1) > O_i(t)\}\right] \\
&= \mathbb{E}\left[\sum_{k=L_C}^{T} \sum_{t=\tau_k}^{\tau_{k+1}-1} \mathbf{1}\{\hat{\mu}_i(t) > x_C\} \cdot \mathbf{1}\{O_i(t+1) > O_i(t)\}\right] \\
&= \mathbb{E}\left[\sum_{k=L_C}^{T} \mathbf{1}\{\hat{\mu}_i(\tau_k) > x_C\}\right] \quad .
\end{aligned} \tag{3.15}$$

The first inequality in (3.15) uses the fact the number of observations starts from at least L_C and increments to at most T. All the T rounds are segmented into multiple intervals and in each interval, only one observation is obtained except for the last time interval during which zero observation may be obtained. The last equality uses the fact that $\mathbf{1}\{O_i(t+1) > O_i(t)\} = 0$ when $t \in \{\tau_k + 1, \cdots, \tau_{k+1} - 1\}$ and $\mathbf{1}\{O_i(t+1) > O_i(t)\} = 1$ only when $t = \tau_k$.

Then, we can use the definition of $\hat{\mu}_i(\tau_k)$ and further have

$$\begin{aligned}
\mathbb{E}\left[\sum_{k=L_C}^{T} \mathbf{1}\{\hat{\mu}_i(\tau_k) > x_C\}\right] &= \sum_{k=L_C}^{T} \mathbb{P}\{\hat{\mu}_i(\tau_k) > x_C\} \\
&= \sum_{k=L_C}^{T} \mathbb{P}\{\tfrac{Q_i(\tau_k)}{k-1+1} > x_C\} \\
&< \sum_{k=L_C}^{T} \mathbb{P}\{\tfrac{Q_i(\tau_k)}{k-1} > x_C\} \\
&\leq \sum_{k=L_C}^{T} e^{-(k-1)\cdot d(x_C,\mu_i)} \\
&< \tfrac{e^{-(L_C-2)\cdot d(x_C,\mu_i)}}{d(x_C,\mu_i)} \\
&\leq e^{-\ln((N_C)^{\eta_C}\cdot T)\cdot \frac{d(x_C,\mu_C^{\max})}{d(x_C,y_C)}} \cdot \tfrac{1}{d(x_C,\mu_C^{\max})} \\
&= \left(\tfrac{1}{(N_C)^{\eta_C}\cdot T}\right)^{\frac{d(x_C,\mu_C^{\max})}{d(x_C,y_C)}} \cdot \tfrac{1}{d(x_C,\mu_C^{\max})} \\
&= \tfrac{1}{N_C} \cdot \tfrac{1}{d(x_C,\mu_C^{\max})} \cdot \left(\tfrac{1}{T}\right)^{\frac{d(x_C,\mu_C^{\max})}{d(x_C,y_C)}} \\
&\leq \tfrac{1}{N_C} \cdot \tfrac{1}{d(x_C,\mu_C^{\max})} \quad .
\end{aligned} \tag{3.16}$$

The first inequality in (3.16) uses the definition of $\hat{\mu}_i(\tau_k)$, the empirical mean of $k-1$ observations. Note that although τ_k is the time stamp when the k-th observation happens, $O_i(t)$ and $Q_i(t)$ will only be updated at the end of round τ_k. This is why we only have $k-1$ observations at round τ_k. The third inequality uses the Chernoff-Hoeffding bound (Fact 1 in [2]). Note that this lemma does not need to use $\eta_C \geq 1$ during the proof. □

Lemma 4. *For any C, we have*

$$\mathbb{E}\left[\sum_{t=1}^{T} \sum_{i\in C} \mathbf{1}\{I_t = i, \overline{E_C^{\mu}(t)}, T_C(t) \geq L_C\}\right] \cdot \Delta_i \leq \tfrac{\Delta_C^{\max}}{d(x_C,\mu_C^{\max})} \quad .$$

Proof of Lemma 4: For clique C, we have

$$
\begin{aligned}
&\mathbb{E}\left[\sum_{t=1}^{T}\sum_{i\in C}\mathbf{1}\{I_t=i,\overline{E_C^{\mu}(t)},T_C(t)\geq L_C\}\right]\cdot\Delta_i\\
&=\mathbb{E}\left[\sum_{t=1}^{T}\sum_{i\in C\backslash\{1\}}\mathbf{1}\{I_t=i,\overline{E_C^{\mu}(t)},T_C(t)\geq L_C\}\right]\cdot\Delta_i\\
&\leq\mathbb{E}\left[\sum_{t=1}^{T}\mathbf{1}\{\exists i\in C\backslash\{1\}\ s.t.\ I_t=i,\overline{E_C^{\mu}(t)},T_C(t)\geq L_C\}\right]\cdot\Delta_C^{\max}\\
&=\mathbb{E}\left[\sum_{t=1}^{T}\mathbf{1}\{\exists i\in C\backslash\{1\}\ s.t.\ I_t=i,\max_{j\in C\backslash\{1\}}\hat{\mu}_j(t)>x_C,T_C(t)\geq L_C\}\right]\cdot\Delta_C^{\max}\\
&\text{(Use the definition of } E_C^{\mu}(t)=\left\{\max_{j\in C\backslash\{1\}}\hat{\mu}_j(t)\leq x_C\right\}\text{ and then the union bound)}\\
&\leq\sum_{j\in C\backslash\{1\}}\mathbb{E}\left[\sum_{t=1}^{T}\mathbf{1}\{\exists i\in C\backslash\{1\}\ s.t.\ I_t=i,\hat{\mu}_j(t)>x_C,T_C(t)\geq L_C\}\right]\cdot\Delta_C^{\max}\\
&=\sum_{j\in C\backslash\{1\}}\sum_{t=1}^{T}\mathbb{P}\{\exists i\in C\backslash\{1\}\ s.t.\ I_t=i,\hat{\mu}_j(t)>x_C,T_C(t)\geq L_C\}\cdot\Delta_C^{\max}\\
&\text{(Pulling any arm in clique } C \text{ makes arm } j \text{ observed)}\\
&\leq\sum_{j\in C\backslash\{1\}}\underbrace{\sum_{t=1}^{T}\mathbb{P}\{\hat{\mu}_j(t)>x_C,O_j(t+1)>O_j(t),O_j(t)\geq L_C\}}_{\text{Lemma 3}}\cdot\Delta_C^{\max}\\
&\leq\sum_{j\in C\backslash\{1\}}\frac{1}{N_C}\cdot\frac{1}{d(x_C,\mu_C^{\max})}\cdot\Delta_C^{\max}\\
&\leq\frac{\Delta_C^{\max}}{d(x_C,\mu_C^{\max})}\quad.
\end{aligned}
\tag{3.17}
$$

□

To analyze the second term in (3.4), i.e., term Ψ_2, we prepare Lemma 5 and Lemma 6. Lemma 5 claims that after an arm $i\in C\backslash\{1\}$ has been observed enough times, i.e., $O_i(t)\geq L_C$, and its empirical mean $\hat{\mu}_i(t)$ is close enough to its true mean, i.e., $\hat{\mu}_i(t)\leq x_C$, it is a rare event that its posterior sample $\theta_i(t)$ is greater than y_C. Lemma 6 uses a union bound over all the arms within clique C based on Lemma 5.

Lemma 5. *For $i\in C\backslash\{1\}$, we have*

$$
\sum_{t=1}^{T}\mathbb{P}\{\hat{\mu}_i(t)\leq x_C,\theta_i(t)>y_C,O_i(t)\geq L_C\}\leq\frac{1}{N_C}\quad.
$$

Proof of Lemma 5: For any sub-optimal arm $i \in C$, we have

$$\begin{aligned}
&\sum_{t=1}^{T} \mathbb{P}\{\hat{\mu}_i(t) \leq x_C, \theta_i(t) > y_C, O_i(t) \geq L_C\} \\
&= \mathbb{E}\left[\sum_{t=1}^{T} \mathbf{1}\{\hat{\mu}_i(t) \leq x_C, \theta_i(t) > y_C, O_i(t) \geq L_C\}\right] \\
&= \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{E}\left[\mathbf{1}\{\hat{\mu}_i(t) \leq x_C, \theta_i(t) > y_C, O_i(t) \geq L_C\} | \mathcal{F}_{t-1}\right]\right] \\
&= \mathbb{E}[\sum_{t=1}^{T} \underbrace{\mathbf{1}\{\hat{\mu}_i(t) \leq x_C, O_i(t) \geq L_C | \mathcal{F}_{t-1}\}}_{\omega(t)} \cdot \underbrace{\mathbb{P}\{\theta_i(t) > y_C | \mathcal{F}_{t-1}\}}_{v(t)}] \quad .
\end{aligned} \tag{3.18}$$

Let $F^{beta}_{\alpha,\beta}(\cdot)$ denote the CDF of Beta(α, β) and $F^{B}_{n,p}(\cdot)$ denote the CDF of binomial distribution with parameter n, p. We categorize all the instantiations of $\mathcal{F}_{t-1}$ into two types based on whether a specific instantiation F_{t-1} can make the indicator function $\omega(t)$ return 1 or not. Let $\gamma(t) := \omega(t) \cdot v(t)$. In each round t, for the instantiation F_{t-1} of $\mathcal{F}_{t-1}$ that makes $\omega(t) = 0$, we have $\gamma(t) = 0$, while for the instantiation F_{t-1} of $\mathcal{F}_{t-1}$ that makes $\omega(t) = 1$, i.e., both events $\hat{\mu}_i(t) \leq x_C$ and $O_i(t) \geq L_C$ are true, we only need to analyze $v(t) = \mathbb{P}\{\theta_i(t) > y_C | \mathcal{F}_{t-1} = F_{t-1}\}$.

Note that $\theta_i(t)$ is sampled from Beta$(Q_i(t)+1, O_i(t)-Q_i(t)+1)$. Then, we have

$$\begin{aligned}
&\mathbb{P}\{\theta_i(t) > y_C | \mathcal{F}_{t-1} = F_{t-1}\} \\
&= 1 - F^{beta}_{Q_i(t)+1, O_i(t)-Q_i(t)+1}(y_C) \\
&= 1 - F^{beta}_{\hat{\mu}_i(t)(O_i(t)+1)+1, (1-\hat{\mu}_i(t))(O_i(t)+1)}(y_C) \\
&\leq 1 - F^{beta}_{x_C(O_i(t)+1)+1, (1-x_C)(O_i(t)+1)}(y_C) \\
&= F^{B}_{O_i(t)+1, y_C}(x_C(O_i(t)+1)) \\
&\leq e^{-(O_i(t)+1)d(x_C, y_C)} \leq e^{-L_C \cdot d(x_C, y_C)} \\
&\leq e^{-\frac{\ln((N_C)^{\eta_C} \cdot T)}{d(x_C, y_C)} \cdot d(x_C, y_C)} \\
&= \frac{1}{(N_C)^{\eta_C} \cdot T} \\
&\leq \frac{1}{N_C \cdot T} \quad .
\end{aligned} \tag{3.19}$$

The third equality uses $F^{beta}_{\alpha,\beta}(y) = 1 - F^{B}_{\alpha+\beta-1, y}(\alpha - 1)$ (Fact 3 in [2]) and the inequality followed by this equality uses the Chernoff-Hoeffding bound again. The last inequality uses $\eta_C \geq 1$. Note that without the condition $\eta_C \geq 1$, it is not easy to make the clique size scale logarithmically with the clique size. Applying $v(t) = \mathbb{P}\{\theta_i(t) > y_C | \mathcal{F}_{t-1} = F_{t-1}\} \leq \frac{1}{N_C \cdot T}$ to (3.18) concludes the proof. □

Lemma 6. *For any C, we have*

$$\mathbb{E}[\sum_{t=1}^{T} \sum_{i \in C} \mathbf{1}\{I_t = i, E^{\mu}_C(t), \overline{E^{\theta}_C(t)}, T_C(t) \geq L_C\}] \cdot \Delta_i \leq \Delta^{\max}_C \quad .$$

Proof of Lemma 6: For clique C, we have

$$\mathbb{E}\left[\sum_{t=1}^{T}\sum_{i\in C}\mathbf{1}\{I_t=i, E_C^{\mu}(t), \overline{E_C^{\theta}(t)}, T_C(t)\geq L_C\}\right]\cdot\Delta_i$$

$$=\mathbb{E}\left[\sum_{t=1}^{T}\sum_{i\in C\setminus\{1\}}\mathbf{1}\{I_t=i, E_C^{\mu}(t), \overline{E_C^{\theta}(t)}, T_C(t)\geq L_C\}\right]\cdot\Delta_i$$

$$\leq\mathbb{E}\left[\sum_{t=1}^{T}\mathbf{1}\{\exists i\in C\setminus\{1\}\ s.\ t.\ I_t=i, E_C^{\mu}(t), \overline{E_C^{\theta}(t)}, T_C(t)\geq L_C\}\right]\cdot\Delta_C^{\max}$$

(Remove the event $\{\exists i\in C\setminus\{1\}\ s.\ t.\ I_t=i\}$ from the indicator function)

$$\leq\mathbb{E}\left[\sum_{t=1}^{T}\mathbf{1}\{E_C^{\mu}(t), \overline{E_C^{\theta}(t)}, T_C(t)\geq L_C\}\right]\cdot\Delta_C^{\max}$$

$$=\mathbb{E}\left[\sum_{t=1}^{T}\mathbf{1}\{E_C^{\mu}(t), \max_{j\in C\setminus\{1\}}\theta_j(t)>y_C, T_C(t)\geq L_C\}\right]\cdot\Delta_C^{\max}$$

(Use the definition $E_C^{\theta}(t)=\left\{\max_{j\in C\setminus\{1\}}\theta_j(t)\leq y_C\right\}$ and then the union bound)

$$\leq\sum_{j\in C\setminus\{1\}}\mathbb{E}\left[\sum_{t=1}^{T}\mathbf{1}\{E_C^{\mu}(t), \theta_j(t)>y_C, T_C(t)\geq L_C\}\right]\cdot\Delta_C^{\max}$$

(Use the definition $E_C^{\mu}(t):=\{\max_{k\in C\setminus\{1\}}\hat{\mu}_k(t)\leq x_C\}$)

$$=\sum_{j\in C\setminus\{1\}}\mathbb{E}\left[\sum_{t=1}^{T}\mathbf{1}\{\max_{k\in C\setminus\{1\}}\hat{\mu}_k(t)\leq x_C, \theta_j(t)>y_C, T_C(t)\geq L_C\}\right]\cdot\Delta_C^{\max}$$

$$\leq\sum_{j\in C\setminus\{1\}}\mathbb{E}\left[\sum_{t=1}^{T}\mathbf{1}\{\hat{\mu}_j(t)\leq x_C, \theta_j(t)>y_C, T_C(t)\geq L_C\}\right]\cdot\Delta_C^{\max}$$

$$\leq\sum_{j\in C\setminus\{1\}}\mathbb{E}\left[\sum_{t=1}^{T}\mathbf{1}\{\hat{\mu}_j(t)\leq x_C, \theta_j(t)>y_C, O_j(t)\geq L_C\}\right]\cdot\Delta_C^{\max}$$

$$=\sum_{j\in C\setminus\{1\}}\underbrace{\sum_{t=1}^{T}\mathbb{P}\{\hat{\mu}_j(t)\leq x_C, \theta_j(t)>y_C, O_j(t)\geq L_C\}}_{\text{Lemma 5}}\cdot\Delta_C^{\max}$$

$$\leq\frac{|C|}{N_C}\cdot\Delta_C^{\max}$$

$$\leq\Delta_C^{\max}\quad.\tag{3.20}$$

□

To analyze the third term in (3.4), i.e., term Ψ_3, we prepare Lemma 7 and Lemma 8. The key techniques in these two lemmas use the ideas in [2] with slight

modifications. For $C \neq \{1\}$, define $p_{c,t} := \mathbb{P}\{\theta_1(t) > y_C | \mathcal{F}_{t-1}\}$ and recall that $\Delta_C^{'} = \mu_1 - y_C$ and $D_C = d(y_C, \mu_1)$.

Lemma 7. *For $i \in C\backslash\{1\}$, we have*

$$\mathbb{P}\{\exists i \in C\backslash\{1\} \; s.\,t.\; I_t = i, E_C^{\mu}(t), E_C^{\theta}(t) | \mathcal{F}_{t-1}\} \leq \tfrac{1-p_{c,t}}{p_{c,t}} \mathbb{P}\{I_t = 1, E_C^{\mu}(t), E_C^{\theta}(t) | \mathcal{F}_{t-1}\}.$$

Proof of Lemma 7: Recall that $p_{c,t} = \mathbb{P}\{\theta_1(t) > y_C | \mathcal{F}_{t-1}\}$. The proof uses a similar idea as when proving Lemma 2.8 in [2]. The key idea behind the proof is to exploit the feature that $\theta_i(t)$ for all $i \in \mathcal{N}$ are generated independently in each round t and, given $\mathcal{F}_{t-1}$, the distribution that generates $\theta_i(t)$ is determined. Recall that the outcome of event $E_C^{\mu}(t)$ is determined by an instantiation F_{t-1} of $\mathcal{F}_{t-1}$. If the instantiation F_{t-1} is the one that makes event $E_C^{\mu}(t)$ false, it is trivial to prove since both sides in Lemma 7 are 0. If the instantiation F_{t-1} is the one that makes event $E_C^{\mu}(t)$ true, then it suffices to prove that for all such instantiations F_{t-1} we have

$$\begin{aligned} &\underbrace{\mathbb{P}\{\exists i \in C\backslash\{1\} \; s.\,t.\; I_t = i | E_C^{\theta}(t), \mathcal{F}_{t-1} = F_{t-1}\}}_{\omega} \\ &\leq \tfrac{1-p_{c,t}}{p_{c,t}} \cdot \underbrace{\mathbb{P}\{I_t = 1 | E_C^{\theta}(t), \mathcal{F}_{t-1} = F_{t-1}\}}_{\gamma} \quad . \end{aligned} \tag{3.21}$$

For clique C, recall that $E_C^{\theta}(t) = \left\{ \max_{i \in C\backslash\{1\}} \theta_i(t) \leq y_C \right\}$. Now, we analyze term ω in (3.21) and have

$$\begin{aligned} & \omega \\ = \; & \mathbb{P}\{\exists i \in C\backslash\{1\} \; s.\,t.\; I_t = i | E_C^{\theta}(t), \mathcal{F}_{t-1} = F_{t-1}\} \\ \leq \; & \mathbb{P}\{\theta_j(t) \leq y_C, \forall j \in \mathcal{N} | E_C^{\theta}(t), \mathcal{F}_{t-1} = F_{t-1}\} \\ = \; & \mathbb{P}\{\theta_1(t) \leq y_C | E_C^{\theta}(t), \mathcal{F}_{t-1} = F_{t-1}\} \\ & \mathbb{P}\{\theta_j(t) \leq y_C, \forall j \in \mathcal{N}\backslash\{1\} | E_C^{\theta}(t), \mathcal{F}_{t-1} = F_{t-1}\} \\ = \; & \mathbb{P}\{\theta_1(t) \leq y_C | \mathcal{F}_{t-1} = F_{t-1}\} \cdot \mathbb{P}\{\theta_j(t) \leq y_C, \forall j \in \mathcal{N}\backslash\{1\} | E_C^{\theta}(t), \mathcal{F}_{t-1} = F_{t-1}\} \\ = \; & (1 - p_{c,t}) \cdot \underbrace{\mathbb{P}\{\theta_j(t) \leq y_C, \forall j \in \mathcal{N}\backslash\{1\} | E_C^{\theta}(t), \mathcal{F}_{t-1} = F_{t-1}\}}_{\beta} \quad . \end{aligned}$$

Now, we analyze term γ in (3.21) and have

$$
\begin{aligned}
& \gamma \\
= \; & \mathbb{P}\{I_t = 1 | E_C^\theta(t), \mathcal{F}_{t-1} = F_{t-1}\} \\
\geq \; & \mathbb{P}\{\theta_1(t) > y_C \geq \theta_j(t), \forall j \in \mathcal{N}\backslash\{1\} | E_C^\theta(t), \mathcal{F}_{t-1} = F_{t-1}\} \\
= \; & \mathbb{P}\{\theta_1(t) > y_C | E_C^\theta(t), \mathcal{F}_{t-1} = F_{t-1}\} \\
& \mathbb{P}\{\theta_j(t) \leq y_C, \forall j \in \mathcal{N}\backslash\{1\} | E_C^\theta(t), \mathcal{F}_{t-1} = F_{t-1}\} \\
= \; & \mathbb{P}\{\theta_1(t) > y_C | \mathcal{F}_{t-1} = F_{t-1}\} \cdot \mathbb{P}\{\theta_j(t) \leq y_C, \forall j \in \mathcal{N}\backslash\{1\} | E_C^\theta(t), \mathcal{F}_{t-1} = F_{t-1}\} \\
= \; & p_{c,t} \cdot \underbrace{\mathbb{P}\{\theta_j(t) \leq y_C, \forall j \in \mathcal{N}\backslash\{1\} | E_C^\theta(t), \mathcal{F}_{t-1} = F_{t-1}\}}_{\beta} \quad .
\end{aligned}
$$

Then, we have

$$
\begin{aligned}
& \mathbb{P}\{\exists i \in C\backslash\{1\} \; s.t. \; I_t = i | E_C^\theta(t), \mathcal{F}_{t-1} = F_{t-1}\} \cdot \tfrac{1}{1-p_{c,t}} \\
\leq \; & \beta \\
\leq \; & \mathbb{P}\{I_t = 1 | E_C^\theta(t), \mathcal{F}_{t-1} = F_{t-1}\} \cdot \tfrac{1}{p_{c,t}} \quad ,
\end{aligned}
$$

which concludes the proof. □

Lemma 8. *For any C, we have*

$$
\begin{aligned}
& \mathbb{E}[\sum_{t=1}^{T} \sum_{i \in C} \mathbf{1}\{I_t = i, E_C^\mu(t), E_C^\theta(t), T_C(t) \geq L_C\}]\Delta_i \\
\leq \; & \tfrac{24\Delta_C^{\max}}{\Delta_C'^2} + O\left(\tfrac{\Delta_C^{\max}}{\Delta_C'^2} + \tfrac{\Delta_C^{\max}}{\Delta_C' D_C} + \tfrac{\Delta_C^{\max}}{\Delta_C'^4}\right) \quad .
\end{aligned}
$$

Proof of Lemma 8: The proof uses a simple fact which is pulling arm 1 means it must be observed. Also, in each round t, the learner can get at most one observation of arm 1. Let τ_k be the time stamp where arm 1 gets the k-th observation and set $\tau_0 = 0$. Note that $p_{c,t}$ cannot change during the rounds when $t \in \{\tau_k + 1, \cdots, \tau_{k+1}\}$ since the beta distribution that generates $\theta_1(t)$ does not change.

For clique C, we have

$$\mathbb{E}\left[\sum_{t=1}^{T}\sum_{i\in C}\mathbf{1}\{I_t=i,E_C^{\mu}(t),E_C^{\theta}(t),T_C(t)\geq L_C\}\right]\cdot\Delta_i$$

$$=\mathbb{E}\left[\sum_{t=1}^{T}\sum_{i\in C\backslash\{1\}}\mathbf{1}\{I_t=i,E_C^{\mu}(t),E_C^{\theta}(t),T_C(t)\geq L_C\}\right]\cdot\Delta_i$$

(Remove event $\{T_C(t)\geq L_C\}$ from the indicator function)

$$\leq\mathbb{E}\left[\sum_{t=1}^{T}\mathbf{1}\{\exists i\in C\backslash\{1\}\ s.\ t.\ I_t=i,E_C^{\mu}(t),E_C^{\theta}(t)\}\right]\cdot\Delta_C^{\max}$$

$$=\sum_{t=1}^{T}\mathbb{P}\{\exists i\in C\backslash\{1\}\ s.\ t.\ I_t=i,E_C^{\mu}(t),E_C^{\theta}(t)\}\cdot\Delta_C^{\max}$$

$$=\sum_{t=1}^{T}\mathbb{E}\left[\mathbb{P}\{\exists i\in C\backslash\{1\}\ s.\ t.\ I_t=i,E_C^{\mu}(t),E_C^{\theta}(t)|\mathcal{F}_{t-1}\}\right]\cdot\Delta_C^{\max}$$

(By using Lemma 7 we can get the following)

$$\leq\sum_{t=1}^{T}\mathbb{E}\left[\frac{1-p_{c,t}}{p_{c,t}}\mathbb{P}\{I_t=1,E_C^{\mu}(t),E_C^{\theta}(t)|\mathcal{F}_{t-1}\}\right]\cdot\Delta_C^{\max}$$

$$=\sum_{t=1}^{T}\mathbb{E}\left[\frac{1-p_{c,t}}{p_{c,t}}\mathbf{1}\{I_t=1,E_C^{\mu}(t),E_C^{\theta}(t)\}\right]\cdot\Delta_C^{\max}$$

$$\leq\sum_{t=1}^{T}\mathbb{E}\left[\frac{1-p_{c,t}}{p_{c,t}}\mathbf{1}\{O_1(t+1)>O_1(t),E_C^{\mu}(t),E_C^{\theta}(t)\}\right]\cdot\Delta_C^{\max}$$

$$\leq\sum_{k=0}^{T}\sum_{t=\tau_k+1}^{\tau_{k+1}}\mathbb{E}\left[\frac{1-p_{c,t}}{p_{c,t}}\cdot\mathbf{1}\{O_1(t+1)>O_1(t),E_C^{\mu}(t),E_C^{\theta}(t)\}\right]\cdot\Delta_C^{\max}$$

(Use the fact that $\mathbf{1}\{O_1(t+1)>O_1(t)\}=0$ when $t\in\{\tau_k+1,\cdots,\tau_{k+1}-1\}$)

$$\leq\sum_{k=0}^{T}\mathbb{E}\left[\frac{1-p_{c,\tau_{k+1}}}{p_{c,\tau_{k+1}}}\right]\cdot\Delta_C^{\max}$$

(Use the fact that $p_{c,t}$ does not change when $t\in\{\tau_k+1,\cdots,\tau_{k+1}\}$)

$$=\underbrace{\sum_{k=0}^{T}\mathbb{E}\left[\frac{1-p_{c,\tau_k+1}}{p_{c,\tau_k+1}}\right]}\cdot\Delta_C^{\max}$$

(Slightly modify Lemma 2.9 in [2] we can get)

$$\leq\sum_{k=0}^{\frac{8}{\Delta_C'}}\frac{3\Delta_C^{\max}}{\Delta_C'}+\sum_{k\geq\frac{8}{\Delta_C'}}^{T}O\left(\frac{\Delta_C^{\max}}{e^{\Delta_C'^2k/2}}+\frac{\Delta_C^{\max}}{(k+1)\Delta_C'^2e^{kD_C}}+\frac{\Delta_C^{\max}}{e^{\Delta_C'^2k/4}-1}\right)$$

$$\leq\frac{24\Delta_C^{\max}}{\Delta_C'^2}+O\left(\frac{\Delta_C^{\max}}{\Delta_C'^2}+\frac{\Delta_C^{\max}}{\Delta_C'D_C}+\frac{\Delta_C^{\max}}{\Delta_C'^4}\right)\quad.$$

The modification is only at the beginning of the proof of Lemma 2.9 in [2]. More specifically, we only need to modify the following: Let $O_i(t) = j$ and $Q_i(t) = s$. Let $y = y_C$. Then we use $p_{c,t} = \mathbb{P}\{\theta_1(t) > y|\mathcal{F}_{t-1}\}$ instead of their $p_{i,t}$ during the proof. Another modification is to let $\tau_j + 1$ be the time stamp after the j-th *observation* of arm 1 instead of the time stamp after the j-th *pull* of arm 1. For the second term in Big-Oh notation in the last inequality, in [2], it is $\Theta\left(\frac{\Delta_C^{\max}}{\Delta_C'^2 D_C}\right)$ originally but it can be improved to $O\left(\frac{\Delta_C^{\max}}{\Delta_C' D_C}\right)$. Also, an alternative way is to pay $O\left(\frac{\Delta_C^{\max}\ln(T\Delta_C')}{(\Delta_C')^2}\right)$. For the last term in Big-Oh notation, instead of paying $O\left(\frac{\Delta_C^{\max}}{\Delta_C'^4}\right)$, an alternative way is to pay $O\left(\frac{\Delta_C^{\max}\ln(T\Delta_C')}{(\Delta_C')^2}\right)$. □

3.7.3 Discussion of UCB-MaxN

Here, we provide more details about the issues with the proof of the regret bound for UCB-MaxN [10]. In the analysis of Theorem 3 of [10] (the regret bound for UCB-MaxN), one of the steps of the proof appears to be problematic. Specifically, there seems to be an issue with the second inequality below inequality (3) (the first inequality just after "The first summation can be bounded using the Chernoff-Hoeffding inequality as before:"). In our understanding, pulling arm k_C, the best sub-optimal arm within clique C, does not mean its upper confidence bound must be greater than that of the globally best arm. An example is that arm k_C may have a neighboring arm j, not belonging to clique C, which has the highest upper confidence bound while, simultaneously, arm k_C has the highest empirical mean among all the neighbors of arm j. In this example, arm k_C is pulled but its upper confidence bound is not necessarily greater than or equal to that of the globally best arm. It is important to note that arm k_C might be collecting observations from pulls of its neighbors that are not neighbors of j. Therefore, it is possible that the upper confidence bound of arm k_C is no greater than that of arm j while simultaneously, the empirical mean of arm k_C is no smaller than that of arm j.

3.7.4 Refined Regret Bound of UCB-N

In this section, we revisit the learning algorithm of UCB-N [10] and derive the following regret guarantee which is inspired by the comments from Prof Kwang-

Sung Jun. The theorem shown in Theorem 12 addresses the concern if $\Delta_C^{\min}$ is too small.

Theorem 12. *The regret of UCB-N [10] is at most*

$$\mathcal{R}(T) \leq \inf_{\mathcal{C}} \left\{ \sum_{C \in \mathcal{C}, C \neq \{1\}} \frac{8\ln(T)}{\left(\Delta_C^{\min}\right)^2} \cdot \Delta_C^{\max} + \min\left\{ \frac{8\ln(|C|)}{\left(\Delta_C^{\min}\right)^2}, 8|C| \right\} \cdot \Delta_C^{\max} + 8\Delta_C^{\max} \right\}. \tag{3.22}$$

Proof of Theorem 12: For each $C \neq \{1\}$, we set $L_C := \frac{8\ln(|C|\cdot T)}{\left(\Delta_C^{\min}\right)^2} + 1$. Then, we have

$$\begin{aligned}
& \mathbb{E}\left[R_C(T)\right] \\
\leq\ & L_C \cdot \Delta_C^{\max} \\
+\ & \sum_{t=1}^{T} \mathbb{E}\left[\mathbf{1}\{\exists i \in C\backslash\{1\}\ s.t.\ \bar{\mu}_i(t) \geq \bar{\mu}_1(t), T_i(t) > T_i(t-1), T_C(t-1) > L_C\}\right] \Delta_C^{\max} \\
\leq\ & L_C \cdot \Delta_C^{\max} \\
+\ & \sum_{t=1}^{T} \mathbb{E}\left[\mathbf{1}\{\exists i \in C\backslash\{1\}\ s.t.\ \hat{\mu}_{i,O_i(t-1)} + \sqrt{\tfrac{2\ln(t)}{O_i(t-1)}} \geq \hat{\mu}_{1,O_1(t-1)} + \sqrt{\tfrac{2\ln(t)}{O_1(t-1)}},\right. \\
& \left. T_i(t) > T_i(t-1), T_C(t-1) > L_C\}\right] \cdot \Delta_C^{\max} \\
\leq\ & L_C \cdot \Delta_C^{\max} \\
+\ & \sum_{t=1}^{T} \sum_{O_C=L_C}^{t-1} \sum_{O_1=1}^{t-1} \mathbb{E}\left[\mathbf{1}\{\exists i \in C\backslash\{1\}\ s.t.\ \hat{\mu}_{i,O_C} + \sqrt{\tfrac{2\ln(t)}{O_C}} \geq \hat{\mu}_{1,O_1} + \sqrt{\tfrac{2\ln(t)}{O_1}},\right. \\
& \left. T_i(t) > T_i(t-1)\}\right] \cdot \Delta_C^{\max} \\
\leq\ & L_C \cdot \Delta_C^{\max} + \sum_{t=1}^{T} \sum_{O_C=L_C}^{t-1} \sum_{O_1=1}^{t-1} \\
& \mathbb{E}\left[\mathbf{1}\left\{\exists i \in C\backslash\{1\}\ s.t.\ \hat{\mu}_{i,O_C} + \sqrt{\tfrac{2\ln(|C|\cdot t)}{O_C}} \geq \hat{\mu}_{1,O_1} + \sqrt{\tfrac{2\ln(t)}{O_1}}\right\}\right] \cdot \Delta_C^{\max} \quad .
\end{aligned} \tag{3.23}$$

If the indicator function returns 1 in the last step of (3.23), it implies at least one of the following events is true:

$$\max_{i \in C\backslash\{1\}} \hat{\mu}_{i,O_C} \geq \mu_C^{\max} + \sqrt{\frac{2\ln(|C|\cdot t)}{O_C}} \quad , \tag{3.24}$$

$$\hat{\mu}_{1,O_1} \leq \mu_1 - \sqrt{\frac{2\ln(t)}{O_1}} \quad , \tag{3.25}$$

$$\sqrt{\frac{2\ln(|C|\cdot t)}{O_C}} > \frac{\Delta_C^{\min}}{2} \quad . \tag{3.26}$$

We now show $\sqrt{\frac{2\ln(|C|\cdot t)}{O_C}} > \frac{\Delta_C^{\min}}{2}$ cannot be true when $O_C \geq L_C$. It is not hard to see we have

$$\sqrt{\frac{2\ln(|C|\cdot t)}{O_C}} \leq \sqrt{\frac{2\ln(|C|\cdot T)}{O_C}} \leq \sqrt{\frac{2\ln(|C|\cdot T)}{L_C}} < \sqrt{\frac{2\ln(|C|\cdot T)}{8\frac{\ln(|C|\cdot T)}{(\Delta_C^{\min})^2}}} = \frac{\Delta_C^{\min}}{2} \quad , \tag{3.27}$$

which yields a contradiction with (3.26).

We now upper bound the probabilities that events shown in (3.24) and (3.25) happen by using Hoeffding's inequality. We have

$$\begin{aligned}
& \mathbb{P}\left\{\max_{i\in C\setminus\{1\}} \hat{\mu}_{i,O_C} \geq \mu_C^{\max} + \sqrt{\tfrac{2\ln(|C|\cdot t)}{O_C}}\right\} \\
\leq & \sum_{i\in C\setminus\{1\}} \mathbb{P}\left\{\hat{\mu}_{i,O_C} \geq \mu_i + \sqrt{\tfrac{2\ln(|C|\cdot t)}{O_C}}\right\} \\
\leq & \sum_{i\in C\setminus\{1\}} e^{-2O_C\cdot\frac{2\ln(|C|\cdot t)}{O_C}} \\
\leq & \sum_{i\in C\setminus\{1\}} \tfrac{1}{|C|^4\cdot t^4} \\
\leq & \tfrac{1}{t^4} \quad .
\end{aligned} \tag{3.28}$$

Similarly, we also have

$$\mathbb{P}\left\{\hat{\mu}_{1,O_1} \leq \mu_1 - \sqrt{\frac{2\ln(t)}{O_1}}\right\} \leq \frac{1}{t^4} \quad . \tag{3.29}$$

We now come back to (3.23). By plugging in (3.28) and (3.29) into (3.23), we have

$$\begin{aligned}
\mathbb{E}\left[R_C(T)\right] &\leq L_C\cdot\Delta_C^{\max} + \sum_{t=1}^{T} t^2\cdot\tfrac{2}{t^4}\cdot\Delta_C^{\max} \\
&\leq \tfrac{8\ln(T)}{\left(\Delta_C^{\min}\right)^2}\cdot\Delta_C^{\max} + \tfrac{8\ln(|C|)}{\left(\Delta_C^{\min}\right)^2}\cdot\Delta_C^{\max} + 8\Delta_C^{\max} \quad .
\end{aligned} \tag{3.30}$$

Let $L_C := \frac{8\ln(T)}{(\Delta_C^{\min})^2} + 1$. Then, we also have

$$\begin{aligned} & \mathbb{E}\left[R_C(T)\right] \\ \leq\ & L_C \cdot \Delta_C^{\max} \\ +\ & \sum_{t=1}^{T} \sum_{O_C=L_C}^{t-1} \sum_{O_1=1}^{t-1} \mathbb{E}\left[\mathbf{1}\{\exists i \in C\backslash\{1\}\ s.t.\ \hat{\mu}_{i,O_C} + \sqrt{\tfrac{2\ln(t)}{O_C}} \geq \hat{\mu}_{1,O_1} + \sqrt{\tfrac{2\ln(t)}{O_1}}\}\right] \cdot \Delta_C^{\max}. \end{aligned} \tag{3.31}$$

If the indicator function is true, it implies at least one of the following is true:

$$\max_{i\in C\backslash\{1\}} \hat{\mu}_{i,O_C} \geq \mu_C^{\max} + \sqrt{\frac{2\ln(t)}{O_C}} \quad , \tag{3.32}$$

$$\hat{\mu}_{1,O_1} \leq \mu_1 - \sqrt{\frac{2\ln(t)}{O_1}} \quad , \tag{3.33}$$

$$\sqrt{\frac{2\ln(t)}{O_C}} > \frac{\Delta_C^{\min}}{2} \quad . \tag{3.34}$$

When $O_C \geq L_C$, we know the event shown in (3.34) cannot be true as

$$\sqrt{\frac{2\ln(t)}{O_C}} \leq \sqrt{\frac{2\ln(T)}{L_C}} < \sqrt{\frac{2\ln(T)}{\frac{8\ln(T)}{(\Delta_C^{\min})^2}}} = \frac{\Delta_C^{\min}}{2} \quad . \tag{3.35}$$

Now, we analyze the probabilities that events shown in (3.32) and (3.33) happen.

We have

$$\begin{aligned} & \mathbb{P}\left\{\max_{i\in C\backslash\{1\}} \hat{\mu}_{i,O_C} \geq \mu_C^{\max} + \sqrt{\tfrac{2\ln(t)}{O_C}}\right\} \\ \leq\ & \sum_{i\in C\backslash\{1\}} \mathbb{P}\left\{\hat{\mu}_{i,O_C} \geq \mu_i + \sqrt{\tfrac{2\ln(t)}{O_C}}\right\} \\ \leq\ & |C| \cdot \tfrac{1}{t^4} \quad . \end{aligned} \tag{3.36}$$

Similarly, we have

$$\mathbb{P}\left\{\hat{\mu}_{1,O_1} \leq \mu_1 - \sqrt{\frac{2\ln(t)}{O_1}}\right\} \leq \frac{1}{t^4} \quad . \tag{3.37}$$

By plugging in (3.36) and (3.37) into (3.31), we have

$$\begin{aligned}\mathbb{E}\left[R_C(T)\right] &\leq L_C \cdot \Delta_C^{\max} + \sum_{t=1}^{T} \sum_{O_C=L_C}^{t-1} \sum_{O_1=1}^{t-1} 2|C| \tfrac{1}{t^4} \Delta_C^{\max} \\ &\leq \tfrac{8\ln(T)}{\left(\Delta_C^{\min}\right)^2} \cdot \Delta_C^{\max} + 8|C| \cdot \Delta_C^{\max} + 8\Delta_C^{\max} \quad .\end{aligned} \tag{3.38}$$

By combining (3.30) and (3.38), we have

$$\mathbb{E}\left[R_C(T)\right] \leq \frac{8\ln(T)}{\left(\Delta_C^{\min}\right)^2} \cdot \Delta_C^{\max} + \min\left\{\frac{8\ln(|C|)}{\left(\Delta_C^{\min}\right)^2}, 8|C|\right\} \cdot \Delta_C^{\max} + 8 \cdot \Delta_C^{\max} \quad , \tag{3.39}$$

which concludes the proof. □

3.7.5 Constant term in Theorem 1.1 of [2]

In this section, we present an explicit version of the constant term shown in Theorem 1.1 in [2]. We use the same notations as the ones used in [2].

We first construct a function $d_1(x, \mu_1) = x \ln\left(\frac{x}{\mu_1}\right) + (1-x)\ln\left(\frac{1-x}{1-\mu_1}\right)$, where $x \in [\mu_i, \mu_1]$. It is not hard to see that $d_1(x, \mu_1)$ is monotonically decreasing when $x \in [\mu_i, \mu_1]$. Also, we have $d_1(\mu_1, \mu_1) = 0$ and $d_1(x, \mu_1) \geq 0$ when $x \in [\mu_i, \mu_1]$.

Since the first derivative of $d_1(x, \mu_1)$ with respect to x is

$$d_1'(x, \mu_1) = \ln\left(\frac{x \cdot (1-\mu_1)}{\mu_1 \cdot (1-x)}\right) \leq 0 \quad , \tag{3.40}$$

and the second derivative is

$$d_1''(x, \mu_1) = \frac{1}{x} + \frac{1}{1-x} > 0 \quad , \tag{3.41}$$

it is not hard to see that $d_1(x, \mu_1)$ is convex.

For a fixed $\epsilon \in (0, 1)$, we let $x_i \in [\mu_i, \mu_1]$ be the unique value such that

$$d_1(x_i, \mu_1) = \frac{d_1(\mu_i, \mu_1)}{1+\epsilon} \quad . \tag{3.42}$$

From (3.42) and the convexity of $d_1(x, \mu_1)$, we have

$$x_i - \mu_i \geq \frac{\epsilon}{1+\epsilon} \cdot \frac{1}{\ln \frac{\mu_1 \cdot (1-\mu_i)}{\mu_i \cdot (1-\mu_1)}} \cdot d_1(\mu_i, \mu_1) \quad . \tag{3.43}$$

Now, we construct another function $d_2(x_i, x) = x_i \ln\left(\frac{x_i}{x}\right) + (1 - x_i) \ln\left(\frac{1-x_i}{1-x}\right)$, where $x \in [\mu_i, \mu_1]$. Note that $d_2(x_i, x_i) = 0$. Also, when $x \in [\mu_i, x_i]$, function $d_2(x_i, x)$ is monotonically decreasing, and, when $x \in [x_i, \mu_1]$, function $d_2(x_i, x)$ is monotonically increasing.

We let $y_i \in [x_i, \mu_1]$ be the unique value such that

$$d_2(x_i, y_i) = \frac{d_2(x_i, \mu_1)}{1+\epsilon} \quad . \tag{3.44}$$

Note that $d_1(x_i, \mu_1) = d_2(x_i, \mu_1)$ and by combining (3.42) and (3.44), we have

$$d_2(x_i, y_i) = \frac{d_1(\mu_i, \mu_1)}{(1+\epsilon)^2} \quad . \tag{3.45}$$

Now, we lower-bound $d_2(x_i, \mu_i)$ by using Pinsker's inequality. From Pinsker's inequality, we have

$$d_2(x_i, \mu_i) \geq 2(x_i - \mu_i)^2 \quad . \tag{3.46}$$

By using (3.43), we have

$$d_2(x_i, \mu_i) \geq 2(x_i - \mu_i)^2 \geq \frac{2\epsilon^2}{(1+\epsilon)^2} \cdot (d_1(\mu_i, \mu_1))^2 \cdot \left(\frac{1}{\ln \frac{\mu_1 \cdot (1-\mu_i)}{\mu_i \cdot (1-\mu_1)}} \right)^2 \quad . \tag{3.47}$$

Then, we have

$$\frac{1}{d_2(x_i, \mu_i)} \leq \frac{(1+\epsilon)^2}{2\epsilon^2} \cdot \ln^2 \left(\frac{\mu_1 \cdot (1-\mu_i)}{\mu_i \cdot (1-\mu_1)} \right) \cdot \frac{1}{(d_1(\mu_i, \mu_1))^2} = \widetilde{O}\left(\frac{1}{\epsilon^2}\right) \quad , \tag{3.48}$$

where $\widetilde{O}(\cdot)$ notation hides problem-dependent constants.

It is important to note that when μ_i approaching 0 or μ_1 approaching 1, the constant term will blow up. Since the proof of our Theorem 10 uses similar ideas as mentioned above, we also consider the cases that the mean rewards of all the arms are away from 0 and 1.

Chapter 4

Differentially Private Stochastic Online Learning

In this chapter, we consider two variants of differentially private stochastic online learning. The first variant is the differentially private stochastic bandits. Previously, [33] devised the DP Successive Elimination (DP-SE) algorithm that achieves the optimal $O\left(\sum_{1\leq j\leq K:\Delta_j>0} \frac{\log T}{\Delta_j} + \frac{K\log T}{\epsilon}\right)$ problem-dependent regret bound, where K is the number of arms, Δ_j is the mean reward gap of arm j, and ϵ is the required privacy parameter. However, like other elimination style algorithms, it is not an anytime algorithm. Until now, it was not known whether UCB-based algorithms could achieve this optimal regret bound. We present an anytime, UCB-based algorithm, Anytime-Lazy-UCB, that achieves optimality. Our experimental results show that the UCB-based algorithm is competitive with DP-SE. The second variant is the full information version of the differentially private stochastic online learning. Specifically, for the problems of decision-theoretic online learning with stochastic rewards, we present the first algorithm that achieves an $O\left(\frac{\log K}{\Delta_{\min}} + \frac{\log K}{\epsilon}\right)$ regret bound, where $\Delta_{\min}$ is the minimum mean reward gap among all the sub-optimal arms.

4.1 Introduction

In this work, we consider the setting of differentially private online learning with stochastic rewards. In particular, we consider the problems of multi-armed bandits (MAB) and decision-theoretic online learning (DTOL) proposed by [18, 40, 41]. In

both settings, we have a Learner, K actions and a stochastic environment. At the beginning of each round, the environment draws random rewards according to fixed but unknown distributions for all the actions. Simultaneously, Learner chooses one action to play and obtains the reward associated with the played action. Regarding the feedback model after playing an action, in an MAB problem, Learner only observes the reward of the played action. In contrast, in a DTOL problem, Learner can observe the rewards of all the actions. Learner's goal is to accumulate as much reward as possible over T rounds.

In these classical settings, Learner can always use the *true* random rewards obtained in previous rounds to guide it to make a future decision. However, this may not be true in some applications. Take clinical tests for example; some patients may not be willing to let others know that they have participated in certain tests due to privacy concerns. Hence, for the sake of privacy, Learner cannot always use the true results in previous tests to design the tests for the future.

Motivated by this kind of practical application, prior works of [23, 20, 32, 37, 1, 33] have explored the setting of differentially private online learning under the framework of event-level differential privacy that was proposed by [15]. Differential privacy provides a tool to tackle privacy concerns by creating plausible deniability of any possible outcomes from a learning algorithm. Simultaneously, it also guarantees that the output of a learning algorithm is almost as accurate as without implementing differential privacy.

Regarding the differentially private stochastic bandits, [32, 37] devised upper confidence bound (UCB)-based algorithms [5]. However, the regret bounds of their algorithms are sub-optimal. We will provide more discussion about their learning algorithms in Section 4.3. Only very recently, [33] devised DP-SE, an optimal Successive Elimination (SE)-based algorithm [17]. However, just like other elimination-style algorithms, the horizon T needs to be known in advance. Hence, DP-SE cannot be an anytime learning algorithm. Also, in some problem instances, SE-based algorithms do not practically perform well compared to UCB-based algorithms. In this paper, we devise an *anytime*, UCB-based algorithm with the optimal $O\left(\sum_{1\leq j\leq K:\Delta_j>0} \frac{\log T}{\Delta_j} + \frac{K\log T}{\epsilon}\right)$ problem-dependent regret bound, where Δ_j is the mean reward gap for a sub-optimal arm j and ϵ is the privacy parameter.

Regarding the differentially private full information setting, [23, 20, 1] derived regret bounds with adversarial losses and the best known regret bound (in terms

of the problem of prediction with expert advice) is $O\left(\sqrt{T\log(K)} + \frac{K\log(K)\log^2(T)}{\epsilon}\right)$ [1]. Note that the term involving ϵ is at least linear in K. However, for the non-private full information setting with stochastic rewards, the best known result is $O\left(\frac{\log(K)}{\Delta_{\min}}\right)$ [40], where $\Delta_{\min}$ is the minimum mean reward gap among all sub-optimal actions. Note that this regret bound does not depend on T. Therefore, it will be interesting to see what regret bound is possible for the differentially private full information setting with stochastic rewards. In this paper, we devise an algorithm with an $O\left(\frac{\log(K)}{\Delta_{\min}} + \frac{\log(K)}{\epsilon}\right)$ regret bound. Note that the term involving privacy parameter ϵ is logarithmic in K.

Recall that in a non-private stochastic environment, learning algorithms typically rely on each action j's empirical mean (the average value among all observations of action j) to make a decision. However, with the consideration of differential privacy, Learner cannot rely on the true empirical means as doing so will violate privacy. Instead, Learner can make a decision based on each action j's *differentially private empirical mean* which is action j's true empirical mean plus some injected noise.

To achieve our goals to devise algorithms with good theoretical guarantees, we use the ideas of "laziness" and "forgetfulness". At first glance, these ideas seem only to make learning algorithms worse. However, they are the key to devise good algorithms in a stochastic environment with differential privacy. The term "laziness" comes from the idea that we only update the differentially private empirical means occasionally (not too often). The term "forgetfulness" comes from the idea that the differentially private empirical mean is only computed based on a certain amount of newly obtained observations instead of all the observations obtained from the very beginning.

The following summarizes the key contributions.

1. We devise the first, anytime, UCB-based algorithm, Anytime-Lazy-UCB, for private stochastic bandits with the optimal $O\left(\sum_{1\leq j\leq K:\Delta_j>0} \frac{\log T}{\Delta_j} + \frac{K\log T}{\epsilon}\right)$ regret bound.

2. We devise the first algorithm, Follow-the-Noisy-Leader (FTNL), for the private stochastic full information setting. FTNL enjoys an $O\left(\frac{\log(K)}{\Delta_{\min}} + \frac{\log(K)}{\epsilon}\right)$ regret bound. Note that the term involving ϵ is only logarithmic in K.

3. We show that a version of a UCB-based algorithm with the Hybrid mech-

anism of [11] obtains an $O\left(\sum_{1\leq j\leq K:\Delta_j>0} \max\left\{\frac{\log(T)}{\Delta_j}, \frac{\log(T)}{\epsilon}\log\left(\frac{\log(T)}{\epsilon\Delta_j}\right)\right\}\right)$ regret bound (in Section 4.4.2). The same result was previously presented by [37]; we explain in Sections 4.3 and 4.4.2 why we opt to present our own result.

4. We conduct experiments to compare the practical performance of our UCB-based algorithm with DP-SE of [33].

4.2 Learning Problem Settings

4.2.1 Stochastic Online Learning

In the stochastic MAB problem, we have an arm set $\mathcal{A}$ with size K and a stochastic environment. At the beginning of round t, the environment generates a reward vector $X(t) := (X_1(t), X_2(t), \ldots, X_K(t))$, where each $X_j(t) \in [0,1]$ is i.i.d. over time from a fixed but unknown distribution. Simultaneously, Learner pulls an arm $J_t \in \mathcal{A}$. Then, Learner observes and obtains $X_{J_t}(t)$, the reward associated with the pulled arm. The goal of Learner is to accumulate as much reward as possible over T rounds.

In the stochastic DTOL problem, we also have an action set $\mathcal{A}$ with size K and a stochastic environment. Different from the bandit setting where Learner can only observe $X_{J_t}(t)$, in a DTOL problem, Learner can observe the complete reward vector $X(t) = (X_1(t), X_2(t), \ldots, X_K(t))$ in every round. The goal of Learner is still to accumulate as much reward as possible over T rounds.[1]

Let $\mu_j := \mathbb{E}\left[X_j(t)\right]$ be the mean reward of an arm (action) $j \in \mathcal{A}$. Without loss of generality, we assume that $\mu_1 > \mu_j$ for all $j \in \mathcal{A}\backslash\{1\}$. Let $\Delta_j := \mu_1 - \mu_j$ be the mean reward gap for a sub-optimal arm j. Let $O_j(t) := \sum_{s=1}^{t} \mathbf{1}\{J_s = j\}$ be the number of pulls of arm $j \in \mathcal{A}$ by the end of round t.

We use (pseudo)-regret $\mathcal{R}(T)$ to measure the performance of Learner's deci-

[1] DTOL is typically studied under losses. To unify the presentation with stochastic bandits, we use reward vectors.

sions. It can be expressed as

$$\begin{aligned} \mathcal{R}(T) &:= \max_{j\in\mathcal{A}} \mathbb{E}\left[\sum_{t=1}^{T} X_j(t) - \sum_{t=1}^{T} X_{J_t}(t)\right] \\ &= T\cdot\mu_1 - \mathbb{E}\left[\sum_{t=1}^{T}\mu_{J_t}\right] \\ &= \sum_{j\in\mathcal{A}} \mathbb{E}[O_j(T)]\cdot\Delta_j \quad . \end{aligned} \tag{4.1}$$

4.2.2 Differential Privacy

In this work, we consider differentially private online learning under the *event-level* framework of [15]. Formally, we say two sequences of reward vectors $X_{1:T}$ and $X'_{1:T}$ are neighbours if they differ in at most one reward vector. If a reward vector encodes information associated with an individual, a differentially private online learning algorithm guarantees that even if an external observer sees the output of the learning algorithm, e.g., the pulled arm sequence, it is very unlikely to infer whether the learning algorithm takes $X_{1:T}$ or $X'_{1:T}$ as input. That is also to say, the information of a single individual cannot impact the output of a private learning algorithm too much. The definition below is the same as the one used in [32, 37, 1, 33].

Definition 4 (Differential Privacy)**.** *An algorithm Π is ϵ-differentially private if for any two neighbouring reward sequences $X_{1:T}$ and $X'_{1:T}$, for any set $\mathcal{D}$ of decisions made from round 1 to T, it holds that $\mathbb{P}\{\Pi(X_{1:T})\in\mathcal{D}\} \le e^{\epsilon}\cdot\mathbb{P}\{\Pi(X'_{1:T})\in\mathcal{D}\}$.*

4.3 Literature

[32] devised the first differentially private UCB algorithm for stochastic bandits by using the Tree-based aggregation mechanism of [15, 11]. In their work, for arm j, a complete binary tree with T leaf nodes is used to track arm j's differentially private empirical mean. Then, Learner constructs upper confidence bounds based on each arm's differentially private empirical mean to decide which arm to pull. In arm j's binary tree, each leaf node holds an obtained observation of arm j. The usage of a tree with T leaf nodes results in a sub-optimal regret bound as the l_1-sensitivity of the tree-based aggregation mechanism with T leaf nodes can be $\log(T)$. Also, since

a complete binary tree with T leaf nodes is maintained, their algorithm cannot be an anytime algorithm.

[37] applied the Hybrid mechanism of [11] for private stochastic bandits. They claimed an $O\left(\frac{K\log(T)}{\Delta}+\frac{K\log(T)}{\epsilon}\log\left(\frac{\log(T)}{\epsilon\cdot\Delta}\right)\right)$ regret bound by using the Hybrid mechanism in UCB. However, we are uncertain about some parts of their analysis (some concerns have also been raised by [33]). Hence, for completeness, we provide our own regret bound for another Hybrid mechanism-based UCB. This bound, appearing in Section 4.4.2, is the same as the stated bound of [37]; Also, in Section 4.4.2, we provide more details about why we present our own result.

[33] devised DP-SE, an optimal differentially private algorithm for stochastic bandits, based on the Successive Elimination (SE) algorithm of [17]. Their $O\left(\frac{K\log T}{\Delta}+\frac{K\log T}{\epsilon}\right)$ regret bound matches the $\Omega\left(\frac{K\log T}{\Delta}+\frac{K\log T}{\epsilon}\right)$ lower bound of [34]. Since DP-SE is built on top an elimination algorithm, DP-SE progresses in epochs and at the end of each epoch, the arms with smaller differentially private empirical means are eliminated. In DP-SE, the number of pulls of each non-eliminated arm in each epoch is carefully designed, which depends on both T and ϵ. Therefore, DP-SE cannot be an anytime algorithm. Also, in some problem instances, DP-SE does not practically perform well compared to algorithms in the UCB family. Our algorithm, Anytime-Lazy-UCB, is the first optimal algorithm in the UCB family. Also, it preserves the typical property that UCB has: the anytime property. The experimental results in Section 4.6 confirm that Anytime-Lazy-UCB is competitive with DP-SE.

For the private stochastic full information setting, until now it was not known whether any algorithm could achieve an $O\left(\frac{\log(K)}{\Delta}+\frac{\log(K)}{\epsilon}\right)$ regret bound. As mentioned in Section 4.1, the best known result is $O\left(\sqrt{T\log(K)}+\frac{K\log(K)\log^2(T)}{\epsilon}\right)$ in the problem of prediction with expert advice with adversarial losses. Our algorithm, Follow-the-Noisy-Leader (FTNL), achieves an $O\left(\frac{\log(K)}{\Delta}+\frac{\log(K)}{\epsilon}\right)$ regret bound in the private stochastic full information setting.

4.4 Bandit Setting

As mentioned in Section 4.3, for the setting of private stochastic bandits, it is possible to have a better regret bound by leveraging the fact that for a sub-optimal arm j, it is wasteful to maintain a complete binary tree with T leaf nodes. Instead, we

can use the following design. We can construct an array with size $O\left(\frac{\log(T)}{\Delta_j \min\{\Delta_j,\epsilon\}}\right)$ to hold the to-be-obtained observations if we knew the gap Δ_j and T in advance. Every time arm j is pulled, the obtained observation will be inserted into an empty slot in the array. When the array is full, we aggregate the values of all the inserted observations and inject a noise variable drawn from Lap $(1/\epsilon)$, where Lap(b) denotes a Laplace distribution centered at 0 with scale b. Intuitively, this idea works, as once a sub-optimal arm j has been observed "enough" times, Learner will not pull it again with high probability.

As practically we do not know the gap Δ_j, and to achieve the goal of devising an algorithm which does not need to know T in advance, we take the strategy of constructing a sequence of arrays with the array size doubling each time. To minimize the amount of noise needed to have an ϵ-differentially private algorithm, we only update the differentially private empirical mean when an array is full of inserted observations, i.e., we update the differential private empirical mean in a lazy way (not too often). Also, the updated differentially private empirical mean is computed only based on observations in a single array, i.e., we abandon all the observations held in the previous arrays.

In this section, we first present our anytime, UCB-based learning algorithm, Anytime-Lazy-UCB. Then, we provide some discussion of Hybrid-UCB, a version of UCB that uses the Hybrid mechanism of [11]. Let $\log(x)$ denote the base-2 logarithm of x and $\ln(x)$ denote the natural logarithm of x. Let $\sum\mathcal{T}$ denote the aggregated values of all inserted observations in an array $\mathcal{T}$.

4.4.1 Anytime-Lazy-UCB

The idea behind Anytime-Lazy-UCB is to create arm-specific arrays sequentially. Every time arm j is pulled, the newly obtained observation will be inserted into an empty slot in an array. For arm j, let $\mathcal{T}_j^{(r)}$ be the r-th array with size $\lambda_j^{(r)} = 2^r$. Let $r_j(t-1)$ be the index of the most recently created array that is full of inserted observations by the end of round $t-1$. Our definition of $r_j(t-1)$ guarantees that array $\mathcal{T}_j^{(r_j(t-1))}$ is full of observations by the end of round $t-1$. Hence, $\sum\mathcal{T}_j^{(r_j(t-1))}$ can be interpreted as the aggregated reward of all observations inserted in $\mathcal{T}_j^{(r_j(t-1))}$.

To have an ϵ-differentially private algorithm, we apply the Laplace mechanism of [16], i.e., we inject a noise variable drawn from Lap $(1/\epsilon)$ to $\sum\mathcal{T}_j^{(r_j(t-1))}$.

Let $\widetilde{\mu}_{j,\lambda_j^{(r_j(t-1))},r_j(t-1)}$ be arm j's differentially private empirical mean among the observations held in $\mathcal{T}_j^{(r_j(t-1))}$, which is

$$\widetilde{\mu}_{j,\lambda_j^{(r_j(t-1))},r_j(t-1)} := \frac{\left(\sum \mathcal{T}_j^{(r_j(t-1))} + \text{Lap}\left(\frac{1}{\epsilon}\right)\right)}{\lambda_j^{(r_j(t-1))}} \quad . \tag{4.2}$$

At the beginning of round t, for each arm j, Learner constructs the upper confidence bound $\overline{\mu}_j(t)$ as

$$\overline{\mu}_j(t) := \widetilde{\mu}_{j,\lambda_j^{(r_j(t-1))},r_j(t-1)} + \sqrt{\frac{3\ln(t)}{\lambda_j^{(r_j(t-1))}}} + \frac{3\ln(t)}{\epsilon \cdot \lambda_j^{(r_j(t-1))}} \tag{4.3}$$

and pulls the arm with the highest upper confidence bound, i.e., Learner pulls arm $J_t \in \arg\max_{j \in \mathcal{A}} \overline{\mu}_j(t)$.

At the end of round t, the reward of the pulled arm, $X_{J_t}(t)$, will be inserted into an empty slot in $\mathcal{T}_{J_t}^{(r_{J_t}(t-1)+1)}$. Next, we check whether it is the "right" time to update the differentially private empirical mean of arm J_t. The rule is that if $\mathcal{T}_{J_t}^{(r_{J_t}(t-1)+1)}$ will be full after inserting $X_{J_t}(t)$, then the differentially private empirical mean of arm J_t will be updated. Also, if array $\mathcal{T}_{J_t}^{(r_{J_t}(t-1)+1)}$ will be full after inserting $X_{J_t}(t)$, we will create a new empty array for the future use.

Algorithm 8 presents Anytime-Lazy-UCB in detail. At first glance, Algorithm 8 may consume $O(T)$ space for each arm. However, as described below, an efficient implementation based on cumulative statistics rather than creating arrays explicitly only needs $O(1)$ space for each arm, i.e., the space complexity of the efficient implementation of Algorithm 8 is $O(K)$, the same as UCB1 [5]. Also, Anytime-Lazy-UCB is simple in the sense that for each arm, it does not need to carefully tune the size of arrays as the array size has no dependency on ϵ.

One of the **efficient implementations of Algorithm 8** is we can compress $\mathcal{T}_j^{(r)}$ to a tuple $\left(N_j^{(r)}, S_j^{(r)}, \lambda_j^{(r)}\right)$ rather than explicitly creating an array $\mathcal{T}_j^{(r)}$, where counter $N_j^{(r)}$ records the number of already inserted observations and $S_j^{(r)}$ records the aggregated rewards among these inserted $N_j^{(r)}$ observations. When $N_j^{(r)}$ hits $\lambda_j^{(r)}$, the differentially private empirical mean of arm j will be updated. Note that Algorithm 8 uses two tuples simultaneously at most. Therefore, the space com-

plexity of Anytime-Lazy-UCB is $O(K)$ if using this efficient implementation as each arm only consumes constant space.

Algorithm 8 Anytime-Lazy-UCB

Initialization: Arm set $\mathcal{A}$, and privacy parameter ϵ ;
for $j \in \mathcal{A}$ **do**
  Set $r_j \leftarrow 0$;
  Pull j once to initialize $\widetilde{\mu}_{j,1,0}$;
  Set $O_j \leftarrow 1$;
  Create $\mathcal{T}_j^{(1)}$ with size $\lambda_j^{(1)} \leftarrow 2$;
end for
for $t = K+1, K+2, \ldots$ **do**
  Construct $\overline{\mu}_j(t)$ based on (4.3) ;
  Pull $J_t \in \max_{j \in \mathcal{A}} \overline{\mu}_j(t)$;
  Set $O_{J_t} \leftarrow O_{J_t} + 1$;
  Insert $X_{J_t}(t)$ to an empty slot in $\mathcal{T}_{J_t}^{(r_{J_t}+1)}$;
  if $O_{J_t} = \sum_{r=0}^{r_{J_t}+1} \lambda_{J_t}^{(r)}$ **then**
    Set $r_{J_t} \leftarrow r_{J_t} + 1$;
    Update $\widetilde{\mu}_{J_t, \lambda_j^{(r_{J_t})}, r_{J_t}}$ based on (4.2) ;
    Create a new array $\mathcal{T}_{J_t}^{(r_{J_t}+1)}$ with size $\lambda_{J_t}^{(r_{J_t}+1)} \leftarrow 2^{r_{J_t}+1}$.
  end if
end for

We now present a privacy guarantee for Algorithm 8.

Theorem 13. *Algorithm 8 is ϵ-differentially private.*

Proof sketch: Intuitively, differential privacy holds because if two neighbouring reward vector sequences differ in a round t, then this difference can be witnessed for only one arm J_t. Ultimately, the affected observation $X_{J_t}(t)$ will be used only via a noisy sum involving Lap $(1/\epsilon)$ noise. Hence, ϵ-differential privacy holds. For completeness, we give a full, mathematical proof in Appendix 4.8.1. □

We now present regret guarantees for Algorithm 8. Theorem 14 provides a problem-dependent regret bound while Theorem 15 provides a problem-independent regret bound.

Theorem 14. *The regret of Algorithm 8 is at most*

$$O\left(\sum_{j\in\mathcal{A}:\Delta_j>0}\frac{\log(T)}{\min\{\Delta_j,\epsilon\}}\right) \quad .$$

Theorem 15. *The regret of Algorithm 8 is also at most*

$$O\left(\sqrt{KT\log(T)}+\frac{K\log(T)}{\epsilon}\right) \quad .$$

Theorem 14 is optimal in the sense that it matches the $\Omega\left(\sum_{j\in\mathcal{A}:\Delta_j>0}\frac{\log T}{\Delta_j}+\frac{K\log T}{\epsilon}\right)$ problem-dependent regret lower bound of [34]. When ϵ is very large, Theorem 14 is the same as the regret bound of non-private stochastic bandits. Our Theorem 15 is the same as the problem-independent regret of DP-SE (shown in Theorem 4.4 of [33]). We mention in passing that apparently both of our problem-independent regret upper bound and that of [33] stand in opposition to a minimax regret lower bound of [7] (see Theorem 3 therein). However, the latter is an unpublished manuscript.

Proof sketch of Theorem 14: Instead of upper bounding $\mathbb{E}\left[O_j(T)\right]$ directly, we take a different approach based on which array the last observation of arm j is inserted into. Let θ_j be the index of the array where the $O_j(T)$-th observation is inserted. Note that θ_j is random. Recall that the array size of $\mathcal{T}_j^{(r)}$ is $\lambda_j^{(r)}=2^r$. Let $\omega_j^{(s)}:=\sum_{r=1}^{s}\lambda_j^{(r)}=\sum_{r=1}^{s}2^r$ be the total length of all arrays up to the array with index s. Let $d_j:=\left\lceil\log\left(\frac{24\ln(T)}{\Delta_j\cdot\min\{\Delta_j,\epsilon\}}\right)\right\rceil$. Then we have

$$\begin{aligned}\mathbb{E}\left[O_j(T)\right] &\le \omega_j^{(d_j)}+\mathbb{E}\left[\sum_{r=d_j+1}^{\theta_j}\lambda_j^{(r)}\right]\\ &\le \omega_j^{(d_j)}+\sum_{s=d_j+1}^{\log(T)}\underbrace{\mathbb{P}\left\{\theta_j=s\right\}}_{\text{bounded by Lemma 9}}\omega_j^{(s)}\\ &\le O\left(\frac{\log(T)}{\Delta_j\cdot\min\left\{\Delta_j,\epsilon\right\}}\right) \quad .\end{aligned} \tag{4.4}$$

The idea of (4.4) is that if the last observation of a sub-optimal arm j is inserted into array $\mathcal{T}_j^{(s)}$, where $s\le d_j$, the number of pulls of j is at most $\sum_{r=1}^{d_j}2^r=\omega_j^{(d_j)}$.

If the last observation of a sub-optimal arm j is inserted into array $\mathcal{T}_j^{(s)}$, where $s \geq d_j + 1$, the number of pulls of arm j is at most $\sum_{r=1}^{s} 2^r = \omega_j^{(s)}$. Our Lemma 9 (in Appendix 4.8.2) further shows that if arm j has been pulled enough, the probability to pull it again is very low. Note that we can create at most $\log(T)$ arrays as the array size doubles each time. □

4.4.2 Hybrid-UCB

Both Anytime-Lazy-UCB and DP-SE use the ideas of laziness and forgetfulness to achieve optimality. They both drop observations. They do not update the differentially private empirical mean of the pulled arm in every round nor use all the observations obtained so far. Therefore, it is a natural question to ask what regret bound is possible if the differentially private empirical mean of the pulled arm is updated at the end of each round by using all the observations obtained so far.

[37] applied the Hybrid mechanism of [11] for differentially private stochastic bandits and devised the DP-UCB-Bound learning algorithm (Algorithm 1 in [37]). In DP-UCB-Bound learning algorithm, the differentially private empirical mean of the pulled arm is updated at the end of each round, and the differentially private empirical mean is computed based on all the obtained observations from the very beginning. They claimed an $O\left(\frac{K\log(T)}{\Delta} + \frac{K\log(T)}{\epsilon}\log\left(\frac{\log(T)}{\Delta\cdot\epsilon}\right)\right)$ regret bound of their DP-UCB-Bound learning algorithm.

However, [33] raised some concerns about their claimed regret bound. We carefully studied the proof of Theorem 3.2 of [37] and found numerous issues, e.g., (A.6), and all steps involving (A.6) such as (A.11), (A.12), (A.14) and afterwards. We do believe that it is possible to have a correct analysis for Algorithm 1 of [37]. However, this new analysis is outside the scope of this thesis.

For completeness, we now show that by extending our array-based design, Algorithm 8, and applying the Hybrid mechanism of [11] for stochastic bandits, yielding Hybrid-UCB, we can have a near-optimal regret bound. Our Hybrid-UCB, a variant of Algorithm 1 of [37], can be found in Algorithm 9. Similar to [37], our Hybrid-UCB also maintains two differential privacy mechanisms simultaneously with each preserving $\epsilon/2$-differential privacy, to track the differentially private empirical mean of each arm. If arm j is pulled in round t, to update arm j's differentially private empirical mean at the end of round t, all arm j's observations obtained so far will be partitioned into two groups and each differential privacy

Algorithm 9 Hybrid-UCB

Initialization: Arm set $\mathcal{A}$ and privacy parameter ϵ ;
for $j \in \mathcal{A}$ **do**
  Pull arm j once ;
  Insert the corresponding observation into an empty array $\mathcal{F}_j^{(0)}$ with size 1 ;
  Set $\widetilde{\mu}_{j,1} \leftarrow \sum \mathcal{F}_j^{(0)} + \text{Lap}\left(\frac{1}{0.5\epsilon}\right)$;
  Set $O_j \leftarrow 1$;
  Set $r_j \leftarrow 1$;
  Create an empty complete binary tree $\mathcal{B}_j^{(r_j)}$ with $\lambda_j^{(r_j)} \leftarrow 2^{r_j}$ leaf nodes and an empty array $\mathcal{F}_j^{(r_j)}$ also with size $\lambda_j^{(r_j)} \leftarrow 2^{r_j}$;
end for
for $t = K+1, K+2, \ldots$ **do**
  Construct upper confidence bound $\overline{\mu}_j(t)$ based on (4.8) for all $j \in \mathcal{A}$;
  Pull arm $J_t \in \max_{j \in \mathcal{A}} \overline{\mu}_j(t)$;
  Set $O_{J_t} \leftarrow O_{J_t} + 1$;
  Insert $X_{J_t}(t)$ into the left-most empty leaf node in $\mathcal{B}_{J_t}^{(r_{J_t})}$ and the left-most empty slot in $\mathcal{F}_{J_t}^{(r_{J_t})}$;
  Noisily sum all observations inserted into arrays $\left(\mathcal{F}_{J_t}^{(0)}, \ldots, \mathcal{F}_{J_t}^{(r_{J_t}-1)}\right)$ to have $\widetilde{F}_{J_t,0:r_{J_t}-1}$ based on (4.5) ;
  Noisily sum all observations inserted into complete binary tree $\mathcal{B}_{J_t}^{(r_{J_t})}$ to have $\widetilde{B}_{J_t,r_{J_t}}$ based on (4.6) ;
  Set $\widetilde{\mu}_{J_t,O_{J_t}} \leftarrow \frac{\widetilde{F}_{J_t,0:r_{J_t}-1} + \widetilde{B}_{J_t,r_{J_t}}}{O_{J_t}}$ based on (4.7) ;
  if $O_{J_t} = \sum_{r=0}^{r_{J_t}} \lambda_{J_t}^{(r)}$ **then**
    Set $r_{J_t} \leftarrow r_{J_t} + 1$;
    Create a new empty complete binary tree $\mathcal{B}_{J_t}^{(r_{J_t})}$ with $\lambda_{J_t}^{(r_{J_t})} \leftarrow 2^{r_{J_t}}$ leaf nodes and a new empty array $\mathcal{F}_{J_t}^{(r_{J_t})}$ also with size $\lambda_{J_t}^{(r_{J_t})} \leftarrow 2^{r_{J_t}}$.
  end if
end for

mechanism takes care of one group of observations.

Let $O_j(t-1)$ be the number of observations of arm j by the end of round $t-1$. In Hybrid-UCB, all $O_j(t-1)$ observations are partitioned into two subsequences, and different subsequences will use different differential privacy mechanisms to inject noise. Every time a new observation of arm j is obtained, we insert this newly obtained observation into both an empty leaf node in a complete binary tree and an empty slot in an array (except the only pull in the initialization phase).

Let $r_j(t-1)$ be the binary tree (and array) index that the $O_j(t-1)$-th observation was inserted into by the end of round $t-1$. Then we know that arrays $\left(\mathcal{F}_j^{(0)}, \mathcal{F}_j^{(1)}, \ldots, \mathcal{F}_j^{(r_j(t-1)-1)}\right)$ will be full of inserted observations, as $\mathcal{F}_j^{(r)}$ and $\mathcal{B}_j^{(r)}$ are always holding the same observations.

The first subsequence contains the observations that have been inserted into arrays $\left(\mathcal{F}_j^{(0)}, \mathcal{F}_j^{(1)}, \ldots, \mathcal{F}_j^{(r_j(t-1)-1)}\right)$ and each array $\mathcal{F}_j^{(r)}$ is injected with a noise variable $Z_j^{(r)} \sim \text{Lap}\left(\frac{1}{0.5\epsilon}\right)$. Then the noisy sum of all these observations inserted into these arrays $\widetilde{F}_{j,0:r_j(t-1)-1}$ can be expressed as

$$\widetilde{F}_{j,0:r_j(t-1)-1} := \sum_{r=0}^{r_j(t-1)-1} \left(\sum \mathcal{F}_j^{(r)} + Z_j^{(r)}\right) \quad . \tag{4.5}$$

The second subsequence contains the remaining observations, i.e., the ones inserted into binary tree $\mathcal{B}_j^{(r_j(t-1))}$. Note that according to the tree-based aggregation mechanism [15, 11], each node in binary tree $\mathcal{B}_j^{(r_j(t-1))}$ is injected with a noise variable $Y_j^{(v)} \sim \text{Lap}\left(\frac{\log\left(\lambda_j^{(r_j(t-1))}\right)}{0.5\epsilon}\right)$, where v is a node in complete binary tree $\mathcal{B}_j^{(r_j(t-1))}$. Node v can be a leaf node or an internal node in $\mathcal{B}_j^{(r_j(t-1))}$. Note that $\log\left(\lambda_j^{(r_j(t-1))}\right) = r_j(t-1)$. The noisy sum of all these observations inserted into the binary tree $\mathcal{B}_{j,r_j(t-1)}$ can be expressed as

$$\widetilde{B}_{j,r_j(t-1)} := \sum \mathcal{B}_j^{(r_j(t-1))} + \sum_{v \in \mathcal{S}_{j,t-1}} Y_j^{(v)} \quad , \tag{4.6}$$

where $\sum \mathcal{B}_j^{(r_j(t-1))}$ is the aggregated values of all observations inserted into $\mathcal{B}_j^{(r_j(t-1))}$

and $\mathcal{S}_{j,t-1}$ is the set of nodes (p-sums) involved in the tree-based aggregation mechanism in order to noisily sum the values of observations inserted into complete binary tree $\mathcal{B}_j^{(r_j(t-1))}$. Note that the size of set $\mathcal{S}_{j,t-1}$ can be at most $\log\left(\lambda_j^{(r_j(t-1))}\right) = \log\left(2^{r_j(t-1)}\right) = r_j(t-1)$.

Let $\widetilde{\mu}_{j,O_j(t-1)}$ be the differentially private empirical of arm j by the end of round $t-1$. It can be expressed as

$$\widetilde{\mu}_{j,O_j(t-1)} := \frac{\widetilde{F}_{j,0:r_j(t-1)-1} + \widetilde{B}_{j,r_j(t-1)}}{O_j(t-1)} \quad . \tag{4.7}$$

At the beginning of round t, for each arm j, we construct the upper confidence bound $\overline{\mu}_j(t)$ as

$$\overline{\mu}_j(t) := \widetilde{\mu}_{j,O_j(t-1)} + \sqrt{\frac{3\log(t)}{O_j(t-1)}} + \frac{6\sqrt{8}\cdot\log(t)\cdot\left\lfloor\log\left(O_j(t-1)+1\right)\right\rfloor}{\epsilon\cdot O_j(t-1)} \tag{4.8}$$

and pull the arm with the highest $\overline{\mu}_j(t)$, i.e., $J_t \in \arg\max_{j\in\mathcal{A}} \overline{\mu}_j(t)$.

At the end of round t, the observation of the pulled arm, $X_{J_t}(t)$, is inserted into the left-most empty slot (and the left-most leaf node) in the most recently created array (and tree). If there is no empty space left after inserting $X_{J_t}(t)$, we will create a new empty array and a new empty binary tree for future use.

We now present privacy and regret guarantees for our Hybrid-UCB.

Theorem 16. *Algorithm 9 is ϵ-differentially private.*

Proof of Theorem 16: As Hybrid-UCB is the composition of our array-based learning algorithm, Algorithm 8 with the required privacy parameter set to 0.5ϵ, and the tree-based aggregation mechanism of [15, 11] with the required privacy parameter set to 0.5ϵ, from the composition theorem of [16] (the property that ϵ's can be added up), we immediately conclude that Algorithm 9 is ϵ-differentially private. □

Theorem 17. *The regret of Algorithm 9 is at most*

$$O\left(\sum_{j\in\mathcal{A}:\Delta_j>0} \max\left\{\frac{\log(T)}{\Delta_j}, \frac{\log(T)}{\epsilon}\log\left(\frac{\log(T)}{\epsilon\cdot\Delta_j}\right)\right\}\right) \quad .$$

From Theorem 17, we know that the regret of Algorithm 9 is sub-optimal compared to the regret of Anytime-Lazy-UCB. However, in terms of T, the sub-optimality is only an extra factor of $\log\log T$.

4.5 Full Information Setting

Different from the bandit setting where only the reward of the pulled arm can be observed, in a full information game a complete reward vector of all actions can be seen. Therefore, it may be harder to preserve ϵ-differential privacy in a full information game. A naive algorithm is to let each action's algorithm that computes the empirical mean be (ϵ/K)-differentially private. Then, Learner runs the naive Follow-the-Noisy-Leader (FTNL). The key difference between Follow-the-Leader (FTL) and the naive FTNL is that in FTL, Learner plays the action with the highest empirical mean while in the naive FTNL, Learner plays the action with the highest differentially private empirical mean. From the Post-Processing Proposition and the Composition Theorem of [16], we immediately have that learning algorithm of Learner is ϵ-differentially private. However, a straightforward analysis of this naive FTNL will result in an $O\left(\max\left\{\frac{\log(K)}{\Delta}, \frac{K\log(K)}{\epsilon}\right\}\right)$ regret bound if we still use the ideas of laziness and forgetfulness to update the differentially private empirical mean [2]. Note that the term involving ϵ is still $\widetilde{O}(K)$. If K is very large, the naive FTNL is very sub-optimal.

To remove the linearity in K, in this section, we introduce a new ingredient in the design of the naive FTNL, which is the algorithm called Report Noisy Max (RNM) [16]. Note that we have presented what RNM is in Section 2.5.

4.5.1 FTNL

Algorithm 10 presents FTNL in detail. FTNL progresses in epochs with epoch s having 2^s rounds and at the end of each epoch, FTNL invokes a procedure called Report Noisy Max (RNM). Let $X(s)$ be the collection of all complete reward vectors belonging to epoch s. In epoch s, Learner plays $J^{(s-1)}$ for 2^s times, where $J^{(s-1)}$ is the output of RNM (Algorithm 11) that takes $(X(s-1), \epsilon)$ as input. At the end of epoch s, RNM takes $X(s)$, all the complete reward vectors within epoch s,

[2] Without using the ideas of laziness and forgetfulness , the regret bound may pay an extra factor of $\log\log T$, which has been described in Section 4.4.2.

as input and outputs *a single action* that has the highest differentially private empirical mean, i.e., $J^{(s)} \leftarrow \arg\max_{j \in \mathcal{A}} \widetilde{\mu}_{j,2^s}$, where $\widehat{\mu}_{j,2^s}$ is the true empirical mean of observations in epoch s and $\widetilde{\mu}_{j,2^s} = \widehat{\mu}_{j,2^s} + \text{Lap}(1/\epsilon)/2^s$ is the differentially private empirical mean of action j. After invoking RNM, FTNL moves to the next epoch and plays the output $J^{(s)}$ for 2^{s+1} times. For initialization purpose, we set $s = 0$ and $J^{(0)} = 1$.

It is important to note that RNM must take fresh observations as input every time. If RNM reuses any observation more than once, then the claim of preserving ϵ-differential privacy is violated. In a private stochastic full information setting, forgetfulness plays an even more important role.

Algorithm 10 FTNL

1: **Initialization:** Action set $\mathcal{A}$ and privacy parameter ϵ ;
   Set $s \leftarrow 0$;
   Set $J \leftarrow 1$;
2: **while** Still have rounds left **do**
3: Play J for 2^s times ;
   Set $J \leftarrow \text{RNM}(X(s), \epsilon)$;
   Set $s \leftarrow s + 1$.
4: **end while**

Algorithm 11 Report Noisy Max (RNM)

1: **Input:** $(X(s), \epsilon)$;
2: **Output:** $J \in \arg\max_{j \in \mathcal{A}} \left(\widehat{\mu}_{j,2^s} + \text{Lap}(1/\epsilon)/2^s\right)$.

We now present FTNL's privacy and regret guarantees.

Theorem 18. *Algorithm 10 is ϵ-differentially private.*

Proof sketch of Theorem 18: Let $\boldsymbol{X}_{1:T} := (X(1), \ldots, X(t), \ldots, X(T))$ be the sequence of original reward vectors and $\boldsymbol{X}'_{1:T} := (X(1), \ldots, X'(t), \ldots, X(T))$ be an arbitrary neighbouring sequence of reward vectors such that $\boldsymbol{X}_{1:T}$ and $\boldsymbol{X}'_{1:T}$ differ in at most one reward vector. Let us say that they differ in the reward vector of round t. Note that the change of a single reward vector may impact the differentially private empirical means of all actions in $\mathcal{A}$. However, by introducing the procedure of RNM to FTNL, the amount of noise needed to preserve ϵ-differential privacy is significantly reduced.

Let epoch s_0 be the one including round t. Then, we know that the decisions made until the last round of s_0 stay the same whether working over $\boldsymbol{X}_{1:T}$ or $\boldsymbol{X}'_{1:T}$, conditioned on the noise injected. Recall that $J^{(s_0)}$ indicates the action output by RNM at the end of epoch s_0 when working over $\boldsymbol{X}_{1:T}$. Let $J'^{(s_0)}$ indicate the corresponding action output by RNM when working over $\boldsymbol{X}'_{1:T}$. As Lap $(1/\epsilon)$ noise is injected to each action j's observations obtained in epoch s_0, from Claim 3.9 (the Report Noisy Max algorithm is ϵ-differentially private) of [16], for any $\sigma \in \mathcal{A}$, we have

$$\mathbb{P}\left\{J^{(s_0)} = \sigma\right\} \leq e^{\epsilon} \cdot \mathbb{P}\left\{J'^{(s_0)} = \sigma\right\} \quad .$$

We defer the full, mathematical proof to Appendix 4.8.5. □

Theorem 19. *The regret of Algorithm 10 is at most*

$$O\left(\frac{\log(K)}{\min\{\epsilon, \Delta_{\min}\}}\right) \quad ,$$

where $\Delta_{\min} = \min_{j \in \mathcal{A}: \Delta_j > 0} \Delta_j$.

Theorem 19 has no dependency on T and is only logarithmic in K. When ϵ is very large, Theorem 19 will be the optimal $O\left(\log(K)/\Delta_{\min}\right)$ regret bound for the non-private stochastic full information setting.

Proof sketch of Theorem 19: Let $(d_1, d_2, \ldots, d_{r_{\max}})$ be a sequence of non-decreasing positive integers, where $r_{\max} := \lceil \log(1/\Delta_{\min}) \rceil$. The choice of these integers can be found in Appendix 4.8.6. We partition all T rounds into $(r_{\max}+1)$ *phases* with a phase $r \leq r_{\max}$ containing epochs from $d_{r-1}+1$ up to d_r and phase $r_{\max}+1$ containing epochs from $d_{r_{\max}}+1$ onwards. We also partition all actions in $\mathcal{A}$ to $r_{\max}$ groups based on the gaps. Let $\Phi^{(r)} := \left\{j \in \mathcal{A} : 0.5^r < \Delta_j \leq 0.5^{r-1}\right\}$, where $1 \leq r \leq r_{\max}$, collect all actions with gaps between $\left(0.5^r, 0.5^{r-1}\right]$. Note that playing any action in $\Phi^{(r)}$ will suffer at most 0.5^{r-1} regret in a single round. We upper bound the regret phase by phase. For all the epochs in phase r, the regret $R^{(r)}$ is at most

$$\begin{aligned} R^{(r)} \quad \leq \quad & \sum_{q=1}^{r-1} \sum_{s=d_{r-1}+1}^{d_r} 2^s \cdot \mathbb{P}\left\{J^{(s-1)} \in \Phi^{(q)}\right\} \cdot 0.5^{q-1} \\ + \quad & \sum_{s=d_{r-1}+1}^{d_r} 2^s \cdot 0.5^{r-1} \quad . \end{aligned} \tag{4.9}$$

For the first term in (4.9), as shown in the proof of Lemma 11 in Appendix 4.8.6,

the probability of playing any action in $\Phi^{(q)}$, where $q \leq r-1$, is very low in any epochs belonging to phase r. The second term in (4.9) uses the idea that playing any action in Φ^q, where $q \geq r$, will suffer at most 0.5^{r-1} regret per round. Taking a summation of $R^{(r)}$ over all phases concludes the proof. □

4.6 Experimental Results

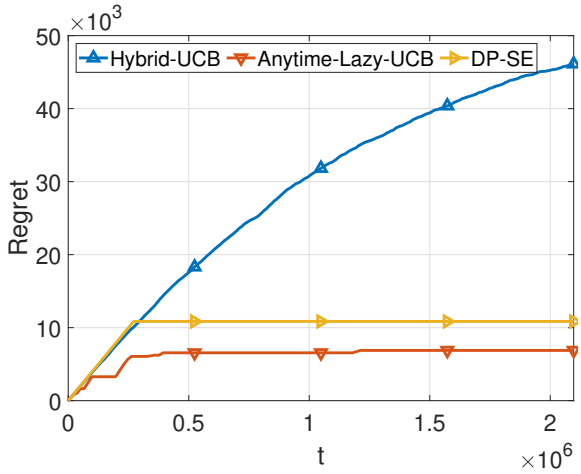

Figure 4.1: Mean reward setting 1: regret with $\epsilon = 0.5$

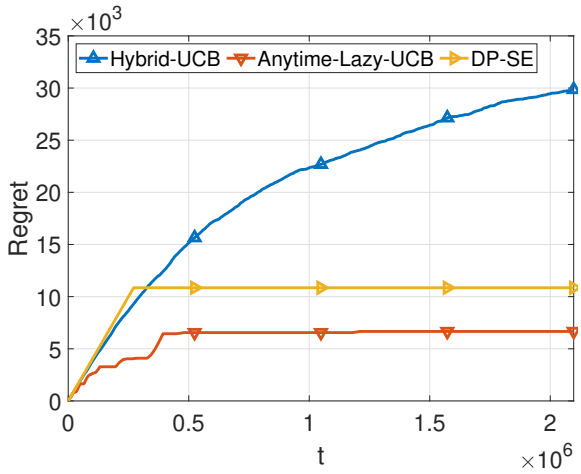

Figure 4.2: Mean reward setting 1: regret with $\epsilon = 1$

We conduct experiments to see the practical performance of Anytime-Lazy-UCB, Hybrid-UCB, and DP-SE of [33]. We set $T = 2 \times 2^{21}$ and each plot is an

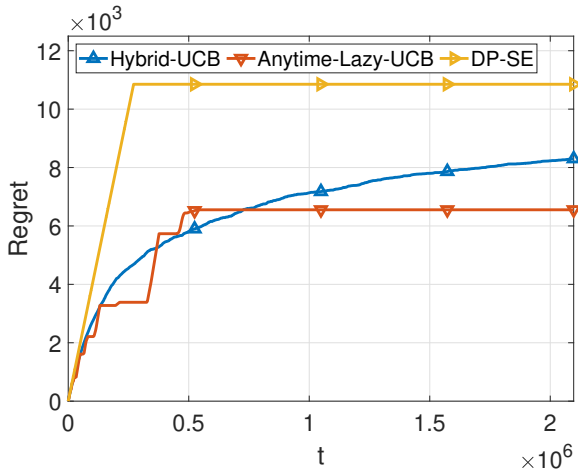

Figure 4.3: Mean reward setting 1: regret with $\epsilon = 8$

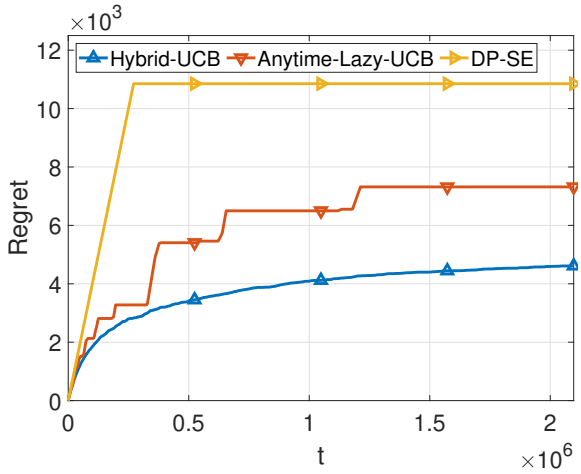

Figure 4.4: Mean reward setting 1: regret with $\epsilon = 64$

average of 15 independent runs. We set $\beta = 1/T$ in DP-SE. For the choices of ϵ, we set $\epsilon = 0.10, 0.25, 0.5, 1, 8, 64, 128$. We set the number of arms $K = 5$ and reuse the same mean reward settings of [33]:

1. Mean reward setting 1: $0.75, 0.70, 0.70, 0.70, 0.70$;
2. Mean reward setting 2: $0.75, 0.625, 0.5, 0.375, 0.25$.

Figures 4.1 and 4.2 show that when ϵ is small ($\epsilon = 0.5, 1$), Anytime-Lazy-UCB is competitive with DP-SE while Hybrid-UCB does not perform well. However, when ϵ is large ($\epsilon = 8, 64$), Hybrid-UCB starts performing well, as shown in Figures 4.3 and 4.4. The larger ϵ is, the better the performance. This fact is just what

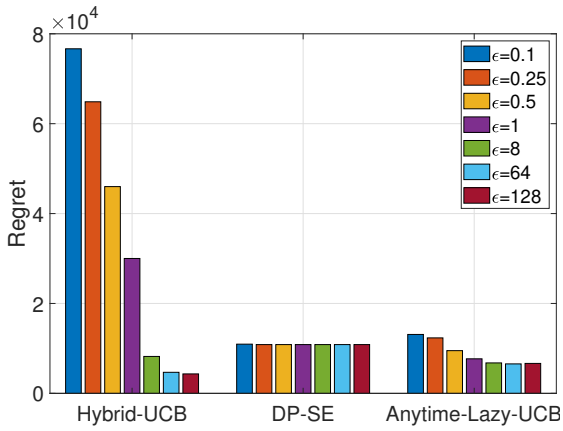

Figure 4.5: Mean reward setting 1: final regret for all ϵ

we have expected, as Hybrid-UCB with a large ϵ value is basically UCB1 of [5]. Recall that Hybrid-UCB uses all the observations obtained so far to compute differentially private empirical mean and updates the differentially private empirical mean of the pulled arm at the end of each round. Figure 4.5 summarizes the final regret (the total regret suffered by the end of round T) for all the considered choices of ϵ and all the compared algorithms. Apparently, for the UCB-based algorithms, Hybrid-UCB and Anytime-Lazy-UCB, the final regret decreases as ϵ increases. However, just as already shown in [33], the final regret of DP-SE with different choices of ϵ tends to stay flat. For some experimental settings (e.g., ϵ is very small), DP-SE practically performs better than Anytime-Lazy-UCB.

Figures 4.6, 4.7, and 4.8 show the performance of all the algorithms in mean reward setting 2 with $\epsilon = 0.5, 8, 64$. Generally, all three algorithms in setting 2 behave very similarly to how they perform in setting 1, e.g., Anytime-Lazy-UCB is competitive with DP-SE and Hybrid-UCB performs well with large ϵ values. Appendix 4.8.7 provides more plots with other choices of ϵ.

4.7 Conclusion

We have introduced the first non-elimination-style algorithm that obtains the optimal problem-dependent regret for differentially private stochastic bandits. Moreover, it is an anytime algorithm, a property that DP-SE (which is also optimal) does not have. While it may be possible to obtain an anytime variant of DP-SE by way

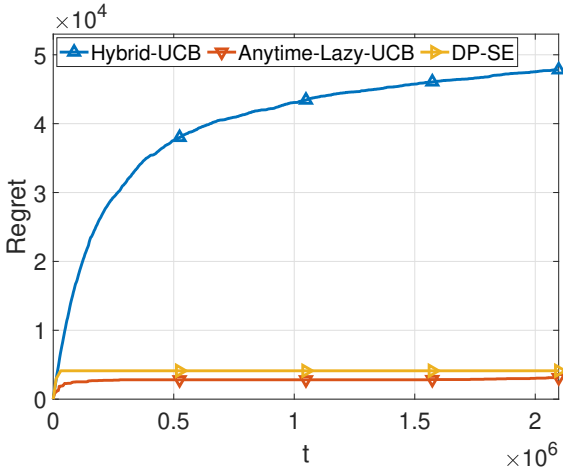

Figure 4.6: Mean reward setting 2: regret with $\epsilon = 0.5$

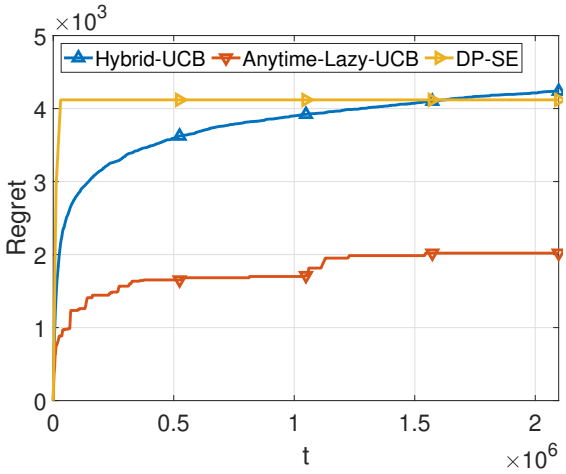

Figure 4.7: Mean reward setting 2: regret with $\epsilon = 8$

of the exponential trick of [9, 6] to preserve the regret bounds, using such tricks is wasteful and may harm the regret by a multiplicative constant between 4 or 8, depending on whether the focus is to preserve problem-dependent regret bounds or problem-independent regret bounds. Despite the above difference, Anytime-Lazy-UCB and DP-SE share in common that they are both lazy and forgetful, i.e., they both update the differentially private empirical mean occasionally and use the newly obtained information only. This raises a natural open question: Is there any optimal algorithm for this setting that avoids being either lazy or forgetful?

Suppose that such an optimal algorithm exists which simultaneously avoids *both* of these properties. Such an algorithm would use all the history information obtained by the end of each round, in the style of classic UCB-style algo-

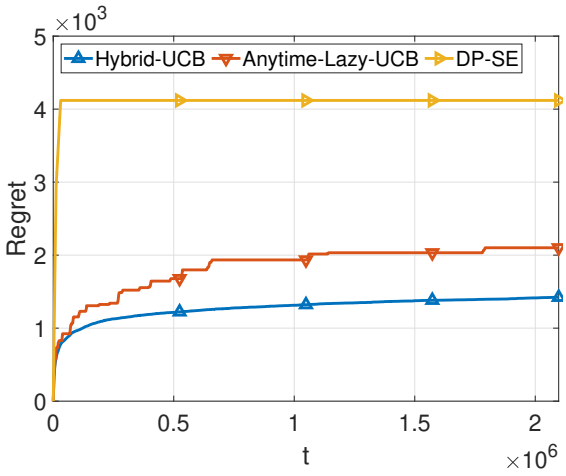

Figure 4.8: Mean reward setting 2: regret with $\epsilon = 64$

rithms. We conjecture that any such algorithm cannot be optimal. Proving this would require a lower bound. We further conjecture that the lower bound is $\Omega\left(\max\left\{\frac{K\log(T)}{\Delta}, \frac{K\log(T)}{\epsilon}\cdot\log\left(\frac{\log(T)}{\epsilon\cdot\Delta}\right)\right\}\right)$. Note that this conjectured lower bound matches the regret upper bound of Hybrid-UCB in Theorem 17.

In the stochastic full information setting, there is no private regret lower bound yet (i.e., the one depending on ϵ). We also conjecture that our $O\left(\frac{\log(K)}{\Delta}+\frac{\log(K)}{\epsilon}\right)$ regret upper bound from Theorem 19 is optimal and leave as an open question proving a matching lower bound.

We believe that, by using the ideas of laziness, forgetfulness, and RNM, it is possible to devise optimal learning algorithms for other differentially private online learning variants such as differentially private Combinatorial multi-armed bandits (CMAB) [13] and differentially private graphical bandits [10]. As we will show in a later Section 6.4, the ideas of forgetfulness, laziness, and RNM will be confirmed to be useful to devise a good differentially private learning algorithm for graphical bandits.

4.8 Appendix of this Chapter

The organization of this appendix is as follows:

4.8.1 - Proofs of Theorem 13 ;

4.8.2 - Proofs of Theorem 14 ;

4.8.3 - Proofs of Theorem 15 ;

4.8.4 - Proofs of Theorem 17 ;

4.8.5 - Proofs of Theorem 18 ;

4.8.6 - Proofs of Theorem 19 ;

4.8.7 - Additional experimental plots .

In these sections, we will often make use of the following facts.

Fact 1. (Fact 3.7 in [16]). If $Y \sim \text{Lap}(b)$, for any $0 < \delta < 1$, we have

$$\mathbb{P}\left\{|Y| > \ln\left(\frac{1}{\delta}\right) \cdot b\right\} = \delta \quad . \tag{4.10}$$

Fact 2. (Corollary 2.9 in [11]). Let $\gamma_1, \gamma_2, \ldots, \gamma_n$ be i.i.d. random variables drawn from $\text{Lap}(b)$. Let $Y = \sum_{i=1}^{n} \gamma_i$. For any $0 < \delta < 1$, and $\chi > b \cdot \max\left\{\sqrt{n}, \sqrt{\ln\left(\frac{2}{\delta}\right)}\right\}$, we have

$$\mathbb{P}\left\{|Y| > \chi \cdot \sqrt{8\ln\left(\frac{2}{\delta}\right)}\right\} \leq \delta \quad . \tag{4.11}$$

4.8.1 Proofs of Theorem 13

Proof of Theorem 13: Let $X_{1:T}$ be the original reward vector sequence and $X'_{1:T}$ be an arbitrary neighbouring reward vector sequence of $X_{1:T}$ such that they can differ in an arbitrary round. Let us say $X_{1:T}$ and $X'_{1:T}$ differ from each other in round t. Let $D_{1:T}$ be the sequence of decisions made through round 1 to round T when working over $X_{1:T}$. Let $D'_{1:T}$ be the sequence of decisions made when working over $X'_{1:T}$.

For an arbitrary $\sigma_{1:T} \in \mathcal{A}^T$, we claim that

$$\mathbb{P}\left\{D_{1:T} = \sigma_{1:T} \mid X_{1:T}\right\} \leq e^{\epsilon} \cdot \mathbb{P}\left\{D'_{1:T} = \sigma_{1:T} \mid X'_{1:T}\right\} \quad . \tag{4.12}$$

To prove this claim, we rewrite both sides in (4.12) first.

The LHS in (4.12) can be expressed as

$$\begin{aligned} &\mathbb{P}\left\{D_{1:T} = \sigma_{1:T} \mid X_{1:T}\right\} \\ &= \mathbb{P}\left\{D_{1:t} = \sigma_{1:t} \mid X_{1:T}\right\} \underbrace{\mathbb{P}\left\{D_{t+1:T} = \sigma_{t+1:T} \mid D_{1:t} = \sigma_{1:t}, X_{1:T}\right\}}_{\eta} \quad . \end{aligned} \tag{4.13}$$

Similarly, we have

$$\begin{aligned} & \mathbb{P}\left\{D'_{1:T} = \sigma_{1:T} \mid X'_{1:T}\right\} \\ = \; & \mathbb{P}\left\{D'_{1:t} = \sigma_{1:t} \mid X'_{1:T}\right\} \underbrace{\mathbb{P}\left\{D'_{t+1:T} = \sigma_{t+1:T} \mid D'_{1:t} = \sigma_{1:t}, X'_{1:T}\right\}}_{\eta'} \quad . \end{aligned} \tag{4.14}$$

Since $X_{1:T}$ and $X'_{1:T}$ only differ in round t at most, the sequence of arms pulled up to round t (including round t) has the same distribution either of $X_{1:T}$ or $X'_{1:T}$. That is also to say, we have

$$\mathbb{P}\left\{D_{1:t} = \sigma_{1:t} \mid X_{1:T}\right\} = \mathbb{P}\left\{D'_{1:t} = \sigma_{1:t} \mid X'_{1:T}\right\} \quad . \tag{4.15}$$

We now only need to prove $\eta \leq e^{\epsilon} \cdot \eta'$ to conclude the proof.

Let r be the index of the array where the observation obtained in round t, i.e., $X_{J_t}(t)$, is inserted, when taking $X_{1:T}$ as input. Note that r is random. Let $t_* \geq t$ be the first round such that, at the end of this round, array $\mathcal{T}_{J_t}^{(r)}$ is full of inserted observations. Note that t_* is also random.

Then η can be expressed as

$$\begin{aligned} \eta \; = \; & \sum_{j \in \mathcal{A}} \sum_{q=1}^{\log(T)} \sum_{s=t}^{T} \mathbb{P}\left\{J_t = j, r = q, t_* = s \mid D_{1:t} = \sigma_{1:t}, X_{1:T}\right\} \\ & \underbrace{\mathbb{P}\left\{D_{t+1:T} = \sigma_{t+1:T} \mid J_t = j, r = q, t_* = s, D_{1:t} = \sigma_{1:t}, X_{1:T}\right\}}_{\alpha} \quad . \end{aligned} \tag{4.16}$$

Similarly, let J'_t be the arm pulled in round t when working over $X'_{1:T}$. Let r' be the index of the array where observation $X'_{J'_t}(t)$ is inserted. Let $t'_* \geq t$ be the first round such that, at the end of this round, array $\mathcal{T}_{J'_t}^{(r')}$ is full of inserted observations.

Then η' can be expressed as

$$\begin{aligned} \eta' \; = \; & \sum_{j \in \mathcal{A}} \sum_{q=1}^{\log(T)} \sum_{s=t}^{T} \mathbb{P}\left\{J'_t = j, r' = q, t'_* = s \mid D'_{1:t} = \sigma_{1:t}, X'_{1:T}\right\} \\ & \underbrace{\mathbb{P}\left\{D'_{t+1:T} = \sigma_{t+1:T} \mid J'_t = j, r' = q, t'_* = s, D'_{1:t} = \sigma_{1:t}, X'_{1:T}\right\}}_{\alpha'} \quad . \end{aligned} \tag{4.17}$$

Since $X_{1:t-1} = X'_{1:t-1}$, $X_{t+1:T} = X'_{t+1:T}$, and the fact that $X_{J_t}(t)$ (or $X'_{J'_t}(t)$) will only be used after array $\mathcal{T}_{J_t}^{(r)}$ (or $\mathcal{T}_{J'_t}^{(r')}$) is full of inserted observations, we have that J_t and J'_t have the same distribution, r and r' have the same distribution, and t_* and

t'_* have the same distribution. Therefore, only α and α' are different in (4.16) and (4.17). We now show $\alpha \leq e^{\epsilon} \cdot \alpha'$.

We now rewrite α as

$$\begin{aligned}
&\alpha \\
=\ & \mathbb{P}\left\{D_{t+1:T} = \sigma_{t+1:T} \mid J_t = j, r = q, t_* = s, D_{1:t} = \sigma_{1:t}, X_{1:T}\right\} \\
=\ & \mathbb{P}\left\{D_{t+1:t_*} = \sigma_{t+1:t_*} \mid J_t = j, r = q, t_* = s, D_{1:t} = \sigma_{1:t}, X_{1:T}\right\} \\
& \underbrace{\mathbb{P}\left\{D_{t_*+1:T} = \sigma_{t_*+1:T} \mid \underbrace{D_{t+1:t_*} = \sigma_{t+1:t_*}}_{=:M}, J_t = j, r = q, t_* = s, D_{1:t} = \sigma_{1:t}, X_{1:T}\right\}}_{\beta}.
\end{aligned} \tag{4.18}$$

Similarly, we rewrite α' as

$$\begin{aligned}
&\alpha' \\
=\ & \mathbb{P}\left\{D'_{t+1:T} = \sigma_{t+1:T} \mid J'_t = j, r' = q, t'_* = s, D'_{1:t} = \sigma_{1:t}, X'_{1:T}\right\} \\
=\ & \mathbb{P}\left\{D'_{t+1:t'_*} = \sigma_{t+1:t'_*} \mid J'_t = j, r' = q, t'_* = s, D'_{1:t} = \sigma_{1:t}, X'_{1:T}\right\} \\
& \underbrace{\mathbb{P}\left\{D'_{t'_*+1:T} = \sigma_{t'_*+1:T} \mid \underbrace{D'_{t+1:t'_*} = \sigma_{t+1:t'_*}}_{=:M'}, J'_t = j, r' = q, t'_* = s, D'_{1:t} = \sigma_{1:t}, X'_{1:T}\right\}}_{\beta'}.
\end{aligned} \tag{4.19}$$

Since $X_{t+1:T} = X'_{t+1:T}$, and the fact that $X_{J_t}(t)$ (or $X'_{J'_t}(t)$) will only be used after array $\mathcal{T}_{J_t}^{(r)}$ (or $\mathcal{T}_{J'_t}^{(r')}$) is full of inserted observations, we have that $D_{t+1:t_*}$ and $D'_{t+1:t'_*}$ have the same conditional distribution. We now show $\beta \leq e^{\epsilon} \cdot \beta'$. The key idea to prove this piece of argument is to use the property that differential privacy is immune to post-processing [15].

Recall that $\widetilde{\mu}_{J_t, \lambda_{J_t}^{(r)}, r}$ is the differentially private empirical mean of arm J_t among all inserted observations in array $\mathcal{T}_{J_t}^{(r)}$ (i.e., at the end of round t_*) when working over $X_{1:T}$. Let $\widetilde{\mu}_{J'_t, \lambda_{J'_t}^{(r')}, r'}$ be the differentially private empirical mean of arm J'_t among all inserted observations in array $\mathcal{T}_{J'_t}^{(r')}$ (i.e., at the end of round t'_*) when working over $X'_{1:T}$. As Lap $\left(\frac{1}{\epsilon}\right)$ noise is injected to each array after the array is full

of inserted observations, for all $\mathcal{I} \subseteq \mathbb{R}$, we have

$$\begin{aligned} & \mathbb{P}\left\{\widetilde{\mu}_{J_t,\lambda_{J_t}^{(r)},r} \cdot \lambda_{J_t}^{(r)} \in \mathcal{I} \mid M\right\} \\ \leq \; & e^{\epsilon} \cdot \mathbb{P}\left\{\widetilde{\mu}_{J'_t,\lambda_{J'_t}^{(r')},r'} \cdot \lambda_{J'_t}^{(r')} \in \mathcal{I} \mid M'\right\} \quad . \end{aligned} \tag{4.20}$$

As $X_{t_*+1:T} = X'_{t'_*+1:T}$ conditioned on $t_* = s, t'_* = s$, we have

$$\begin{aligned} & \mathbb{P}\left\{D_{t_*+1:T} = \sigma_{t_*+1:T} \mid \widetilde{\mu}_{J_t,\lambda_{J_t}^{(r)},r} \cdot \lambda_{J_t}^{(r)} \in \mathcal{I}, M\right\} \\ = \; & \mathbb{P}\left\{D'_{t'_*+1:T} = \sigma_{t'_*+1:T} \mid \widetilde{\mu}_{J'_t,\lambda_{J'_t}^{(r')},r'} \cdot \lambda_{J'_t}^{(r')} \in \mathcal{I}, M'\right\} \quad . \end{aligned} \tag{4.21}$$

By combining (4.20) and (4.21), we have $\beta \leq e^{\epsilon} \cdot \beta'$, which concludes the proof. □

4.8.2 Proofs of Theorem 14

Proof of Theorem 14: To prove Theorem 14, instead of upper bounding the expected number of pulls of a sub-optimal arm j directly, we take a different approach. Our new approach is based on which array the last observation of arm j is inserted into. Let θ_j be index of the array where the $O_j(T)$-th observation is inserted. Note that θ_j is random. Let $d_j := \left\lceil \log\left(\frac{24\ln(T)}{\Delta_j \cdot \min\{\Delta_j, \epsilon\}}\right)\right\rceil$ and $\omega_j^{(r)} := \sum_{s=1}^{r} \lambda_j^{(s)} = \sum_{s=1}^{r} 2^r = 2^{r+1} - 2$. Then we have $\omega_j^{(d_j)} = O\left(\frac{\log(T)}{\Delta_j \cdot \min\{\Delta_j, \epsilon\}}\right)$.

We have that the number of pulls of a sub-optimal arm j by the end of round T (except for the pull in the initialization phase) is at most

$$O_j(T) \; \leq \sum_{r=1}^{\theta_j} \lambda_j^{(r)} = \sum_{r=1}^{d_j} \lambda_j^{(r)} + \sum_{r=d_j+1}^{\theta_j} \lambda_j^{(r)} = \omega_j^{(d_j)} + \sum_{r=d_j+1}^{\theta_j} \lambda_j^{(r)} \quad . \tag{4.22}$$

By taking the expectation of both sides, we have

$$\begin{aligned}
\mathbb{E}\left[O_j(T)\right] &\leq \mathbb{E}\left[\sum_{r=1}^{\theta_j} \lambda_j^{(r)}\right] \\
&\leq \omega_j^{(d_j)} + \sum_{s=d_j+1}^{\log(T)} \mathbb{P}\left\{\theta_j = s\right\} \cdot \sum_{r=d_j+1}^{s} \lambda_j^{(r)} \\
&\leq \omega_j^{(d_j)} + \sum_{s=d_j+1}^{\log(T)} \underbrace{\mathbb{P}\left\{\theta_j = s\right\} \cdot \omega_j^{(s)}}_{=O(1), \text{ Lemma 9}} \\
&\leq O\left(\frac{\log(T)}{\Delta_j \cdot \min\left\{\Delta_j, \epsilon\right\}}\right) + O\left(\log(T)\right) \\
&= O\left(\frac{\log(T)}{\Delta_j \cdot \min\left\{\Delta_j, \epsilon\right\}}\right) \quad .
\end{aligned} \tag{4.23}$$

The second term in the third inequality of (4.23) uses the fact that $\sum_{r=d_j+1}^{s} \lambda_j^{(r)} \leq \omega_j^{(s)}$. Also, if the last observation of arm j is inserted into array $\mathcal{T}_j^{(s)}$, where $s \geq d_j + 1$, it implies array $\mathcal{T}_j^{(s-1)}$ has no empty slots left at all, i.e., the number of inserted observations in array $\mathcal{T}_j^{(s-1)}$ is $\lambda_j^{(s-1)} = 2^{s-1} \geq 2^{d_j+1-1} \geq \frac{24\ln(T)}{\Delta_j \cdot \min\{\Delta_j, \epsilon\}}$. This amount of i.i.d. observations makes the probability that arm j is pulled in the future very low. We formalize this intuition in Lemma 9.

Lemma 9. *For a fixed sub-optimal arm j and $s \geq d_j + 1$, we have $\mathbb{P}\left\{\theta_j = s\right\} \cdot \omega_j^{(s)} \leq O(1)$.*

We now prove Theorem 14 and defer the proof of Lemma 9 to the end of this section.

From (4.1), we have the regret

$$\begin{aligned}
\mathcal{R}(T) &= \sum_{j \in \mathcal{A}: \Delta_j > 0} \mathbb{E}\left[O_j(T)\right] \cdot \Delta_j \\
&\leq \sum_{j \in \mathcal{A}: \Delta_j > 0} O\left(\frac{\log(T)}{\Delta_j \cdot \min\left\{\Delta_j, \epsilon\right\}}\right) \cdot \Delta_j \\
&= \sum_{j \in \mathcal{A}: \Delta_j > 0} O\left(\frac{\log(T)}{\min\left\{\Delta_j, \epsilon\right\}}\right) \quad ,
\end{aligned} \tag{4.24}$$

which concludes the proof. □

Proof of Lemma 9: Recall that $d_j = \left\lceil \log\left(\frac{24\ln(T)}{\Delta_j \cdot \min(\Delta_j, \epsilon)}\right)\right\rceil$ and $\omega_j^{(r)} = \sum_{s=1}^{r} \lambda_j^{(s)} = \sum_{s=1}^{r} 2^r =$

$2^{r+1}-2$. Then we have $\log\left(\frac{24\ln(T)}{\Delta_j\cdot\min(\Delta_j,\epsilon)}\right) \leq 2^{d_j} \leq 2\log\left(\frac{24\ln(T)}{\Delta_j\cdot\min(\Delta_j,\epsilon)}\right)$. We upper bound the LHS of Lemma 9 as

$$
\begin{aligned}
& \mathbb{P}\left\{\theta_j = s\right\}\cdot\omega_j^{(s)} \\
\leq\ & \mathbb{P}\left\{\exists t\in\{\omega_j^{(s-1)}+1+K,\ldots,T\} : J_t = j, r_j(t-1) = s-1\right\}\cdot\omega_j^{(s)} \\
\leq\ & \sum_{t=\omega_j^{(s-1)}}^{T} \mathbb{P}\left\{\overline{\mu}_j(t)\geq\overline{\mu}_1(t), r_j(t-1)=s-1\right\}\cdot\omega_j^{(s)} \\
\leq\ & \sum_{t=\omega_j^{(s-1)}}^{T} \mathbb{E}\left[\mathbf{1}\left\{\overline{\mu}_j(t)\geq\overline{\mu}_1(t), r_j(t-1)=s-1\right\}\right]\cdot\omega_j^{(s)} \\
\leq\ & \sum_{t=\omega_j^{(s-1)}}^{T}\ \sum_{\tau=0}^{\lfloor\log(t)\rfloor-1} \\
& \mathbb{E}\left[\mathbf{1}\left\{\underbrace{\widetilde{\mu}_{j,2^{s-1},s-1}+\sqrt{\frac{\ln(t^3)}{2^{s-1}}}+\frac{\ln(t^3)}{\epsilon 2^{s-1}} \geq \widetilde{\mu}_{1,2^\tau,\tau}+\sqrt{\frac{\ln(t^3)}{2^\tau}}+\frac{\ln(t^3)}{\epsilon 2^\tau}}_{\Psi}\right\}\right]\omega_j^{(s)} \quad .
\end{aligned}
\tag{4.25}
$$

If Ψ is true, then at least one of the following is true:

$$
\widetilde{\mu}_{j,2^{s-1},s-1} \geq \mu_j + \sqrt{\frac{\ln(t^3)}{2^{s-1}}}+\frac{\ln(t^3)}{\epsilon 2^{s-1}} \quad ; \tag{4.26}
$$

$$
\widetilde{\mu}_{1,2^\tau,\tau} \leq \mu_1 - \sqrt{\frac{\ln(t^3)}{2^\tau}}-\frac{\ln(t^3)}{\epsilon 2^\tau} \quad ; \tag{4.27}
$$

$$
\frac{\Delta_j}{2} < \sqrt{\frac{\ln(t^3)}{2^{s-1}}}+\frac{\ln(t^3)}{\epsilon 2^{s-1}} \quad . \tag{4.28}
$$

For (4.26), we apply inequality (5.9) and Hoeffding's inequality. We have

$$
\begin{aligned}
& \mathbb{P}\left\{\widetilde{\mu}_{j,2^{s-1},s-1} \geq \mu_j + \sqrt{\tfrac{\ln(t^3)}{2^{s-1}}}+\tfrac{\ln(t^3)}{\epsilon 2^{s-1}}\right\} \\
\leq\ & \underbrace{\mathbb{P}\left\{\widetilde{\mu}_{j,2^{s-1},s-1} \geq \widehat{\mu}_{j,2^{s-1},s-1}+\frac{\ln(t^3)}{\epsilon 2^{s-1}}\right\}}_{\text{Inequality (5.9)}} + \underbrace{\mathbb{P}\left\{\widehat{\mu}_{j,2^{s-1},s-1} \geq \mu_j+\sqrt{\frac{\ln(t^3)}{2^{s-1}}}\right\}}_{\text{Hoeffding's inequality}} \\
=\ & O\left(\tfrac{1}{t^3}\right) \quad .
\end{aligned}
\tag{4.29}
$$

Similarly, for (4.27), we have

$$\mathbb{P}\left\{\widetilde{\mu}_{1,2^\tau,\tau} \leq \mu_1 - \sqrt{\frac{\ln(t^3)}{2^\tau}} - \frac{\ln(t^3)}{\epsilon 2^\tau}\right\} = O\left(\frac{1}{t^3}\right) \quad . \tag{4.30}$$

We now prove that inequality (4.28) cannot be true by using contradiction. The RHS in (4.28) is upper bounded as

$$\begin{aligned} & \sqrt{\frac{\ln(t^3)}{2^{s-1}}} + \frac{\ln(t^3)}{\epsilon 2^{s-1}} \\ \leq & \sqrt{\frac{\ln(T^3)}{2^{d_j}}} + \frac{\ln(T^3)}{\epsilon 2^{d_j}} \quad \text{(Note that } s-1 \geq d_j\text{)} \\ \leq & \sqrt{\frac{\ln(T^3)}{\frac{24\ln(T)}{\Delta_j \cdot \min\left\{\Delta_j, \epsilon\right\}}}} + \frac{\ln(T^3)}{\epsilon \cdot \frac{24\ln(T)}{\Delta_j \cdot \min\left\{\Delta_j, \epsilon\right\}}} \quad \text{(Use the lower bound of } 2^{d_j}\text{)} \\ < & \ 0.5\Delta_j \quad , \end{aligned} \tag{4.31}$$

which implies that (4.28) cannot be true.

We now come back to (4.25) and have

$$
\begin{aligned}
& \mathbb{P}\left\{\theta_j = s\right\} \omega_j^{(s)} \\
\leq & \sum_{t=\omega_j^{(s-1)}}^{T} \sum_{\tau=0}^{\lfloor \log(t) \rfloor - 1} \\
& \mathbb{E}\left[\underbrace{\mathbf{1}\left\{ \widetilde{\mu}_{j,2^{s-1},s-1} + \sqrt{\frac{\ln(t^3)}{2^{s-1}}} + \frac{\ln(t^3)}{\epsilon 2^{s-1}} \geq \widetilde{\mu}_{1,2^{\tau},\tau} + \sqrt{\frac{\ln(t^3)}{2^{\tau}}} + \frac{\ln(t^3)}{\epsilon 2^{\tau}} \right\}}_{\Psi} \right] \omega_j^{(s)} \\
\leq & \sum_{t=\omega_j^{(s-1)}}^{T} \sum_{\tau=0}^{\lfloor \log(t) \rfloor - 1} \omega_j^{(s)} \left(\underbrace{\mathbb{P}\left\{ \widetilde{\mu}_{j,2^{s-1},s-1} \geq \mu_j + \sqrt{\frac{\ln(t^3)}{2^{s-1}}} + \frac{\ln(t^3)}{\epsilon 2^{s-1}} \right\}}_{(=O(1/t^3))} \right. \\
& \left. + \underbrace{\mathbb{P}\left\{ \widetilde{\mu}_{1,2^{\tau},\tau} \leq \mu_1 - \sqrt{\frac{\ln(t^3)}{2^{\tau}}} - \frac{\ln(t^3)}{\epsilon 2^{\tau}} \right\}}_{(=O(1/t^3))} \right) \\
\leq & \sum_{t=\omega_j^{(s-1)}}^{T} \sum_{\tau=0}^{\lfloor \log(t) \rfloor - 1} O\left(\frac{1}{t^3}\right) \cdot \omega_j^{(s)} \\
\leq & \sum_{t=\omega_j^{(s-1)}}^{T} O\left(\frac{1}{t^2}\right) \cdot \omega_j^{(s)} \\
\leq & \int_{\omega_j^{(s-1)}}^{T} O\left(\frac{1}{t^2}\right) dt \cdot \omega_j^{(s)} \\
< & O\left(\frac{\omega_j^{(s)}}{\omega_j^{(s-1)}}\right) \\
\leq & O(1) \quad .
\end{aligned}
\tag{4.32}
$$

The second to last inequality uses the following fact. When $r \geq 1$, we have $\frac{\omega_j^{(r+1)}}{\omega_j^{(r)}} = \frac{2^{r+2}-2}{2^{r+1}-2} = \frac{4 \cdot 2^r - 2}{2 \cdot 2^r - 2} = \frac{2 \cdot 2^r - 1}{2^r - 1} = \frac{2 \cdot 2^r - 2 + 1}{2^r - 1} = 2 + \frac{1}{2^r - 1} \leq 3.$ □

4.8.3 Proofs of Theorem 15

Proof of Theorem 15: Let $\Delta^* := \sqrt{\frac{K \log(T)}{T}}$ be the critical gap threshold and, for each $j \in \mathcal{A}$, let $N_j := \mathbb{E}\left[O_j(T)\right]$. We use the following basic decomposition of the

pseudo-regret:

$$\sum_{j\in\mathcal{A}:\Delta_j>0} N_j\Delta_j = \sum_{j\in\mathcal{A}:\Delta_j\geq\Delta^*} N_j\Delta_j + \sum_{j\in\mathcal{A}:0<\Delta_j<\Delta^*} N_j\Delta_j \quad . \tag{4.33}$$

We now bound each summation in turn.

The first summation in (4.33) can be upper bounded as

$$\begin{aligned}\sum_{j\in\mathcal{A}:\Delta_j\geq\Delta^*} N_j\Delta_j &= \sum_{j\in\mathcal{A}:\Delta_j\geq\Delta^*} O\left(\frac{\log(T)}{\min\{\Delta_j,\varepsilon\}}\right)\\ &\leq \sum_{j\in\mathcal{A}:\Delta_j\geq\Delta^*} O\left(\max\left\{\frac{\log(T)}{\Delta^*},\frac{\log(T)}{\varepsilon}\right\}\right)\\ &\leq O\left(K\max\left\{\frac{\log(T)}{\Delta^*},\frac{\log(T)}{\varepsilon}\right\}\right)\\ &\leq O\left(\max\left\{\sqrt{KT\log(T)},\frac{K\log(T)}{\varepsilon}\right\}\right) \quad ,\end{aligned} \tag{4.34}$$

where the first equality uses the problem-dependent regret bound of Algorithm 8.

The second summation in (4.33) admits the bound:

$$\sum_{j\in\mathcal{A}:0<\Delta_j<\Delta^*} N_j\Delta_j \leq \Delta^*\cdot\sum_{j\in\mathcal{A}:0<\Delta_j<\Delta^*} N_j \leq T\Delta^* = \sqrt{KT\log(T)} \quad . \tag{4.35}$$

Combining (4.34) and (4.35) concludes the proof. □

4.8.4 Proofs of Theorem 17

Proof of Theorem 17: Before we step into the detailed proof, we first give a few definitions and then we decompose the regret.

Let $d_j := \left\lceil \log\left(\max\left\{\frac{256\log(T)}{\Delta_j^2},\frac{256\log(T)}{\epsilon\cdot\Delta_j}\cdot\log\left(\frac{256\log(T)}{\epsilon\cdot\Delta_j}\right)\right\}\right)\right\rceil$.

Let $\omega_j^{(r)} := \sum_{s=0}^{r}\lambda_j^{(r)} = \sum_{s=0}^{r} 2^r = 2^{r+1}-1$.

Then we have $\omega_j^{(d_j)} = O\left(\max\left\{\frac{\log(T)}{\Delta_j^2},\frac{\log(T)}{\epsilon\cdot\Delta_j}\cdot\log\left(\frac{\log(T)}{\epsilon\cdot\Delta_j}\right)\right\}\right)$. Let θ_j be the array/tree index that the $O_j(T)$-th observation of arm j is inserted into by the end of round T. Note that θ_j is random. Then we have the number of pulls of a sub-

optimal arm j by the end of round T is at most

$$\begin{aligned} O_j(T) &\leq \sum_{r=0}^{\theta_j} \lambda_j^{(r)} \\ &= \sum_{r=0}^{d_j} \lambda_j^{(r)} + \sum_{r=d_j+1}^{\theta_j} \lambda_j^{(r)} \\ &= \omega_j^{(d_j)} + \sum_{r=d_j+1}^{\theta_j} \lambda_j^{(r)} \quad . \end{aligned} \tag{4.36}$$

As θ_j is random, by taking the expectation of both sides, we have

$$\begin{aligned} \mathbb{E}\left[O_j(T)\right] &\leq \omega_j^{(d_j)} + \mathbb{E}\left[\sum_{r=d_j+1}^{\theta_j} \lambda_j^{(r)}\right] \\ &\leq \omega_j^{(d_j)} + \sum_{s=d_j+1}^{\log(T)} \underbrace{\mathbb{P}\left\{\theta_j = s\right\} \cdot \omega_j^{(s)}}_{\text{Lemma 10}} \quad . \end{aligned} \tag{4.37}$$

To upper bound (4.37), we introduce Lemma 10 first, which states that after a sub-optimal j has been observed "enough" times, the probability that it will be pulled again is very low.

Lemma 10. *For a fixed sub-optimal arm j and $s \geq d_j + 1$, we have $\mathbb{P}\left\{\theta_j = s\right\} \cdot \omega_j^{(s)} = O(1)$.*

We now prove Theorem 17. The proof of Lemma 10 is deferred to the end of this section.

From (4.1), we have

$$\begin{aligned} \mathcal{R}(T) &= \sum_{j \in \mathcal{A}: \Delta_j > 0} \mathbb{E}\left[O_j(T)\right] \cdot \Delta_j \\ &\leq \sum_{j \in \mathcal{A}: \Delta_j > 0} \left(\omega_j^{(d_j)} \cdot \Delta_j + \sum_{s=d_j+1}^{\log(T)} \underbrace{\mathbb{P}\left\{\theta_j = s\right\} \cdot \omega_j^{(s)}}_{=O(1), \text{ Lemma 10}} \cdot \Delta_j \right) \\ &= \sum_{j \in \mathcal{A}: \Delta_j > 0} O\left(\max\left\{\frac{\log(T)}{\Delta_j}, \frac{\log(T)}{\epsilon} \cdot \log\left(\frac{\log(T)}{\epsilon \cdot \Delta_j}\right)\right\}\right) \quad , \end{aligned} \tag{4.38}$$

which concludes the proof. □

Proof of Lemma 10: The idea behind the proof uses the fact that if the last observa-

tion of arm j in inserted into array $\mathcal{F}_j^{(s)}$ (tree $\mathcal{B}_j^{(s)}$), it implies that arm j is pulled again after it has already been pulled $\omega_j^{(s-1)}$ times. To calculate the differentially private empirical mean of these $\omega_j^{(s-1)}$ observations, Hybrid-UCB uses observations inserted into arrays $\left(\mathcal{F}_j^{(0)}, \mathcal{F}_j^{(1)}, \ldots, \mathcal{F}_j^{(s-2)}\right)$ and observations inserted into $\mathcal{B}_j^{(s-1)}$. Note that based on our learning algorithm, all arrays $\left(\mathcal{F}_j^{(0)}, \mathcal{F}_j^{(1)}, \ldots, \mathcal{F}_j^{(s-2)}\right)$ and binary tree $\mathcal{B}_j^{(s-1)}$ are full of inserted observations.

We now start the proof of Lemma 10. We have

$$\begin{aligned}
& \mathbb{P}\left\{\theta_j = s\right\} \cdot \omega_j^{(s)} \\
\leq\ & \mathbb{P}\left\{\exists t \in \{\omega_j^{(s-1)} + K, \ldots, T\} : J_t = j, O_j(t-1) = \omega_j^{(s-1)}\right\} \cdot \omega_j^{(s)} \\
\leq\ & \sum_{t=\omega_j^{(s-1)}}^{T} \mathbb{E}\left[\mathbf{1}\left\{\overline{\mu}_j(t) \geq \overline{\mu}_1(t), O_j(t-1) = \omega_j^{(s-1)}\right\}\right] \cdot \omega_j^{(s)} \\
\leq\ & \sum_{t=\omega_j^{(s-1)}}^{T} \sum_{h=1}^{t-1} \mathbb{E}\left[\mathbf{1}\left\{\widetilde{\mu}_{j,\omega_j^{(s-1)}} + \sqrt{\frac{3\log(t)}{\omega_j^{(s-1)}}} + \frac{6\sqrt{8}\log(t)\cdot s}{\epsilon\cdot\omega_j^{(s-1)}} \geq \right.\right. \\
& \left.\left. \widetilde{\mu}_{1,h} + \sqrt{\frac{3\log(t)}{h}} + \frac{6\sqrt{8}\log(t)(\lfloor\log(h+1)\rfloor)}{\epsilon\cdot h}\right\}\right] \cdot \omega_j^{(s)} \quad .
\end{aligned} \tag{4.39}$$

Let Ψ be the indication function in the last inequality of (4.39). If Ψ is true, then at least one of the following is true:

$$\widetilde{\mu}_{j,\omega_j^{(s-1)}} \geq \mu_j + \sqrt{\frac{3\log(t)}{\omega_j^{(s-1)}}} + \frac{6\sqrt{8}\log(t)\cdot s}{\epsilon\cdot\omega_j^{(s-1)}} \quad ; \tag{4.40}$$

$$\widetilde{\mu}_{1,h} \leq \mu_1 - \sqrt{\frac{3\log(t)}{h}} - \frac{6\sqrt{8}\log(t)(\lfloor\log(h+1)\rfloor)}{\epsilon\cdot h} \quad ; \tag{4.41}$$

$$\frac{\Delta_j}{2} < \sqrt{\frac{3\log(t)}{\omega_j^{(s-1)}}} + \frac{6\sqrt{8}\log(t)\cdot s}{\epsilon\cdot\omega_j^{(s-1)}} \quad . \tag{4.42}$$

Before upper bounding Ψ, let us first introduce some notation. Let $\widehat{F}_{j,0:r_j(t-1)-1} := \sum_{r=0}^{r_j(t-1)-1} \sum \mathcal{F}_j^{(r)}$ be the aggregated reward of all observations inserted into arrays $\left(\mathcal{F}_j^{(0)}, \mathcal{F}_j^{(1)}, \ldots, \mathcal{F}_j^{(r_j(t-1)-1)}\right)$, i.e., the non-noisy version of $\widetilde{F}_{j,0:r_j(t-1)-1}$. Similarly, let $\widehat{B}_{j,r_j(t-1)} = \sum \mathcal{B}_j^{(r_j(t-1))}$ be the non-noisy version of $\widetilde{B}_{j,r_j(t-1)}$. Also, let $\widehat{\mu}_{j,O_j(t-1)} = \frac{\widehat{F}_{j,0:r_j(t-1)-1} + \widehat{B}_{j,r_j(t-1)}}{O_j(t-1)}$ be the true empirical mean of all these $O_j(t-1)$ observations,

i.e., the non-noisy version of $\widetilde{\mu}_{j,O_j(t-1)}$. After these preparations, we now analyze (4.40), (4.41), and (4.42) one by one.

For (4.40), we use Hoeffding's inequality and inequality (4.11). We have

$$\begin{aligned}
& \mathbb{P}\left\{\widetilde{\mu}_{j,\omega_j^{(s-1)}} \geq \mu_j + \sqrt{\frac{3\log(t)}{\omega_j^{(s-1)}}} + \frac{6\sqrt{8}\log(t)\cdot s}{\epsilon\cdot\omega_j^{(s-1)}}\right\} \\
\leq\ & \mathbb{P}\left\{\widehat{\mu}_{j,\omega_j^{(s-1)}} \geq \mu_j + \sqrt{\frac{3\log(t)}{\omega_j^{(s-1)}}}\right\} + \mathbb{P}\left\{\widetilde{\mu}_{j,\omega_j^{(s-1)}} \geq \widehat{\mu}_{j,\omega_j^{(s-1)}} + \frac{6\sqrt{8}\log(t)\cdot s}{\epsilon\cdot\omega_j^{(s-1)}}\right\} \\
\leq\ & \mathbb{P}\left\{\widehat{\mu}_{j,\omega_j^{(s-1)}} \geq \mu_j + \sqrt{\frac{3\log(t)}{\omega_j^{(s-1)}}}\right\} + \mathbb{P}\left\{\widetilde{F}_{j,0:s-2} \geq \widehat{F}_{j,0:s-2} + \frac{3\sqrt{8}\log(t)\cdot s}{0.5\epsilon}\right\} \\
+\ & \mathbb{P}\left\{\widetilde{B}_{j,s-1} \geq \widehat{B}_{j,s-1} + \frac{3\sqrt{8}\log(t)\cdot s}{0.5\epsilon}\right\} \\
\leq\ & \mathbb{P}\left\{\widehat{\mu}_{j,\omega_j^{(s-1)}} \geq \mu_j + \sqrt{\frac{4\ln(t)}{\omega_j^{(s-1)}}}\right\} + \mathbb{P}\left\{\widetilde{F}_{j,0:s-2} \geq \widehat{F}_{j,0:s-2} + \frac{3\sqrt{8}\log(t)\cdot(s-1)}{0.5\epsilon}\right\} \\
+\ & \mathbb{P}\left\{\widetilde{B}_{j,s-1} \geq \widehat{B}_{j,s-1} + \frac{3\sqrt{8}\log(t)\cdot(s-1)}{0.5\epsilon}\right\} \\
=\ & O\left(\frac{1}{t^4}\right) \quad .
\end{aligned} \tag{4.43}$$

In the last inequality of (4.43), for the first term, we use Hoeffding's inequality. Note that $3\log(t) > 4\ln(t)$. We now provide more details about how to apply inequality (4.11) for the second and the third terms. For the second term, we set $\chi = \frac{1}{0.5\epsilon}\cdot\frac{\sqrt{8}\log(t^3)(s-1)}{\sqrt{8\ln(t^4)}} = \frac{1}{0.5\epsilon}\cdot\frac{3\sqrt{\ln(t)}\cdot(s-1)}{2\ln(2)} > \frac{1}{0.5\epsilon}\cdot 2\sqrt{\ln(t)}(s-1) > \frac{1}{0.5\epsilon}\max\left\{\sqrt{s-1}, \sqrt{\ln(t^4)}\right\}$. For the third term, we set $\chi = \frac{s-1}{0.5\epsilon}\cdot\frac{\sqrt{8}\log(t^3)}{\sqrt{8\ln(t^4)}} > \frac{s-1}{0.5\epsilon}\cdot\sqrt{4\ln(t)} = \frac{s-1}{0.5\epsilon}\max\left\{\sqrt{s-1}, \sqrt{\ln(t^4)}\right\}$. Note that we always have $s-1 < \log(2^s - 1) < \log(t) < \ln(t^4)$.

Similarly, for (4.41), we have

$$\mathbb{P}\left\{\widetilde{\mu}_{1,h} \leq \mu_1 - \sqrt{\frac{3\log(t)}{h}} - \frac{6\sqrt{8}\log(t)(\lfloor\log(h+1)\rfloor)}{\epsilon\cdot h}\right\} = O\left(\frac{1}{t^4}\right) \quad . \tag{4.44}$$

We now show that (4.42) cannot be true by using contradiction.

As $2^{d_j} \geq \max\left\{\frac{256\log(T)}{\Delta_j^2}, \frac{256\log(T)}{\epsilon\cdot\Delta_j}\cdot\log\left(\frac{256\log(T)}{\epsilon\cdot\Delta_j}\right)\right\} \geq \frac{256\log(T)}{\Delta_j^2}$, the first term

of the RHS in (4.42) is upper bounded as

$$\sqrt{\frac{3\log(t)}{\omega_j^{(s-1)}}} < \sqrt{\frac{3\log(T)}{\lambda_j^{(s-1)}}} = \sqrt{\frac{3\log(T)}{2^{s-1}}} \leq \sqrt{\frac{3\log(T)}{2^{d_j+1-1}}} \leq \sqrt{\frac{3\log(T)}{\frac{256\log(T)}{\Delta_j^2}}} = \sqrt{\frac{3}{256}}\Delta_j = \frac{\sqrt{3}\Delta_j}{16} \quad . \tag{4.45}$$

To upper bound the second term in the RHS of (4.42), we construct the following decreasing function first. Let $f(x) = \frac{x}{2^x}$, where $x \geq 2$. Then we have $f'(x) = \frac{1}{2^x} - \frac{x\ln(2)}{2^x} < 0$. Let $z_j := \log\left(\frac{256\log(T)}{\epsilon\cdot\Delta_j}\cdot\log\left(\frac{256\log(T)}{\epsilon\cdot\Delta_j}\right)\right)$. Then we have $s \geq d_j + 1 \geq z_j + 1 > z_j$. Therefore, we also have

$$\frac{s}{2^s} < \frac{z_j}{2^{z_j}} = \frac{\log\left(\frac{256\log(T)}{\epsilon\cdot\Delta_j}\cdot\log\left(\frac{256\log(T)}{\epsilon\cdot\Delta_j}\right)\right)}{2^{\log\left(\frac{256\log(T)}{\epsilon\cdot\Delta_j}\cdot\log\left(\frac{256\log(T)}{\epsilon\cdot\Delta_j}\right)\right)}} \leq \frac{2\log\left(\frac{256\log(T)}{\epsilon\cdot\Delta_j}\right)}{\frac{256\log(T)}{\epsilon\cdot\Delta_j}\cdot\log\left(\frac{256\log(T)}{\epsilon\cdot\Delta_j}\right)} = \frac{\epsilon\cdot\Delta_j}{128\log(T)} \quad . \tag{4.46}$$

The second term in the RHS of (4.42) is upper bounded as

$$\frac{6\sqrt{8}\log(t)\cdot s}{\epsilon\cdot\omega_j^{(s-1)}} < \frac{6\sqrt{8}\log(t)\cdot s}{\epsilon\cdot\lambda_j^{(s-1)}} = \frac{2\cdot 6\sqrt{8}\log(t)}{\epsilon}\cdot\frac{s}{2^s} \leq \frac{12\sqrt{8}\log(T)}{\epsilon}\cdot\frac{\epsilon\cdot\Delta_j}{128\log(T)} = \frac{3\sqrt{2}\Delta_j}{16} \quad . \tag{4.47}$$

From (4.45) and (4.47), we know that the RHS in (4.42) is at most $\frac{\sqrt{3}\Delta_j}{16} + \frac{3\sqrt{2}\Delta_j}{16} < \frac{\Delta_j}{2}$, which implies (4.42) cannot be true.

We now come back to (4.39). We have

$$\begin{aligned}
& \mathbb{P}\left\{\theta_j = s\right\} \cdot \omega_j^{(s)} \\
\leq & \sum_{t=\omega_j^{(s-1)}}^{T} \sum_{h=1}^{t-1} \mathbb{E}\left[\mathbf{1}\left\{\widetilde{\mu}_{j,\omega_j^{(s-1)}} + \sqrt{\frac{3\log(t)}{\omega_j^{(s-1)}}} + \frac{6\sqrt{8}\log(t)\cdot s}{\epsilon\cdot\omega_j^{(s-1)}} \geq \right.\right. \\
& \left.\left. \widetilde{\mu}_{1,h} + \sqrt{\frac{3\log(t)}{h}} + \frac{6\sqrt{8}\log(t)(\lfloor\log(h+1)\rfloor)}{\epsilon\cdot h}\right\}\right] \cdot \omega_j^{(s)} \\
\leq & \sum_{t=\omega_j^{(s-1)}}^{T} \sum_{h=1}^{t-1} \left(\mathbb{P}\left\{(4.40)\right\} + \mathbb{P}\left\{(4.41)\right\}\right) \cdot \omega_j^{(s)} \\
\leq & \sum_{t=\omega_j^{(s-1)}}^{T} \sum_{h=1}^{t-1} O\left(\frac{1}{t^4}\right) \cdot \omega_j^{(s)} \\
\leq & \sum_{t=\omega_j^{(s-1)}}^{T} O\left(\frac{1}{t^3}\right) \cdot \omega_j^{(s)} \\
\leq & O\left(\frac{\omega_j^{(s)}}{\omega_j^{(s-1)}}\right) \\
\leq & O(1) \quad .
\end{aligned} \tag{4.48}$$

Note that $\frac{\omega_j^{(s)}}{\omega_j^{(s-1)}} = \frac{2^{s+1}-1}{2^s-1} \leq 4$. □

4.8.5 Proofs of Theorem 18

Proof of Theorem 18: Let $\boldsymbol{X}_{1:T}$ be the original reward vector sequence and $\boldsymbol{X}'_{1:T}$ be an arbitrary neighbouring reward vector sequence of $\boldsymbol{X}_{1:T}$ such that they differ in an arbitrary reward vector from each other at most. Let us say they differ in the reward vector in round t. Let $D_{1:T}$ be the sequence of decisions made through round 1 to T when taking $\boldsymbol{X}_{1:T}$ as input. Similarly, let $D'_{1:T}$ be the sequence of decisions made when taking $\boldsymbol{X}'_{1:T}$ as input.

Let $\sigma_{1:T} \in \mathcal{A}^T$. We claim that

$$\mathbb{P}\left\{D_{1:T} = \sigma_{1:T} \mid X_{1:T}\right\} \leq e^{\epsilon} \cdot \mathbb{P}\left\{D'_{1:T} = \sigma_{1:T} \mid X'_{1:T}\right\} \quad . \tag{4.49}$$

Let $\tau^{(r)}$ be the last round of epoch r, i.e., at the end of round $\tau^{(r)}$, a new action will be output by RNM. Also, let r_0 be the epoch including round t. Hence, we have $t \in \left\{\tau^{(r_0-1)}+1, \tau^{(r_0-1)}+2, \ldots, \tau^{(r_0)}\right\}$. We set $\tau^{(0)} = 0$. Note that for a fixed t, r_0 is also fixed.

The LHS of (4.50) can be rewritten as

$$\begin{aligned}\mathbb{P}\left\{D_{1:T}=\sigma_{1:T}\mid X_{1:T}\right\} &= \prod_{1\le r\le r_0}\mathbb{P}\left\{D_{\tau^{(r-1)}+1:\tau^{(r)}}=\sigma_{\tau^{(r-1)}+1:\tau^{(r)}}\mid X_{1:T}\right\}\\ &\quad \underbrace{\mathbb{P}\left\{D_{\tau^{(r_0)}+1:\tau^{(r_0+1)}}=\sigma_{\tau^{(r_0)}+1:\tau^{(r_0+1)}}\mid X_{1:T}\right\}}_{\alpha}\\ &\quad \prod_{r\ge r_0+2}\mathbb{P}\left\{D_{\tau^{(r-1)}+1:\tau^{(r)}}=\sigma_{\tau^{(r-1)}+1:\tau^{(r)}}\mid X_{1:T}\right\} \quad .\end{aligned} \tag{4.50}$$

Similarly, the RHS of (4.50) can be rewritten as

$$\begin{aligned}\mathbb{P}\left\{D'_{1:T}=\sigma_{1:T}\mid X'_{1:T}\right\} &= \prod_{1\le r\le r_0}\mathbb{P}\left\{D'_{\tau^{(r-1)}+1:\tau^{(r)}}=\sigma_{\tau^{(r-1)}+1:\tau^{(r)}}\mid X'_{1:T}\right\}\\ &\quad \underbrace{\mathbb{P}\left\{D'_{\tau^{(r_0)}+1:\tau^{(r_0+1)}}=\sigma_{\tau^{(r_0)}+1:\tau^{(r_0+1)}}\mid X'_{1:T}\right\}}_{\alpha'}\\ &\quad \prod_{r\ge r_0+2}\mathbb{P}\left\{D'_{\tau^{(r-1)}+1:\tau^{(r)}}=\sigma_{\tau^{(r-1)}+1:\tau^{(r)}}\mid X'_{1:T}\right\} \quad .\end{aligned} \tag{4.51}$$

The idea behind (4.50) and (4.54) is that the decisions made in epoch r only depends on the reward vector sequence in epoch $r-1$, i.e., the decisions made in epoch r have no dependency on the decisions made in epoch $r-1$.

Since $\boldsymbol{X}_{1:T}$ and $\boldsymbol{X}'_{1:T}$ only differ in round t at most, the decisions made from round 1 to round $\tau^{(r_0)}$ (including round $\tau^{(r_0)}$) and from round $\tau^{(r_0+1)}+1$ to the end stay the same under either $\boldsymbol{X}_{1:T}$ or $\boldsymbol{X}'_{1:T}$ conditioning on the noise injected. Therefore, in (4.50) and (4.54), only α and α' can be different. Now we analyze the relationship of α and α'.

Recall that $J^{(r)}$ indicates the action output by RNM at the end of epoch r over $\boldsymbol{X}_{1:T}$. Let $J'^{(r)}$ indicate the action output by RNM at the end of epoch r over $\boldsymbol{X}'_{1:T}$. As Lap $\left(\frac{1}{\epsilon}\right)$ noise is injected to each action j's fresh observations obtained in epoch r_0, from Claim 3.9 of [16], we have

$$\mathbb{P}\left\{J^{(r_0)}=\sigma_{\tau^{(r_0)}+1}\right\}\le e^{\epsilon}\cdot\mathbb{P}\left\{J'^{(r_0)}=\sigma_{\tau^{(r_0)}+1}\mid X_{1:T}\right\} \quad , \tag{4.52}$$

which implies

$$\mathbb{P}\left\{D_{\tau^{(r_0)}+1}=\sigma_{\tau^{(r_0)}+1}\right\}\le e^{\epsilon}\cdot\mathbb{P}\left\{D'_{\tau^{(r_0)}+1}=\sigma_{\tau^{(r_0)}+1}\mid X'_{1:T}\right\} \quad . \tag{4.53}$$

As the decisions made in rounds $\left\{\tau^{(r_0)}+1,\tau^{(r_0)}+2,\ldots,\tau^{(r_0+1)}\right\}$ stay the same,

we have

$$\begin{aligned} & \mathbb{P}\left\{D_{\tau^{(r_0)}+1:\tau^{(r_0+1)}} = \sigma_{\tau^{(r_0)}+1:\tau^{(r_0+1)}} \mid D_{\tau^{(r_0)}+1} = \sigma_{\tau^{(r_0)}+1}, X_{1:T}\right\} \\ = \; & \mathbb{P}\left\{D'_{\tau^{(r_0)}+1:\tau^{(r_0+1)}} = \sigma_{\tau^{(r_0)}+1:\tau^{(r_0+1)}} \mid D'_{\tau^{(r_0)}+1} = \sigma_{\tau^{(r_0)}+1}, X'_{1:T}\right\} \quad . \end{aligned} \tag{4.54}$$

By combining (4.53) and (4.54), we have $\alpha \leq e^{\epsilon}\alpha'$, which concludes the proof. □

4.8.6 Proofs of Theorem 19

Proof of Theorem 19: Let $r_{\max} := \left\lceil \log\left(\frac{1}{\Delta_{\min}}\right)\right\rceil$. We partition all actions in $\mathcal{A}$ to $r_{\max}$ groups based on the mean reward gaps. Let $\Phi^{(r)} := \left\{j \in \mathcal{A} : 0.5^r < \Delta_j \leq 0.5^{r-1}\right\}$, where $1 \leq r \leq r_{\max}$, collect all actions with gaps between $\left(0.5^r, 0.5^{r-1}\right]$. Note that playing any action in $\Phi^{(r)}$ will suffer at most 0.5^{r-1} regret in a single round. Let $\lambda^{(r)} := \frac{8\log(K)}{\min\{0.5^{2r}, \epsilon\cdot 0.5^r\}}$, where $1 \leq r \leq r_{\max}$. Let $(d_1, d_2, \ldots, d_{r_{\max}})$ be a sequence of non-decreasing positive integers, where $d_r := \left\lceil \log\left(\lambda^{(r)}\right)\right\rceil$.

We partition all T rounds into $(r_{\max}+1)$ phases. For a phase $r \leq r_{\max}$, it contains epochs $d_{r-1}+1, d_r+2, \ldots, d_r$. For the last phase $r = r_{\max}+1$, it contains all the epochs from epoch $d_{r_{\max}}+1$ to the end. We set $d_0 = 0$.

We upper bound the regret phase by phase. For all epochs in phase $r \leq r_{\max}$, the total regret $R^{(r)}$ is at most

$$R^{(r)} \leq \sum_{s=d_{r-1}+1}^{d_r} 2^s \cdot 0.5^{r-1} + \sum_{s=d_{r-1}+1}^{d_r} 2^s \cdot \sum_{q=1}^{r-1} \mathbb{P}\left\{J^{(s-1)} \in \Phi^{(q)}\right\} \cdot 0.5^{q-1} \quad . \tag{4.55}$$

Note that the first term in (4.55) uses the fact that playing any action in groups $\Phi^{(r)}, \Phi^{(r+1)}, \ldots, \Phi^{(r_{\max})}$ will suffer at most 0.5^{r-1} regret in a single round.

For all the epochs in the the last phase $r = r_{\max}+1$, the total regret is at most

$$R^{(r_{\max}+1)} \leq \sum_{s=d_{r_{\max}}+1}^{\log(T)} 2^s \cdot \sum_{q=1}^{r_{\max}} \mathbb{P}\left\{J^{(s-1)} \in \Phi^{(q)}\right\} \cdot 0.5^{q-1} \quad . \tag{4.56}$$

By taking a summation of the regret over all phases, we know that the regret by

the end of round T is at most

$$
\begin{aligned}
& \mathcal{R}(T) \\
\leq & \sum_{r=1}^{r_{\max}} R^{(r)} + R^{(r_{\max}+1)} \\
\leq & \sum_{r=1}^{r_{\max}} \left(\sum_{s=d_{r-1}+1}^{d_r} 2^s \cdot 0.5^{r-1} + \sum_{s=d_{r-1}+1}^{d_r} 2^s \cdot \sum_{q=1}^{r-1} \mathbb{P}\left\{ J^{(s-1)} \in \Phi^{(q)} \right\} \cdot 0.5^{q-1} \right) \\
+ & \sum_{s=d_{r_{\max}}+1}^{\log(T)} 2^s \cdot \sum_{q=1}^{r_{\max}} \mathbb{P}\left\{ J^{(s-1)} \in \Phi^{(q)} \right\} \cdot 0.5^{q-1} \\
\leq & \underbrace{\sum_{r=1}^{r_{\max}} \sum_{s=d_{r-1}+1}^{d_r} 2^s \cdot 0.5^{r-1}}_{=:\Gamma} \\
+ & \underbrace{\sum_{r=1}^{r_{\max}} \sum_{s=d_{r-1}+1}^{d_r} 2^s \sum_{q=1}^{r-1} \mathbb{P}\left\{ J^{(s-1)} \in \Phi^{(q)} \right\} 0.5^{q-1}}_{=:\Lambda_1} \\
+ & \underbrace{\sum_{s=d_{r_{\max}}+1}^{\log(T)} 2^s \sum_{q=1}^{r_{\max}} \mathbb{P}\left\{ J^{(s-1)} \in \Phi^{(q)} \right\} 0.5^{q-1}}_{=:\Lambda_2} \quad .
\end{aligned}
\tag{4.57}
$$

We now upper bound term Γ in (4.57) as

$$
\begin{aligned}
\Gamma &= \sum_{r=1}^{r_{\max}} \sum_{s=d_{r-1}+1}^{d_r} 2^s \cdot 0.5^{r-1} \\
&= \sum_{r=1}^{r_{\max}} \left(2^{d_{r-1}+1} + 2^{d_{r-1}+2} + \ldots + 2^{d_r} \right) \cdot 0.5^{r-1} \\
&= \left(2^1 + 2^2 + \ldots + 2^{d_1} \right) + \left(2^{d_1} + 2^{d_1+1} + \ldots + 2^{d_2-1} \right) \\
&+ \left(2^{d_2-1} + 2^{d_2} + \ldots + 2^{d_3-2} \right) \\
&+ \left(2^{d_3-2} + 2^{d_3-1} + \ldots + 2^{d_4-3} \right) + \ldots \\
&+ \left(2^{d_{r_{\max}-1}+2-r_{\max}} + 2^{d_{r_{\max}-1}+3-r_{\max}} + \ldots + 2^{d_{r_{\max}}+1-r_{\max}} \right) \\
&\leq 2 \left(2^1 + 2^2 + \ldots 2^{d_{r_{\max}}+1-r_{\max}} \right) \\
&\leq 4 \cdot 2^{d_{r_{\max}}+1-r_{\max}} \\
&= 8 \cdot 2^{d_{r_{\max}}-r_{\max}} \\
&\leq O\left(\frac{\log(K)}{\min\{\Delta_{\min}, \epsilon\}} \right) \quad .
\end{aligned}
\tag{4.58}
$$

The last inequality in (4.58) uses the inequalities that $2^{-r_{\max}} \leq \Delta_{\min}$ and $2^{d_{r_{\max}}} \leq \frac{64 \log(K)}{\Delta_{\min} \cdot \min\{\Delta_{\min}, \epsilon\}}$. For the proof of the first inequality, from $\log\left(\frac{1}{\Delta_{\min}}\right) \leq r_{\max} \leq$

$\log\left(\frac{1}{\Delta_{\min}}\right)+1$, we have $\frac{1}{\Delta_{\min}} \leq 2^{r_{\max}} \leq \frac{2}{\Delta_{\min}}$ and $\Delta_{\min} \geq 0.5^{r_{\max}} \geq 0.5\Delta_{\min}$. For the proof of the second inequality, we have

$$\begin{aligned} 2^{d_{r_{\max}}} &= 2^{\lceil \log\left(\lambda^{(r_{\max})}\right) \rceil} \leq 2 \cdot 2^{\log\left(\lambda^{(r_{\max})}\right)} = 2\lambda^{(r_{\max})} \\ &= \frac{16\log(K)}{\min\{0.5^{2r_{\max}}, \epsilon \cdot 0.5^{r_{\max}}\}} = \frac{16 \cdot 2^{r_{\max}} \cdot \log(K)}{\min\{0.5^{r_{\max}}, \epsilon\}} < \frac{16 \cdot \frac{2}{\Delta_{\min}} \cdot \log(K)}{\min\{0.5\Delta_{\min}, 0.5\epsilon\}} = \frac{64\log(K)}{\Delta_{\min} \cdot \min\{\Delta_{\min}, \epsilon\}} \quad . \end{aligned}$$

Let $\Lambda := \Lambda_1 + \Lambda_2$. We now upper bound Λ in (4.57) as

$$\begin{aligned} \Lambda &= \sum_{r=1}^{r_{\max}} \sum_{s=d_{r-1}+1}^{d_r} 2^s \cdot \sum_{q=1}^{r-1} \mathbb{P}\left\{J^{(s-1)} \in \Phi^{(q)}\right\} \cdot 0.5^{q-1} \\ &+ \sum_{s=d_{r_{\max}}+1}^{\log(T)} 2^s \cdot \sum_{q=1}^{r_{\max}} \mathbb{P}\left\{J^{(s-1)} \in \Phi^{(q)}\right\} \cdot 0.5^{q-1} \\ &= \sum_{q=1}^{r_{\max}} \sum_{s=d_q+1}^{\log(T)} 2^s \cdot \mathbb{P}\left\{J^{(s-1)} \in \Phi^{(q)}\right\} \cdot 0.5^{q-1} \\ &= \sum_{r=1}^{r_{\max}} \sum_{s=d_r+1}^{\log(T)} 2^s \cdot \mathbb{P}\left\{J^{(s-1)} \in \Phi^{(r)}\right\} \cdot 0.5^{r-1} \\ &\leq \sum_{r=1}^{r_{\max}} \sum_{j \in \Phi^{(r)}} \underbrace{\sum_{s=d_r+1}^{\log(T)} 2^s \cdot \mathbb{P}\left\{J^{(s-1)} = j\right\}}_{\zeta} \cdot 0.5^{r-1} \quad . \end{aligned} \tag{4.59}$$

The second equality in (4.59) simply rearranges the summation order. In the last inequality, we use the union bound and Lemma 11 to upper bound the term ζ. Lemma 11 states that in all epochs from phase $r+1$ onwards, the probability of playing any action in group $\Phi^{(r)}$ is very low.

Lemma 11. *For any $j \in \Phi^{(r)}$, we have*

$$\sum_{s=d_r+1}^{\log(T)} 2^s \cdot \mathbb{P}\left\{J^{(s-1)} = j\right\} \leq O\left(\frac{1}{K \cdot \min\{0.5^{2r}, \epsilon \cdot 0.5^r\}}\right) \quad . \tag{4.60}$$

We defer the proof of Lemma 11 to the end of this section. We now upper bound

the term Λ in (4.57). We have

$$\begin{aligned}
\Lambda &= \sum_{r=1}^{r_{\max}} \sum_{j\in\Phi^{(r)}} \sum_{s=d_r+1}^{\log(T)} 2^s \cdot \mathbb{P}\left\{J^{(s-1)} = j\right\} \cdot 0.5^{r-1} \\
&\leq \sum_{r=1}^{r_{\max}} \sum_{j\in\Phi^{(r)}} O\left(\frac{1}{K\cdot\min\{0.5^{2r}, 0.5^r\epsilon\}}\right) \cdot 0.5^{r-1} \\
&\leq \sum_{r=1}^{r_{\max}} \sum_{j\in\Phi^{(r)}} O\left(\frac{1}{K\cdot\min\{0.5^r, \epsilon\}}\right) \\
&\leq \sum_{r=1}^{r_{\max}} \sum_{j\in\Phi^{(r)}} O\left(\frac{1}{K\cdot\min\{\Delta_{\min}, \epsilon\}}\right) \\
&\leq O\left(\frac{1}{\min\{\Delta_{\min}, \epsilon\}}\right) \quad .
\end{aligned} \tag{4.61}$$

We now prove Theorem 19. By plugging (4.58) and (4.61) into (4.57), we have

$$\mathcal{R}(T) \leq O\left(\frac{\log(K)}{\min\{\Delta_{\min}, \epsilon\}}\right) + O\left(\frac{1}{\min\{\Delta_{\min}, \epsilon\}}\right) = O\left(\frac{\log(K)}{\min\{\Delta_{\min}, \epsilon\}}\right) \quad , \tag{4.62}$$

which concludes the proof of Theorem 19. □

Proof of Lemma 11: Recall that $\widehat{\mu}_{j,2^{s-1}}$ is the true empirical mean of action j among 2^{s-1} observations. We have

$$\begin{aligned}
& \mathbb{P}\left\{J^{(s-1)} = j\right\} \\
\leq\ & \mathbb{P}\left\{\widetilde{\mu}_{j,2^{s-1}} \geq \widetilde{\mu}_{1,2^{s-1}}\right\} \\
\leq\ & \mathbb{P}\left\{\widehat{\mu}_{j,2^{s-1}} + \frac{Y_j}{2^{s-1}} \geq \widehat{\mu}_{1,2^{s-1}} + \frac{Y_1}{2^{s-1}}\right\} \\
\leq\ & \mathbb{P}\left\{\widehat{\mu}_{j,2^{s-1}} + \frac{Y_j}{2^{s-1}} \geq \mu_j + \frac{\Delta_j}{2}\right\} + \mathbb{P}\left\{\widehat{\mu}_{1,2^{s-1}} + \frac{Y_1}{2^{s-1}} \leq \mu_1 - \frac{\Delta_j}{2}\right\} \\
\leq\ & \mathbb{P}\left\{\widehat{\mu}_{j,2^{s-1}} \geq \mu_j + \frac{\Delta_j}{4}\right\} + \mathbb{P}\left\{\frac{Y_j}{2^{s-1}} \geq \frac{\Delta_j}{4}\right\} \\
+\ & \mathbb{P}\left\{\widehat{\mu}_{1,2^{s-1}} \leq \mu_1 - \frac{\Delta_j}{4}\right\} + \mathbb{P}\left\{\frac{Y_1}{2^{s-1}} \leq -\frac{\Delta_j}{4}\right\} \quad ,
\end{aligned} \tag{4.63}$$

where Y_j and Y_1 are i.i.d. according to Lap $\left(\frac{1}{\epsilon}\right)$.

Hoeffding's inequality can be applied to the first and third term in the last inequality of (4.63), giving

$$\mathbb{P}\left\{\widehat{\mu}_{j,2^{s-1}} \geq \mu_j + \frac{\Delta_j}{4}\right\} \leq O\left(e^{-2^{s-1}\cdot\Delta_j^2\cdot\frac{1}{8}}\right) \tag{4.64}$$

and

$$\mathbb{P}\left\{\widehat{\mu}_{1,2^{s-1}} \leq \mu_1 - \frac{\Delta_j}{4}\right\} \leq O\left(e^{-2^{s-1}\cdot\Delta_j^2\cdot\frac{1}{8}}\right) \quad . \tag{4.65}$$

From (5.9), we upper bound the second and fourth term in the last inequality of (4.63) as

$$\mathbb{P}\left\{\frac{Y_j}{2^{s-1}} \geq \frac{\Delta_j}{4}\right\} \leq O\left(e^{-\epsilon\Delta_j\cdot 2^{s-1}\cdot\frac{1}{4}}\right) \tag{4.66}$$

and

$$\mathbb{P}\left\{\frac{Y_1}{2^{s-1}} \leq -\frac{\Delta_j}{4}\right\} \leq O\left(e^{-\epsilon\Delta_j\cdot 2^{s-1}\cdot\frac{1}{4}}\right) \quad . \tag{4.67}$$

By plugging (4.64), (4.65), (4.66), and (4.67) to (4.63), we have

$$\mathbb{P}\left\{J^{(s-1)} = j\right\} \leq O\left(e^{-2^{s-1}\cdot\min\left\{\Delta_j^2,\epsilon\Delta_j\right\}\cdot\frac{1}{8}}\right) \quad . \tag{4.68}$$

We now come back to the proof of Lemma 11. We have

$$
\begin{aligned}
\text{LHS of (4.60)} &= \sum_{s=d_r+1}^{\log(T)} 2^s \cdot \mathbb{P}\left\{J^{(s-1)} = j\right\} \\
&\leq \sum_{s=d_r+1}^{\log(T)} 2^s \cdot O\left(e^{-2^{s-1}\cdot\min\left\{\Delta_j^2,\epsilon\Delta_j\right\}\cdot\frac{1}{8}}\right) \\
&\leq \sum_{s=d_r+1}^{\log(T)} O\left(2^s \cdot e^{-2^{s-1}\cdot\min\left\{\Delta_j^2,\epsilon\Delta_j\right\}\cdot\frac{1}{8}}\right) \\
&\leq \sum_{s=d_r}^{\log(T)} O\left(2^s \cdot e^{-2^{s}\cdot\min\left\{\Delta_j^2,\epsilon\Delta_j\right\}\cdot\frac{1}{8}}\right) \\
&\leq \int_{d_r}^{\log(T)} O\left(2^s \cdot e^{-2^{s}\cdot\min\left\{\Delta_j^2,\epsilon\Delta_j\right\}\cdot\frac{1}{8}}\right) d_s \\
&= O\left(-\frac{e^{-\min\left\{\Delta_j^2,\epsilon\Delta_j\right\}2^s\cdot\frac{1}{8}}}{\frac{1}{8}\cdot\min\left\{\Delta_j^2,\epsilon\Delta_j\right\}\ln(2)}\Big|_{d_r}^{\log(T)}\right) \\
&\leq O\left(\frac{e^{-\min\left\{\Delta_j^2,\epsilon\Delta_j\right\}2^{d_r}\cdot\frac{1}{8}}}{\min\left\{\Delta_j^2,\epsilon\Delta_j\right\}}\right) \\
&\leq O\left(\frac{e^{-\min\left\{\Delta_j^2,\epsilon\Delta_j\right\}\cdot\lambda^{(r)}\cdot\frac{1}{8}}}{\min\left\{\Delta_j^2,\epsilon\Delta_j\right\}}\right) \\
&= O\left(\frac{e^{-\log(K)}}{\min\left\{\Delta_j^2,\epsilon\Delta_j\right\}}\right) \\
&= O\left(\frac{1}{K\cdot\min\left\{\Delta_j^2,\epsilon\Delta_j\right\}}\right) \\
&\leq O\left(\frac{1}{K\cdot\min\{0.5^{2r},\epsilon\cdot0.5^r\}}\right) \quad ,
\end{aligned}
\tag{4.69}
$$

which concludes the proof. □

4.8.7 Additional experimental results

Figure 4.9, Figure 4.10, and Figure 4.11 show the performance for mean reward setting 1 when setting $\epsilon = 0.1, 0.25, 128$. Figure 4.12 shows the final regret of all the algorithms with all choices of ϵ in the mean reward setting 2 while Figure 4.13, Figure 4.14, Figure 4.15, and Figure 4.16 show the individual performance when setting $\epsilon = 0.1, 0.25, 1, 128$.

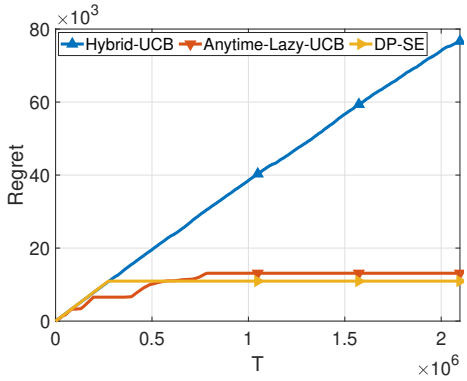

Figure 4.9: regret with $\epsilon = 0.1$ for mean reward setting 1

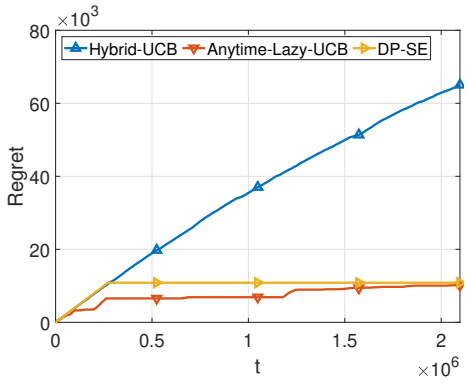

Figure 4.10: regret with $\epsilon = 0.25$ for mean reward setting 1

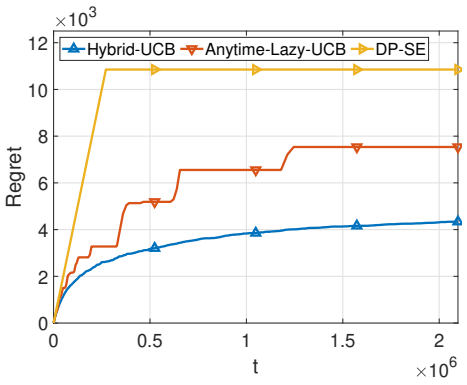

Figure 4.11: regret with $\epsilon = 128$ for mean reward setting 1

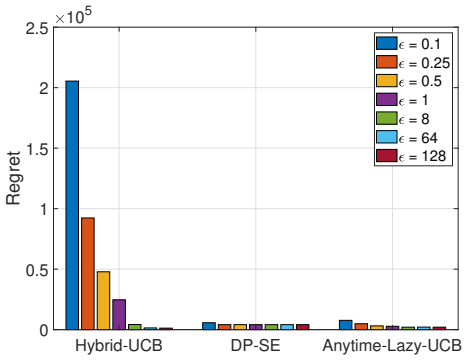

Figure 4.12: Mean reward setting 2: final regret for all ϵ

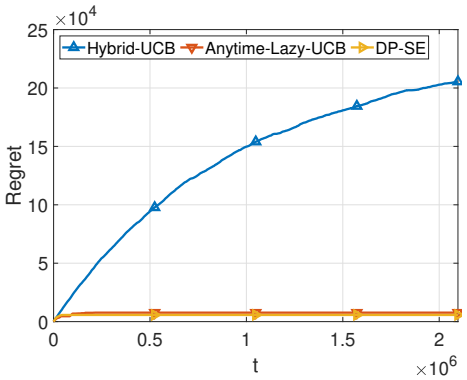

Figure 4.13: regret with $\epsilon = 0.1$ for mean reward setting 2

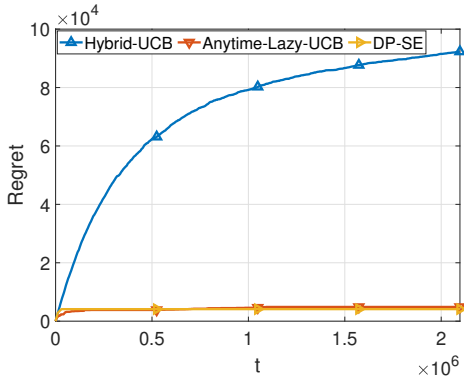

Figure 4.14: regret with $\epsilon = 0.25$ for mean reward setting 2

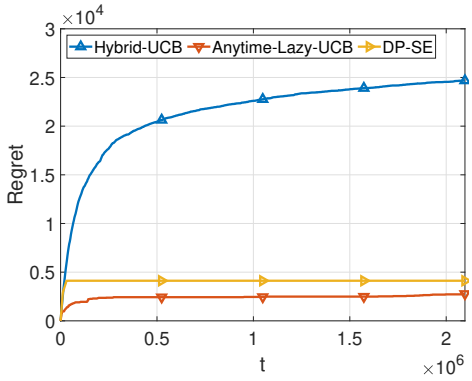

Figure 4.15: regret with $\epsilon = 1$ for mean reward setting 2

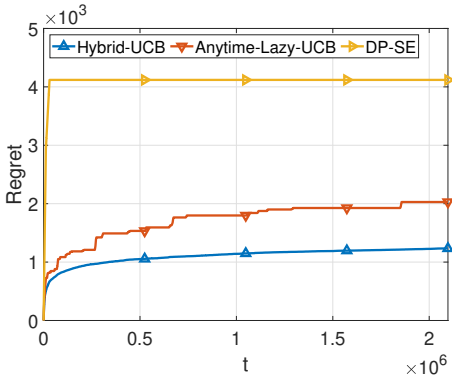

Figure 4.16: regret with $\epsilon = 128$ for mean reward setting 2

Chapter 5

Bi-Level Bandits: A Multi-Armed Bandit Problem with Unknown Arms

This chapter investigates a novel variant of stochastic multi-armed bandits: bi-level bandits with unknown arms. In this hierarchical setting, there are two levels of arms but only Level-I arms are visible to Learner. Level-II arms remain hidden and cannot be pulled nor observed directly by Learner. However, each Level-II arm is managed by a Level-I arm. Regarding the learning protocol, in each round, Learner first pulls a Level-I arm. Then, the selected Level-I arm pulls a Level-II arm. Note that the environment only reveals the reward of the pulled Level-II arm to the selected Level-I arm instead of to Learner. The goals of both Learner and the selected Level-I arm are to accumulate reward as much as possible. In this chapter, we also investigate a differential private version of bi-level bandits. Our design guarantees that both the algorithms of Learner and Level-I arms are differentially private.

5.1 Introduction

The stochastic multi-armed bandit problem is a classical sequential learning game. In this game, we have a Learner, a fixed and known arm set $\mathcal{A}$ with size K, and a stochastic environment. In round t, the environment generates a reward vector $X_t := (X_1(t), \ldots, X_K(t))$ according to fixed but unknown probability distribution(s). Simultaneously, Learner pulls an arm I_t directly from the arm set. Then, the environment reveals the reward of the pulled arm to Learner and Learner obtains

a random reward $X_{I_t}(t)$. Learner plays this game repeatedly for T rounds with the goal to accumulate as much reward as possible. In this classical setting, Learner can always pull an arm directly from the known arm set and use the obtained rewards to make future decisions. However, in some cases, Learner may not be able to pull an individual arm directly nor observe the outcome of the pulled arm.

A motivating application is a sequential investment problem in the medical sector. Suppose there are multiple drugs that are under testing (each drug being an arm and pulling an arm meaning conducting a trial on that drug) by multiple pharmaceutical companies, and a foundation (Learner) plans to invest resources in these pharmaceutical companies to assist the testing of drugs. Each period, the foundation selects a company and provides it some support. Then, the selected company can use the support to test the drugs. Critically, the selected company may not want to reveal the outcomes of a specific drug, particularly, a drug that is potentially very inferior, as it may prevent the company from receiving further support from the foundation. Therefore, it is vital that the outcomes of a particular drug are invisible to the foundation. Since the selected company would like to use the allocated support to test the drugs that will succeed with high chance and the foundation would like to invest in the companies that also succeed with high chance, they both have the goal to accumulate reward as much as possible. Note that if a clinical trial is successful, we say that a reward is obtained.

In this work, we consider a new variant of stochastic bandits in which Learner does not directly pull and learn the outcomes of a specific arm —*Bi-Level Bandits with Unknown Arms*. In this setting, we have *two levels of arms*. The arms in the first level are known to Learner and can be pulled directly by Learner (e.g., the pharmaceutical companies in the aforementioned example). In contrast, the arms in the second level are unknown to Learner, i.e., these arms cannot be pulled nor be observed directly by Learner (e.g., the drugs in the aforementioned example). However, each arm in the second level is managed by an arm in the first level. In each round, Learner first pulls an arm in the first level and then the selected first-level arm pulls an arm in the second level. At the end of the round, the environment reveals the reward of the pulled second-level arm only to the selected first-level arm instead of to Learner. Learner can have a reward reported by the selected first-level arm without disclosing the identity of the pulled second-level arm. By introducing the intermediate level, the outcomes of a particular (second-level) arm can be hidden.

Since drug trials involve testing over volunteers, the act of revealing outcomes directly to Learner might also compromise the privacy of individuals. Therefore, the learning algorithm run by each intermediate level and Learner should also preserve the privacy of individuals. Fortunately, differentially private learning algorithms [16] provide a tool to tackle this kind of event-level privacy concern. In this work, we also consider a differentially private version of bi-level bandits, which guarantees that an external observer is very unlikely to infer the true information associated with an individual from the output of the learning algorithms. Motivated by other real-world applications, various works [32, 33, 37, 21] have explored differentially private stochastic multi-armed bandit problems.

It is important to note that although the rewards for each second-level arm are i.i.d. over time, the rewards for a specific first-level arm are non-i.i.d. as the first-level arm's decision about which second-level arm to pull may change over time. Therefore, it is non-trivial to devise a learning algorithm with an $O(\log(T))$ regret bound for bi-level bandits. In this paper, we show the possibility to have a learning algorithm with an $O(\log(T))$ regret bound in this non-i.i.d. environment. The idea behind our learning algorithm is to construct asymmetric confidence intervals for the first-level arms. To tackle the challenge that a first-level arm does not have a fixed true mean, we construct a confidence interval around the mean reward of the best second-level arm managed by this specific first-level arm.

We now list the key contributions in this paper:

1. We propose a new bandit setting, bi-level bandits with unknown arms, and devise a two-level elimination algorithm with

$$\sum_{j\in\mathcal{A}} \sum_{i\in\mathcal{A}_j:\max\left\{\Delta_j,\Delta_i^{(j)}\right\}>0} O\left(\frac{\log(KT)}{\max\left\{\Delta_j,\Delta_i^{(j)}\right\}}\right)$$

   regret bound, where $\Delta_i^{(j)}$ is the performance loss when a first-level arm j pulls a sub-optimal second-level arm (j,i) instead of the optimal second-level arm (j,i^*), Δ_j is the minimum performance loss when Learner pulls a sub-optimal first-level arm j, and K is the total number of arms in the second level ;

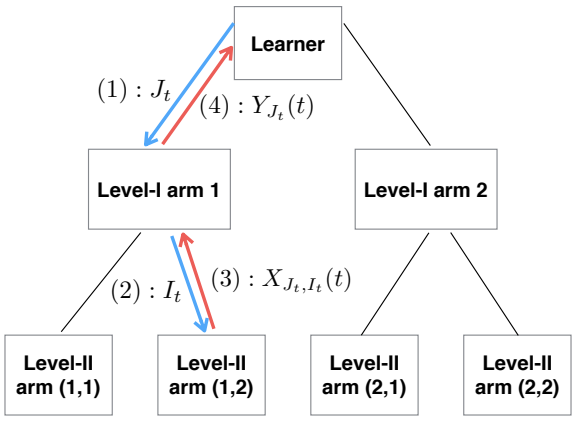

Figure 5.1: Learning Model of Bi-Level Bandits

2. We present a differentially private two-level elimination algorithm for which

$$\sum_{j\in\mathcal{A}} \sum_{i\in\mathcal{A}_j:\max\left\{\Delta_j,\Delta_i^{(j)}\right\}>0} O\left(\log(KT)\cdot\max\left\{\frac{1}{\max\left\{\Delta_j,\Delta_i^{(j)}\right\}},\frac{1}{\epsilon}\right\}\right)$$

regret bound is achievable, where $\epsilon > 0$ is the required privacy parameter ;

3. Our proposed differentially private two-level elimination algorithm can be converted to a new, simpler, and optimal private elimination algorithm for stochastic bandits that is different from the one proposed in [33] ;

4. We conduct experiments to show the practical performance of our proposed learning algorithms and to verify our theoretical results .

5.2 Bi-Level Bandits

In this section, we first present the learning problem of bi-level bandits. Then, we present our proposed learning algorithm, Two-Level Elimination Algorithm. At the end of this section, we present regret analysis of our proposed algorithm.

5.2.1 Bi-Level Bandits Learning Problem

In a stochastic bi-level bandits problem, we have a hierarchical arm structure with two levels of arms. Level-I arms are visible to Learner and can be pulled directly.

These arms are collected into a set $\mathcal{A}$ with size m. Each Level-I arm $j \in \mathcal{A}$ manages a set $\mathcal{A}_j$ of size k_j Level-II arms that are unknown to Learner and can only be pulled by Level-I arm j. Note that all Level-II arm sets are disjoint.

At the beginning of round $t = 1, 2, \ldots, T$, the environment generates a random reward $X_{j,i}(t) \in [0, 1]$ from an unknown but fixed distribution with mean $\mu_{j,i}$ for each Level-II arm (j, i). Simultaneously, Learner pulls a Level-I arm $J_t \in \mathcal{A}$ first and then J_t pulls a Level-II arm $I_t \in \mathcal{A}_{J_t}$. Let (J_t, I_t) indicate the decision pair in round t to index the pulled Level-II arm. After pulling (J_t, I_t), the environment reveals the reward of the pulled Level-II $X_{J_t,I_t}(t)$ only to Level-I arm J_t. At the end of round t, Learner realizes a reward $Y_{J_t}(t) = X_{J_t,I_t}(t)$ reported by J_t. Although the rewards $X_{J_t,I_t}(t)$ are i.i.d. over time, at Learner's side, the rewards $Y_{J_t}(t)$ are non-i.i.d. over time as J_t's decision about which Level-II arm to pull may change over time. This game will be played repeatedly for T rounds. Both Learner and the selected Level-I arm would like to accumulate as much reward as possible. Figure 5.1 provides a graphical illustration of the learning protocol. Note that I_t's identity is never disclosed to Learner. Consequently, even if Learner learns $Y_{J_t}(t)$, it cannot infer I_t nor $X_{J_t,I_t}(t)$.

For each Level-I arm $j \in \mathcal{A}$, let $(j, i^*) \leftarrow \arg\max_{i \in \mathcal{A}_j} \mu_{j,i}$ be the best Level-II arm managed by Level-I arm j. Let $\Delta_i^{(j)} := \mu_{j,i^*} - \mu_{j,i}$ be the mean reward gap between a sub-optimal Level-II arm (j, i) and the best Level-II arm (j, i^*). The gap $\Delta_i^{(j)}$ measures the performance loss in a single round when Level-I arm j pulls (j, i) instead of (j, i^*). Let $j^* = \arg\max_{j \in \mathcal{A}} \mu_{j,i^*}$ be the best Level-I arm, which is defined as the Level-I arm that has the Level-II arm with the highest mean reward. Let $\Delta_j := \mu_{j^*,i^*} - \mu_{j,i^*}$ be the mean reward gap of sub-optimal Level-I arm j. The gap Δ_j measures the minimum performance loss in a single round when Learner pulls a sub-optimal Level-I arm j. Note that the mean reward gap between Level-II arms (j^*, i^*) and (j, i) is $\mu_{j^*,i^*} - \mu_{j,i} = \Delta_j + \Delta_i^{(j)}$. To ease the presentation, we assume that j^* and each (j, i^*) are unique. However, this assumption can be removed by using arguments in [6].

We use (pseudo)-regret $\mathcal{R}(T)$ to measure the performance of our developed learning algorithms, which is defined as the expected cumulative performance loss

between pulling (j^*, i^*) and (J_t, I_t). Mathematically, the regret can be expressed as

$$\begin{aligned} \mathcal{R}(T) &= \max_{j\in\mathcal{A}, i\in\mathcal{A}_j} \mathbb{E}\left[\sum_{t=1}^{T} X_{j,i}(t) - \sum_{t=1}^{T} X_{J_t,I_t}(t)\right] \\ &= \sum_{j\in\mathcal{A}}\sum_{i\in\mathcal{A}_j} \mathbb{E}\left[\sum_{t=1}^{T} \mathbf{1}\{(J_t, I_t) = (j,i)\}\right] \cdot \left(\Delta_j + \Delta_i^{(j)}\right) \quad . \end{aligned} \tag{5.1}$$

5.2.2 Two-Level Elimination Algorithm

One may think that a naive approach that lets both Learner and each Level-I arm run the *round*-based UCB may work. In reality, this naive combination may not have a provable non-trivial regret bound as each Level-I arm does not have a fixed true mean due to the random decisions made about which Level-II arm to pull. Note that one of the key reasons why UCB works well both theoretically and practically is that the upper confidence bound of a specific arm is no smaller than the corresponding true mean with high probability. However, in bi-level bandits, since a Level-I arm does not have a fixed true mean, it is very challenging to construct a confidence interval for a Level-I arm.

We now present our two-level elimination learning algorithm, Algorithm 12. The key idea behind the algorithm is to construct an asymmetric confidence interval around μ_{j,i^*}, the best achievable mean reward for Level-I arm j, by using the idea of *abandoning observations*. The dropping of observations contributes to controlling the deviation of the empirical mean of a Level-I arm.

The learning algorithm progresses in epochs and in epoch r, Learner maintains an active Level-I arm set $\mathcal{A}^{(r)} \subseteq \mathcal{A}$ and each active Level-I arm $j \in \mathcal{A}^{(r)}$ maintains an active Level-II arm set $\mathcal{A}_j^{(r)} \subseteq \mathcal{A}_j$. Let $K := \sum_{j=1}^{m} k_j$ be the total number of Level-II arms and $L^{(r)} := 2\log(KT) \cdot 2^{2r}$, where $r \geq 1$, be the number of pulls of an active Level-II arm in epoch r. When epoch r starts, Learner allocates each active Level-I arm j exactly $L^{(r)} \cdot \left|\mathcal{A}_j^{(r)}\right|$ rounds to ensure that Level-I arm j can pull each of its active Level-II arm exactly $L^{(r)}$ times (Line 4 in Algorithm 12).

Line 5 in Algorithm 12 presents the algorithm run by each active Level-I arm j. Let $\mathcal{O}_{j,i}^{(r)}$ collect all the obtained observations from a Level-II arm $i \in \mathcal{A}_j^{(r)}$ within *only* epoch r and $\mathcal{O}_j^{(r)}$ collect all the obtained observations accounting for all active Level-II arms managed by $j \in \mathcal{A}^{(r)}$ within only epoch r. After collecting these observations, each active Level-I arm $j \in \mathcal{A}^{(r)}$ first reports the empirical mean $\widehat{\theta}_j^{(r)}$

among all observations collected in $\mathcal{O}_j^{(r)}$ [1]. Then, the active Level-I arm $j \in \mathcal{A}^{(r)}$ will eliminate the bad Level-II arm $i \in \mathcal{A}_j^{(r)}$ if its empirical mean $\widehat{\mu}_{j,i}^{(r)}$ (among observations collected in $\mathcal{O}_{j,i}^{(r)}$) satisfies the rule:

$$\widehat{\mu}_{j,i}^{(r)} + \tfrac{1}{2^r} < \max_{i' \in \mathcal{A}_j^{(r)}} \left(\widehat{\mu}_{j,i'}^{(r)} - \tfrac{1}{2^r} \right) \quad . \tag{5.2}$$

To compute $\widehat{\mu}_{j,i}^{(r)}$, we only consider those observations obtained in epoch r, i.e., all observations collected in $\mathcal{O}_{j,i}^{(r)}$; see (5.5) and (5.6) later for more details about why dropping observations obtained in previous epochs contributes to controlling the deviation of the empirical mean of an active Level-I arm.

To achieve a good theoretical guarantee, Learner eliminates a Level-I arm $j \in \mathcal{A}^{(r)}$ based on $\widehat{\theta}_j^{(r)}$, the empirical mean reported in epoch r by an active Level-I arm $j \in \mathcal{A}^{(r)}$. Learner will eliminate a bad Level-I arm $j \in \mathcal{A}^{(r)}$ if its empirical mean $\widehat{\theta}_j^{(r)}$ satisfies the rule:

$$\widehat{\theta}_j^{(r)} + 2 \cdot 8 \cdot \tfrac{1}{2^r} < \max_{j' \in \mathcal{A}^{(r)}} \left(\widehat{\theta}_{j'}^{(r)} - 8 \cdot \tfrac{1}{2^r} \right) \quad . \tag{5.3}$$

Note that the absolute values of the additive terms at both sides in (5.3) are not equal. Let $\theta_j^{(r)} := \mathbb{E}\left[\widehat{\theta}_j^{(r)}\right]$ be the expected value of $\widehat{\theta}_j^{(r)}$. Two arguments support the idea in (5.3).

The first argument is, from Hoeffding's inequality, with high probability, we have

$$\theta_j^{(r)} - 8 \cdot \tfrac{1}{2^r} \leq \widehat{\theta}_j^{(r)} \leq \theta_j^{(r)} + 8 \cdot \tfrac{1}{2^r} \quad , \tag{5.4}$$

which states that the empirical mean $\widehat{\theta}_j^{(r)}$ will not be too far from its expected value $\theta_j^{(r)}$.

The second argument is, with high probability, we have

$$\mu_{j,i^*} - 8 \cdot \tfrac{1}{2^r} \leq \theta_j^{(r)} \leq \mu_{j,i^*} \quad , \tag{5.5}$$

which states that $\theta_j^{(r)}$ will not be too far from its best achievable true mean μ_{j,i^*}.

[1] It is equivalent to reporting the aggregated reward among all these $\mathcal{O}_j^{(r)}$ observations, as Learner knows the total number of observations.

The idea behind this argument is that with high probability, a Level-I arm j will have already eliminated a Level-II arm $i \in \mathcal{A}_j$ by the end of epoch $r-1$ if its mean reward $\mu_{j,i}$ satisfies $\mu_{j,i} < \mu_{j,i^*} - 8 \cdot \frac{1}{2^r}$ (see Lemma 13 in Appendix 5.7.1).

By combining (5.4) and (5.5), we construct an asymmetric confidence interval for the empirical mean of Level-I arm $j \in \mathcal{A}^{(r)}$, i.e., with high probability, we have

$$\mu_{j,i^*} - 2 \cdot 8 \cdot \tfrac{1}{2^r} \leq \widehat{\theta}_j^{(r)} \leq \mu_{j,i^*} + 8 \cdot \tfrac{1}{2^r} \quad . \tag{5.6}$$

With the progression of epochs, the confidence interval shrinks, and after enough epochs, Learner will eliminate a bad Level-I arm with high probability. Lemma 16 in Appendix 5.7.1 proves this claim. Once a Level-I arm is eliminated, none of its Level-II arms can be pulled.

Algorithm 12 Two-Level Elimination Algorithm

1: **Input:** Level-I arm set $\mathcal{A}$, Level-II arm set $\mathcal{A}_j$ for all $j \in \mathcal{A}$, and T ;
2: **Initialization:** $r \leftarrow 1$, $\mathcal{A}^{(1)} \leftarrow \mathcal{A}$, and $\mathcal{A}_j^{(1)} \leftarrow \mathcal{A}_j$ for all $j \in \mathcal{A}$;
3: **while** T rounds have not been consumed yet **do**
4: Learner sets $L^{(r)} = 2\log(KT)2^{2r}$;
Learner allocates $L^{(r)} \left|\mathcal{A}_j^{(r)}\right|$ rounds for each Level-I arm $j \in \mathcal{A}^{(r)}$;
5: Each Level-I arm $j \in \mathcal{A}^{(r)}$:
(i) pulls each Level-II arm $i \in \mathcal{A}_j^{(r)}$ for $L^{(r)}$ times and puts all the obtained observations in $\mathcal{O}_j^{(r)}$ and $\mathcal{O}_{j,i}^{(r)}$, respectively ;
(ii) reports $\widehat{\theta}_j^{(r)}$;
(iii) eliminates a Level-I arm $i \in \mathcal{A}_j^{(r)}$ based on inequality (5.2) and reports $\left|\mathcal{A}_j^{(r+1)}\right|$;
6: Learner eliminates Level-I arm $j \in \mathcal{A}^{(r)}$ based on inequality (5.3) and gets $\mathcal{A}^{(r+1)}$;
7: $r \leftarrow r+1$.
8: **end while**

5.2.3 Regret Analysis

We now present a regret guarantee for Algorithm 12.

Theorem 20. *The regret of Algorithm 12 is at most*

$$\sum_{j\in\mathcal{A}} \sum_{i\in\mathcal{A}_j:\max\left\{\Delta_j,\Delta_i^{(j)}\right\}>0} O\left(\frac{\log(KT)}{\max\left\{\Delta_j,\Delta_i^{(j)}\right\}}\right) \quad .$$

Several remarks are in order for Theorem 20. Recall that the mean reward gap between Level-II arms (j^*,i^*) and (j,i) is $\Delta_j+\Delta_i^{(j)}$. Suppose we place all these K Level-II arms only in one level, and run the standard elimination algorithm [6] over these K arms. Then the regret is at most $\sum_{j\in\mathcal{A}} \sum_{i\in\mathcal{A}_j:\max\left\{\Delta_j,\Delta_i^{(j)}\right\}>0} O\left(\frac{\log(T)}{\Delta_j+\Delta_i^{(j)}}\right)$ instead of $O\left(\frac{\log(T)}{\max\left\{\Delta_j,\Delta_i^{(j)}\right\}}\right)$ for each sub-optimal (j,i). Since $\frac{1}{\max\left\{\Delta_j,\Delta_i^{(j)}\right\}} \leq \frac{2}{\Delta_j+\Delta_i^{(j)}}$, the multiplicative factor of performance loss is at most 2 when converting a one-level bandit problem to a bi-level bandit problem.

Proof Sketch of Theorem 20: For a sub-optimal Level-II arm i managed by Level-I arm j, we upper bound the expected number of pulls based on the quantities Δ_j and $\Delta_i^{(j)}$. If $\Delta_j \leq \Delta_i^{(j)}$, our technical Lemma 14 in Appendix 5.7.1 proves that each sub-optimal Level-II arm i can be in Level-I arm j's active arm set for at most $\left\lceil \log\left(64/\Delta_i^{(j)}\right)\right\rceil$ epochs with high probability. If $\Delta_j > \Delta_i^{(j)}$, Learner will have already eliminated Level-I arm j before the Level-I arm j eliminates Level-II arm i with high probability. Therefore, all the Level-II arms managed by the Level-I arm j (obviously, including Level-II arm i itself) will not be pulled after the round when Level-I arm j is eliminated. Our technical Lemma 16 in Appendix 5.7.1 formally proves that each sub-optimal Level-I arm j can be in Learner's active arm set for at most $\lceil \log\left(48/\Delta_j\right)\rceil$ epochs with high probability. □

5.3 Differentially Private Bi-Level Bandits

In this section, we present a differentially private learning algorithm for bi-level bandits along with both privacy and regret guarantees.

5.3.1 Differential Privacy

Let $X_{1:T}$ be the sequence of reward vectors from round 1 to round T that are fed into the learning algorithm. Before presenting the formal definition of differential privacy, we first define the notion of a neighbouring sequence. We say $X'_{1:T}$ is a neighbouring sequence of $X_{1:T}$ if $X_{1:T}$ and $X'_{1:T}$ differ in at most one reward vector.

Definition 5. *(Differential privacy). A learning algorithm Π is ϵ-differentially private if for any decision set $\mathcal{J} \subseteq Range\,(\Pi)$, we have*

$$\mathbb{P}\left\{\Pi\left(X_{1:T}\right) \in \mathcal{J}\right\} \leq \mathbb{P}\left\{\Pi\left(X'_{1:T}\right) \in \mathcal{J}\right\} \cdot e^{\epsilon} \quad .$$

The definition of differential privacy guarantees that from the output of the learning algorithm, i.e., $((J_1, I_1), (J_2, I_2) \ldots, (J_T, I_T))$, an outside observer is very unlikely to infer whether the learning algorithm takes the original reward sequence $X_{1:T}$ or a neighbouring reward sequence $X'_{1:T}$ as input. This property implies that the information in any single round t has almost no impact on the output of the learning algorithm.

5.3.2 Differentially Private Two-Level Elimination Algorithm

By modifying Algorithm 12, we now present a differentially private algorithm; this algorithm is shown in Algorithm 13. The high-level idea behind our private learning algorithm is to have each Level-I arm's algorithm be ϵ-differentially private. Then, from the property that differential privacy is immune to post-processing (Proposition 2.1 in [16], it is also shown in Proposition 1), the algorithm run by Learner is ϵ-differentially private.

Recall that in Algorithm 12, each active Level-I arm j reports both the empirical mean $\widehat{\theta}_j^{(r)}$ and the updated size of the active Level-II arm set $|\mathcal{A}_j^{(r+1)}|$. To have a private version of this step, we use the Laplace mechanism (Definition 3.3 in [16], it is also shown in Definition 3) to inject noise. The key challenge to devise a differentially private learning algorithm that has a good theoretical guarantee is the controlling of the noise variables. A naive way is to inject a noise variable drawn from Lap $\left(\frac{1}{0.5\epsilon}\right)$ to the observations obtained for each active Level-II arm. Then, each Level-I arm computes its differentially private empirical mean and conducts

the differentially private elimination to have an updated size of the active Level-II arm set that will be used for the future epoch. However, this naive way will result in a sub-optimal regret bound due to the fact the computed differentially private empirical mean may contain multiple noise variables.

To limit the amount of noise variables included in the differentially private empirical mean of a Level-I arm, we take the way of composing two differentially private algorithms at each Level-I arm's side with each 0.5ϵ-differentially private. The first private algorithm computes a differentially private version of $\widehat{\theta}_j^{(r)}$ and the second private algorithm conducts a private elimination to have a private version of $|\mathcal{A}_j^{(r+1)}|$. Then, from the basic composition theorem that ϵ's can be added-up (Theorem 3.16 in [16], it is also shown in Theorem 7), we claim that the composed algorithm run by each Level-I arm is ϵ-differentially private.

We still set $L^{(r)} = 2\log(KT) \cdot 2^{2r}$, the number of pulls of an active Level-II arm in epoch r. For the first private algorithm, as our goal is to preserve 0.5ϵ-differential privacy, we inject a noise $Z_j^{(r)} \sim \text{Lap}\left(\frac{1}{0.5\epsilon}\right)$ to all the obtained observations of Level-I arm $j \in \mathcal{A}^{(r)}$, i.e., $\mathcal{O}_j^{(r)}$. Let $\widetilde{\theta}_j^{(r)} := \widehat{\theta}_j^{(r)} + \frac{Z_j^{(r)}}{|\mathcal{A}_j^{(r)}| \cdot L^{(r)}}$ be the differentially private empirical mean of a Level-I arm $j \in \mathcal{A}_j^{(r)}$. For the second private algorithm, we inject a noise $Z_{j,i}^{(r)} \sim \text{Lap}\left(\frac{1}{0.5\epsilon}\right)$ to the obtained observations of each Level-II arm $i \in \mathcal{A}_j^{(r)}$, i.e., $\mathcal{O}_{j,i}^{(r)}$. Let $\widetilde{\mu}_{j,i}^{(r)} := \widehat{\mu}_{j,i}^{(r)} + \frac{Z_{j,i}^{(r)}}{L^{(r)}}$ be the differentially private empirical mean of a Level-II arm $i \in \mathcal{A}_j^{(r)}$. We now come back to the elimination rules set by each active Level-I arm j and Learner.

The differentially private elimination rule at Level-I arm j's side is now modified to

$$\widetilde{\mu}_{j,i}^{(r)} + \frac{1}{2^r} + \frac{3}{\epsilon \cdot 2^{2r}} < \max_{i' \in \mathcal{A}_j^{(r)}} \left(\widetilde{\mu}_{j,i'}^{(r)} - \frac{1}{2^r} - \frac{3}{\epsilon \cdot 2^{2r}} \right) \quad . \tag{5.7}$$

Compared with the non-private elimination rule (5.2), the above private elimination rule includes an extra term at each side. These extra terms use the fact that with high probability, the amount of noise injected per observation is not too much (shown in (5.9) in the appendix of this chapter).

The differentially private elimination rule at Learner's side is now modified to

$$\widetilde{\theta}_j^{(r)} + 2 \cdot 8 \cdot \frac{1}{2^r} + \frac{48}{\epsilon \cdot 2^{2r}} + \frac{3}{\epsilon \cdot |\mathcal{A}_j^{(r)}| \cdot 2^{2r}} < \max_{j' \in \mathcal{A}^{(r)}} \left(\widetilde{\theta}_{j'}^{(r)} - 8 \cdot \frac{1}{2^r} - \frac{3}{\epsilon \cdot |\mathcal{A}_{j'}^{(r)}| \cdot 2^{2r}} \right) \quad . \tag{5.8}$$

Besides the argument shown in (5.4), several other arguments support this private elimination rule. The first argument is that the amount of noise injected is not too much (shown in (5.9) in the appendix of this chapter). The second argument is that with high probability, the expected reward $\theta_j^{(r)} = \mathbb{E}\left[\widehat{\theta}_j^{(r)}\right]$ of an active Level-I arm j is not too far from its best achievable mean reward μ_{j,i^*} (shown in claim (*iii*) in Lemma 20 in Appendix 5.7.3). By combining (5.4) and claim (*iii*) in Lemma 20, we construct an asymmetric confidence interval round the empirical mean $\widehat{\theta}_j^{(r)}$, which is quite similar to the one shown in (5.6). Note that when $\epsilon \to \infty$, (5.7) will be same as (5.2) and (5.8) will be the same as (5.3).

Algorithm 13 Differentially Private Two-Level Elimination Algorithm

1: **Input:** Level-I arm set $\mathcal{A}$, Level-II arm set $\mathcal{A}_j$ for all $j \in \mathcal{A}$, T, and privacy parameter ϵ ;
2: **Initialization:** $r \leftarrow 1$, $\mathcal{A}^{(1)} \leftarrow \mathcal{A}$, $\mathcal{A}_j^{(1)} \leftarrow \mathcal{A}_j$ for all $j \in \mathcal{A}$;
3: **while** T rounds have not been consumed yet **do**
4: Learner sets $L^{(r)} = 2\log(KT)2^{2r}$;
Learner allocates $L^{(r)}\left|\mathcal{A}_j^{(r)}\right|$ rounds for each Level-I arm $j \in \mathcal{A}^{(r)}$;
5: Each Level-I arm $j \in \mathcal{A}^{(r)}$:
(i) pulls each active Level-II arm $i \in \mathcal{A}_j^{(r)}$ for $L^{(r)}$ times to have $\mathcal{O}_j^{(r)}$ and $\mathcal{O}_{j,i}^{(r)}$, respectively ;
(ii) injects $Z_j^{(r)} \sim \text{Lap}\left(\frac{1}{0.5\epsilon}\right)$ to $\mathcal{O}_j^{(r)}$ and reports $\widetilde{\theta}_j^{(r)}$;
(iii) injects $Z_{j,i}^{(r)} \sim \text{Lap}\left(\frac{1}{0.5\epsilon}\right)$ to $\mathcal{O}_{j,i}^{(r)}$ for each $i \in \mathcal{A}_j^{(r)}$, eliminates a Level-I arm $i \in \mathcal{A}_j^{(r)}$ based on inequality (5.7), and reports $\left|\mathcal{A}_j^{(r+1)}\right|$;
6: Learner eliminates Level-I arm $j \in \mathcal{A}^{(r)}$ based on inequality (5.8) and gets $\mathcal{A}^{(r+1)}$;
7: $r \leftarrow r + 1$.
8: **end while**

5.3.3 Privacy and Regret Analysis

In this subsection, we provide privacy and regret guarantees for Algorithm 13.

Theorem 21. *Algorithm 13 is* 1.5ϵ*-differentially private.*

Proof sketch of Theorem 21: The high-level idea behind the proof is to show that the algorithm run by Learner is ϵ-differentially private and the algorithm run by Level-

I arms is 0.5ϵ-differentially private. By composing these two private learning algorithms together, we can claim that Algorithm 13 is 1.5ϵ-differentially private.

The challenging part for the proof is to show that the algorithm run by Learner is ϵ-differentially private. The idea behind the proof is to use the following two arguments repeatedly. One argument is that differentially private algorithms can be composed, i.e., a composition theorem saying that ϵ's can be added-up (Theorem 3.16 in [16], it is also shown in Theorem 7). The other argument is that differential privacy is immune to post-processing (Proposition 2.1 in [16], it is also shown in Proposition 1). Recall that each active Level-I arm reports both the differentially private empirical mean and the updated size of the active Level-II arm set, and Learner relies on these two parameters from all active Level-I arms to do the elimination. Since the algorithm run by each active Level-I arm to compute the differentially private empirical mean is 0.5ϵ-differentially private and the algorithm to compute the updated size of the active Level-II arm set is also 0.5ϵ-differentially private, from the composition theorem, we claim that the composed algorithm run by each Level-I arm is ϵ-differentially private. Since Learner relies on the differentially private empirical means and the sizes of the active Level-II arm set to do the elimination, from the post-processing proposition, we can conclude that the algorithm run by Learner is ϵ-differentially private. □

Theorem 22. *The regret of Algorithm 13 is at most*

$$\sum_{j\in\mathcal{A}} \sum_{i\in\mathcal{A}_j:\max\left\{\Delta_j,\Delta_i^{(j)}\right\}>0} O\left(\max\left\{\frac{\log(KT)}{\max\left\{\Delta_j,\Delta_i^{(j)}\right\}},\frac{\log(KT)}{\epsilon}\right\}\right) .$$

Several remarks are in order for Algorithm 13 and Theorem 22. First, when setting $\epsilon' = \frac{2}{3}\epsilon$, Algorithm 13 is ϵ'-differentially privacy; when $\epsilon \to \infty$, the term involving privacy parameter ϵ will vanish, i.e., the regret bound shown in Theorem 22 will be the same as the one shown in Theorem 20. Therefore, ∞-differentially private bi-level bandits can be viewed as the basic bi-level bandits shown in Section 5.2. Second, if we only have one Level-I arm and it has K Level-II arms, the setting of differentially private bi-level bandits boils down to the setting of differentially private stochastic bandits and Algorithm 13 achieves the optimal $\sum_{i\in\mathcal{A}_1:\Delta_i^{(1)}>0} O\left(\frac{\log(T)}{\Delta_i^{(1)}}+\frac{\log(T)}{\epsilon}\right)$ regret bound. Note that our Algorithm 13 is a sim-

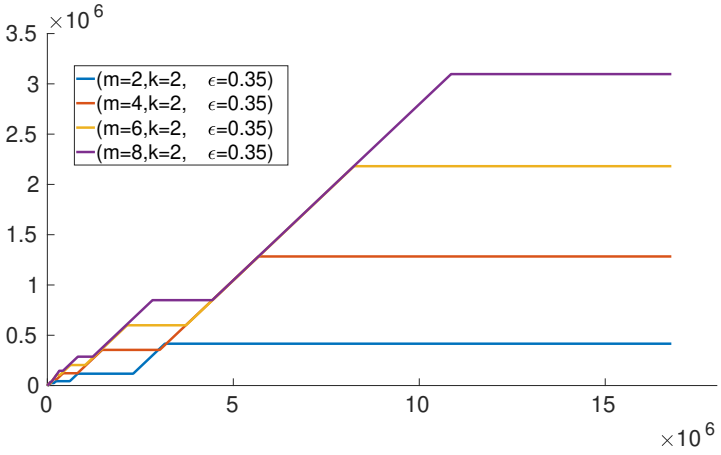

Figure 5.2: The impact of m: $m = 2, 4, 6, 8$

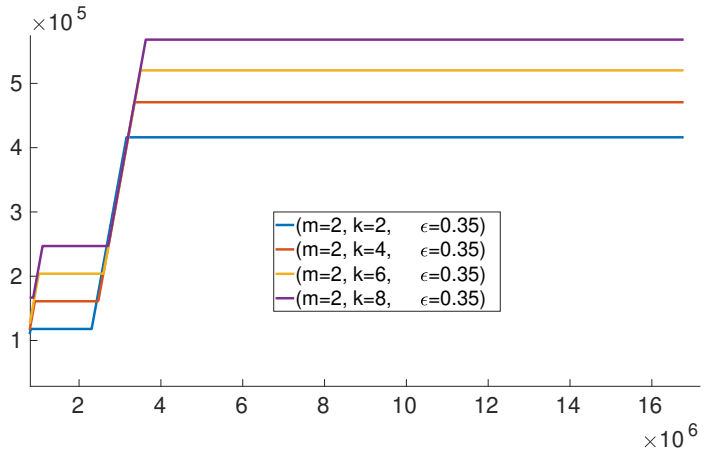

Figure 5.3: The impact of k: $k = 2, 4, 6, 8$

pler version of the differentially private elimination-style algorithm than the one that has been presented in [33]. The simplicity comes from the fact in our differentially private elimination algorithm, we do not need to compute the number of pulls of (Level-II) arms in a private way. Instead, we simply quadruple the number of pulls in each epoch.

5.4 Experimental Results

In this section, we present experimental results to show the practical performance of our learning algorithms and to verify our theoretical results. Our results are the average of 10 independent runs. Recall that m is the number of Level-I arms and

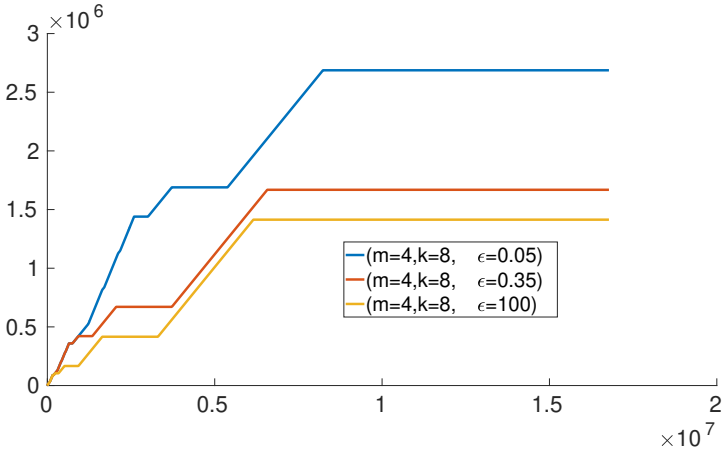

Figure 5.4: The impact of ϵ: $\epsilon = 0.05, 0.35, 100$

k_j is the number of Level-II arms managed by a Level-I arm j.

The first experiment is to see the linearity of regret in the number of Level-I arms. To remove the impact of ϵ and mean reward gaps, i.e., $\max\left\{\Delta_j, \Delta_i^{(j)}\right\}$, we set $\Delta_j = \Delta_i^{(j)} = \epsilon = 0.35$. Then, we fix the number of Level-II arms $k_j = 2$ for all $j \leq m$. Let us say $k := k_j$. Now, only parameter m can impact the regret. We increase the number of Level-I arms linearly, e.g., setting $m = 2, 4, 6, 8$. Figure 5.2 shows the performance comparison in this set of settings. Just as expected, the regret grows linearly with the growth of m.

The second experiment is to see the linearity of regret in the number of Level-II arms. We still fix $\Delta_j = \Delta_i^{(j)} = \epsilon = 0.35$. Then, we fix the number of Level-I arms $m = 2$. We now increase the number of Level-II arms linearly to see the practical performance. We set the number of Level-II arms $k_j = 2, 4, 6, 8$ for all $j \leq m$. From Figure 5.3, we can see that the regret grows linearly with the growth of k_j. The third experiment is to see the impact of privacy parameter ϵ. We set $m = 4$, $k_j = 8$ for all $j \leq m$, and $\Delta_j = \Delta_i^{(j)} = 0.35$. Then, we set $\epsilon = 0.05, 0.35, 100$ to see the impact of privacy parameter ϵ on the regret. Figure 5.4 shows the regret in this set of settings. When setting a very small ϵ, e.g., $\epsilon = 0.05$, the learning algorithm suffers more regret as the term $\frac{1}{\epsilon}$ plays a dominating role. When setting a very large ϵ, e.g., $\epsilon = 100$, the learning algorithm basically boils down to a non-private setting and the term $1/\max\left\{\Delta_j, \Delta_i^{(j)}\right\}$ dominates the $\frac{1}{\epsilon}$ term. More experimental results can be found in Appendix 5.7.4.

5.5 Literature

Differentially private multi-armed bandit problems have been well studied in [20, 32, 34, 33, 37, 7, 23, 1, 21], and our work is most related to the variants of differentially private stochastic bandits. Regarding problem-dependent regret bounds for differentially private stochastic bandits, the authors of [32] devised the first differentially private UCB and Thompson Sampling (TS) algorithms but with sub-optimal regret bounds. Recently, the authors of [33] and [21] devised the differentially private elimination algorithm and UCB-based algorithm, respectively, with the optimal $O\left(K\log(T)/\Delta\right)+O\left(K\log(T)/\epsilon\right)$ regret bound.

Regarding the setting of (differentially private) bi-level bandits, only very recently, we discovered a very interesting setting in [27]: federated private bandits with multiple agents. At first glance, our learning setting seems to be very related to theirs in that our Level-I arms can be viewed as their agents and our Level-II arms can be viewed as their arms. However, when taking a detailed look, these two learning settings are quite different. One of the fundamental differences is that in bi-level bandits, all Level-II arms are unknown to Learner while in federated private bandits, Learner necessarily knows of all the arms. Besides, for the learning setting in [27], in each round t, *each* agent pulls an arm while in our setting, only one of the Level-I arms is allowed to pull a Level-II arm. The authors of [35] proposed a decentralized online learning setting where there are multiple cooperative learners. When comparing their setting to ours, our Level-I arms can also be viewed as their learners and our Level-II arms can be viewed as their arms. However, their learners work in a cooperative way while our Level-I arms work independently.

5.6 Discussion

In this work, we have presented a new bandit setting: bi-level bandits, and novel two-level elimination algorithms to solve this problem. Actually, instead of running the elimination style algorithm, Learner actually can run a UCB-based algorithm by computing the upper confidence bound based on our developed asymmetric confidence intervals to decide which Level-I arm to pull. One may ask what is the advantage to running the elimination style algorithm at a Level-I arm's side. Recall that both Learner and each Level-I arm are at the same side, i.e., they both

want to accumulate as much reward as possible. The advantage of running the elimination style algorithm at each Level-I arm j's side is that the number of pulls of the best Level-II arm (j, i^*) is always guaranteed to be no smaller than the pulls of any other sub-optimal Level-II arm. Along with the fact that the bad Level-II arms will be eliminated gradually, the empirical mean of a Level-I arm j becomes closer and closer to the best achievable mean reward μ_{j,i^*}. If a Level-I arm runs an algorithm that does not guarantee the number of pulls of the best Level-II arm (j, i^*) (e.g., running the round-based UCB), the empirical mean of a specific Level-I arm might be quite under-estimated, which will make it eliminated at an early stage by Learner. Therefore, it is not wise for a smart Level-I arm to choose to run the round-based UCB algorithm, as it may not have any benefit.

From the perspective of an intermediate level entity, one of the advantages of the bi-level bandit learning model is that the learning algorithm has a biased positive opinion of such an entity (such as drug company) that has good average performance but, meanwhile, has some poorly performing drugs (an outcome which is only known to the company after it has begun clinical trials). Bi-level bandits enables drug developers to be able to focus on high-risk drugs that might also have high reward for society, without the potential downside of a funding organization reducing funding after discovering that a drug performs very poorly. On the other side, due to the feature that bi-level bandit model enables more risk taking, a company's ability to better mask its failures from society can naturally also pose a risk to society, and this is a concern that should be considered when using such a framework.

5.7 Appendix of this Chapter

The organization of the appendix is as follows:

For both the proofs of Theorem 20 and Theorem 22, we will Fact 1 below and Lemma 12.

Fact 1. If $Y \sim \text{Lap}(b)$, for any $0 < \delta < 1$, we have

$$\mathbb{P}\left\{|Y| > \ln\left(\frac{1}{\delta}\right) \cdot b\right\} = \delta \quad . \tag{5.9}$$

For a Level-I arm j, let $\mathcal{E}_j^{(r)} := \left\{\left|\widehat{\mu}_{j,i}^{(l)} - \mu_{j,i}\right| < \frac{1}{2^l}, \forall l \in \{1,2,\ldots,r\}, \forall i \in \mathcal{A}_j^{(l)}\right\}$ be the event that in all epochs $l \in \{1,2,\ldots,r\}$, for all Level-II arm $i \in \mathcal{A}_j^{(l)}$, all the confidence intervals hold simultaneously. Let $\overline{\mathcal{E}_j^{(r)}}$ be the complementary event of $\mathcal{E}_j^{(r)}$. Note that we set $\mathbf{1}\left\{\mathcal{E}_j^{(0)}\right\} = 1$ for all $j \in \mathcal{A}$. As we will show in Lemma 12 below, event $\overline{\mathcal{E}_j^{(r)}}$ is a low probability one.

Lemma 12. *In any epoch $r \geq 1$, for a Level-I arm $j \in \mathcal{A}^{(r)}$, we have $\mathbb{P}\left\{\overline{\mathcal{E}_j^{(r)}}\right\} = O\left(\frac{1}{K^3T^3}\right)$.*

Proof of Lemma 12: The proof first uses a union bound and then uses the Hoeffding's inequality. We have

$$\begin{aligned}
\mathbb{P}\left\{\overline{\mathcal{E}_j^{(r)}}\right\} &= \mathbb{P}\left\{\exists l \in \{1,2,\ldots,r\}, \exists i \in \mathcal{A}_j^{(l)} \text{ s.t. } \left|\widehat{\mu}_{j,i}^{(l)} - \mu_{j,i}\right| \geq \tfrac{1}{2^l}\right\} \\
&\leq \sum_{l\in\{1,2,\ldots,r\}} \sum_{i\in\mathcal{A}_j} \underbrace{\mathbb{P}\left\{\left|\widehat{\mu}_{j,i}^{(l)} - \mu_{j,i}\right| \geq \frac{1}{2^l}\right\}}_{\text{Hoeffding's inequality}} \\
&\leq \sum_{l=1}^{T} \sum_{i\in\mathcal{A}_j} 2e^{-2\cdot 2\log(KT)\cdot 2^{2l}\cdot\frac{1}{2^{2l}}} \\
&\leq O\left(\frac{1}{K^3T^3}\right) \quad ,
\end{aligned} \tag{5.10}$$

which concludes the proof.

Note that $\widehat{\mu}_{j,i}^{(l)}$ is the empirical mean of $2\log(KT)\cdot 2^{2l}$ observations that are i.i.d. according to a fixed distribution with mean $\mu_{j,i}$. □

5.7.1 Proofs of Theorem 20

We first present some lemmas and then give the proof of Theorem 20. Recall that $\theta_j^{(r)} = \mathbb{E}\left[\widehat{\theta}_j^{(r)}\right]$ is the expected mean reward of an active Level-I arm $j \in \mathcal{A}^{(r)}$ after

collecting all the observations within epoch r. As will be shown in the last claim of Lemma 13 below, $\theta_j^{(r)}$ will be very close to its best achievable mean reward μ_{j,i^*} with high probability.

Lemma 13. *In any epoch $r \geq 1$, for a Level-I arm $j \in \mathcal{A}^{(r)}$, if event $\mathcal{E}_j^{(r)}$ is true, we have the following claims:*

(i): The best Level-II arm (j, i^*) will not eliminated by the end of epoch r, i.e., $(j, i^*) \in \mathcal{A}_j^{(r+1)}$;

(ii): A sub-optimal Level-II arm (j, i) with mean reward $\mu_{j,i}$ will not be in $\mathcal{A}_j^{(r+1)}$ if $\mu_{j,i} \leq \mu_{j,i^*} - 4 \cdot \frac{1}{2^r}$;

(iii): $\mu_{j,i^*} - 8 \cdot \frac{1}{2^r} < \theta_j^{(r)} \leq \mu_{j,i^*}$.

From Lemma 13, we have the following lemma, which states that once the number of completed epochs is enough, a sub-optimal Level-II arm (j, i) with a mean reward gap $\Delta_i^{(j)}$ will be eliminated with high probability.

Lemma 14. *For a sub-optimal Level-II arm (j, i) with a mean reward gap $\Delta_i^{(j)}$, in any epoch $r \geq \left\lceil \log\left(\frac{64}{\Delta_i^{(j)}}\right)\right\rceil$, if event $\mathcal{E}_j^{(r)}$ is true, this sub-optimal Level-II arm (j, i) will not be in $\mathcal{A}_j^{(r+1)}$.*

Let $\mathcal{V}^{(r)} := \left\{\mu_{j,i^*} - 2 \cdot 8 \cdot \frac{1}{2^l} < \widehat{\theta}_j^{(l)} < \mu_{j,i^*} + 8 \cdot \frac{1}{2^l}, \forall l \in \{1, 2, \ldots, r\}, \forall j \in \mathcal{A}^{(l)}\right\}$ be the event that in all epochs $l \in \{1, 2, \ldots, r\}$, the empirical mean of a Level-I arm $j \in \mathcal{A}^{(l)}$, i.e., $\widehat{\theta}_j^{(l)}$, is not far from its best achievable true mean μ_{j,i^*}. It is important to note that the confidence interval of $\widehat{\theta}_j^{(l)}$ is asymmetric. Let $\overline{\mathcal{V}^{(r)}}$ be the complementary event of $\mathcal{V}^{(r)}$. Note that we set $\mathbf{1}\left\{\mathcal{V}^{(0)}\right\} = 1$. As we will show in Lemma 15 below, event $\overline{\mathcal{V}^{(r)}}$ is a low probability one.

Lemma 15. *In any epoch $r \geq 1$, we have $\mathbb{P}\left\{\overline{\mathcal{V}^{(r)}}\right\} = O\left(\frac{1}{K^2T^2}\right)$.*

For each sub-optimal Level-I arm $j \in \mathcal{A}\backslash\{j^*\}$, define $\lambda_j := \log\left(\frac{48}{\Delta_j}\right)$. The intuitive understanding of λ_j is that if the number of epochs has progressed enough, i.e., more than λ_j epochs have transpired, Learner can safely eliminate a sub-optimal Level-I arm j with a mean reward gap Δ_j with high probability. We formalize this intuition in the next lemma.

Lemma 16. *When $r \geq \lambda_j$ and event $\mathcal{V}^{(r)}$ is true, we have that a sub-optimal Level-I arm j with a mean reward gap Δ_j will not be in $\mathcal{A}^{(r+1)}$.*

With these preparations in place, we now prove Theorem 20.

Proof of Theorem 20: The regret shown in (5.1) can be further decomposed as

$$\begin{aligned}
&\mathcal{R}(T) \\
=& \sum_{j\in\mathcal{A}\backslash\{j^*\}} \underbrace{\sum_{t=1}^{T} \mathbb{E}\left[\mathbf{1}\{(J_t,I_t)=(j,i^*)\}\right]\cdot\Delta_j}_{\text{Lemma 17}} \\
+& \sum_{j\in\mathcal{A}} \sum_{i\in\mathcal{A}_j\backslash\{(j,i^*)\}} \underbrace{\sum_{t=1}^{T} \mathbb{E}\left[\mathbf{1}\{(J_t,I_t)=(j,i)\}\right]\cdot\left(\Delta_j+\Delta_i^{(j)}\right)}_{\text{Lemma 18}} \quad .
\end{aligned} \tag{5.11}$$

For the first term in (5.11), we prepare Lemma 17 to upper bound it, and for the second term in (5.11), we prepare Lemma 18 to upper bound it.

Lemma 17. *For a sub-optimal Level-I arm $j \in \mathcal{A}\backslash\{j^*\}$, we have*

$$\sum_{t=1}^{T} \mathbb{E}\left[\mathbf{1}\{(J_t,I_t)=(j,i^*)\}\right]\cdot\Delta_j = O\left(\frac{\log(KT)}{\Delta_j}\right) + O\left(\frac{1}{K}\right). \tag{5.12}$$

Lemma 18. *For a sub-optimal Level-II arm $(j,i) \in \mathcal{A}_j\backslash\{(j,i^*)\}$, we have*

$$\sum_{t=1}^{T} \mathbb{E}\left[\mathbf{1}\{(J_t,I_t)=(j,i)\}\right]\cdot\left(\Delta_i^{(j)}+\Delta_j\right) = O\left(\frac{\log(KT)}{\max\left\{\Delta_j,\Delta_i^{(j)}\right\}}\right) + O\left(\frac{1}{K}\right). \tag{5.13}$$

By plugging Lemma 17 and Lemma 18 into (5.11), we have

$$\begin{aligned}
\mathcal{R}(T) &\leq \sum_{j\in\mathcal{A}\backslash\{j^*\}} O\left(\frac{\log(KT)}{\Delta_j}\right) + \sum_{j\in\mathcal{A}}\sum_{i\in\mathcal{A}_j\backslash\{(j,i^*)\}} O\left(\frac{\log(KT)}{\max\left\{\Delta_j,\Delta_i^{(j)}\right\}}\right) \\
&= \sum_{j\in\mathcal{A}\backslash\{j^*\}}\sum_{i=(j,i^*)} O\left(\frac{\log(KT)}{\max\left\{\Delta_j,\Delta_i^{(j)}\right\}}\right) + \sum_{j\in\mathcal{A}}\sum_{i\in\mathcal{A}_j\backslash\{(j,i^*)\}} O\left(\frac{\log(KT)}{\max\left\{\Delta_j,\Delta_i^{(j)}\right\}}\right) \\
&= \sum_{j\in\mathcal{A}}\sum_{i\in\mathcal{A}_j:\max\left\{\Delta_j,\Delta_i^{(j)}\right\}>0} O\left(\frac{\log(KT)}{\max\left\{\Delta_j,\Delta_i^{(j)}\right\}}\right) \quad ,
\end{aligned} \tag{5.14}$$

which concludes the proof of Theorem 20. □

We now present the proofs of Lemmas 13 through 18.

Proof of Lemma 13:

Claim (i): We use contradiction to prove this argument. Suppose that a Level-I arm j eliminates the best Level-II arm (j, i^*) in an epoch $s \leq r$ and event $\mathcal{E}_j^{(r)}$ is true. Recall that the elimination rule of each active Level-I arm j, which is shown in (5.2), is:

$$\widehat{\mu}_{j,i}^{(s)} + \frac{1}{2^s} < \max_{i' \in \mathcal{A}_j^{(s)}} \left(\widehat{\mu}_{j,i'}^{(s)} - \frac{1}{2^s} \right) \quad .$$

If $\mathcal{E}_j^{(r)}$ is true, for the best Level-II arm (j, i^*), we have

$$\widehat{\mu}_{j,i^*}^{(s)} + \frac{1}{2^s} > \left(\mu_{j,i^*} - \frac{1}{2^s} \right) + \frac{1}{2^s} = \mu_{j,i^*} \quad . \tag{5.15}$$

Let $(j, i_*^{(s)}) \leftarrow \arg\max_{i' \in \mathcal{A}_j^{(s)}} \left(\widehat{\mu}_{j,i'}^{(s)} - \frac{1}{2^s} \right)$. If $\mathcal{E}_j^{(r)}$ is true, simultaneously, we also have

$$\widehat{\mu}_{j,i_*^{(s)}}^{(s)} - \frac{1}{2^s} < \left(\mu_{j,i_*^{(s)}} + \frac{1}{2^s} \right) - \frac{1}{2^s} = \mu_{j,i_*^{(s)}} \quad . \tag{5.16}$$

As $\mu_{j,i^*} \geq \mu_{j,i_*^{(s)}}$ always holds, it means that the best Level-II arm (j, i^*) will not be eliminated in epoch s, which yields a contradiction. Both the first inequalities in (5.15) and (5.16) use the fact that if event $\mathcal{E}_j^{(r)}$ is true, it means $\widehat{\mu}_{j,i^*}^{(s)} > \mu_{j,i^*} - \frac{1}{2^s}$ and $\widehat{\mu}_{j,i_*^{(s)}}^{(s)} < \mu_{j,i_*^{(s)}} + \frac{1}{2^s}$ are true.

Claim (ii): If $\mathcal{E}_j^{(r)}$ is true, we show that, for an epoch $s \leq r$, a sub-optimal Level-II arm $(j, i) \in \mathcal{A}_j^{(s)}$ with mean reward $\mu_{j,i} \leq \mu_{j,i^*} - 4 \cdot \frac{1}{2^s}$ will not be in $\mathcal{A}_j^{(s+1)}$.

If $\mathcal{E}_j^{(r)}$ is true, for a sub-optimal Level-II arm $(j, i) \in \mathcal{A}_j^{(s)}$ with mean reward $\mu_{j,i} \leq \mu_{j,i^*} - 4 \cdot \frac{1}{2^s}$, we have

$$\widehat{\mu}_{j,i}^{(s)} + \frac{1}{2^s} < \left(\mu_{j,i} + \frac{1}{2^s} \right) + \frac{1}{2^s} = \mu_{j,i} + \frac{2}{2^s} \leq \mu_{j,i^*} - \frac{2}{2^s} \quad . \tag{5.17}$$

From claim (i), we know that the best Level-II arm (j, i^*) is always in the active Level-II arm set of Level-I arm j.

Therefore, if $\mathcal{E}_j^{(r)}$ is true, simultaneously, we also have

$$\widehat{\mu}_{j,i^*}^{(s)} - \frac{1}{2^s} > \left(\mu_{j,i^*} - \frac{1}{2^s} \right) - \frac{1}{2^s} = \mu_{j,i^*} - \frac{2}{2^s} \quad , \tag{5.18}$$

which implies a sub-optimal Level-II arm (j,i) with mean reward $\mu_{j,i} \leq \mu_{j,i^*} - 4 \cdot \frac{1}{2^s}$ will not be in $\mathcal{A}_j^{(s+1)}$.

Claim (iii): Let $\mathcal{F}_j^{(r-1)}$ collect all the history information by the end of epoch $r-1$, i.e., collecting the pulled Level-II arms by Level-I arm j and their rewards. Then we rewrite $\theta_j^{(r)}$ as

$$\begin{aligned}
\theta_j^{(r)} &= \mathbb{E}\left[\widehat{\theta}_j^{(r)}\right] \\
&= \mathbb{E}\left[\mathbb{E}\left[\widehat{\theta}_j^{(r)} \mid \mathcal{F}_j^{(r-1)}\right]\right] \\
&= \mathbb{E}\left[\mathbb{E}\left[\frac{1}{\left|\mathcal{A}_j^{(r)}\right|} \cdot \sum_{i \in \mathcal{A}_j^{(r)}} \widehat{\mu}_{j,i}^{(r)} \mid \mathcal{F}_j^{(r-1)}\right]\right] \\
&= \mathbb{E}\left[\frac{1}{\left|\mathcal{A}_j^{(r)}\right|} \sum_{i \in \mathcal{A}_j^{(r)}} \mathbb{E}\left[\widehat{\mu}_{j,i}^{(r)} \mid \mathcal{F}_j^{(r-1)}\right]\right] \\
&= \mathbb{E}\left[\frac{1}{\left|\mathcal{A}_j^{(r)}\right|} \sum_{i \in \mathcal{A}_j^{(r)}} \mu_{j,i}\right] \quad .
\end{aligned} \tag{5.19}$$

The upper bound of $\theta_j^{(r)}$ is trivial to prove as we have

$$\theta_j^{(r)} = \mathbb{E}\left[\frac{1}{\left|\mathcal{A}_j^{(r)}\right|} \sum_{i \in \mathcal{A}_j^{(r)}} \mu_{j,i}\right] \leq \mathbb{E}\left[\frac{1}{\left|\mathcal{A}_j^{(r)}\right|} \sum_{i \in \mathcal{A}_j^{(r)}} \mu_{j,i^*}\right] = \mu_{j,i^*} \quad .$$

The proof of the lower bound of $\theta_j^{(r)}$ uses claim (ii). We have

$$\theta_j^{(r)} = \mathbb{E}\left[\frac{1}{\left|\mathcal{A}_j^{(r)}\right|} \sum_{i \in \mathcal{A}_j^{(r)}} \mu_{j,i}\right] > \mathbb{E}\left[\frac{1}{\left|\mathcal{A}_j^{(r)}\right|} \sum_{i \in \mathcal{A}_j^{(r)}} \left(\mu_{j,i^*} - 4 \cdot \frac{1}{2^{r-1}}\right)\right] = \mu_{j,i^*} - \frac{8}{2^r}. \qquad \square$$

Proof of Lemma 14: When $r \geq \left\lceil \log\left(\frac{64}{\Delta_i^{(j)}}\right)\right\rceil$, we have $\mu_{j,i} = \mu_{j,i^*} - \Delta_i^{(j)} < \mu_{j,i^*} - 4 \cdot \frac{1}{2^r}$. From claim (ii) in Lemma 13, we know that this sub-optimal Level-II arm (j,i) will not be in $\mathcal{A}_j^{(r+1)}$. $\square$

Proof of Lemma 15: In certain steps during the proof, we use Lemma 12 and claim (iii) in Lemma 13. We have

$$\begin{aligned}
\mathbb{P}\left\{\overline{\mathcal{V}^{(r)}}\right\} &\leq \sum_{l=1}^{r}\sum_{j\in\mathcal{A}} \mathbb{P}\left\{\widehat{\theta}_j^{(l)} \leq \mu_{j,i^*} - 2\cdot 8\cdot\tfrac{1}{2^l}, \mathcal{E}_j^{(l)}\right\} + \mathbb{P}\left\{\widehat{\theta}_j^{(l)} \geq \mu_{j,i^*} + 8\cdot\tfrac{1}{2^l}, \mathcal{E}_j^{(l)}\right\} \\
&+ \sum_{l=1}^{r}\sum_{j\in\mathcal{A}} \mathbb{P}\left\{\overline{\mathcal{E}_j^{(l)}}\right\} \\
&\leq \sum_{l=1}^{r}\sum_{j\in\mathcal{A}} \mathbb{P}\left\{\widehat{\theta}_j^{(l)} \leq \mu_{j,i^*} - 2\cdot 8\cdot\tfrac{1}{2^l}, \quad \mu_{j,i^*} - 8\cdot\tfrac{1}{2^l} < \theta_j^{(l)} \leq \mu_{j,i^*}\right\} \\
&+ \sum_{l=1}^{r}\sum_{j\in\mathcal{A}} \mathbb{P}\left\{\widehat{\theta}_j^{(l)} \geq \mu_{j,i^*} + 8\cdot\tfrac{1}{2^l}, \quad \mu_{j,i^*} - 8\cdot\tfrac{1}{2^l} < \theta_j^{(l)} \leq \mu_{j,i^*}\right\} \\
&+ \sum_{l=1}^{r}\sum_{j\in\mathcal{A}} \underbrace{\mathbb{P}\left\{\overline{\mathcal{E}_j^{(l)}}\right\}}_{=O\left(\frac{1}{K^3T^3}\right), \text{ Lemma 12}} \\
&\leq \sum_{l=1}^{r}\sum_{j\in\mathcal{A}} \underbrace{\mathbb{P}\left\{\widehat{\theta}_j^{(l)} \leq \theta_j^{(l)} - 8\cdot\frac{1}{2^l}\right\} + \mathbb{P}\left\{\widehat{\theta}_j^{(l)} \geq \theta_j^{(l)} + 8\cdot\frac{1}{2^l}\right\}}_{\text{Hoeffding's inequality}} + O\left(\tfrac{1}{K^2T^2}\right) \\
&\leq O\left(\tfrac{1}{K^2T^2}\right) \quad .
\end{aligned} \tag{5.20}$$

For the second to last inequality, recall that $\theta_j^{(l)} = \mathbb{E}\left[\widehat{\theta}_j^{(l)}\right]$ and $\widehat{\theta}_j^{(l)}$ is the empirical mean of $\left|\mathcal{A}_j^{(l)}\right| \cdot 2\log(KT)2^{2l}$ independent observations with each in $[0,1]$. From Hoeffding's inequality, we have

$$\mathbb{P}\left\{\left|\widehat{\theta}_j^{(l)} - \mathbb{E}\left[\widehat{\theta}_j^{(l)}\right]\right| \geq 8\cdot\frac{1}{2^l}\right\} \leq 2e^{-2\left|\mathcal{A}_j^{(l)}\right|\cdot 2\log(KT)\cdot 2^{2l}\cdot 64\cdot\frac{1}{2^{2l}}} \leq O\left(\frac{1}{K^3T^3}\right).$$

□

Proof of Lemma 16: There are two steps needed to complete the proof. The first step is to prove that the best Level-I arm j^* is in $\mathcal{A}^{(r+1)}$ if event $\mathcal{V}^{(r)}$ is true. We prove this argument by using contradiction. The second step is to prove that a sub-optimal Level-I arm j with a mean reward gap Δ_j will not be in $\mathcal{A}^{(r+1)}$ when $r \geq \lambda_j$ and event $\mathcal{V}^{(r)}$ is true.

Step 1: Recall that $j^* = \arg\max_{j\in\mathcal{A}} \mu_{j,i^*}$. Suppose that Learner eliminates j^* in an epoch $s \leq r$ and event $\mathcal{V}^{(r)}$ is true. Also, recall that, as already shown in (5.3),

Learner will only eliminate the best Level-I arm j^* in epoch s if

$$\widehat{\theta}_{j^*}^{(s)} + 2 \cdot 8 \cdot \tfrac{1}{2^s} \quad < \quad \max_{j' \in \mathcal{A}^{(s)}} \left(\widehat{\theta}_{j'}^{(s)} - 8 \cdot \tfrac{1}{2^s} \right) \quad . \tag{5.21}$$

If event $\mathcal{V}^{(r)}$ is true, for the best Level-I arm j^*, we have

$$\widehat{\theta}_{j^*}^{(s)} + 2 \cdot 8 \cdot \tfrac{1}{2^s} \quad > \quad \left(\mu_{j^*,i^*} - 2 \cdot 8 \cdot \tfrac{1}{2^s} \right) + 2 \cdot 8 \cdot \tfrac{1}{2^s} = \mu_{j^*,i^*} \quad . \tag{5.22}$$

The first inequality in (5.22) uses the fact that if event $\mathcal{V}^{(r)}$ is true, we have $\widehat{\theta}_{j^*}^{(s)} > \mu_{j^*,i^*} - 2 \cdot 8 \cdot \frac{1}{2^s}$.

Let $j^{(s)*} \in \arg\max_{j' \in \mathcal{A}^{(s)}} \left(\widehat{\theta}_{j'}^{(s)} - 8 \cdot \frac{1}{2^s} \right)$. If event $\mathcal{V}^{(r)}$ is true, simultaneously, we also have

$$\widehat{\theta}_{j^{(s)*}}^{(s)} - 8 \cdot \tfrac{1}{2^s} \quad < \quad \left(\mu_{j^{(s)*},i^*} + 8 \cdot \tfrac{1}{2^s} \right) - 8 \cdot \tfrac{1}{2^s} < \mu_{j^{(s)*},i^*} \quad . \tag{5.23}$$

The last inequality in (5.23) uses the fact that if event $\mathcal{V}^{(r)}$ is true, we have $\widehat{\theta}_{j^{(s)*}}^{(s)} < \mu_{j^{(s)*},i^*} + 8 \cdot \frac{1}{2^s}$.

The arguments of (5.22) and (5.23) together imply that the best Level-I arm j^* has no chance to be eliminated in epoch s, which yields a contradiction.

Step 2: As we have shown that $j^* \in \mathcal{A}^{(r)}$, we now show that a sub-optimal Level-I arm $j \in \mathcal{A}^{(r)}$ with a mean reward gap Δ_j will not be in $\mathcal{A}^{(r+1)}$ when $r \geq \lambda_j$ and event $\mathcal{V}^{(r)}$ is true.

For a sub-optimal Level-I arm $j \in \mathcal{A}^{(r)}$, if event $\mathcal{V}^{(r)}$ is true, we have

$$\widehat{\theta}_{j}^{(r)} + 2 \cdot 8 \cdot \tfrac{1}{2^r} \quad < \quad \left(\mu_{j,i^*} + 8 \cdot \tfrac{1}{2^r} \right) + 2 \cdot 8 \cdot \tfrac{1}{2^r} = \mu_{j,i^*} + \tfrac{24}{2^r} \leq \mu_{j,i^*} + \tfrac{\Delta_j}{2} \quad . \tag{5.24}$$

The first inequality in (5.24) uses the fact that if event $\mathcal{V}^{(r)}$ is true, we have $\widehat{\theta}_{j}^{(r)} < \mu_{j,i^*} + 8 \cdot \frac{1}{2^r}$. The second inequality uses the fact that $\frac{1}{2^r} \leq \frac{1}{2^{\lambda_j}} \leq \frac{\Delta_j}{48}$ when $r \geq \lambda_j$.

Simultaneously, if event $\mathcal{V}^{(r)}$ is true, then for the best Level-I arm j^*, we also have

$$\widehat{\theta}_{j^*}^{(r)} - 8 \cdot \tfrac{1}{2^r} \quad > \quad \left(\mu_{j^*,i^*} - 2 \cdot 8 \cdot \tfrac{1}{2^r} \right) - 8 \cdot \tfrac{1}{2^r} = \mu_{j^*,i^*} - \tfrac{24}{2^r} \geq \mu_{j^*,i^*} - \tfrac{\Delta_j}{2} = \mu_{j,i^*} + \tfrac{\Delta_j}{2}, \tag{5.25}$$

which indicates that the sub-optimal Level-I arm $j \in \mathcal{A}^{(r)} \backslash \{j^*\}$ will be eliminated by the end of epoch r, when $r \geq \lambda_j$. The first inequality in (5.25) uses the fact that

if event $\mathcal{V}^{(r)}$ is true, we have $\widehat{\theta}_{j^*}^{(r)} > \mu_{j^*,i^*} - 2 \cdot 8 \cdot \frac{1}{2^r}$. The second inequality uses the fact that $-\frac{1}{2^r} \geq -\frac{1}{2^{\lambda_j}} \geq -\frac{\Delta_j}{48}$ when $r \geq \lambda_j$. □

Proof of Lemma 17: All T rounds can be divided into at most $\log(T)$ epochs. Let $\tau_r + 1$ be the first round of epoch r, i.e., all rounds $t \in \{\tau_r + 1, \tau_r + 2, \ldots, \tau_{r+1}\}$ are in epoch r. Note that each active Level-II arm will be pulled exactly $2\log(KT) \cdot 2^{2r}$ times in epoch r. We set $\tau_1 = 0$. We have

$$
\begin{aligned}
\text{LHS in (5.12)} &= \sum_{t=1}^{T} \mathbb{E}\left[\mathbf{1}\left\{(J_t, I_t) = (j, i^*)\right\}\right] \cdot \Delta_j \\
&\leq \sum_{r=1}^{\log(T)} \mathbb{E}\left[\sum_{t=\tau_r+1}^{\tau_{r+1}} \mathbf{1}\left\{(J_t, I_t) = (j, i^*)\right\}\right] \cdot \Delta_j \\
&\leq \sum_{r=1}^{\lceil \lambda_j \rceil} \mathbb{E}\left[\sum_{t=\tau_r+1}^{\tau_{r+1}} \mathbf{1}\left\{(J_t, I_t) = (j, i^*)\right\}\right] \cdot \Delta_j \\
&+ \sum_{r=\lceil \lambda_j \rceil+1}^{\log(T)} \mathbb{E}\left[\sum_{t=\tau_r+1}^{\tau_{r+1}} \mathbf{1}\left\{(J_t, I_t) = (j, i^*)\right\}\right] \cdot \Delta_j \\
&\leq \sum_{r=1}^{\lceil \lambda_j \rceil} 2\log(KT) \cdot 2^{2r} \cdot \Delta_j \\
&+ \sum_{r=\lceil \lambda_j \rceil+1}^{\log(T)} \mathbb{E}\left[\sum_{t=\tau_r+1}^{\tau_{r+1}} \mathbf{1}\left\{(J_t, I_t) = (j, i^*)\right\}\right] \cdot \Delta_j \\
&\leq O\left(\frac{\log(KT)}{\Delta_j}\right) + \underbrace{\sum_{r=\lceil \lambda_j \rceil+1}^{\log(T)} \mathbb{E}\left[\sum_{t=\tau_r+1}^{\tau_{r+1}} \mathbf{1}\left\{(J_t, I_t) = (j, i^*)\right\}\right] \cdot \Delta_j}_{(\zeta)} \quad .
\end{aligned}
\tag{5.26}
$$

Then we decompose term (ζ) in (5.26) into two parts based on events $\mathcal{V}^{(r-1)}$ and $\overline{\mathcal{V}^{(r-1)}}$. We have

$$
\begin{aligned}
(\zeta) &= \sum_{r=\lceil \lambda_j \rceil + 1}^{\log(T)} \mathbb{E}\left[\sum_{t=\tau_r+1}^{\tau_{r+1}} \mathbf{1}\left\{(J_t, I_t) = (j, i^*)\right\}\right] \cdot \Delta_j \\
&\leq \sum_{r=\lceil \lambda_j \rceil + 1}^{\log(T)} \mathbb{E}\left[\sum_{t=\tau_r+1}^{\tau_{r+1}} \underbrace{\mathbf{1}\left\{(J_t, I_t) = (j, i^*), \mathcal{V}^{(r-1)}\right\}}_{=0, \text{ Lemma 16}}\right] \cdot \Delta_j \\
&+ \underbrace{\sum_{r=\lceil \lambda_j \rceil + 1}^{\log(T)} \mathbb{E}\left[\sum_{t=\tau_r+1}^{\tau_{r+1}} \mathbf{1}\left\{\overline{\mathcal{V}^{(r-1)}}\right\}\right] \cdot \Delta_j}_{\leq O\left(\frac{1}{K}\right), \text{ Lemma 15}} \\
&\leq O\left(\frac{1}{K}\right) \quad .
\end{aligned}
\tag{5.27}
$$

By plugging the upper bound of (ζ) into (5.26), we conclude the proof. □

Proof of Lemma 18: All T rounds can be divided into at most $\log(T)$ epochs. Let $\tau_r + 1$ be the first round when epoch r starts. This implies that all rounds $t \in \{\tau_r + 1, \tau_r + 2, \ldots, \tau_{r+1}\}$ are in epoch r and each active Level-II arm will be pulled exactly $2\log(KT) \cdot 2^{2r}$ times. We set $\tau_1 = 0$. We have

$$
\begin{aligned}
\text{LHS in (5.13)} &= \sum_{t=1}^{T} \mathbb{E}\left[\mathbf{1}\left\{(J_t, I_t) = (j, i)\right\}\right] \cdot \left(\Delta_i^{(j)} + \Delta_j\right) \\
&\leq \sum_{r=1}^{\log(T)} \mathbb{E}\left[\sum_{t=\tau_r+1}^{\tau_{r+1}} \mathbf{1}\left\{(J_t, I_t) = (j, i)\right\}\right] \cdot \left(\Delta_i^{(j)} + \Delta_j\right) \\
&\leq \underbrace{\sum_{r=1}^{\log(T)} \mathbb{E}\left[\sum_{t=\tau_r+1}^{\tau_{r+1}} \mathbf{1}\left\{(J_t, I_t) = (j, i)\right\}\right] \cdot 2 \cdot \max\left\{\Delta_i^{(j)}, \Delta_j\right\}}_{(\eta)} \quad .
\end{aligned}
\tag{5.28}
$$

We decompose term (η) based on events $\mathcal{E}_j^{(r-1)}$, $\mathcal{V}^{(r-1)}$, and their complementary events. We have

$$
\begin{aligned}
(\eta) \;\leq\; & \sum_{r=1}^{\log(T)} \mathbb{E}\left[\sum_{t=\tau_r+1}^{\tau_{r+1}} \mathbf{1}\left\{(J_t, I_t) = (j,i), \mathcal{E}_j^{(r-1)}, \mathcal{V}^{(r-1)}\right\}\right] \cdot 2 \cdot \max\left\{\Delta_i^{(j)}, \Delta_j\right\} \\
& + \underbrace{\sum_{r=1}^{\log(T)} \mathbb{E}\left[\sum_{t=\tau_r+1}^{\tau_{r+1}} \mathbf{1}\left\{\overline{\mathcal{E}_j^{(r-1)}}\right\}\right] \cdot 2 \cdot \max\left\{\Delta_i^{(j)}, \Delta_j\right\}}_{\leq O\left(\frac{1}{K}\right), \text{ Lemma 12}} \\
& + \underbrace{\sum_{r=1}^{\log(T)} \mathbb{E}\left[\sum_{t=\tau_r+1}^{\tau_{r+1}} \mathbf{1}\left\{\overline{\mathcal{V}^{(r-1)}}\right\}\right] \cdot 2 \cdot \max\left\{\Delta_i^{(j)}, \Delta_j\right\}}_{\leq O\left(\frac{1}{K}\right), \text{ Lemma 15}} \\
\leq\; & \underbrace{\sum_{r=1}^{\log(T)} \mathbb{E}\left[\sum_{t=\tau_r+1}^{\tau_{r+1}} \mathbf{1}\left\{(J_t, I_t) = (j,i), \mathcal{E}_j^{(r-1)}, \mathcal{V}^{(r-1)}\right\}\right] \cdot 2 \cdot \max\left\{\Delta_i^{(j)}, \Delta_j\right\}}_{(\omega)} \\
& + O\left(\frac{1}{K}\right).
\end{aligned}
\tag{5.29}
$$

We consider two cases to upper bound term (ω) based on whether $\Delta_i^{(j)} \leq \Delta_j$ or $\Delta_i^{(j)} > \Delta_j$.

Case I: if $\Delta_j = \max\left\{\Delta_i^{(j)}, \Delta_j\right\}$, we have

$$
\begin{aligned}
(\omega) \;=\; & \sum_{r=1}^{\log(T)} \mathbb{E}\left[\sum_{t=\tau_r+1}^{\tau_{r+1}} \mathbf{1}\left\{(J_t, I_t) = (j,i), \mathcal{E}_j^{(r-1)}, \mathcal{V}^{(r-1)}\right\}\right] \cdot 2 \cdot \Delta_j \\
\leq\; & \sum_{r=1}^{\log(T)} \mathbb{E}\left[\sum_{t=\tau_r+1}^{\tau_{r+1}} \mathbf{1}\left\{(J_t, I_t) = (j,i), \mathcal{V}^{(r-1)}\right\}\right] \cdot 2 \cdot \Delta_j \\
=\; & \sum_{r=1}^{\lceil \lambda_j \rceil} \mathbb{E}\left[\sum_{t=\tau_r+1}^{\tau_{r+1}} \mathbf{1}\left\{(J_t, I_t) = (j,i), \mathcal{V}^{(r-1)}\right\}\right] \cdot 2 \cdot \Delta_j \\
& + \sum_{r=\lceil \lambda_j \rceil + 1}^{\log(T)} \mathbb{E}\left[\sum_{t=\tau_r+1}^{\tau_{r+1}} \underbrace{\mathbf{1}\left\{(J_t, I_t) = (j,i), \mathcal{V}^{(r-1)}\right\}}_{=0, \text{ Lemma 16}}\right] \cdot 2 \cdot \Delta_j \\
\leq\; & \sum_{r=1}^{\lceil \lambda_j \rceil} 2\log(KT) \cdot 2^{2r} \cdot 2 \cdot \Delta_j \\
\leq\; & O\left(\frac{\log(KT)}{\Delta_j}\right) \\
=\; & O\left(\frac{\log(KT)}{\max\left\{\Delta_j, \Delta_i^{(j)}\right\}}\right).
\end{aligned}
\tag{5.30}
$$

Case II: if $\Delta_i^{(j)} = \max\left\{\Delta_i^{(j)}, \Delta_j\right\}$, we have

$$
\begin{aligned}
(\omega) &= \sum_{r=1}^{\log(T)} \mathbb{E}\left[\sum_{t=\tau_r+1}^{\tau_{r+1}} \mathbf{1}\left\{(J_t, I_t) = (j,i), \mathcal{E}_j^{(r-1)}, \mathcal{V}^{(r-1)}\right\}\right] \cdot 2 \cdot \Delta_i^{(j)} \\
&\leq \sum_{r=1}^{\log(T)} \mathbb{E}\left[\sum_{t=\tau_r+1}^{\tau_{r+1}} \mathbf{1}\left\{(J_t, I_t) = (j,i), \mathcal{E}_j^{(r-1)}\right\}\right] \cdot 2 \cdot \Delta_i^{(j)} \\
&\leq \sum_{r=1}^{\left\lceil \log\left(\frac{64}{\Delta_i^{(j)}}\right)\right\rceil} \mathbb{E}\left[\sum_{t=\tau_r+1}^{\tau_{r+1}} \mathbf{1}\left\{(J_t, I_t) = (j,i)\right\}\right] \cdot 2 \cdot \Delta_i^{(j)} \\
&+ \sum_{r=\left\lceil \log\left(\frac{64}{\Delta_i^{(j)}}\right)\right\rceil + 1}^{\log(T)} \mathbb{E}\left[\sum_{t=\tau_r+1}^{\tau_{r+1}} \underbrace{\mathbf{1}\left\{(J_t, I_t) = (j,i), \mathcal{E}_j^{(r-1)}\right\}}_{=0, \text{ Lemma 14}}\right] \cdot 2 \cdot \Delta_i^{(j)} \qquad (5.31) \\
&\leq \sum_{r=1}^{\left\lceil \log\left(\frac{64}{\Delta_i^{(j)}}\right)\right\rceil} 2\log(KT) \cdot 2^{2r} \cdot 2 \cdot \Delta_i^{(j)} \\
&= O\left(\frac{\log(KT)}{\Delta_i^{(j)}}\right) \\
&= O\left(\frac{\log(KT)}{\max\left\{\Delta_j, \Delta_i^{(j)}\right\}}\right).
\end{aligned}
$$

□

5.7.2 Proofs of Theorem 21

Proof of Theorem 21: Recall that $X_{1:T}$ is the original reward vector sequence and $X'_{1:T}$ is an arbitrary neighbouring reward vector sequence of $X_{1:T}$ such that they can differ in an arbitrary round. Let us say $X_{1:T}$ and $X'_{1:T}$ differ from each other in round t.

Let $D_{1:T} \in \mathcal{A}^T$ be the sequence of decisions made by Learner through round 1 to round T when working over $X_{1:T}$. Let $D'_{1:T} \in \mathcal{A}^T$ be the sequence of decisions made by Learner when working over $X'_{1:T}$. Let set $\mathcal{K}$ collect all the Level-II arms. Let $W_{1:T} \in \mathcal{K}^T$ be the sequence of decisions made by Level-I arms through round 1 to round T when taking $X_{1:T}$ as input. Let $W'_{1:T} \in \mathcal{K}^T$ be the sequence of decisions made by Level-I arms through round 1 to round T when taking $X'_{1:T}$ as input.

For an arbitrary pair $(\sigma_{1:T}, \rho_{1:T}) \in \mathcal{A}^T \times \mathcal{K}^T$, we claim that

$$\mathbb{P}\left\{(D_{1:T}, W_{1:T}) = (\sigma_{1:T}, \rho_{1:T}) \mid X_{1:T}\right\} \leq e^{1.5\epsilon} \cdot \mathbb{P}\left\{\left(D'_{1:T}, W'_{1:T}\right) = (\sigma_{1:T}, \rho_{1:T}) \mid X'_{1:T}\right\}. \tag{5.32}$$

The RHS in (5.32) can be rewritten as

$$\begin{aligned} & \mathbb{P}\left\{(D_{1:T}, W_{1:T}) = (\sigma_{1:T}, \rho_{1:T}) \mid X_{1:T}\right\} \\ = \; & \mathbb{P}\left\{(D_{1:t}, W_{1:t}) = (\sigma_{1:t}, \rho_{1:t}) \mid X_{1:T}\right\} \\ & \underbrace{\mathbb{P}\left\{(D_{t+1:T}, W_{t+1:T}) = (\sigma_{t+1:T}, \rho_{t+1:T}) \mid (D_{1:t}, W_{1:t}) = (\sigma_{1:t}, \rho_{1:t}), X_{1:T}\right\}}_{\psi} \quad . \end{aligned} \tag{5.33}$$

Similarly, we have

$$\begin{aligned} & \mathbb{P}\left\{(D'_{1:T}, W'_{1:T}) = (\sigma_{1:T}, \rho_{1:T}) \mid X'_{1:T}\right\} \\ = \; & \mathbb{P}\left\{(D'_{1:t}, W'_{1:t}) = (\sigma_{1:t}, \rho_{1:t}) \mid X'_{1:T}\right\} \\ & \underbrace{\mathbb{P}\left\{\left(D'_{t+1:T}, W'_{t+1:T}\right) = (\sigma_{t+1:T}, \rho_{t+1:T}) \mid \left(D'_{1:t}, W'_{1:t}\right) = (\sigma_{1:t}, \rho_{1:t}), X'_{1:T}\right\}}_{\psi'} \quad . \end{aligned} \tag{5.34}$$

Since $X_{1:t-1} = X'_{1:t-1}$, we have

$$\mathbb{P}\left\{(D_{1:t}, W_{1:t}) = (\sigma_{1:t}, \rho_{1:t}) \mid X_{1:T}\right\} = \mathbb{P}\left\{\left(D'_{1:t}, W'_{1:t}\right) = (\sigma_{1:t}, \rho_{1:t}) \mid X'_{1:T}\right\} \quad . \tag{5.35}$$

Now, we only need to show $\psi \leq e^{1.5\epsilon}\psi'$ to conclude the proof.
We first rewrite ψ as

$$\begin{aligned} \psi \; = \; & \mathbb{P}\left\{(D_{t+1:T}, W_{t+1:T}) = (\sigma_{t+1:T}, \rho_{t+1:T}) \mid (D_{1:t}, W_{1:t}) = (\sigma_{1:t}, \rho_{1:t}), X_{1:T}\right\} \\ = \; & \underbrace{\mathbb{P}\left\{D_{t+1:T} = \sigma_{t+1:T} \mid (D_{1:t}, W_{1:t}) = (\sigma_{1:t}, \rho_{1:t}), X_{1:T}\right\}}_{\zeta} \\ & \underbrace{\mathbb{P}\left\{W_{t+1:T} = \rho_{t+1:T} \mid D_{t+1:T} = \sigma_{t+1:T}, (D_{1:t}, W_{1:t}) = (\sigma_{1:t}, \rho_{1:t}), X_{1:T}\right\}}_{\phi} \quad . \end{aligned} \tag{5.36}$$

Similarly, we rewrite ψ' as

$$\begin{aligned}\psi' &= \mathbb{P}\left\{\left(D'_{t+1:T}, W'_{t+1:T}\right) = (\sigma_{t+1:T}, \rho_{t+1:T}) \mid (D'_{1:t}, W'_{1:t}) = (\sigma_{1:t}, \rho_{1:t}), X'_{1:T}\right\} \\ &= \underbrace{\mathbb{P}\left\{D'_{t+1:T} = \sigma_{t+1:T} \mid \left(D'_{1:t}, W'_{1:t}\right) = (\sigma_{1:t}, \rho_{1:t}), X'_{1:T}\right\}}_{\zeta'} \\ &\quad \underbrace{\mathbb{P}\left\{W'_{t+1:T} = \rho_{t+1:T} \mid D'_{t+1:T} = \sigma_{t+1:T}, \left(D'_{1:t}, W'_{1:t}\right) = (\sigma_{1:t}, \rho_{1:t}), X'_{1:T}\right\}}_{\phi'} .\end{aligned} \tag{5.37}$$

Recall that (J_t, I_t) is the decision pair in round t. Let r be the epoch that includes round t when taking $X_{1:T}$ as input. Let H be the first round such that, at the end of this round, observation $X_{J_t,I_t}(t)$ will be used by Level-I arm J_t, i.e., Level-I arm J_t will finish pulling all the managed active Level-II arms in epoch r at the end of round H and do the elimination to update the active Level-II arm set. Let Q be the first round such that, at the end of this round, observation $Y_{J_t}(t) = X_{J_t,I_t}(t)$ will be used by Learner, i.e., Learner will finish pulling all the active Level-I arms in epoch r at the end of round Q and do the elimination to update the active Level-I arm set.

Similarly, let (J'_t, I'_t), r', H' and Q' be the corresponding parameters when working over $X'_{1:T}$.

Now, we show $\zeta \leq e^{\epsilon}\zeta'$ and $\phi \leq e^{0.5\epsilon}\phi'$, respectively.

Proofs of $\zeta \leq e^{\epsilon}\zeta'$:

We rewrite ζ first and have

$$\begin{aligned}&\zeta \\ =\ & \mathbb{P}\left\{D_{t+1:T} = \sigma_{t+1:T} \mid (D_{1:t}, W_{1:t}) = (\sigma_{1:t}, \rho_{1:t}), X_{1:T}\right\} \\ =\ & \sum_{j\in\mathcal{A}}\sum_{i\in\mathcal{K}}\sum_{s=1}^{\log(T)}\sum_{h=t}^{T}\sum_{q=h}^{T} \\ & \mathbb{P}\left\{\underbrace{(J_t, I_t) = (j,i), r = s, H = h, Q = q}_{=:G} \mid (D_{1:t}, W_{1:t}) = (\sigma_{1:t}, \rho_{1:t}), X_{1:T}\right\} \\ & \underbrace{\mathbb{P}\left\{D_{t+1:T} = \sigma_{t+1:T} \mid \underbrace{G, (D_{1:t}, W_{1:t}) = (\sigma_{1:t}, \rho_{1:t}), X_{1:T}}_{=:M}\right\}}_{\alpha} .\end{aligned} \tag{5.38}$$

Similarly, ζ' can be rewritten as

$$\begin{aligned}
& \zeta' \\
= \; & \mathbb{P}\left\{D'_{t+1:T} = \sigma_{t+1:T} \mid (D'_{1:t}, W'_{1:t}) = (\sigma_{1:t}, \rho_{1:t}), X'_{1:T}\right\} \\
= \; & \sum_{j\in\mathcal{A}} \sum_{i\in\mathcal{K}} \sum_{s=1}^{\log(T)} \sum_{h=t}^{T} \sum_{q=h}^{T} \\
& \underbrace{\mathbb{P}\left\{ \underbrace{(J'_t, I'_t) = (j,i), r' = s, H' = h, Q' = q}_{=:G'} \mid (D'_{1:t}, W'_{1:t}) = (\sigma_{1:t}, \rho_{1:t}), X'_{1:T} \right\}}_{\alpha'} \\
& \underbrace{\mathbb{P}\left\{ D'_{t+1:T} = \sigma_{t+1:T} \mid \underbrace{G', (D'_{1:t}, W'_{1:t}) = (\sigma_{1:t}, \rho_{1:t}), X'_{1:T}}_{=:M'} \right\}}_{\alpha'} .
\end{aligned} \tag{5.39}$$

Note that only α and α' may be different in ζ and ζ'. We now rewrite α and α', and have

$$\alpha = \mathbb{P}\left\{D_{t+1:Q} = \sigma_{t+1:Q} \mid M\right\} \cdot \underbrace{\mathbb{P}\left\{D_{Q+1:T} = \sigma_{Q+1:T} \mid M, D_{t+1:Q} = \sigma_{t+1:Q}\right\}}_{\gamma} , \tag{5.40}$$

and

$$\alpha' = \mathbb{P}\left\{D'_{t+1:Q'} = \sigma_{t+1:Q'} \mid M'\right\} \cdot \underbrace{\mathbb{P}\left\{D'_{Q'+1:T} = \sigma_{Q'+1:T} \mid M', D'_{t+1:Q'} = \sigma_{t+1:Q'}\right\}}_{\gamma'} . \tag{5.41}$$

We now only need to analyze γ and γ', and we will show that $\gamma \leq e^{\epsilon}\gamma'$ by using the properties that differential privacy is immune to post-processing (Proposition 2.1 in [16], it is also shown in Proposition 1) and differentially private algorithms can be composed (Theorem 3.16 in [16], it is also shown in Theorem 7).

Recall that $\widetilde{\theta}^{(r)}_{J_t}$ is the differentially private empirical mean of an active Level-I arm J_t among all the collected observations in $\mathcal{O}^{(r)}_{J_t}$ at the end of epoch r when working over $X_{1:T}$. Let $\widetilde{\theta}^{(r')}_{J'_t}$ be the differentially private empirical mean of an active Level-I arm J'_t among all the collected observations in $\mathcal{O}^{(r')}_{J'_t}$ at the end of epoch r' when working over $X'_{1:T}$.

Recall that the first private algorithm (at Level-I arm's side) is to compute the differentially private empirical mean among all the collected observations in $\mathcal{O}^{(r)}_{J_t}$.

As a noise drawn from Lap $\left(\frac{1}{0.5\epsilon}\right)$ is injected, from Theorem 3.6 in [16] (it is also shown in Theorem 5), we know that, for any $a \in \mathbb{R}$, we have

$$\begin{aligned} & \mathbb{P}\left\{\widetilde{\theta}_{J_t}^{(r)} \cdot \left|\mathcal{A}_{J_t}^{(r)}\right| \cdot L^{(r)} = a \mid M, D_{t+1:Q} = \sigma_{t+1:Q}\right\} \\ \leq \; & e^{0.5\epsilon} \cdot \mathbb{P}\left\{\widetilde{\theta}_{J_t'}^{(r')} \cdot \left|\mathcal{A}_{J_t'}^{(r')}\right| \cdot L^{(r')} = a \mid M', D'_{t+1:Q'} = \sigma_{t+1:Q'}\right\} \quad . \end{aligned} \tag{5.42}$$

For the remaining active Level-I arms except the one pulled in round t, the conditional distribution of the differentially private empirical means cannot be impacted by the changed reward vector in round t.

Recall that $\widetilde{\mu}_{J_t,I_t}^{(r)}$ is the differentially private empirical mean of an active Level-II arm (J_t, I_t) among all the collected observations in $\mathcal{O}_{J_t,I_t}^{(r)}$ at the end of epoch r (i.e., at the end of round H). Let $\widetilde{\mu}_{J_t',I_t'}^{(r')}$ be the differentially private empirical mean of an active Level-II arm (J_t', I_t') among all the collected observations in $\mathcal{O}_{J_t',I_t'}^{(r')}$ at the end of epoch r' (i.e., at the end of round H').

As a noise drawn from Lap $\left(\frac{1}{0.5\epsilon}\right)$ is injected, from Theorem 3.6 in [16] (it is also shown in Theorem 5), for any $b \in \mathbb{R}$, we have

$$\begin{aligned} & \mathbb{P}\left\{\widetilde{\mu}_{J_t,I_t}^{(r)} \cdot L^{(r)} = b \mid M, D_{t+1:Q} = \sigma_{t+1:Q}\right\} \\ \leq \; & e^{0.5\epsilon} \cdot \mathbb{P}\left\{\widetilde{\mu}_{J_t',I_t'}^{(r')} \cdot L^{(r')} = b \mid M', D'_{t+1:Q'} = \sigma_{t+1:Q'}\right\} \quad . \end{aligned} \tag{5.43}$$

For the remaining active Level-II arms except the pulled one in round t, the conditional distribution of the differentially private empirical means cannot be impacted by the changed reward vector in round t.

Recall that the second private algorithm (at Level-I arm's side) is to do the private elimination and compute the updated size of the active Level-II arm set. From the property that differential privacy is immune to post-processing (Proposition 2.1 in [16], it is also shown in Propostion 1), for any $c \in \mathbb{N}$, we have

$$\begin{aligned} & \mathbb{P}\left\{\left|\mathcal{A}_{J_t}^{(r+1)}\right| = c \mid M, D_{t+1:Q} = \sigma_{t+1:Q}\right\} \\ \leq \; & e^{0.5\epsilon} \cdot \mathbb{P}\left\{\left|\mathcal{A}_{J_t'}^{(r'+1)}\right| = c \mid M', D'_{t+1:Q'} = \sigma_{t+1:Q'}\right\} \quad . \end{aligned} \tag{5.44}$$

For the remaining active Level-I arms except the pulled one in round t, the distribution of the updated size of the active Level-II arm set cannot be impacted by the changed reward vector in round t.

As $X_{Q+1:T} = X'_{Q'+1:T}$ conditioned on $Q = q$ and $Q' = q$, from the property that

differential privacy is immune to post-processing (Proposition 2.1 in [16]), we have

$$\mathbb{P}\left\{D_{Q+1:T}=\sigma_{Q+1:T} \mid M, D_{t+1:Q}=\sigma_{t+1:Q},\, \widetilde{\theta}_{J_t}^{(r)} \left|\mathcal{A}_{J_t}^{(r)}\right| L^{(r)}=a, \left|\mathcal{A}_{J_t}^{(r+1)}\right|=c\right\} = \mathbb{P}\left\{D'_{Q'+1:T}=\sigma_{Q'+1:T} \mid M', D'_{t+1:Q'}=\sigma_{t+1:Q'},\, \widetilde{\theta}_{J'_t}^{(r')} \left|\mathcal{A}_{J'_t}^{(r')}\right| L^{(r')}=a, \left|\mathcal{A}_{J'_t}^{(r'+1)}\right|=c\right\}. \tag{5.45}$$

By combining (5.42), (5.44), and (5.45), we have $\gamma \leq e^{\epsilon}\gamma'$, which implies $\zeta \leq e^{\epsilon}\zeta'$.

Proofs of $\phi \leq e^{0.5\epsilon}\phi'$:

We first rewrite ϕ as

$$\begin{aligned}
&\phi \\
=\;& \mathbb{P}\left\{W_{t+1:T}=\rho_{t+1:T} \mid \underbrace{D_{t+1:T}=\sigma_{t+1:T}, (D_{1:t}, W_{1:t})=(\sigma_{1:t},\rho_{1:t}), X_{1:T}}_{=:Y}\right\} \\
=\;& \sum_{j\in\mathcal{A}}\sum_{i\in\mathcal{K}}\sum_{s=1}^{\log(T)}\sum_{h=t}^{T}\sum_{q=h}^{T} \mathbb{P}\left\{\underbrace{(J_t,I_t)=(j,i), r=s, H=h, Q=q}_{=:V} \mid Y\right\} \\
& \underbrace{\mathbb{P}\left\{W_{t+1:T}=\rho_{t+1:T} \mid \underbrace{V, D_{t+1:T}=\sigma_{t+1:T}, (D_{1:t}, W_{1:t})=(\sigma_{1:t},\rho_{1:t}), X_{1:T}}_{=:Z}\right\}}_{\eta}.
\end{aligned} \tag{5.46}$$

Similarly, we have

$$\begin{aligned}
&\phi' \\
=\;& \mathbb{P}\left\{W'_{t+1:T}=\rho_{t+1:T} \mid \underbrace{D'_{t+1:T}=\sigma_{t+1:T}, (D'_{1:t}, W'_{1:t})=(\sigma_{1:t},\rho_{1:t}), X'_{1:T}}_{=:Y'}\right\} \\
=\;& \sum_{j\in\mathcal{A}}\sum_{i\in\mathcal{K}}\sum_{s=1}^{\log(T)}\sum_{h=t}^{T}\sum_{q=h}^{T} \mathbb{P}\left\{\underbrace{(J'_t,I'_t)=(j,i), r'=s, H'=h, Q'=q}_{=:V'} \mid Y'\right\} \\
& \underbrace{\mathbb{P}\left\{W'_{t+1:T}=\rho_{t+1:T} \mid \underbrace{V', D'_{t+1:T}=\sigma_{t+1:T}, (D'_{1:t}, W'_{1:t})=(\sigma_{1:t},\rho_{1:t}), X'_{1:T}}_{=:Z'}\right\}}_{\eta'}.
\end{aligned} \tag{5.47}$$

Now, we show $\eta \leq e^{0.5\epsilon}\eta'$ to conclude the proof.

We rewrite η as

$$\begin{aligned}\eta &= \mathbb{P}\left\{W_{t+1:T} = \rho_{t+1:T} \mid Z\right\} \\ &= \mathbb{P}\left\{W_{t+1:H} = \rho_{t+1:H} \mid Z\right\} \cdot \underbrace{\mathbb{P}\left\{W_{H+1:T} = \rho_{H+1:T} \mid W_{t+1:H} = \rho_{t+1:H}, Z\right\}}_{\chi}.\end{aligned} \tag{5.48}$$

Similarly, we have

$$\begin{aligned}\eta' &= \mathbb{P}\left\{W'_{t+1:T} = \rho_{t+1:T} \mid Z'\right\} \\ &= \mathbb{P}\left\{W'_{t+1:H'} = \rho_{t+1:H'} \mid Z'\right\} \cdot \underbrace{\mathbb{P}\left\{W'_{H'+1:T} = \rho_{H'+1:T} \mid W'_{t+1:H'} = \rho_{t+1:H'}, Z'\right\}}_{\chi'}.\end{aligned} \tag{5.49}$$

By using similar arguments that have been shown in (5.43) and (5.44), we have $\chi \leq e^{0.5\epsilon}\chi'$, which concludes the proof of $\phi \leq e^{0.5\epsilon}\phi'$.

By plugging $\zeta \leq e^{\epsilon}\zeta'$ and $\phi \leq e^{0.5\epsilon}\phi'$ into (5.36) and (5.36), respectively, we conclude the proof of Theorem 21.

□

5.7.3 Proofs of Theorem 22

Let $\mathcal{N}_j^{(r)} := \left\{\left|Z_{j,i}^{(l)}\right| < \frac{6\log(KT)}{\epsilon}, \forall l \in \{1,2\ldots,r\}, \forall i \in \mathcal{A}_j^{(l)}\right\}$ be the event that in all epochs $l \in \{1,2,\ldots,r\}$, for all the Level-II arms, the noise injected is not too much. Let $\overline{\mathcal{N}_j^{(r)}}$ be the complementary event of $\mathcal{N}_j^{(r)}$. Note that we define $\mathbf{1}\left\{\mathcal{N}_j^{(0)}\right\} = 1$. As we will show in Lemma 19 below, event $\overline{\mathcal{N}_j^{(r)}}$ is a low probability one.

Lemma 19. *In any epoch $r \geq 1$, we have $\mathbb{P}\left\{\overline{\mathcal{N}_j^{(r)}}\right\} = O\left(\frac{1}{K^3T^3}\right)$.*

Recall that $\theta_j^{(r)} = \mathbb{E}\left[\widehat{\theta}_j^{(r)}\right]$. As will be shown in the last claim of Lemma 20 below, $\theta_j^{(r)}$ is not too far from its best achievable mean reward μ_{j,i^*} with high probability.

Lemma 20. *In an epoch $r \geq 1$, for a Level-I arm $j \in \mathcal{A}^{(r)}$, if both events $\mathcal{E}_j^{(r)}$ and $\mathcal{N}_j^{(r)}$ are true, we have the following claims:*

(i): The best Level-II arm (j, i^*) will not be eliminated by the end of epoch r, i.e., $(j, i^*) \in \mathcal{A}_j^{(r+1)}$;

(ii): A sub-optimal Level-II arm $(j,i) \in \mathcal{A}_j^{(r)}$ *with mean reward* $\mu_{j,i}$ *will not be in* $\mathcal{A}_j^{(r+1)}$ *if* $\mu_{j,i} \leq \mu_{j,i^*} - 4 \cdot \frac{1}{2^r} - 12 \cdot \frac{1}{\epsilon \cdot 2^{2r}}$ *;*

(iii): $\mu_{j,i^*} - 8 \cdot \frac{1}{2^r} - 48 \cdot \frac{1}{\epsilon \cdot 2^{2r}} < \theta_j^{(r)} \leq \mu_{j,i^*}$.

Let $\gamma_{j,i} := \max\left\{\log\left(\frac{16}{\Delta_i^{(j)}}\right), \log\left(\sqrt{\frac{16}{\epsilon \cdot \Delta_i^{(j)}}}\right)\right\}$. The intuitive understanding of $\gamma_{j,i}$ is that once enough epochs have transpired, i.e., more than $\gamma_{j,i}$ epochs, a Level-I arm j can safely eliminate a sub-optimal Level-II arm (j,i) with a mean reward gap $\Delta_i^{(j)}$ with high probability. We formalize this intuition in the next lemma.

Lemma 21. *For a sub-optimal Level-II arm* (j,i)*, in any epoch* $r \geq \gamma_{j,i}$*, if events* $\mathcal{E}_j^{(r)}$ *and* $\mathcal{N}_j^{(r)}$ *are both true, we have that a Level-II arm* (j,i) *with a mean reward gap* $\Delta_i^{(j)}$ *will not be in* $\mathcal{A}_j^{(r+1)}$.

Let $\mathcal{V}^{(r)} := \left\{\mu_{j,i^*} - 2 \cdot \frac{8}{2^l} - \frac{48}{\epsilon \cdot 2^{2l}} < \widehat{\theta}_j^{(l)} < \mu_{j,i^*} + \frac{8}{2^l}, \forall l \in \{1,2,\ldots,r\}, \forall j \in \mathcal{A}^{(l)}\right\}$ be the event that in all epochs $l \in \{1,2,\ldots,r\}$, the empirical mean of a Level-I arm $j \in \mathcal{A}^{(l)}$ is not far from its best achievable true mean μ_{j,i^*}. Let $\overline{\mathcal{V}^{(r)}}$ be the complementary event of $\mathcal{V}^{(r)}$. Note that we set $\mathbf{1}\left\{\mathcal{V}^{(0)}\right\} = 1$. As we will show in Lemma 22 below, $\overline{\mathcal{V}^{(r)}}$ is a low probability one.

Lemma 22. *In any epoch* $r \geq 1$*, we have* $\mathbb{P}\left\{\overline{\mathcal{V}^{(r)}}\right\} = O\left(\frac{1}{K^2T^2}\right)$.

Let $\mathcal{N}^{(r)} := \left\{\left|Z_j^{(l)}\right| < \frac{6\log(KT)}{\epsilon}, \forall l \in \{1,2\ldots,r\}, \forall j \in \mathcal{A}^{(l)}\right\}$ be the event that in all epochs $l \in \{1,2,\ldots,r\}$, for all the Level-I arms, the noise injected is not too much. Let $\overline{\mathcal{N}^{(r)}}$ be the complementary event of $\mathcal{N}^{(r)}$. Note that we define $\mathbf{1}\left\{\mathcal{N}^{(0)}\right\} = 1$. As we will show in Lemma 23, event $\overline{\mathcal{N}^{(r)}}$ is a low probability one.

Lemma 23. *In any epoch* $r \geq 1$*, we have* $\mathbb{P}\left\{\overline{\mathcal{N}^{(r)}}\right\} = O\left(\frac{1}{K^3T^3}\right)$.

For a Level-I arm $j \in \mathcal{A}\backslash\{j^*\}$, let $\lambda_j := \max\left\{\log\left(\frac{216}{\Delta_j}\right), \log\left(\sqrt{\frac{216}{\epsilon \cdot \Delta_j}}\right)\right\}$. The intuitive understanding of λ_j is that if the epoch number has not progressed to λ_j, Learner can hardly make a decision to eliminate this sub-optimal Level-I arm j. However, if enough epochs have transpired, Learner can safely remove Level-I arm j from the active arm set with high probability. We formalize this intuition in the next lemma.

Lemma 24. *When* $r \geq \lambda_j$*, if events* $\mathcal{V}^{(r)}$ *and* $\mathcal{N}^{(r)}$ *are both true, we have that a sub-optimal Level-I arm* j *with a mean reward gap* Δ_j *will not be in* $\mathcal{A}^{(r+1)}$.

After all these preparations, we now prove Theorem 22.

Proof of Theorem 22: The regret shown in (5.1) can be further decomposed as

$$\begin{aligned}
&\mathcal{R}(T) \\
=& \sum_{j\in\mathcal{A}\backslash\{j^*\}} \underbrace{\sum_{t=1}^{T} \mathbb{E}\left[\mathbf{1}\{(J_t,I_t)=(j,i^*)\}\right]\cdot\Delta_j}_{\text{Lemma 25}} \\
+& \sum_{j\in\mathcal{A}} \sum_{i\in\mathcal{A}_j\backslash\{(j,i^*)\}} \underbrace{\sum_{t=1}^{T} \mathbb{E}\left[\mathbf{1}\{(J_t,I_t)=(j,i)\}\right]\cdot\left(\Delta_j+\Delta_i^{(j)}\right)}_{\text{Lemma 26}} \quad .
\end{aligned} \tag{5.50}$$

For the first term in (5.50), we prepare Lemma 25 to upper bound it, and for the second term in (5.50), we prepare Lemma 26 to upper bound it.

Lemma 25. *For a sub-optimal Level-I arm $j \in \mathcal{A}\backslash\{j^*\}$, we have*

$$\sum_{t=1}^{T} \mathbb{E}\left[\mathbf{1}\{(J_t,I_t)=(j,i^*)\}\right]\cdot\Delta_j = O\left(\max\left\{\frac{\log(KT)}{\Delta_j},\frac{\log(KT)}{\epsilon}\right\}\right)+O\left(\frac{1}{K}\right). \tag{5.51}$$

Lemma 26. *For a sub-optimal Level-II arm $(j,i) \in \mathcal{A}_j\backslash\{(j,i^*)\}$, we have*

$$\begin{aligned}
&\sum_{t=1}^{T} \mathbb{E}\left[\mathbf{1}\{(J_t,I_t)=(j,i)\}\right]\cdot\left(\Delta_i^{(j)}+\Delta_j\right) \\
=& \; O\left(\max\left\{\frac{\log(KT)}{\max\left\{\Delta_j,\Delta_i^{(j)}\right\}},\frac{\log(KT)}{\epsilon}\right\}\right)+O\left(\frac{1}{K}\right).
\end{aligned} \tag{5.52}$$

By plugging Lemma 25 and Lemma 26 into (5.50), we have

$$\begin{aligned}
\mathcal{R}(T) \;\leq\; & \sum_{j\in\mathcal{A}\backslash\{j^*\}} O\left(\max\left\{\frac{\log(KT)}{\Delta_j},\frac{\log(KT)}{\epsilon}\right\}\right) \\
&+ \sum_{j\in\mathcal{A}} \sum_{i\in\mathcal{A}_j\backslash\{(j,i^*)\}} O\left(\max\left\{\frac{\log(KT)}{\max\left\{\Delta_j,\Delta_i^{(j)}\right\}},\frac{\log(KT)}{\epsilon}\right\}\right) \\
=\; & \sum_{j\in\mathcal{A}} \sum_{i\in\mathcal{A}_j:\max\left\{\Delta_j,\Delta_i^{(j)}\right\}>0} O\left(\max\left\{\frac{\log(KT)}{\max\left\{\Delta_j,\Delta_i^{(j)}\right\}},\frac{\log(KT)}{\epsilon}\right\}\right) \quad ,
\end{aligned} \tag{5.53}$$

which concludes the proof of Theorem 22. □

We now present the proofs of Lemmas 19 through 26.

Proof of Lemma 19: From (5.9), we know that with probability at most $O\left(\frac{1}{K^4T^4}\right)$, we have $\left|Z_{j,i}^{(l)}\right| \geq \frac{6\log(KT)}{\epsilon} = \frac{3\ln(KT)}{\ln(2)\cdot 0.5\cdot\epsilon} > \frac{4\ln(KT)}{0.5\epsilon}$. Note that $Z_{j,i}^{(l)} \sim \text{Lap}\left(\frac{1}{0.5\epsilon}\right)$. Then, taking a union bound over all Level-II arms and over all the epochs concludes the proof. □

Proof of Lemma 20:

Claim (i): We use contradiction to prove this argument. Suppose that a Level-I arm j eliminates the best Level-II arm (j, i^*) in an epoch $s \leq r$ and events $\mathcal{E}_j^{(r)}$ and $\mathcal{N}_j^{(r)}$ are true. Recall that the elimination rule used by Level-I arm j in epoch s, which is shown in (5.7), is:

$$\widetilde{\mu}_{j,i}^{(s)} + \frac{1}{2^s} + \frac{3}{\epsilon\cdot 2^{2s}} < \max_{i'\in\mathcal{A}_j^{(s)}}\left(\widetilde{\mu}_{j,i'}^{(s)} - \frac{1}{2^s} - \frac{3}{\epsilon\cdot 2^{2s}}\right) \quad .$$

Recall that $L^{(s)} = 2\log(KT)\cdot 2^{2s}$.

If $\mathcal{E}_j^{(r)}$ and $\mathcal{N}_j^{(r)}$ are both true, then for the best Level-II arm (j, i^*), we have

$$\begin{aligned} & \widetilde{\mu}_{j,i^*}^{(s)} + \tfrac{1}{2^s} + \tfrac{3}{\epsilon\cdot 2^{2s}} \\ = & \left(\widehat{\mu}_{j,i^*}^{(s)} + \frac{Z_{j,i^*}^{(s)}}{L^{(s)}}\right) + \tfrac{1}{2^s} + \tfrac{3}{\epsilon\cdot 2^{2s}} \\ > & \left(\mu_{j,i^*} - \tfrac{1}{2^s}\right) + \left(-\tfrac{3}{\epsilon\cdot 2^{2s}}\right) + \tfrac{1}{2^s} + \tfrac{3}{\epsilon\cdot 2^{2s}} \\ = & \mu_{j,i^*} \quad . \end{aligned} \tag{5.54}$$

The first equality in (5.54) uses the fact that $\widetilde{\mu}_{j,i^*}^{(s)} = \widehat{\mu}_{j,i^*}^{(s)} + \frac{Z_{j,i}^{(s)}}{L^{(s)}}$. The first inequality uses the facts that if event $\mathcal{E}_j^{(r)}$ is true, we have $\widehat{\mu}_{j,i^*}^{(s)} > \mu_{j,i^*} - \frac{1}{2^s}$, and if event $\mathcal{N}_j^{(r)}$ is true, we have $Z_{j,i^*}^{(s)} > -\frac{6\log(KT)}{\epsilon}$, which implies $\frac{Z_{j,i^*}^{(s)}}{L^{(s)}} > \frac{3}{\epsilon\cdot 2^{2s}}$.

Let $(j, i_*^{(s)}) \leftarrow \arg\max_{i'\in\mathcal{A}_j^{(s)}}\left(\widetilde{\mu}_{j,i'}^{(s)} - \frac{1}{2^s} - \frac{3}{\epsilon\cdot 2^{2s}}\right)$. If $\mathcal{E}_j^{(r)}$ and $\mathcal{N}_j^{(r)}$ are both true, simul-

taneously, we also have

$$\begin{aligned} & \widetilde{\mu}_{j,i_*^{(s)}}^{(s)} - \tfrac{1}{2^s} - \tfrac{3}{\epsilon \cdot 2^{2s}} \\ < & \left(\mu_{j,i_*^{(s)}} + \tfrac{1}{2^s}\right) + \left(\tfrac{3}{\epsilon \cdot 2^{2s}}\right) - \tfrac{1}{2^s} - \tfrac{3}{\epsilon \cdot 2^{2s}} \\ = & \mu_{j,i_*^{(s)}} \quad . \end{aligned} \tag{5.55}$$

As $\mu_{j,i^*} \geq \mu_{j,i_*^{(s)}}$ always holds, it means that the best Level-II arm (j, i^*) will not be eliminated in epoch s, which yields a contradiction.

Claim (ii): For an epoch $s \leq r$, we show that a sub-optimal Level-II arm $(j, i) \in \mathcal{A}_j^{(s)}$ with mean reward $\mu_{j,i} < \mu_{j,i^*} - 4 \cdot \frac{1}{2^s} - 12 \cdot \frac{1}{\epsilon \cdot 2^{2s}}$ will not be in $\mathcal{A}_j^{(s+1)}$ if events $\mathcal{E}_j^{(r)}$ and $\mathcal{N}_j^{(r)}$ are true.

If $\mathcal{E}_j^{(r)}$ and $\mathcal{N}_j^{(r)}$ are true, for a sub-optimal Level-II arm $(j, i) \in \mathcal{A}_j^{(s)}$ with mean reward $\mu_{j,i} \leq \mu_{j,i^*} - 4 \cdot \frac{1}{2^s} - 12 \cdot \frac{1}{\epsilon \cdot 2^{2s}}$, we have

$$\begin{aligned} & \widetilde{\mu}_{j,i}^{(s)} + \tfrac{1}{2^s} + \tfrac{3}{\epsilon \cdot 2^{2s}} \\ < & \left(\mu_{j,i} + \tfrac{1}{2^s}\right) + \left(\tfrac{3}{\epsilon \cdot 2^{2s}}\right) + \tfrac{1}{2^s} + \tfrac{3}{\epsilon \cdot 2^{2s}} \\ = & \mu_{j,i} + \tfrac{2}{2^s} + \tfrac{6}{\epsilon \cdot 2^{2s}} \\ \leq & \mu_{j,i^*} - \tfrac{2}{2^s} - \tfrac{6}{\epsilon \cdot 2^{2s}} \quad . \end{aligned} \tag{5.56}$$

From claim (i), we know that the best Level-II arm (j, i^*) is always in the active Level-II arm set. Therefore, simultaneously, if $\mathcal{E}_j^{(r)}$ and $\mathcal{N}_j^{(r)}$ are both true, we also have

$$\begin{aligned} & \widetilde{\mu}_{j,i^*}^{(s)} - \tfrac{1}{2^s} - \tfrac{3}{\epsilon \cdot 2^{2s}} \\ > & \left(\mu_{j,i^*} - \tfrac{1}{2^s}\right) + \left(-\tfrac{3}{\epsilon \cdot 2^{2s}}\right) - \tfrac{1}{2^s} - \tfrac{3}{\epsilon \cdot 2^{2s}} \\ = & \mu_{j,i^*} - \tfrac{2}{2^s} - \tfrac{6}{\epsilon \cdot 2^{2s}} \quad , \end{aligned} \tag{5.57}$$

which means a sub-optimal Level-II arm (j, i) with mean reward $\mu_{j,i} \leq \mu_{j,i^*} - 4 \cdot \frac{1}{2^s} - 12 \cdot \frac{1}{\epsilon \cdot 2^{2s}}$ will not be in $\mathcal{A}_j^{(s+1)}$.

Claim (iii): Recall that $\theta_j^{(r)} = \mathbb{E}\left[\widehat{\theta}_j^{(r)}\right]$. Let $\mathcal{F}_j^{(r-1)}$ collect all the history information by the end of epoch $r-1$, i.e., collecting the pulled Level-II arms by Level-I

arm j, their rewards, and the injected noise. Then, $\theta_j^{(r)}$ can be rewritten as

$$\begin{aligned}
\theta_j^{(r)} &= \mathbb{E}\left[\widehat{\theta}_j^{(r)}\right] \\
&= \mathbb{E}\left[\mathbb{E}\left[\widehat{\theta}_j^{(r)} \mid \mathcal{F}_j^{(r-1)}\right]\right] \\
&= \mathbb{E}\left[\mathbb{E}\left[\frac{1}{\left|\mathcal{A}_j^{(r)}\right|} \cdot \sum_{i \in \mathcal{A}_j^{(r)}} \widehat{\mu}_{j,i}^{(r)} \mid \mathcal{F}_j^{(r-1)}\right]\right] \\
&= \mathbb{E}\left[\frac{1}{\left|\mathcal{A}_j^{(r)}\right|} \sum_{i \in \mathcal{A}_j^{(r)}} \mathbb{E}\left[\widehat{\mu}_{j,i}^{(r)} \mid \mathcal{F}_j^{(r-1)}\right]\right] \\
&= \mathbb{E}\left[\frac{1}{\left|\mathcal{A}_j^{(r)}\right|} \sum_{i \in \mathcal{A}_j^{(r)}} \mu_{j,i}\right] \quad .
\end{aligned} \tag{5.58}$$

The upper bound of $\theta_j^{(r)}$ is trivial to prove as we have

$$\theta_j^{(r)} = \mathbb{E}\left[\frac{1}{\left|\mathcal{A}_j^{(r)}\right|} \sum_{i \in \mathcal{A}_j^{(r)}} \mu_{j,i}\right] \leq \mathbb{E}\left[\frac{1}{\left|\mathcal{A}_j^{(r)}\right|} \sum_{i \in \mathcal{A}_j^{(r)}} \mu_{j,i^*}\right] = \mu_{j,i^*} \quad .$$

The proof of the lower bound of $\theta_j^{(r)}$ uses claim (ii). We have

$$\begin{aligned}
\theta_j^{(r)} &= \mathbb{E}\left[\frac{1}{\left|\mathcal{A}_j^{(r)}\right|} \sum_{i \in \mathcal{A}_j^{(r)}} \mu_{j,i}\right] \\
&> \mathbb{E}\left[\frac{1}{\left|\mathcal{A}_j^{(r)}\right|} \sum_{i \in \mathcal{A}_j^{(r)}} \left(\mu_{j,i^*} - 4 \cdot \frac{1}{2^{r-1}} - 12 \cdot \frac{1}{\epsilon \cdot 2^{2(r-1)}}\right)\right] \\
&= \mu_{j,i^*} - 8 \cdot \frac{1}{2^r} - 48 \cdot \frac{1}{\epsilon \cdot 2^{2r}} \quad .
\end{aligned}$$

□

Proof of Lemma 21: When $r \geq \gamma_{j,i}$, we have $4 \cdot \frac{1}{2^r} + 12 \cdot \frac{1}{\epsilon \cdot 2^{2r}} \leq 4 \cdot \frac{1}{2^{\gamma_{j,i}}} + 12 \cdot \frac{1}{\epsilon \cdot 2^{2\gamma_{j,i}}} =$

$\Delta_i^{(j)}$. Then, we have $\mu_{j,i} = \mu_{j,i^*} - \Delta_i^{(j)} \leq \mu_{j,i^*} - 4 \cdot \frac{1}{2^r} - 12 \cdot \frac{1}{\epsilon \cdot 2^{2r}}$. From claim (ii) in Lemma 20, we know that Level-II arm (j,i) with a mean reward gap $\Delta_i^{(j)}$ will not be in $\mathcal{A}_j^{(r+1)}$. □

Proof of Lemma 22: In certain steps during the proof, we use claim (iii) in Lemma 20 and Hoeffding's inequality. We have

$$\begin{aligned}
& \mathbb{P}\left\{\overline{\mathcal{V}^{(r)}}\right\} \\
\leq & \sum_{l=1}^{r} \sum_{j \in \mathcal{A}} \left(\mathbb{P}\left\{ \widehat{\theta}_j^{(l)} \leq \mu_{j,i^*} - 2 \cdot 8 \cdot \frac{1}{2^l} - 48 \cdot \frac{1}{\epsilon \cdot 2^{2l}} \right\} + \mathbb{P}\left\{ \widehat{\theta}_j^{(l)} \geq \mu_{j,i^*} + 8 \cdot \frac{1}{2^l} \right\} \right) \\
\leq & \sum_{l=1}^{r} \sum_{j \in \mathcal{A}} \underbrace{\mathbb{P}\left\{ \widehat{\theta}_j^{(l)} \leq \mu_{j,i^*} - 2 \cdot 8 \cdot \frac{1}{2^l} - 48 \cdot \frac{1}{\epsilon \cdot 2^{2l}}, \mathcal{E}_j^{(l)}, \mathcal{N}_j^{(l)} \right\}}_{(\zeta_1)} \\
+ & \sum_{l=1}^{r} \sum_{j \in \mathcal{A}} \underbrace{\mathbb{P}\left\{ \widehat{\theta}_j^{(l)} \geq \mu_{j,i^*} + 8 \cdot \frac{1}{2^l}, \mathcal{E}_j^{(l)}, \mathcal{N}_j^{(l)} \right\}}_{(\zeta_2)} \\
+ & \underbrace{\sum_{l=1}^{r} \sum_{j \in \mathcal{A}} \mathbb{P}\left\{ \overline{\mathcal{E}_j^{(l)}} \right\} + \sum_{l=1}^{r} \sum_{j \in \mathcal{A}} \mathbb{P}\left\{ \overline{\mathcal{N}_j^{(l)}} \right\}}_{\leq O\left(\frac{1}{K^2T^2}\right), \text{ Lemma 12 and Lemma 19}} .
\end{aligned} \tag{5.59}$$

We now upper-bound term (ζ_1). We have

$$\begin{aligned}
(\zeta_1) &= \mathbb{P}\left\{ \widehat{\theta}_j^{(l)} \leq \mu_{j,i^*} - 2 \cdot 8 \cdot \frac{1}{2^l} - 48 \cdot \frac{1}{\epsilon \cdot 2^{2l}}, \mathcal{E}_j^{(l)}, \mathcal{N}_j^{(l)} \right\} \\
&\leq \mathbb{P}\left\{ \widehat{\theta}_j^{(l)} \leq \mu_{j,i^*} - 2 \cdot 8 \cdot \frac{1}{2^l} - 48 \cdot \frac{1}{\epsilon \cdot 2^{2l}}, \quad \mu_{j,i^*} - 8 \cdot \frac{1}{2^l} - \frac{48}{\epsilon \cdot 2^{2l}} \leq \theta_j^{(l)} \leq \mu_{j,i^*} \right\} \\
&\leq \mathbb{P}\left\{ \widehat{\theta}_j^{(l)} \leq \theta_j^{(l)} - 8 \cdot \frac{1}{2^l} \right\} \\
&\leq e^{-\left|\mathcal{A}_j^{(l)}\right| \cdot 2\log(KT) \cdot 2^{2l} \cdot 64 \cdot \frac{1}{2^{2l}}} \\
&\leq O\left(\frac{1}{K^3T^3}\right) .
\end{aligned} \tag{5.60}$$

Similarly, we have

$$\begin{aligned}
(\zeta_2) &= \mathbb{P}\left\{ \widehat{\theta}_j^{(l)} \geq \mu_{j,i^*} + 8 \cdot \frac{1}{2^l}, \mathcal{E}_j^{(l)}, \mathcal{N}_j^{(l)} \right\} \\
&\leq \mathbb{P}\left\{ \widehat{\theta}_j^{(l)} \geq \mu_{j,i^*} + 8 \cdot \frac{1}{2^l}, \quad \mu_{j,i^*} - 8 \cdot \frac{1}{2^l} - \frac{48}{\epsilon \cdot 2^{2l}} \leq \theta_j^{(l)} \leq \mu_{j,i^*} \right\} \\
&\leq \mathbb{P}\left\{ \widehat{\theta}_j^{(l)} \geq \theta_j^{(l)} + 8 \cdot \frac{1}{2^l} \right\} \\
&\leq O\left(\frac{1}{K^3T^3}\right) .
\end{aligned} \tag{5.61}$$

The first inequalities in (5.60) and (5.61) use claim (iii) in Lemma 20. The last inequalities in (5.60) and (5.61) use Hoeffding's inequality. Note that $\widehat{\theta}_j^{(l)}$ is the empirical mean of $\left|\mathcal{A}_j^{(l)}\right| \cdot 2\log(KT) \cdot 2^{2l}$ observations with each observation in $[0,1]$. □

Proof of Lemma 23: From (5.9), we know that with probability at most $O\left(\frac{1}{K^4T^4}\right)$, we have $\left|Z_j^{(l)}\right| \geq \frac{6\log(KT)}{\epsilon} > \frac{4\ln(KT)}{0.5\epsilon}$. Note that $Z_j^{(r)} \sim \text{Lap}\left(\frac{1}{0.5\epsilon}\right)$. Then, taking a union bound over all Level-I arms and over all the epochs concludes the proof. □

Proof of Lemma 24: There are two steps needed to complete the proof. We first prove that the best Level-I arm j^* is will not be eliminated by the end of epoch r if events $\mathcal{V}^{(r)}$ and $\mathcal{N}^{(r)}$ are both true. We prove this argument by using contradiction. Then, we prove that a sub-optimal Level-I arm j with a mean reward gap Δ_j will not be in $\mathcal{A}^{(r+1)}$ when $r \geq \lambda_j$, and events $\mathcal{V}^{(r)}$ and $\mathcal{N}^{(r)}$ are both true.

Step 1:

Recall that $j^* = \arg\max_{j\in\mathcal{A}} \mu_{j,i^*}$. Suppose that Learner eliminates j^* in an epoch $s \leq r$ and events $\mathcal{V}^{(r)}$ and $\mathcal{N}^{(r)}$ are both true. Also, recall that according to (5.8), Learner will only eliminate Level-I arm $j^* \in \mathcal{A}^{(s)}$ in epoch s if

$$\widetilde{\theta}_{j^*}^{(s)} + 2\cdot 8\cdot\tfrac{1}{2^s} + 48\cdot\tfrac{1}{\epsilon\cdot 2^{2s}} + \tfrac{6\log(KT)}{\epsilon\cdot\left|\mathcal{A}_{j^*}^{(s)}\right|\cdot L^{(s)}} \quad < \quad \max_{j'\in\mathcal{A}^{(s)}} \left(\widetilde{\theta}_{j'}^{(s)} - 8\cdot\tfrac{1}{2^s} - \tfrac{6\log(KT)}{\epsilon\cdot\left|\mathcal{A}_{j'}^{(s)}\right|\cdot L^{(s)}}\right) \quad .$$

If events $\mathcal{V}^{(r)}$ and $\mathcal{N}^{(r)}$ are both true, for the best Level-I arm j^*, we have

$$\begin{aligned}
&\widetilde{\theta}_{j^*}^{(s)} + 2\cdot 8\cdot\tfrac{1}{2^s} + 48\cdot\tfrac{1}{\epsilon\cdot 2^{2s}} + \tfrac{6\log(KT)}{\epsilon\cdot\left|\mathcal{A}_{j^*}^{(s)}\right|\cdot L^{(s)}} \\
= \;& \left(\widehat{\theta}_{j^*}^{(s)} + \tfrac{Z_{j^*}^{(s)}}{\left|\mathcal{A}_{j^*}^{(s)}\right|\cdot L^{(s)}}\right) + 2\cdot 8\cdot\tfrac{1}{2^s} + 48\cdot\tfrac{1}{\epsilon\cdot 2^{2s}} + \tfrac{6\log(KT)}{\epsilon\cdot\left|\mathcal{A}_{j^*}^{(s)}\right|\cdot L^{(s)}} \\
= \;& \left(\widehat{\theta}_{j^*}^{(s)} + 2\cdot 8\cdot\tfrac{1}{2^s} + 48\cdot\tfrac{1}{\epsilon\cdot 2^{2s}}\right) + \left(\tfrac{Z_{j^*}^{(s)}}{\left|\mathcal{A}_{j^*}^{(s)}\right|\cdot L^{(s)}} + \tfrac{6\log(KT)}{\epsilon\cdot\left|\mathcal{A}_{j^*}^{(s)}\right|\cdot L^{(s)}}\right) \\
> \;& \mu_{j^*,i^*} \quad .
\end{aligned} \tag{5.62}$$

The last inequality in (5.62) uses the fact that if event $\mathcal{V}^{(r)}$ is true, we have $\widehat{\theta}_{j^*}^{(s)} + 2\cdot 8\cdot\frac{1}{2^s} + 48\cdot\frac{1}{\epsilon\cdot 2^{2s}} > \mu_{j^*,i^*}$, and if event $\mathcal{N}^{(r)}$ is true, we have $Z_{j^*}^{(s)} + \frac{6\log(KT)}{\epsilon} > 0$.

Let $j^{(s)*} \in \arg\max_{j'\in\mathcal{A}^{(s)}} \left(\widetilde{\theta}_{j'}^{(s)} - 8\cdot\frac{1}{2^s} - \frac{6\log(KT)}{\epsilon\cdot\left|\mathcal{A}_{j'}^{(s)}\right|\cdot L^{(s)}}\right)$. If events $\mathcal{V}^{(r)}$ and $\mathcal{N}^{(r)}$ are both

true, simultaneously, we also have

$$\begin{aligned} & \widetilde{\theta}^{(s)}_{j^{(s)*}} - 8\cdot\tfrac{1}{2^s} - \frac{6\log(KT)}{\epsilon\cdot\left|\mathcal{A}^{(s)}_{j^{(s)*}}\right|\cdot L^{(s)}} \\ = & \left(\widehat{\theta}^{(s)}_{j^{(s)*}} - 8\cdot\tfrac{1}{2^s}\right) + \left(\frac{Z^{(s)}_{j^{(s)*}}}{\left|\mathcal{A}^{(s)}_{j^{(s)*}}\right|\cdot L^{(s)}} - \frac{6\log(KT)}{\epsilon\cdot\left|\mathcal{A}^{(s)}_{j^{(s)*}}\right|\cdot L^{(s)}}\right) \\ < & \mu_{j^{(s)*},i^*} \quad . \end{aligned} \tag{5.63}$$

The last inequality in (5.63) uses the fact that if event $\mathcal{V}^{(r)}$ is true, we have $\widehat{\theta}^{(s)}_{j^{(s)*}} - 8\cdot\frac{1}{2^s} < \mu_{j^{(s)*},i^*}$, and if event $\mathcal{N}^{(r)}$ is true, we have $Z^{(s)}_{j^{(s)*}} - \frac{6\log(KT)}{\epsilon} < 0$.

The arguments of (5.62) and (5.63) together imply that the best Level-I arm j^* has no chance to be eliminated in epoch $s \le r$, which yields a contradiction.

Step 2: As we have shown that j^* will not be eliminated, we now show that a sub-optimal Level-I arm with a mean reward gap Δ_j will not be in $\mathcal{A}^{(r+1)}$ when $r \ge \lambda_j$, and both events $\mathcal{V}^{(r)}$ and $\mathcal{N}^{(r)}$ are true.

If events $\mathcal{V}^{(r)}$ and $\mathcal{N}^{(r)}$ are both true, for a sub-optimal Level-I arm $j \in \mathcal{A}^{(r)}$, we have

$$\begin{aligned} & \widetilde{\theta}^{(r)}_j + 2\cdot 8\cdot\tfrac{1}{2^r} + 48\cdot\tfrac{1}{\epsilon\cdot 2^{2r}} + \frac{6\log(KT)}{\epsilon\cdot\left|\mathcal{A}^{(r)}_j\right|\cdot L^{(r)}} \\ = & \left(\widehat{\theta}^{(r)}_j + \frac{Z^{(r)}_j}{\epsilon\cdot\left|\mathcal{A}^{(r)}_j\right|\cdot L^{(r)}}\right) + 2\cdot 8\cdot\tfrac{1}{2^r} + 48\cdot\tfrac{1}{\epsilon\cdot 2^{2r}} + \frac{6\log(KT)}{\epsilon\cdot\left|\mathcal{A}^{(r)}_j\right|\cdot L^{(r)}} \\ = & \widehat{\theta}^{(r)}_j + 2\cdot 8\cdot\tfrac{1}{2^r} + 48\cdot\tfrac{1}{\epsilon\cdot 2^{2r}} + \left(\frac{Z^{(r)}_j}{\epsilon\cdot\left|\mathcal{A}^{(r)}_j\right|\cdot L^{(r)}} + \frac{6\log(KT)}{\epsilon\cdot\left|\mathcal{A}^{(r)}_j\right|\cdot L^{(r)}}\right) \\ < & \left(\mu_{j,i^*} + 8\cdot\tfrac{1}{2^r}\right) + 2\cdot 8\cdot\tfrac{1}{2^r} + 48\cdot\tfrac{1}{\epsilon\cdot 2^{2r}} + 2\cdot\frac{6\log(KT)}{\epsilon\cdot\left|\mathcal{A}^{(r)}_j\right|\cdot L^{(r)}} \\ < & \mu_{j,i^*} + 8\cdot\tfrac{1}{2^r} + 2\cdot 8\cdot\tfrac{1}{2^r} + 48\cdot\tfrac{1}{\epsilon\cdot 2^{2r}} + 2\cdot\frac{6\log(KT)}{\epsilon\cdot 2\log(KT)\cdot 2^{2r}} \\ < & \mu_{j,i^*} + 54\left(\tfrac{1}{2^r} + \tfrac{1}{\epsilon\cdot 2^{2r}}\right) \\ \le & \mu_{j,i^*} + \tfrac{1}{2}\cdot\Delta_j \quad . \end{aligned} \tag{5.64}$$

The first inequality in (5.64) uses the fact that if event $\mathcal{V}^{(r)}$ is true, we have $\widehat{\theta}^{(r)}_j < \mu_{j,i^*} + 8\cdot\frac{1}{2^r}$, and if event $\mathcal{N}^{(r)}$ is true, we have $Z^{(r)}_j < \frac{6\log(KT)}{\epsilon}$. The last inequality uses the fact that $\frac{1}{2^r} + \frac{1}{\epsilon\cdot 2^{2r}} \le \frac{1}{2^{\lambda_j}} + \frac{1}{\epsilon\cdot 2^{2\cdot\lambda_j}} \le \frac{\Delta_j}{108}$, when $r \ge \lambda_j$.

If events $\mathcal{V}^{(r)}$ and $\mathcal{N}^{(r)}$ are both true, simultaneously, for the best Level-I arm

j^*, we also have

$$\begin{aligned}
& \widetilde{\theta}_{j^*}^{(r)} - 8 \cdot \tfrac{1}{2^r} - \tfrac{6\log(KT)}{\epsilon \cdot \left|\mathcal{A}_{j^*}^{(r)}\right| \cdot L^{(r)}} \\
= & \left(\widehat{\theta}_{j^*}^{(r)} + \tfrac{Z_{j^*}^{(r)}}{\left|\mathcal{A}_{j^*}^{(r)}\right| \cdot L^{(r)}}\right) - 8 \cdot \tfrac{1}{2^r} - \tfrac{6\log(KT)}{\epsilon \cdot \left|\mathcal{A}_{j^*}^{(r)}\right| \cdot L^{(r)}} \\
> & \left(\mu_{j^*,i^*} - 2 \cdot 8 \cdot \tfrac{1}{2^r} - 48 \cdot \tfrac{1}{\epsilon \cdot 2^{2r}}\right) - 8 \cdot \tfrac{1}{2^r} - \tfrac{6\log(KT)}{\epsilon \cdot \left|\mathcal{A}_{j^*}^{(r)}\right| \cdot L^{(r)}} - \tfrac{6\log(KT)}{\epsilon \cdot \left|\mathcal{A}_{j^*}^{(r)}\right| \cdot L^{(r)}} \\
> & \mu_{j^*,i^*} - 2 \cdot 8 \cdot \tfrac{1}{2^r} - 48 \cdot \tfrac{1}{\epsilon \cdot 2^{2r}} - 8 \cdot \tfrac{1}{2^r} - \tfrac{6\log(KT)}{\epsilon \cdot 2\log(KT) \cdot 2^{2r}} - \tfrac{6\log(KT)}{\epsilon \cdot 2\log(KT) \cdot 2^{2r}} \\
= & \mu_{j^*,i^*} - 3 \cdot 8 \cdot \tfrac{1}{2^r} - 48 \cdot \tfrac{1}{\epsilon \cdot 2^{2r}} - 2 \cdot \tfrac{6\log(KT)}{\epsilon \cdot 2\log(KT) \cdot 2^{2r}} \\
> & \mu_{j^*,i^*} - 54\left(\tfrac{1}{2^r} + \tfrac{1}{\epsilon \cdot 2^{2r}}\right) \\
\geq & \mu_{j^*,i^*} - \tfrac{1}{2} \cdot \Delta_j \\
= & \mu_{j,i^*} + \tfrac{1}{2} \cdot \Delta_j \quad ,
\end{aligned} \tag{5.65}$$

which indicates that the sub-optimal Level-I arm $j \in \mathcal{A}^{(r)} \backslash \{j^*\}$ will be eliminated by the end of epoch r when $r \geq \lambda_j$. The first inequality in (5.65) uses the fact that if event $\mathcal{V}^{(r)}$ is true, we have $\widehat{\theta}_{j^*}^{(r)} > \mu_{j^*,i^*} - 2 \cdot 8 \cdot \frac{1}{2^r} - 48 \cdot \frac{1}{\epsilon \cdot 2^{2r}}$, and if event $\mathcal{N}^{(r)}$ is true, we have $Z_{j^*}^{(r)} > -\frac{6\log(KT)}{\epsilon}$. □

Proof of Lemma 25: All T rounds can be divided into at most $\log(T)$ epochs. Let $\tau_r + 1$ be the first round of epoch r, i.e., all rounds $t \in \{\tau_r + 1, \tau_r + 2, \ldots, \tau_{r+1}\}$ are in epoch r. Note that each active Level-II arm will be pulled exactly $2\log(KT) \cdot 2^{2r}$ times in epoch r. We set $\tau_1 = 0$. We have

$$
\begin{aligned}
& \text{LHS in (5.51)} \\
= & \sum_{t=1}^{T} \mathbb{E}\left[\mathbf{1}\left\{(J_t, I_t) = (j, i^*)\right\}\right] \cdot \Delta_j \\
\leq & \sum_{r=1}^{\log(T)} \mathbb{E}\left[\sum_{t=\tau_r+1}^{\tau_{r+1}} \mathbf{1}\left\{(J_t, I_t) = (j, i^*)\right\}\right] \cdot \Delta_j \\
\leq & \sum_{r=1}^{\lceil \lambda_j \rceil} \mathbb{E}\left[\sum_{t=\tau_r+1}^{\tau_{r+1}} \mathbf{1}\left\{(J_t, I_t) = (j, i^*)\right\}\right] \cdot \Delta_j \\
+ & \sum_{r=\lceil \lambda_j \rceil+1}^{\log(T)} \mathbb{E}\left[\sum_{t=\tau_r+1}^{\tau_{r+1}} \mathbf{1}\left\{(J_t, I_t) = (j, i^*)\right\}\right] \cdot \Delta_j \\
\leq & \sum_{r=1}^{\lceil \lambda_j \rceil} 2\log(KT) \cdot 2^{2r} \cdot \Delta_j + \sum_{r=\lceil \lambda_j \rceil+1}^{\log(T)} \mathbb{E}\left[\sum_{t=\tau_r+1}^{\tau_{r+1}} \mathbf{1}\left\{(J_t, I_t) = (j, i^*)\right\}\right] \cdot \Delta_j \\
\leq & O\left(\max\left\{\frac{\log(KT)}{\Delta_j}, \frac{\log(KT)}{\epsilon}\right\}\right) + \underbrace{\sum_{r=\lceil \lambda_j \rceil+1}^{\log(T)} \mathbb{E}\left[\sum_{t=\tau_r+1}^{\tau_{r+1}} \mathbf{1}\left\{(J_t, I_t) = (j, i^*)\right\}\right] \cdot \Delta_j}_{(\zeta)} \quad .
\end{aligned}
\tag{5.66}
$$

Then we decompose term (ζ) in (5.66) based on events $\mathcal{V}^{(r-1)}, \mathcal{N}^{(r-1)}$, and their complementary events. We have

$$
\begin{aligned}
(\zeta) = & \sum_{r=\lceil \lambda_j \rceil+1}^{\log(T)} \mathbb{E}\left[\sum_{t=\tau_r+1}^{\tau_{r+1}} \mathbf{1}\left\{(J_t, I_t) = (j, i^*)\right\}\right] \cdot \Delta_j \\
\leq & \sum_{r=\lceil \lambda_j \rceil+1}^{\log(T)} \mathbb{E}\left[\sum_{t=\tau_r+1}^{\tau_{r+1}} \underbrace{\mathbf{1}\left\{(J_t, I_t) = (j, i^*), \mathcal{V}^{(r-1)}, \mathcal{N}^{(r-1)}\right\}}_{=0, \text{ Lemma 24}}\right] \cdot \Delta_j \\
+ & \underbrace{\sum_{r=\lceil \lambda_j \rceil+1}^{\log(T)} \mathbb{E}\left[\sum_{t=\tau_r+1}^{\tau_{r+1}} \mathbf{1}\left\{\overline{\mathcal{V}^{(r-1)}}\right\}\right] \cdot \Delta_j}_{\leq O\left(\frac{1}{K}\right), \text{ Lemma 22}} + \underbrace{\sum_{r=\lceil \lambda_j \rceil+1}^{\log(T)} \mathbb{E}\left[\sum_{t=\tau_r+1}^{\tau_{r+1}} \mathbf{1}\left\{\overline{\mathcal{N}^{(r-1)}}\right\}\right] \cdot \Delta_j}_{\leq O\left(\frac{1}{K}\right), \text{ Lemma 23}} \\
\leq & O\left(\frac{1}{K}\right) \quad .
\end{aligned}
\tag{5.67}
$$

By plugging the upper bound of (ζ) into (5.66), we conclude the proof. □

Proof of Lemma 26: All T rounds can be divided into at most $\log(T)$ epochs. Let $\tau_r + 1$ be the first round when epoch r starts. This implies that all rounds $t \in \{\tau_r + 1, \tau_r + 2, \ldots, \tau_{r+1}\}$ are in epoch r and each active Level-II arm will be pulled

exactly $2\log(KT)\cdot 2^{2r}$ times. We set $\tau_1 = 0$. We have

$$\begin{aligned}
\text{LHS in (5.52)} &= \sum_{t=1}^{T} \mathbb{E}\left[\mathbf{1}\left\{(J_t, I_t) = (j,i)\right\}\right] \cdot \left(\Delta_i^{(j)} + \Delta_j\right) \\
&\leq \sum_{r=1}^{\log(T)} \mathbb{E}\left[\sum_{t=\tau_r+1}^{\tau_{r+1}} \mathbf{1}\left\{(J_t, I_t) = (j,i)\right\}\right] \cdot \left(\Delta_i^{(j)} + \Delta_j\right) \\
&\leq \underbrace{\sum_{r=1}^{\log(T)} \mathbb{E}\left[\sum_{t=\tau_r+1}^{\tau_{r+1}} \mathbf{1}\left\{(J_t, I_t) = (j,i)\right\}\right] \cdot 2 \cdot \max\left\{\Delta_i^{(j)}, \Delta_j\right\}}_{(\eta)} \quad .
\end{aligned} \tag{5.68}$$

We decompose term (η) based on events $\mathcal{E}_j^{(r-1)}$, $\mathcal{N}_j^{(r-1)}$, $\mathcal{V}^{(r-1)}$, and $\mathcal{N}^{(r-1)}$ and their complementary events. We have

$$\begin{aligned}
(\eta) \leq\ & \omega \\
&+ \underbrace{\sum_{r=1}^{\log(T)} \mathbb{E}\left[\sum_{t=\tau_r+1}^{\tau_{r+1}} \mathbf{1}\left\{(J_t, I_t) = (j,i), \overline{\mathcal{N}_j^{(r-1)}}\right\}\right] \cdot 2 \cdot \max\left\{\Delta_i^{(j)}, \Delta_j\right\}}_{\leq O\left(\frac{1}{K}\right), \text{ Lemma 19}} \\
&+ \underbrace{\sum_{r=1}^{\log(T)} \mathbb{E}\left[\sum_{t=\tau_r+1}^{\tau_{r+1}} \mathbf{1}\left\{(J_t, I_t) = (j,i), \overline{\mathcal{E}_j^{(r-1)}}\right\}\right] \cdot 2 \cdot \max\left\{\Delta_i^{(j)}, \Delta_j\right\}}_{\leq O\left(\frac{1}{K}\right), \text{ Lemma 12}} \\
&+ \underbrace{\sum_{r=1}^{\log(T)} \mathbb{E}\left[\sum_{t=\tau_r+1}^{\tau_{r+1}} \mathbf{1}\left\{(J_t, I_t) = (j,i), \overline{\mathcal{V}^{(r-1)}}\right\}\right] \cdot 2 \max\left\{\Delta_i^{(j)}, \Delta_j\right\}}_{\leq O\left(\frac{1}{K}\right), \text{ Lemma 22}} \\
&+ \underbrace{\sum_{r=1}^{\log(T)} \mathbb{E}\left[\sum_{t=\tau_r+1}^{\tau_{r+1}} \mathbf{1}\left\{(J_t, I_t) = (j,i), \overline{\mathcal{N}^{(r-1)}}\right\}\right] \cdot 2 \cdot \max\left\{\Delta_i^{(j)}, \Delta_j\right\}}_{\leq O\left(\frac{1}{K}\right), \text{ Lemma 23}} \\
\leq\ & \omega + O\left(\frac{1}{K}\right) \quad ,
\end{aligned} \tag{5.69}$$

where $\omega :=$

$$2\sum_{r=1}^{\log(T)} \mathbb{E}\left[\sum_{t=\tau_r+1}^{\tau_{r+1}} \mathbf{1}\left\{(J_t, I_t) = (j,i), \mathcal{E}_j^{(r-1)}, \mathcal{N}_j^{(r-1)}, \mathcal{V}^{(r-1)}, \mathcal{N}^{(r-1)}\right\}\right] \max\left\{\Delta_i^{(j)}, \Delta_j\right\}.$$

We consider two cases to upper bound term (ω) based on whether $\Delta_i^{(j)} \leq \Delta_j$ or

$\Delta_i^{(j)} > \Delta_j$.

Case I: if $\Delta_j = \max\left\{\Delta_i^{(j)}, \Delta_j\right\}$, we have

$$
\begin{aligned}
(\omega) &= \sum_{r=1}^{\log(T)} \mathbb{E}\left[\sum_{t=\tau_r+1}^{\tau_{r+1}} \mathbf{1}\left\{(J_t, I_t) = (j,i), \mathcal{E}_j^{(r-1)}, \mathcal{N}_j^{(r-1)}, \mathcal{V}^{(r-1)}, \mathcal{N}^{(r-1)}\right\}\right] \cdot 2 \cdot \Delta_j \\
&\leq \sum_{r=1}^{\log(T)} \mathbb{E}\left[\sum_{t=\tau_r+1}^{\tau_{r+1}} \mathbf{1}\left\{(J_t, I_t) = (j,i), \mathcal{V}^{(r-1)}, \mathcal{N}^{(r-1)}\right\}\right] \cdot 2 \cdot \Delta_j \\
&= \sum_{r=1}^{\lceil \lambda_j \rceil} \mathbb{E}\left[\sum_{t=\tau_r+1}^{\tau_{r+1}} \mathbf{1}\left\{(J_t, I_t) = (j,i), \mathcal{V}^{(r-1)}, \mathcal{N}^{(r-1)}\right\}\right] \cdot 2 \cdot \Delta_j \\
&+ \sum_{r=\lceil \lambda_j \rceil + 1}^{\log(T)} \mathbb{E}\left[\sum_{t=\tau_r+1}^{\tau_{r+1}} \underbrace{\mathbf{1}\left\{(J_t, I_t) = (j,i), \mathcal{V}^{(r-1)}, \mathcal{N}^{(r-1)}\right\}}_{=0, \text{ Lemma 24}}\right] \cdot 2 \cdot \Delta_j \\
&\leq \sum_{r=1}^{\lceil \lambda_j \rceil} 2\log(KT) \cdot 2^{2r} \cdot 2 \cdot \Delta_j + 0 \\
&= O\left(\max\left\{\frac{\log(KT)}{\Delta_j}, \frac{\log(KT)}{\epsilon}\right\}\right) \\
&= O\left(\max\left\{\frac{\log(KT)}{\max\left\{\Delta_j, \Delta_i^{(j)}\right\}}, \frac{\log(KT)}{\epsilon}\right\}\right).
\end{aligned}
\tag{5.70}
$$

Case II: if $\Delta_i^{(j)} = \max\left\{\Delta_j\right\}, \Delta_i^{(j)}$, we have

$$
\begin{aligned}
(\omega) &= \sum_{r=1}^{\log(T)} \mathbb{E}\left[\sum_{t=\tau_r+1}^{\tau_{r+1}} \mathbf{1}\left\{(J_t, I_t) = (j,i), \mathcal{E}_j^{(r-1)}, \mathcal{N}_j^{(r-1)}, \mathcal{V}^{(r-1)}, \mathcal{N}^{(r-1)}\right\}\right] \cdot 2 \cdot \Delta_i^{(j)} \\
&\leq \sum_{r=1}^{\log(T)} \mathbb{E}\left[\sum_{t=\tau_r+1}^{\tau_{r+1}} \mathbf{1}\left\{(J_t, I_t) = (j,i), \mathcal{E}_j^{(r-1)}, \mathcal{N}_j^{(r-1)}, \right\}\right] \cdot 2 \cdot \Delta_i^{(j)} \\
&\leq \sum_{r=1}^{\lceil \gamma_{j,i} \rceil} \mathbb{E}\left[\sum_{t=\tau_r+1}^{\tau_{r+1}} \mathbf{1}\left\{(J_t, I_t) = (j,i)\right\}\right] \cdot 2 \cdot \Delta_i^{(j)} \\
&+ \sum_{r=\lceil \gamma_{j,i} \rceil + 1}^{\log(T)} \mathbb{E}\left[\sum_{t=\tau_r+1}^{\tau_{r+1}} \underbrace{\mathbf{1}\left\{(J_t, I_t) = (j,i), \mathcal{E}_j^{(r-1)}, \mathcal{N}_j^{(r-1)}\right\}}_{=0, \text{ Lemma 21}}\right] \cdot 2 \cdot \Delta_i^{(j)} \\
&\leq \sum_{r=1}^{\lceil \gamma_{j,i} \rceil} 2\log(KT) \cdot 2^{2r} \cdot 2 \cdot \Delta_i^{(j)} \\
&= O\left(\max\left\{\frac{\log(KT)}{\Delta_i^{(j)}}, \frac{\log(KT)}{\epsilon}\right\}\right) \\
&= O\left(\max\left\{\frac{\log(KT)}{\max\left\{\Delta_j, \Delta_i^{(j)}\right\}}, \frac{\log(KT)}{\epsilon}\right\}\right).
\end{aligned}
\tag{5.71}
$$

□

5.7.4 Additional experimental results

Table 5.1: Mean reward setting with 5 Level-II arms

	Level-II arm 1	Level-II arms 2 to 3	Level-II arms 4 to 5
Level-I arm 1	0.90	0.65	0.40
Level-I arm 2	0.65	0.40	0.15
Level-I arm 3	0.65	0.40	0.15
Level-I arm 4	0.65	0.40	0.15

Table 5.2: Mean reward setting with 9 Level-II arms

	Level-II arm 1	Level-II arms 2 to 5	Level-II arms 6 to 9
Level-I arm 1	0.90	0.65	0.40
Level-I arm 2	0.65	0.40	0.15
Level-I arm 3	0.65	0.40	0.15
Level-I arm 4	0.65	0.40	0.15

Table 5.3: Mean reward setting with 13 Level-II arms

	Level-II arm 1	Level-II arms 2 to 7	Level-II arms 8 to 13
Level-I arm 1	0.90	0.65	0.40
Level-I arm 2	0.65	0.40	0.15
Level-I arm 3	0.65	0.40	0.15
Level-I arm 4	0.65	0.40	0.15

We consider a setting where the number of Level-I arms is $m = 4$ and the number of Level-II arms associated with each Level-I arm is $k = 5, 9, 13$. The mean rewards with different number of Level-II arms are shown in Table 5.1,Table 5.2, and Table 5.3. Note that in all these three mean reward settings, we have $\Delta_j = 0.25$. For half of the Level-II arms, we set $\Delta_i^{(j)} = 0.25$ and for the remaining half, we set $\Delta_i^{(j)} = 0.5$.

The first experiment is to see the performance comparison when we set a very small ϵ, e.g., $\epsilon = 0.1$. Figure 5.5 shows the experimental results when we set $\epsilon = 0.1$. Note that in this setting, the value of ϵ is smaller than Δ_j and $\Delta_i^{(j)}$. The second experiment is the case where we set $\epsilon = 0.25$. In this setting, almost half of the Level-II arms have mean reward gaps $\Delta_i^{(j)}$ that are equal to ϵ while for the remaining half, their mean reward gaps $\Delta_i^{(j)}$ are greater than ϵ. Figure 5.6 shows the experimental results when $\epsilon = 0.25$. The third experiment is the case where we set $\epsilon = 0.5$. In this setting, almost half of the Level-II arms have mean reward gaps $\Delta_i^{(j)}$ that are equal to ϵ while for the remaining half, their mean reward gaps $\Delta_i^{(j)}$ are smaller than ϵ. Figure 5.7 shows the experimental results when $\epsilon = 0.5$. The fourth experiment is to see the performance comparison when we set a very large ϵ, e.g., $\epsilon = 1$. Figure 5.8 shows the experimental results when $\epsilon = 1$. In all these experiments, we can see that the regret grows linearly as the number of Level-II arms increases.

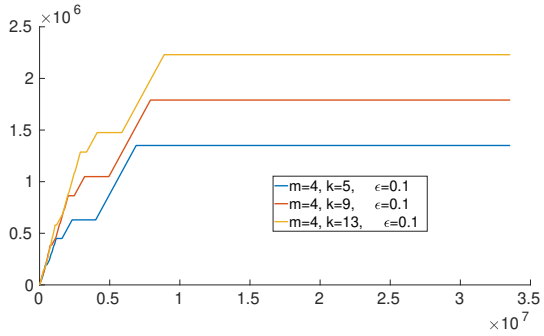

Figure 5.5: $\epsilon = 0.1$ and the impact of k: $k = 5, 9, 13$

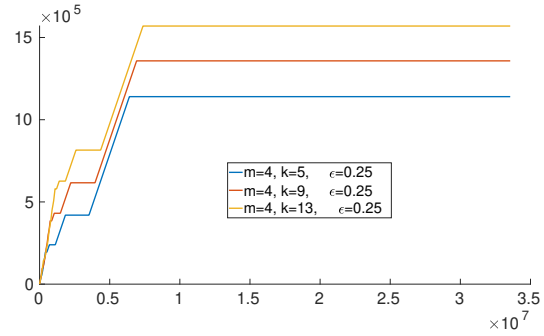

Figure 5.6: $\epsilon = 0.25$ and the impact of k: $k = 5, 9, 13$

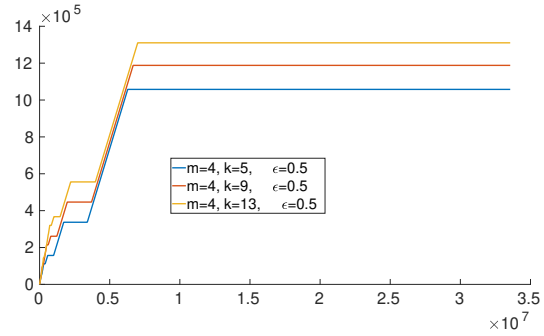

Figure 5.7: $\epsilon = 0.5$ and the impact of k: $k = 5, 9, 13$

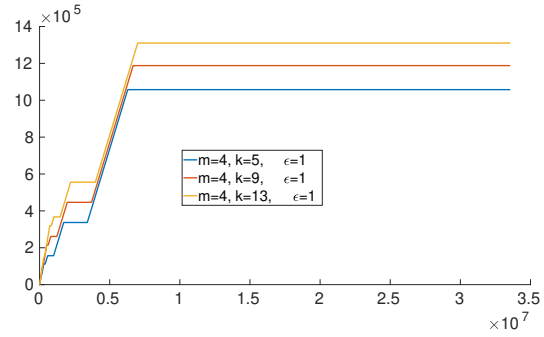

Figure 5.8: $\epsilon = 1.0$ and the impact of k: $k = 5, 9, 13$

Chapter 6

Differentially Private Graphical Bandits

This chapter addresses the differentially private version of the learning problem that has been presented in Section 3.2, i.e., the differentially private stochastic multi-armed bandit problem with undirected feedback graphs. In this chapter, we also provide our own understanding of how to devise efficient differentially private online learning algorithms in a stochastic environment.

6.1 Introduction

In Section 3.1, we have presented why we are interested in investigating learning algorithms for graphical bandits and in Section 4.1, we have presented why differentially private online learning such as private bandits and private full information setting are attracting consistent attention. In this chapter, we will investigate a learning problem that is broader than the private bandit setting and the full information setting, which is *differentially private graphical bandits*. Note that differentially private graphical bandits can also be viewed as a private version of the learning problem that has been presented in Chapter 3. So far, to the best of our knowledge, there is no prior work investigating this learning problem either under stochastic rewards or adversarial rewards.

The key difficulty in having a good private algorithm for graphical bandits is the control of the l_1-sensitivity of the algorithm that computes the empirical means and meanwhile maintaining a good exploration-vs-exploitation balance. For both

the private bandit setting and the full information setting, we do not have this issue due to the fact that in each round, either 1 observation or K observations is revealed, i.e., a fixed amount of observations are obtained in each round. However, in graphical bandits, the number of observations revealed can be any number between 1 and K, depending on the pulled arm and the feedback model. In Section 6.3, we will discuss in detail why the changing number of observations make it non-trivial to design efficient learning algorithms for differentially private graphical bandits.

In this chapter, we present a novel UCB-based algorithm, DP-UCB-N, for differentially private graphical bandits with undirected feedback graphs. Our algorithm relies on a specific clique covering $\mathcal{C}$ as input, and obtains a regret bound which is linear in the size of the given clique covering $\mathcal{C}$ up to logarithmic factors both for the leading and constant terms, i.e., an $O\left(\sum_{1\leq i\leq|\mathcal{C}|:\Delta_i^{\min}>0}\frac{\log(|C_i|\cdot T)}{\min\left\{\Delta_i^{\min},\,\epsilon\right\}}\right)$ regret bound, where $\Delta_i^{\min}$ indicates the minimum mean reward gap among all suboptimal arms within a clique $C_i \in \mathcal{C}$. The high-level idea behind DP-UCB-N is to let Learner run Anytime-time-UCB over a set of cliques instead of over arms directly, and each clique runs a modified version of RNM that has been presented in Section 2.5 and Subsection 4.5.1. By using this combination, the goals of controlling the sensitivity and balancing exploration-vs-exploitation can be achieved simultaneously.

6.2 Learning Problem

In this section, we first recap the learning problem of stochastic graphical bandits and then we present the definition of differentially private graphical bandits.

6.2.1 Stochastic Graphical Bandits

We consider a stochastic multi-armed bandit problem with an undirected feedback graph. In this game, we have a fixed arm set $\mathcal{A}$ with size K, a stochastic environment, and an undirected graph $G = (\mathcal{A}, \mathcal{E})$ representing all the feedback relationships among all arms in $\mathcal{A}$. At the beginning of round t, the environment generates random rewards $X_j(t) \in [0,1]$ for all $j \in \mathcal{A}$ independently from fixed but unknown distributions. Simultaneously, Learner pulls an arm $J_t \in \mathcal{A}$.

Graph $G = (\mathcal{A}, \mathcal{E})$ denotes the feedback relationships over arm set $\mathcal{A}$. An edge $\{i, j\} \in \mathcal{E}$ means that Learner can get an observation of arm j when pulling arm i, and vice versa. Note that pulling arm i always lets Learner observe the reward of arm i itself, i.e., G includes self-loops. We assume that graph G does not vary over time. For an arm $j \in \mathcal{A}$, let set $\mathcal{N}_j$ collect all arm j's neighbouring nodes in G including arm j itself. At the end of round t, Learner obtains a reward $X_{J_t}(t)$ and observes the reward of each arm in $\mathcal{N}_{J_t}$. The goal of Learner is to pull arms sequentially to accumulate as much reward as possible over T rounds.

Let $\mu_j := \mathbb{E}\left[X_j(t)\right]$ be the mean reward of arm j. We assume that the first arm is the unique best arm, i.e., $\mu_1 > \mu_j, \forall j \neq 1$. Let $\Delta_j := \mu_1 - \mu_j$ be the mean reward gap between arm j and the best arm. To measure the quality of our learning algorithms, we use the (pseudo-)regret $\mathcal{R}(T)$, which is defined as

$$\mathcal{R}(T) = \mathbb{E}\left[\sum_{t=1}^{T} \mu_1 - \mu_{J_t}\right] \quad . \tag{6.1}$$

6.2.2 Differential Privacy

Different from the definitions that have been shown in Definitions 4 and 5, for differentially private graphical bandits, the feedback graph also impacts the decisions made by Learner. Therefore, we present a generalized version of Definition 4 for fitting differentially private graphical bandits. Let us say $X_{1:T}$ and $X'_{1:T}$ are neighbouring reward sequences such that they differ in at most one reward vector.

Definition 6 (Differential Privacy)**.** *An algorithm Π is ϵ-differentially private if for any two neighbouring reward sequences $X_{1:T}$ and $X'_{1:T}$, for any set $\mathcal{D}$ of decisions made from round 1 to T, for any feedback graph G, it holds that*

$$\mathbb{P}\left\{\Pi(X_{1:T}, G) \in \mathcal{D}\right\} \leq e^{\epsilon} \cdot \mathbb{P}\left\{\Pi(X'_{1:T}, G) \in \mathcal{D}\right\} \quad . \tag{6.2}$$

Several remarks are in order for Definition 6. If G only contains isolated nodes, i.e., $\mathcal{E} = \{\{i, i\} : i \in \mathcal{A}\}$, then the differentially private graphical bandits setting boils down to the differentially private bandit setting and Section 4.4 has presented an optimal UCB-based algorithm, Anytime-Lazy-UCB (Algorithm 8), for this special case. If G is a complete graph, i.e, $\mathcal{E} = \{\{i, j\} : i, j \in \mathcal{A}\}$, then the differentially

private graphical bandit setting boils down to the differentially private full information setting and Section 4.5 has presented a good learning algorithm, FTNL (Algorithm 10), for this special case.

6.3 Discussion

So far, to the best of our knowledge, there does not exist any work on investigating the learning problems of differentially private graphical bandits either with stochastic rewards or adversarial rewards.

The challenge to design a good differentially private learning algorithm for stochastic graphical bandits is brought by the changing number of observations revealed in each round. The varying number of observations revealed over time makes it difficult to decide the needed noise level to have an ϵ-differentially private learning algorithm.

In the differentially private online learning, particularly, the settings where both exploration and exploitation exist, we need to maintain a proper trade-off from two perspectives. The first trade-off is the one between exploration and exploitation (accumulating reward vs gaining information). The second trade-off is the one between a privacy guarantee and a regret guarantee. Intuitively, the more information revealed, the harder to satisfy the privacy requirement. Suppose the following extreme case. If we do not disclose any information, then there is no privacy concern. However, Learner has no chance to maintain the exploration-vs-exploitation trade-off. Consequentially, it is not surprising that Learner will suffer a regret that is linear in T.

Recall the two differentially private online learning settings that have been discussed in Chapter 4, the private bandit setting and the private full information setting. They both have the property that the number of observations obtained in each round t is fixed. This property plays an important role to decide the needed noise to have ϵ-differentially private learning algorithms.

In the differentially private stochastic bandit setting, by using the ideas of laziness and forgetfulness, since only one reward is revealed, the l_1-sensitivity for the composed algorithm that computes the empirical mean of all arms is 1. Note that the single change of any reward vector can only impact the empirical mean of one arm once. From Definition 3, we know that we only need to inject a noise drawn

from Lap $\left(\frac{1}{\epsilon}\right)$ to each arm's observations.

In the differentially private full information setting with stochastic rewards, recall that exploration is not needed as the reward vector (K observations) can be seen in each round. By using Report Noisy Max (shown in Section 2.5), we also only need to inject a noise variable drawn from Lap $\left(\frac{1}{\epsilon}\right)$. Although K observations are revealed, the presented algorithm for full information game, Algorithm 10, still has a regret bound that is logarithmic in K for the term involving privacy parameter ϵ. We can say that in these two cases, we inject the minimum amount of noise.

However, in graphical bandits setting, the number of observations revealed at the end of each round is changing over time. It can be any value between 1 and K, depending on the out-degree of the pulled arm (the number of outgoing edges of the pulled arm). Take the undirected feedback graph shown in Figure 6.1 for example. In round s, if arm 4 is pulled, i.e., $J_s = 4$, then Learner obtains observations from arms $\{4, 5, 7\}$. In round s', if arm 6 is pulled, i.e., $J_{s'} = 6$, then Learner obtains observations from arms $\{6, 1, 2, 7\}$. Note that Learner can obtain an observation of arm 7 if an arm from $\{3, 4, 5, 6, 7\}$ is pulled. Therefore, **how can we decide the amount of noise needed for the observations of arm 7?**

We can always use a safe but naive strategy which is injecting a noise drawn from Lap $\left(\frac{K}{\epsilon}\right)$ to each arm's obtained observations. Note that this naive strategy does not leverage the feedback graph to decide the noise level at all. Then, from the standard composition theorem that ϵ's can be added up (Theorem 7), we know that the learning algorithm that computes the empirical means is ϵ-differentially private. It is not surprising to see that this naive strategy will result in a very sub-optimal regret bound, as the injected noise is much more than Lap $\left(\frac{1}{\epsilon}\right)$. A better strategy is to inject a noise drawn from Lap $\left(\frac{m_j}{\epsilon}\right)$, where $m_j := \max_{i \in \mathcal{N}_j} |\mathcal{N}_i|$ is the maximum degree among all the neighbouring nodes of arm j. Intuitively, this strategy will have a better regret bound than the naive strategy, as $m_j \leq K$. However, the injected noise is still more than Lap $\left(\frac{1}{\epsilon}\right)$.

In order to have a differentially private learning algorithm with a good theoretical guarantee, ideally, we only want to inject a noise drawn from Lap $\left(\frac{1}{\epsilon}\right)$ up to a universal constant. Recall that for the regret bounds of private bandit setting and full information setting, the terms involving ϵ are $O\left(\frac{K \log(T)}{\epsilon}\right)$ (Theorem 14) and $O\left(\frac{\log(K)}{\epsilon}\right)$ (Theorem 19), respectively. Motivated by the regret bound of UCB-NE

that has been shown in Section 3.4, we would like the regret bound for the privacy term takes the form of $O\left(\sum_{C\in\mathcal{C}} \frac{\log(|C|\cdot T)}{\epsilon}\right)$, which does not depend on K.

Since graphical bandits lie in between the bandit setting and the full information setting, a natural question is: **is it possible to use the techniques that have been used to design the optimal differentially private bandit learning algorithm and full information algorithm to devise a good learning algorithm for differentially private graphical bandits?**

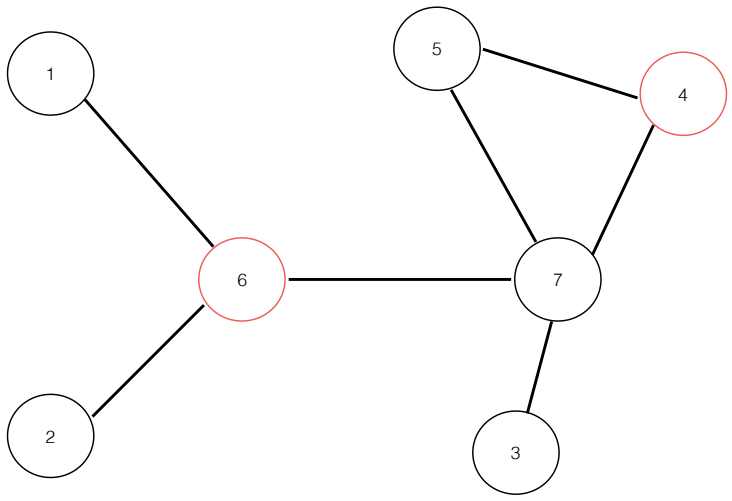

Figure 6.1: Undirected Feedback Graph

6.4 Differentially Private Algorithm

By leveraging the ideas shown in Sections 4.4 and 4.5, we can devise a good learning algorithm for differentially private graphical bandits for which the term involving ϵ in the regret bound can be much better than $O(K)$ for some feedback graphs.

In this section, we first present a differentially private learning algorithm for graphical bandits: DP-UCB-N. Then, we present privacy and regret analysis for DP-UCB-N. It is important to note that DP-UCB-N relies on a specific clique covering $\mathcal{C}$ as input.

6.4.1 DP-UCB-N

The high-level idea behind DP-UCB-N is to let Learner run private UCB (Anytime-Lazy-UCB) over a set of cliques instead of over all arms. To help Learner achieve the tradeoff between exploration and exploitation, we use the "information minimization principle" [16], by letting each clique only disclose the highest differentially private empirical mean and its arm ID. For each clique, it runs a modified version of RNM that has been introduced in Section 2.5. More specifically, each clique returns a tuple with two attributes. The first attribute is the arm ID with the highest differentially private empirical mean. The second attribute is the highest differentially private empirical mean.

Note that in graphical bandits, the exploration-vs-exploitation trade-off is still needed to be maintained for most of the cases. We can view how to achieve this trade-off from two perspectives. Returning the highest differentially private empirical mean helps Learner to construct the upper confidence bound, to achieve the exploration-vs-exploitation trade-off at a clique-level. Returning the arm ID with the highest differentially private empirical mean is to help Learner to do pure exploitation within a clique. Note that the information of all the remaining arms within the clique other than the disclosed arm are hidden to Learner.

Before presenting the learning algorithm, let us have some notations and definitions first. Let $\mathcal{C}$ be the input clique covering of graph G which is a set of cliques that cover all nodes in G. Let $C_i \in \mathcal{C}$ be a clique. We choose a clique covering $\mathcal{C}$ such that there is no overlapping nodes among cliques. Let $\beta(G) = |\mathcal{C}|$ be the size of the clique covering $\mathcal{C}$, i.e., the number of cliques covering all nodes in graph G.

As we will still use the ideas of forgetfulness and laziness that have been presented in Section 4.4.1, let $O_j(t-1)$ be the *effective number of observations* by the end of round $t-1$ of arm j. The effective number of observations is not the same as the total number of observations for a specific arm j. The effective number of observations is the amount of observations that have been used to compute the differentially private empirical mean $\widetilde{\mu}_{j,O_j(t-1)} := \widehat{\mu}_{j,O_j(t-1)} + \frac{Z_j}{O_j(t-1)}$, where $Z_j \sim \text{Lap}\left(\frac{1}{0.5\epsilon}\right)$. That is also to say, the number of effective observations doubles every time when computing the differentially private empirical mean.

Let $i^*(t-1) \leftarrow \arg\max_{j \in C_i} \widetilde{\mu}_{j,O_j(t-1)}$ be the arm within clique C_i that has the highest differentially private empirical mean by the end of round $t-1$. It is important to note that, just as have mentioned in Section 2.5, Report Noisy max (RNM)

must take fresh observations as input, i.e., reusing observations is not allowed. Let $\left(i^*(t-1), \widetilde{\mu}_{i^*(t-1),O_{i^*(t-1)}(t-1)}\right)$ indicates a tuple reported from clique C_i by the end of round $t-1$.

Algorithm 14 presents the learning algorithm in detail. For the initialization, for each clique, we choose one arm and pull it once to have an observation for all the arms within the clique. Then, we inject a noise drawn from Lap $\left(\frac{1}{0.5\epsilon}\right)$ to each arm's observation. Each clique reports a tuple $(i^*, \widetilde{\mu}_{i^*,1})$ with the first attribute indicating the arm within clique C_i that has the highest differentially private empirical mean and the second attribute in the tuple indicating the highest differentially private empirical mean itself. Note that in Section 2.5, we have presented the ideas about why we need to inject a noise drawn from Lap $\left(\frac{1}{0.5\epsilon}\right)$.

After the initialization phase, we have tuples $(i^*, \widetilde{\mu}_{i^*,1})$ returned by each clique. For rounds $t = \beta(G)+1, \beta(G)+2, \ldots,$ Learner first constructs the upper confidence bounds based on all the reported differentially private empirical means from each clique. That is also to say, for clique C_i, Learner constructs the upper confidence bound $\overline{\mu}_i(t)$ as

$$\overline{\mu}_i(t) := \widetilde{\mu}_{i^*(t-1),O_{i^*(t-1)}(t-1)} + \sqrt{\frac{3\log\left(|C_i|\cdot t\right)}{O_{i^*(t-1)}(t-1)}} + \frac{6\log\left(|C_i|\cdot t\right)}{\epsilon\cdot O_{i^*(t-1)}(t-1)} \quad . \tag{6.3}$$

Learner first locates the clique C_{I_t} with the highest upper confidence bound, i.e., $C_{I_t} \in \arg\max_{C_i\in\mathcal{C}} \overline{\mu}_i(t)$. Then, Learner pulls the arm that has been reported by the located clique, i.e., it pulls arm $J_t \leftarrow I_t^*(t-1)$.

It is important to note that for each clique, only the highest differentially private empirical mean can be seen by Learner. The remaining differentially private empirical means are still hidden. This satisfies the principle of information minimization. After pulling arm $J_t \leftarrow I_t^*(t-1)$, Learner gets an observation for all arms in clique C_{I_t}.

6.4.2 Privacy and Regret Guarantees

In this subsection, we present privacy and regret guarantees for DP-UCB-N.

Theorem 23. *Algorithm 14 preserves ϵ-differential privacy.*

Proof sketch of Theorem 23: Let us say that the reward vector in round t are different between $X_{1:T}$ and $X'_{1:T}$. Fix a sequence of noise variables. It is not hard to see that

Algorithm 14 DP-UCB-N

Input: An arm set $\mathcal{A}$, undirected feedback graph G, a clique covering $\mathcal{C}$, and privacy parameter ϵ ;
Initialization: For each clique $C_i \in \mathcal{C}$, do the following:
  Select one arm and pull it once ;
  Set $O_j \leftarrow 1$ for all $j \in C_i$; % the effective number of observations
  Inject a noise drawn from Lap $\left(\frac{1}{0.5\epsilon}\right)$ to each arm j's observation ;
  Report a tuple $\left(i^*, \widetilde{\mu}_{i^*,O_{i^*}}\right)$;
  Set $r_i \leftarrow 0$; % epoch index at clique-level
  Set $N_j \leftarrow 1$ for all $j \in C_i$; % global counter
for $t = |\mathcal{C}|+1, |\mathcal{C}|+2, \ldots$ **do**
  Learner locates the index of the clique with the highest upper confidence bound based on (6.3), i.e., setting $I_t \leftarrow \arg\max_{1\le i\le|\mathcal{C}|} \overline{\mu}_i(t)$;
  Learner pulls the arm reported by clique C_{I_t}, the arm with the highest differentially private empirical mean within clique C_{I_t}, i.e., pulling $J_t \leftarrow I_t^*$;
  for $j \in C_{I_t}$ **do**
    Set $N_j \leftarrow N_j + 1$; % increment global counter
    if $N_j = \sum_{r=0}^{r_{I_t}+1} 2^r$ **then**
      $O_j \leftarrow 2^{r_{I_t}+1}$; % Update the number of effective observations
    end if
  end for
  if $O_{J_t} = 2^{r_{I_t}+1}$ **then**
    for $j \in C_{I_t}$ **do**
      Inject a fresh noise variable drawn from Lap $\left(\frac{1}{0.5\epsilon}\right)$ to all these O_j observations ;
      Update the differentially private empirical mean $\widetilde{\mu}_{j,O_j}$;
    end for
    Clique C_{I_t} reports a tuple $\left(I_t^*, \widetilde{\mu}_{I_t^*,O_{I_t^*}}\right)$, where $I_t^* \leftarrow \arg\max_{j\in C_{I_t}} \widetilde{\mu}_{j,O_j}$;
    Set $r_{I_t} \leftarrow r_{I_t} + 1$.
  end if
end for

the information (including the located cliques and the pulled arms) from round 1 to t (including round t) are the same whether taking $X_{1:T}$ or $X'_{1:T}$ as input. Let us say $I_t = I'_t = i$, where I'_t indicates the index of the located clique in round t when taking $X'_{1:T}$ as input. Note that the reward vector in round t may only impact the statistics of all arms within the located clique in round t, i.e., clique C_i.

Let $t_0 \geq t$ the round at the end of which clique C_i will report a new tuple. Since $X_{t+1:t_0} = X'_{t+1:t_0}$, we know that the decisions made from round $t+1$ to t_0 (including round t_0) are the same whether taking $X_{1:T}$ or $X'_{1:T}$ as input.

The decisions made from round $t_0 + 1$ may start to diverge between taking $X_{1:T}$ and $X'_{1:T}$ as input. We now show that even if from round $t_0 + 1$, the decisions are almost the same by using the Laplace mechanism (Definition 3.3 in [16]). As a noise variable drawn from Lap $\left(\frac{1}{0.5\epsilon}\right)$ is injected to the obtained observations of each arm $j \in C_i$, from Theorem 3.6 [16] that Laplace mechanism preserves ϵ-differential privacy and Claim 3.9 that the Report Noisy Max is ϵ-differentially private [16], we know that the reported arm by clique C_i at the end of t_0 is almost the same whether taking $X_{1:T}$ or $X'_{1:T}$ as input. Also, the differentilly private empirical mean of the reported arm is almost the same whether taking $X_{1:T}$ or $X'_{1:T}$ as input.

Since the reward vector in round t cannot impact the statistics of arms other than those in clique C_i, and even for arms in clique C_i, Learner can only see the statistics of the arm with the highest differentially private empirical mean, the decisions made from round $t_0 + 1$ are exactly the same conditioned on that the impacted clique outputs the same arm and the same differentially private empirical mean of that arm. □

We now present a regret bound for Algorithm 14. Let $\Delta_i^{\min} = \min_{j \in C_i : \Delta_j > 0} \Delta_j$ be the minimum mean reward gap among all sub-optimal arms in clique C_i.

Theorem 24. *The regret of Algorithm 14 is at most*

$$O\left(\sum_{1 \leq i \leq |\mathcal{C}|} \frac{\log\left(|C_i| \cdot T\right)}{\min\left\{\Delta_i^{\min}, \epsilon\right\}}\right) \quad .$$

Several remarks on are in order for Algorithm 14 and Theorem 24.

1. Both the leading and constant terms are linear in the size of clique covering $\mathcal{C}$ up to logarithmic factors ;

2. When setting $\epsilon \to \infty$, the differentially private graphical bandits setting will be the non-private graphical bandit setting that has been studied in Chapter 3. However, for each clique, the regret bound shown in Theorem 24 will be $O\left(\frac{\log(|C_i|\cdot T)}{\Delta_i^{\min}}\right)$ instead of the $O\left(\frac{\log(|C_i|\cdot T)\cdot\Delta_i^{\max}}{\left(\Delta_i^{\min}\right)^2}\right)$ bound that is shown in Theorem 9. The improvement comes from a more refined analysis that is presented in Appendix 6.6.2. Note that $\Delta_i^{\max} := \max_{j\in C_i}\Delta_j$ is the maximum mean reward gap among all sub-optimal arms in clique C_i.

6.5 Conclusion and Future Work

In this section, we discuss more about differentially private graphical bandits. One of the downsides of Algorithm 14, DP-UCB-N, is that it relies on a specific clique covering $\mathcal{C}$ as input, and the regret bound only holds for the input clique covering. There may be computational issue. However, for the non-private graphical bandit setting, Algorithm 5, UCB-NE, does not rely on $\mathcal{C}$ as input, and thus the regret bound holds for all the possible clique coverings and there is no computational issue. A natural question is: **is it possible to have a private learning algorithm for graphical bandits that does not rely on $\mathcal{C}$ as input? That is also to say, is there any computationally efficient learning algorithm for differentially private graphical bandits?**

Before answering this question, let us briefly summarize why DP-UCB-N succeeds to tackle the challenge that has been discussed in Section 6.3. Briefly, DP-UCB-N runs a private learning algorithm over a set of cliques, and each clique runs Follow-the-Noisy-Leader and returns a tuple with two attributes to help Learner to achieve the exploration-vs-exploitation trade-off. This combination makes the learning algorithm have a lower and unchanged l_1-sensitivity. Even if a reward vector is changed, it can only impact the statistics of at most one clique regardless of how many arms are covered by that clique. That is also to say, the degrees of the nodes of the graph cannot impact the l_1-sensitivity directly.

In contrast, if we run a private algorithm directly over the arm set, when changing a reward vector, the number of impacted arms highly depends on the degrees of the nodes of the graph. Therefore, our conjecture is that *a good private learning algorithm for graphical bandits requires Learner not to run a private algorithm directly over the arm set. Instead, Learner can run a private algorithm over groups that cover all*

arms. For DP-UCB-N, each clique can be viewed as a group.

Now, to answer the aforementioned question, we only need to find a computationally efficient way to form groups that cover all the arms. Ideally, the number of groups is linear in the independence number, another important quantity of the graph. Fortunately, by modifying the idea of ALPHASAMPLE in [14], we can form groups in an efficient way. Then, Learner can run a private learning algorithm over these formed groups, and each group runs a similar algorithm to FTNL. However, by using the idea of ALPHASAMPLE, an extra $\log(K)$ factor is expected in the regret bound.

6.6 Appendix of this Chapter

The organization of this appendix is as follows:

6.6.1 - Proofs of Theorem 23 ;

6.6.2 - Proofs of Theorem 24 .

To prove Theorem 24, we use the following fact.

Fact 1. (Fact 3.7 in [16]). If $Y \sim \text{Lap}(b)$, for any $0 < \delta < 1$, we have

$$\mathbb{P}\left\{|Y| > \ln\left(\frac{1}{\delta}\right) \cdot b\right\} = \delta \quad . \tag{6.4}$$

6.6.1 Proofs of Theorem 23

Proof of Theorem 23: Let $X_{1:T}$ be the original sequence of reward vectors and $X'_{1:T}$ be an arbitrary neighbouring sequence of reward vectors of $X_{1:T}$ such that $X_{1:T}$ and $X'_{1:T}$ differ in round t at most. Let $D_{1:T}$ be a sequence of decisions made from round 1 to round T when taking $X_{1:T}$ as input and let $D'_{1:T}$ be a sequence of decisions made when taking $X'_{1:T}$ as input.

We claim that for any $\sigma_{1:T} \in \mathcal{A}^T$, we have

$$\mathbb{P}\left\{D_{1:T} = \sigma_{1:T} \mid X_{1:T}, G\right\} \leq e^{\epsilon} \cdot \mathbb{P}\left\{D'_{1:T} = \sigma_{1:T} \mid X'_{1:T}, G\right\} \quad . \tag{6.5}$$

As G is fixed, we drop G during the proof. Since $X_{1:T}$ and $X'_{1:T}$ only differs in round t, the arms pulled up to round t (including round t) has the same distribu-

tion under either of $X_{1:T}$ or $X'_{1:T}$. That is also to say, for any $j \in \mathcal{A}$, we have

$$\mathbb{P}\left\{J_t = j, D_{1:t} = \sigma_{1:t} \mid X_{1:T}\right\} = \mathbb{P}\left\{J'_t = j, D'_{1:t} = \sigma_{1:t} \mid X'_{1:T}\right\} \quad , \tag{6.6}$$

where J'_t indicates the arm pulled in round t when taking $X'_{1:T}$ as input.

The LHS of (6.5) can be rewritten as

$$\begin{aligned} & \mathbb{P}\left\{D_{1:T} = \sigma_{1:T} \mid X_{1:T}\right\} \\ = & \sum_{j \in \mathcal{A}} \mathbb{P}\left\{J_t = j, D_{1:t} = \sigma_{1:t} \mid X_{1:T}\right\} \underbrace{\mathbb{P}\left\{D_{t+1:T} = \sigma_{t+1:T} \mid J_t = j, D_{1:t} = \sigma_{1:t}, X_{1:T}\right\}}_{\alpha}. \end{aligned} \tag{6.7}$$

Similarly, we have

$$\begin{aligned} & \mathbb{P}\left\{D'_{1:T} = \sigma_{1:T} \mid X'_{1:T}\right\} \\ = & \sum_{j \in \mathcal{A}} \mathbb{P}\left\{J'_t = j, D'_{1:t} = \sigma_{1:t} \mid X'_{1:T}\right\} \underbrace{\mathbb{P}\left\{D'_{t+1:T} = \sigma_{t+1:T} \mid J'_t = j, D'_{1:t} = \sigma_{1:t}, X'_{1:T}\right\}}_{\alpha'}. \end{aligned} \tag{6.8}$$

By plugging (6.6) into (6.7) and (6.8), we only need to analyze the relationships between α and α' in (6.7) and (6.8).

Let $t_0 \geq t$ be the first round such that, at the end of round t_0, the differentially private empirical means of all arms in clique C_{I_t} will be updated and a new tuple $\left(I_t^*(t_0), \widetilde{\mu}_{I_t^*(t_0), O_{I_t^*(t_0)}(t_0)}\right)$ will be reported. Note that t_0 is random. Similarly, let $t'_0 \geq t$ be the first round such that, at the end of round t'_0, the differentially private empirical means of all arms in clique $C_{I'_t}$ will be updated and a new tuple $\left(I_t'^*(t'_0), \widetilde{\mu}_{I_t'^*(t'_0), O_{I_t'^*(t'_0)}}\right)$ will be reported. Note that $C_{I'_t}$ is clique that covers J'_t when taking $X'_{1:T}$ as input.

For any $1 \leq i \leq \beta(G)$, we have

$$\begin{aligned} & \mathbb{P}\left\{I_t = i \mid J_t = j, D_{1:t} = \sigma_{1:t}, X_{1:t} = \sigma_{1:t}\right\} \\ = & \mathbb{P}\left\{I'_t = i \mid J'_t = j, D'_{1:t} = \sigma_{1:t}, X'_{1:t} = \sigma_{1:t}\right\} \quad . \end{aligned} \tag{6.9}$$

For any $t \leq s \leq T$, we have

$$\begin{aligned} & \mathbb{P}\left\{t_0 = s \mid J_t = j, I_t = i, D_{1:t} = \sigma_{1:t}, X_{1:t} = \sigma_{1:t}\right\} \\ = & \mathbb{P}\left\{t'_0 = s \mid I'_t = i, J'_t = j, D'_{1:t} = \sigma_{1:t}, X'_{1:t} = \sigma_{1:t}\right\} \quad . \end{aligned} \tag{6.10}$$

We now rewrite α as

$$\begin{aligned}\alpha &= \mathbb{P}\left\{D_{1:T} = \sigma_{1:T} \mid J_t = j, D_{1:t} = \sigma_{1:t}, X_{1:T}\right\} \\ &= \sum_{s=t}^{T}\sum_{i=1}^{\beta(G)} \mathbb{P}\left\{t_0 = s, I_t = i \mid J_t = j, D_{1:t} = \sigma_{1:t}, X_{1:T}\right\} \\ &\quad \underbrace{\mathbb{P}\left\{D_{t+1:T} = \sigma_{t+1:T} \mid t_0 = s, I_t = i, J_t = j, D_{1:t} = \sigma_{1:t}, X_{1:T}\right\}}_{\beta} \quad .\end{aligned} \tag{6.11}$$

Similarly, α' can be rewritten as

$$\begin{aligned}\alpha' &= \mathbb{P}\left\{D'_{1:T} = \sigma_{1:T} \mid J'_t = j, D'_{1:t} = \sigma_{1:t}, X'_{1:T}\right\} \\ &= \sum_{s=t}^{T}\sum_{i=1}^{\beta(G)} \mathbb{P}\left\{t'_0 = s, I'_t = i \mid J'_t = j, D'_{1:t} = \sigma_{1:t}, X'_{1:T}\right\} \\ &\quad \underbrace{\mathbb{P}\left\{D'_{t+1:T} = \sigma_{t+1:T} \mid t'_0 = s, I'_t = i, J'_t = j, D'_{1:t} = \sigma_{1:t}, X'_{1:T}\right\}}_{\beta'} \quad .\end{aligned} \tag{6.12}$$

By plugging (6.9) and (6.10) into (6.11) and (6.12), respectively, we know that only β and β' can be different. We now show that $\beta \leq e^{\epsilon} \cdot \beta'$.

For β, we have

$$\begin{aligned}\beta &= \mathbb{P}\left\{D_{t+1:T} = \sigma_{t+1:T} \mid t_0 = s, I_t = i, J_t = j, D_{1:t} = \sigma_{1:t}, X_{1:T}\right\} \\ &= \mathbb{P}\left\{D_{t+1:t_0} = \sigma_{t+1:t_0} \mid t_0 = s, I_t = i, J_t = j, D_{1:t} = \sigma_{1:t}, X_{1:T}\right\} \\ &\quad \underbrace{\mathbb{P}\left\{D_{t_0+1:T} = \sigma_{t_0+1:T} \mid \underbrace{t_0 = s, I_t = i, J_t = j, D_{1:t_0} = \sigma_{1:t_0}, X_{1:T}}_{:=M}\right\}}_{\gamma} \quad .\end{aligned} \tag{6.13}$$

Similarly, for β', we have

$$\begin{aligned}\beta' &= \mathbb{P}\left\{D'_{t+1:T} = \sigma_{t+1:T} \mid t'_0 = s, I'_t = i, J'_t = j, D'_{1:t} = \sigma_{1:t}, X'_{1:T}\right\} \\ &= \mathbb{P}\left\{D'_{t+1:t'_0} = \sigma_{t+1:t'_0} \mid t'_0 = s, I'_t = i, J'_t = j, D'_{1:t} = \sigma_{1:t}, X'_{1:T}\right\} \\ &\quad \underbrace{\mathbb{P}\left\{D'_{t'_0+1:T} = \sigma_{t'_0+1:T} \mid \underbrace{t'_0 = s, I'_t = i, J'_t = j, D'_{1:t'_0} = \sigma_{1:t'_0}, X'_{1:T}}_{:=M'}\right\}}_{\gamma'} \quad .\end{aligned} \tag{6.14}$$

Note that $X_{1:T}$ and $X'_{1:T}$ only differ in round t, and any reward observed in

round t will not be used in the algorithm's decision-making until after round t_0 and t'_0. Therefore, $D_{t+1:t_0}$ and $D'_{t+1:t'_0}$ have the same conditional distribution. We now show $\gamma \leq e^{\epsilon} \cdot \gamma'$.

As a noise drawn from Lap $\left(\frac{1}{0.5\epsilon}\right)$ is injected to the newly obtained observations, from Claim 3.9 in [16], for any $a \in \mathcal{A}$, we have

$$\mathbb{P}\left\{I_t^*(t_0) = a \mid M\right\} \leq \mathbb{P}\left\{I_t'^*(t'_0) = a \mid M'\right\} \cdot e^{0.5\epsilon} \quad . \tag{6.15}$$

Meanwhile, is for any $\mathcal{I} \subseteq \mathbb{R}$, we also have

$$\begin{aligned} & \mathbb{P}\left\{\widetilde{\mu}_{I_t^*(t_0), O_{I_t^*(t_0)}(t_0)} \cdot O_{I_t^*(t_0)}(t_0) \in \mathcal{I} \mid M, I_t^*(t_0) = a\right\} \\ \leq \quad & \mathbb{P}\left\{\widetilde{\mu}_{I_t'^*(t'_0), O_{I_t'^*(t'_0)}(t'_0)} \cdot O_{I_t'^*(t'_0)}(t'_0) \in \mathcal{I} \mid M', I_t'^*(t'_0) = a\right\} \cdot e^{0.5\epsilon} \quad . \end{aligned} \tag{6.16}$$

By combining (6.15) and (6.16), we have

$$\begin{aligned} & \mathbb{P}\left\{I_t^*(t_0) = a, \quad \widetilde{\mu}_{I_t^*(t_0), O_{I_t^*(t_0)}(t_0)} \cdot O_{I_t^*(t_0)}(t_0) \in \mathcal{I} \mid M\right\} \\ \leq \quad & e^{\epsilon} \cdot \mathbb{P}\left\{I_t'^*(t'_0) = a, \quad \widetilde{\mu}_{I_t'^*(t'_0), O_{I_t'^*(t'_0)}(t'_0)} \cdot O_{I_t'^*(t'_0)}(t'_0) \in \mathcal{I} \mid M'\right\} \quad . \end{aligned} \tag{6.17}$$

As $X_{t_0+1:T} = X'_{t'_0+1:T}$, we have

$$\begin{aligned} & \mathbb{P}\left\{D_{t_0+1:T} = \sigma_{t_0+1:T} \mid I_t^*(t_0) = a, \quad \widetilde{\mu}_{I_t^*(t_0), O_{I_t^*(t_0)}(t_0)} \cdot O_{I_t^*(t_0)}(t_0) \in \mathcal{I}, M\right\} \\ = \quad & \mathbb{P}\left\{D'_{t'_0+1:T} = \sigma_{t'_0+1:T} \mid I_t'^*(t'_0) = a, \quad \widetilde{\mu}_{I_t'^*(t'_0), O_{I_t'^*(t'_0)}(t'_0)} \cdot O_{I_t'^*(t'_0)}(t'_0) \in \mathcal{I}, M'\right\}. \end{aligned} \tag{6.18}$$

By combining (6.17) and (6.18), we have $\gamma \leq e^{\epsilon}\gamma'$, which concludes the proof. □

6.6.2 Proofs of Theorem 24

Proof of Theorem 24: We first rewrite the regret and have

$$\mathcal{R}(T) \quad = \quad \sum_{C_i \in \mathcal{C}} \underbrace{\sum_{j \in C_i : \Delta_j > 0} \mathbb{E}\left[\sum_{t=1}^{T} \mathbf{1}\left\{J_t = j\right\}\right] \Delta_j}_{=: R_i(T)} \quad . \tag{6.19}$$

Let $R_i(T)$ be the regret suffered by pulling sub-optimal arms within clique C_i over T rounds. Let us say that the unique best arm is in C_1. We now analyze the regret by pulling any sub-optimal arm in $C_i \in \mathcal{C}$, i.e., upper bounding $R_i(t)$ for all $1 \leq i \leq |\mathcal{C}|$.

Before upper bounding the regret, let us introduce some notations first. For any clique $C_i \in \mathcal{C}$, let $\Delta_i^{\min} := \min_{j \in C_i : \Delta_j > 0} \Delta_j$ indicate the minimum mean reward gap among all sub-optimal arms within clique C_i and let $r_{\max}^{(i)} := \left\lceil \log\left(\frac{1}{\Delta_i^{(\min)}}\right) \right\rceil$. For the proof, to have a refined regret bound analysis, we take two partitions. One partition is taken over the arms in each clique C_i and the other partition is taken over all T rounds.

For each clique C_i, we partition all arms in C_i into multiple groups based on the mean reward gaps. For each $1 \leq r \leq r_{\max}^{(i)}$, let $\Phi_r^{(i)} := \left\{ j \in C_i : 0.5^r \leq \Delta_j < 0.5^{r-1} \right\} \subseteq C_i$ collect all arms with mean reward gaps between $\left[0.5^r, 0.5^{r-1}\right)$. Note that pulling any arm in $\Phi_r^{(i)}$ will suffer 0.5^{r-1} regret at most per round.

We also partition all T rounds into multiple phases, and each phase may have multiple reported tuples. It is important to note that each reported tuple must be computed based on fresh observations.

Let $\lambda_r^{(i)} := \frac{64 \cdot \log(|C_i| \cdot T)}{\min\{0.5^{2r},\, \epsilon \cdot 0.5^r\}}$. Let $\left(d_0^{(i)}, d_1^{(i)}, d_2^{(i)}, \ldots, d_{r_{\max}^{(i)}}^{(i)}, d_{r_{\max}^{(i)}+1}^{(i)} \right)$ be a sequence of non-decreasing non-negative integers that partitions all T rounds into multiple phases. For each $1 \leq r \leq r_{\max}^{(i)}$, we set $d_r^{(i)} := \left\lceil \log\left(\lambda_r^{(i)}\right) \right\rceil$. We set $d_0^{(i)} = 0$ and $d_{r_{\max}^{(i)}+1}^{(i)} = \log(T)$.

Let $\tau_i^{(s-1)}$ be the round at the end of which the s-th tuple from clique C_i will be reported. Note that Learner will use the newly reported tuple to construct the upper confidence bounds for all the rounds until after a new tuple is reported, i.e., for all rounds in $\left\{ \tau_i^{(s-1)} + 1, \tau_i^{(s-1)} + 2, \ldots, \tau_i^{(s)} \right\}$, Learner will use the tuple reported by round $\tau_i^{(s-1)}$ to construct the upper confidence bounds.

Let $N_i(t-1) := \sum_{h=1}^{t-1} \mathbf{1}\left\{ J_h \in C_i \right\}$ be the total number of pulls of arms in clique C_i by the end of round $t-1$.

For any clique $C_i \in \mathcal{C}$, we have

$$\begin{aligned}
& R_i(T) \\
= & \sum_{j \in C_i} \mathbb{E}\left[\sum_{t=1}^{T} \mathbf{1}\left\{J_t = j\right\}\right] \cdot \Delta_j \\
\leq & \sum_{r=1}^{r_{\max}^{(i)}} \mathbb{E}\left[\sum_{t=1}^{T} \mathbf{1}\left\{\overline{\mu}_i(t) \geq \overline{\mu}_1(t), N_i(t) > N_i(t-1), \exists j \in \Phi_r^{(i)} \; s.t. \; J_t = j\right\}\right] \cdot 0.5^{r-1} \\
\leq & \sum_{r=1}^{r_{\max}^{(i)}} \sum_{s=1}^{\log(T)} 0.5^{r-1} \\
& \mathbb{E}\left[\sum_{t=\tau_i^{(s-1)}+1}^{\tau_i^{(s)}} \mathbf{1}\left\{\overline{\mu}_i(t) \geq \overline{\mu}_1(t), N_i(t) > N_i(t-1), \exists j \in \Phi_r^{(i)} \; s.t. \; J_t = j\right\}\right] \quad .
\end{aligned} \tag{6.20}$$

The first inequality in (6.20) uses the fact pulling any arm in group $\Phi_r^{(i)}$ will suffer at most 0.5^{r-1} regret per round. The second inequality partitions all T rounds into multiple intervals based on whether a new tuple will be reported by clique C_i or not. Note that each clique can report at most $\log(T)$ tuples during the entire learning.

We now partition all these $\log(T)$ reported tuples further by using the aforementioned non-decreasing non-negative integers $\left(d_0^{(i)}, d_1^{(i)}, \ldots, d_{r_{\max}^{(i)}+1}^{(i)}\right)$. We have

$$\begin{aligned}
R_i(T) \leq & \sum_{r=1}^{r_{\max}^{(i)}} \sum_{q=0}^{r_{\max}^{(i)}} \sum_{s=d_q^{(i)}+1}^{d_{q+1}^{(i)}} 0.5^{r-1} \\
& \mathbb{E}\left[\sum_{t=\tau_i^{(s-1)}+1}^{\tau_i^{(s)}} \mathbf{1}\left\{\overline{\mu}_i(t) \geq \overline{\mu}_1(t), N_i(t) > N_i(t-1), \exists j \in \Phi_r^{(i)} \; s.t. \; J_t = j\right\}\right] \\
\leq & \sum_{q=0}^{r_{\max}^{(i)}} \sum_{s=d_q^{(i)}+1}^{d_{q+1}^{(i)}} 2^s \cdot 0.5^q \\
& + \sum_{q=1}^{r_{\max}^{(i)}} \sum_{s=d_q^{(i)}+1}^{d_{q+1}^{(i)}} 0.5^{q-1} \\
& \underbrace{\mathbb{E}\left[\sum_{t=\tau_i^{(s-1)}+1}^{\tau_i^{(s)}} \mathbf{1}\left\{\overline{\mu}_i(t) \geq \overline{\mu}_1(t), N_i(t) > N_i(t-1), \exists j \in \Phi_q^{(i)} \; s.t. \; J_t = j\right\}\right]}_{\zeta} \quad .
\end{aligned} \tag{6.21}$$

We now analyze term ζ in (6.21). We have

$$
\begin{aligned}
& \zeta \\
= \; & \mathbb{E}\left[\sum_{t=\tau_i^{(s-1)}+1}^{\tau_i^{(s)}} \mathbf{1}\left\{\overline{\mu}_i(t) \geq \overline{\mu}_1(t), N_i(t) > N_i(t-1), \exists j \in \Phi_q^{(i)} \; s.t. \; J_t = j\right\}\right] \\
\leq \; & \mathbb{E}\left[\sum_{t=\tau_i^{(s-1)}+1}^{\tau_i^{(s)}} \mathbf{1}\left\{N_i(t) > N_i(t-1),\right.\right. \\
& \left.\left.\max_{j \in \Phi_q^{(i)}} \widetilde{\mu}_{j,O_j(t-1)} + \sqrt{\frac{3\log(|C_i|\cdot t)}{O_j(t-1)}} + \frac{6\log(|C_i|\cdot t)}{\epsilon \cdot O_j(t-1)} \geq \overline{\mu}_1(t)\right\}\right] \\
\leq \; & \mathbb{E}\left[\sum_{t=\tau_i^{(s-1)}+1}^{\tau_i^{(s)}} \mathbf{1}\left\{N_i(t) > N_i(t-1), \max_{j \in \Phi_q^{(i)}} \widetilde{\mu}_{j,2^{s-1}} + \sqrt{\frac{3\log(|C_i|\cdot t)}{2^{s-1}}} + \frac{6\log(|C_i|\cdot t)}{\epsilon \cdot 2^{s-1}} \geq \right.\right. \\
& \left.\left.\widetilde{\mu}_{1,O_1(t-1)} + \sqrt{\frac{3\log(|C_1|\cdot t)}{O_1(t-1)}} + \frac{6\log(|C_1|\cdot t)}{\epsilon \cdot O_1(t-1)}\right\}\right] \\
\leq \; & \mathbb{E}\left[\sum_{t=\tau_i^{(s-1)}+1}^{\tau_i^{(s)}} \sum_{h=0}^{\log(t-1)} \mathbf{1}\left\{N_i(t) > N_i(t-1),\right.\right. \\
& \left.\left.\max_{j \in \Phi_q^{(i)}} \widetilde{\mu}_{j,2^{s-1}} + \sqrt{\frac{3\log(|C_i|\cdot t)}{2^{s-1}}} + \frac{6\log(|C_i|\cdot t)}{\epsilon \cdot 2^{s-1}} \geq \widetilde{\mu}_{1,2^h} + \sqrt{\frac{3\log(|C_1|\cdot t)}{2^h}} + \frac{6\log(|C_1|\cdot t)}{\epsilon \cdot 2^h}\right\}\right].
\end{aligned}
\tag{6.22}
$$

The second inequality in (6.22) uses the facts that the number of effective observations for any arm within clique C_i is 2^{s-1} for all rounds in $\left\{\tau_i^{(s-1)}+1, \ldots, \tau_i^{(s)}\right\}$, and $\overline{\mu}_1(t) = \max_{k \in C_1} \widetilde{\mu}_{k,O_k(t-1)} + \sqrt{\frac{3\log(|C_1|\cdot t)}{O_k(t-1)}} + \frac{6\log(|C_1|\cdot t)}{O_k(t-1)} \geq \widetilde{\mu}_{1,O_1(t-1)} + \sqrt{\frac{3\log(|C_1|\cdot t)}{O_1(t-1)}} + \frac{6\log(|C_1|\cdot t)}{O_1(t-1)}$.

If the indicator function in the last inequality of (6.22) is true, it implies that at least one of the followings is true:

$$
\max_{j \in \Phi_q^{(i)}} \widetilde{\mu}_{j,2^{s-1}} - \sqrt{\frac{3\log\left(|C_i|\cdot t\right)}{2^{s-1}}} - \frac{6\log\left(|C_i|\cdot t\right)}{\epsilon \cdot 2^{s-1}} \geq \mu_1 - 0.5^q \quad , \tag{6.23}
$$

$$
\widetilde{\mu}_{1,2^h} + \sqrt{\frac{3\log\left(|C_1|\cdot t\right)}{2^h}} + \frac{6\log\left(|C_1|\cdot t\right)}{\epsilon \cdot 2^h} \leq \mu_1 \quad , \tag{6.24}
$$

$$2\left(\sqrt{\frac{3\log\left(|C_i|\cdot t\right)}{2^{s-1}}}+\frac{6\log\left(|C_i|\cdot t\right)}{\epsilon\cdot 2^{s-1}}\right)>0.5^q \quad . \tag{6.25}$$

Let $Z_j^{(s)}\sim \text{Lap}\left(\frac{1}{0.5\epsilon}\right)$ be the noise injected to the 2^s observations of arm j. For inequalities (6.23) and (6.24), we apply the Hoeffding's inequality and use inequality (6.4) to upper bound the probability that these events happen. We have

$$\begin{aligned}
&\mathbb{P}\left\{\max_{j\in\Phi_q^{(i)}}\widetilde{\mu}_{j,2^{s-1}}-\sqrt{\frac{3\log(|C_i|\cdot t)}{2^{s-1}}}-\frac{6\log(|C_i|\cdot t)}{\epsilon\cdot 2^{s-1}}\geq\mu_1-0.5^q\right\}\\
\leq&\sum_{j\in\Phi_q^{(i)}}\mathbb{P}\left\{\widetilde{\mu}_{j,2^{s-1}}-\sqrt{\frac{3\log(|C_i|\cdot t)}{2^{s-1}}}-\frac{6\log(|C_i|\cdot t)}{\epsilon\cdot 2^{s-1}}\geq\mu_1-0.5^q\right\}\\
\leq&\sum_{j\in\Phi_q^{(i)}}\mathbb{P}\left\{\widetilde{\mu}_{j,2^{s-1}}-\sqrt{\frac{3\log(|C_i|\cdot t)}{2^{s-1}}}-\frac{6\log(|C_i|\cdot t)}{\epsilon\cdot 2^{s-1}}\geq\mu_j\right\}\\
=&\sum_{j\in\Phi_q^{(i)}}\mathbb{P}\left\{\widehat{\mu}_{j,2^{s-1}}+\frac{Z_j^{(s-1)}}{2^{s-1}}-\sqrt{\frac{3\log(|C_i|\cdot t)}{2^{s-1}}}-\frac{6\log(|C_i|\cdot t)}{\epsilon\cdot 2^{s-1}}\geq\mu_j\right\}\\
\leq&\sum_{j\in\Phi_q^{(i)}}\left(\underbrace{\mathbb{P}\left\{\widehat{\mu}_{j,2^{s-1}}-\sqrt{\frac{3\log\left(|C_i|t\right)}{2^{s-1}}}\geq\mu_j\right\}}_{\text{Hoeffding's inequality}}+\underbrace{\mathbb{P}\left\{Z_j^{(s-1)}-\frac{3\log\left(|C_i|t\right)}{0.5\cdot\epsilon}\geq 0\right\}}_{\text{Inequality (6.4)}}\right)\\
\leq&\frac{2|\Phi_q^{(i)}|}{\left(|C_i|\cdot t\right)^3} \quad .
\end{aligned} \tag{6.26}$$

Similarly, we have

$$\mathbb{P}\left\{\widetilde{\mu}_{1,2^h}+\sqrt{\frac{3\log\left(|C_1|\cdot t\right)}{2^h}}+\frac{6\log\left(|C_1|\cdot t\right)}{\epsilon\cdot 2^h}\leq\mu_1\right\}\leq\frac{2}{\left(|C_1|\cdot t\right)^3} \quad . \tag{6.27}$$

We now show that (6.25) cannot be true by using contradiction when $s-1\geq d_q^{(i)}$. Note that $2^{s-1}\geq 2^{d_q^{(i)}}=2^{\left\lceil\log\left(\lambda_q^{(i)}\right)\right\rceil}=2^{\left\lceil\log\left(\frac{64\log\left(|C_i|\cdot T\right)}{\min\left\{0.5^{2q},\epsilon\cdot 0.5^q\right\}}\right)\right\rceil}\geq\frac{64\log(|C_i|\cdot T)}{\min\left\{0.5^{2q},\epsilon\cdot 0.5^q\right\}}$.

We have

$$\begin{aligned}
\text{LHS in (6.25)} &= 2\left(\sqrt{\frac{3\log(|C_i|\cdot t)}{2^{s-1}}}+\frac{6\log(|C_i|\cdot t)}{\epsilon\cdot 2^{s-1}}\right)\\
&\le 2\left(\sqrt{\frac{3\log(|C_i|\cdot T)}{\frac{64\log(|C_i|\cdot T)}{\min\left\{0.5^{2q},\epsilon\cdot 0.5^{q}\right\}}}}+\frac{6\log(|C_i|\cdot T)}{\epsilon\cdot\frac{64\log(|C_i|\cdot T)}{\min\left\{0.5^{2q},\epsilon\cdot 0.5^{q}\right\}}}\right)\\
&\le 2\left(\sqrt{\frac{3\min\left\{0.5^{2q},\epsilon\cdot 0.5^{q}\right\}}{64}}+\frac{6\cdot\min\left\{0.5^{2q},\epsilon\cdot 0.5^{q}\right\}}{64\epsilon}\right)\\
&\le 2\left(\frac{\sqrt{3}\cdot 0.5^{q}}{8}+\frac{6\cdot 0.5^{q}}{64}\right)\\
&< 0.5^{q} \quad ,
\end{aligned} \tag{6.28}$$

which yields contradiction.

Therefore, we have

$$\begin{aligned}
\zeta &\le \mathbb{E}\left[\sum_{t=\tau_i^{(s-1)}+1}^{\tau_i^{(s)}}\sum_{h=0}^{\log(t-1)}\mathbf{1}\left\{N_i(t)>N_i(t-1),\right.\right.\\
&\qquad \left.\left.\max_{j\in\Phi_q^{(i)}}\widetilde{\mu}_{j,2^{s-1}}+\sqrt{\frac{3\log(|C_i|\cdot t)}{2^{s-1}}}+\frac{6\log(|C_i|\cdot t)}{\epsilon\cdot 2^{s-1}}\ge\widetilde{\mu}_{1,2^h}+\sqrt{\frac{3\log(|C_1|\cdot t)}{2^h}}+\frac{6\log(|C_1|\cdot t)}{\epsilon\cdot 2^h}\right\}\right]\\
&\le \mathbb{E}\left[\sum_{t=2^{s-1}+1}^{T}\sum_{h=0}^{\log(t-1)}\mathbf{1}\left\{N_i(t)>N_i(t-1),\right.\right.\\
&\qquad \left.\left.\max_{j\in\Phi_q^{(i)}}\widetilde{\mu}_{j,2^{s-1}}+\sqrt{\frac{3\log(|C_i|\cdot t)}{2^{s-1}}}+\frac{6\log(|C_i|\cdot t)}{\epsilon\cdot 2^{s-1}}\ge\widetilde{\mu}_{1,2^h}+\sqrt{\frac{3\log(|C_1|\cdot t)}{2^h}}+\frac{6\log(|C_1|\cdot t)}{\epsilon\cdot 2^h}\right\}\right]\\
&\le \sum_{t=2^{s-1}+1}^{T}\sum_{h=0}^{\log(t-1)}\left(\mathbb{P}\left\{(6.23)\text{ is true}\right\}+\mathbb{P}\left\{(6.24)\text{ is true}\right\}\right)\\
&\le \sum_{t=2^{s-1}+1}^{T}\sum_{h=0}^{\log(t-1)}\frac{4}{t^3}\\
&\le \sum_{t=1}^{T}t\cdot\frac{4}{t^3}\\
&\le 7 \quad .
\end{aligned} \tag{6.29}$$

By plugging the upper bound of ζ to (6.20), we have

$$\begin{aligned} R_i(T) &\leq \sum_{q=0}^{r_{\max}^{(i)}} \sum_{s=d_q^{(i)}+1}^{d_{q+1}^{(i)}} 2^s \cdot 0.5^q + \sum_{q=1}^{r_{\max}^{(i)}} \sum_{s=d_q^{(i)}+1}^{d_{q+1}^{(i)}} 7 \cdot 0.5^{q-1} \\ &\leq 5 \sum_{q=0}^{r_{\max}^{(i)}} \sum_{s=d_q^{(i)}+1}^{d_{q+1}^{(i)}} 2^s \cdot 0.5^q \\ &\leq O\left(\frac{\log(|C_i| \cdot T)}{\min\left\{\Delta_i^{(\min)}, \epsilon\right\}} \right) \quad . \end{aligned} \tag{6.30}$$

□

Bibliography

[1] Naman Agarwal and Karan Singh. The price of differential privacy for online learning. In *International Conference on Machine Learning*, pages 32–40. PMLR, 2017.

[2] Shipra Agrawal and Navin Goyal. Near-optimal regret bounds for Thompson sampling. *Journal of the ACM (JACM)*, 64(5):1–24, 2017.

[3] Noga Alon, Nicolo Cesa-Bianchi, Ofer Dekel, and Tomer Koren. Online learning with feedback graphs: Beyond bandits. In *Conference on Learning Theory*, pages 23–35. PMLR, 2015.

[4] Jean-Yves Audibert and Sébastien Bubeck. Minimax policies for adversarial and stochastic bandits. In *COLT*, volume 7, pages 1–122, 2009.

[5] Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2):235–256, 2002.

[6] Peter Auer and Ronald Ortner. UCB revisited: Improved regret bounds for the stochastic multi-armed bandit problem. *Periodica Mathematica Hungarica*, 61(1-2):55–65, 2010.

[7] Debabrota Basu, Christos Dimitrakakis, and Aristide Tossou. Differential privacy for multi-armed bandits: What is it and what is its cost? *arXiv preprint arXiv:1905.12298*, 2019.

[8] Donald A Berry and Bert Fristedt. Bandit problems: sequential allocation of experiments (monographs on statistics and applied probability). *London: Chapman and Hall*, 5(71-87):7–7, 1985.

[9] Lilian Besson and Emilie Kaufmann. What doubling tricks can and can't do for multi-armed bandits. *arXiv preprint arXiv:1803.06971*, 2018.

[10] Stéphane Caron, Branislav Kveton, Marc Lelarge, and Smriti Bhagat. Leveraging side observations in stochastic bandits. *Uncertainty in Artificial Intelligence (UAI)*, 2012.

[11] T-H Hubert Chan, Elaine Shi, and Dawn Song. Private and continual release of statistics. *ACM Transactions on Information and System Security (TISSEC)*, 14(3):1–24, 2011.

[12] Olivier Chapelle and Lihong Li. An empirical evaluation of Thompson sampling. *Advances in neural information processing systems*, 24:2249–2257, 2011.

[13] Xiaoyu Chen, Kai Zheng, Zixin Zhou, Yunchang Yang, Wei Chen, and Liwei Wang. (Locally) differentially private combinatorial semi-bandits. In *International Conference on Machine Learning*, pages 1757–1767. PMLR, 2020.

[14] Alon Cohen, Tamir Hazan, and Tomer Koren. Online learning with feedback graphs without the graphs. In *International Conference on Machine Learning*, pages 811–819. PMLR, 2016.

[15] Cynthia Dwork, Moni Naor, Toniann Pitassi, and Guy N Rothblum. Differential privacy under continual observation. In *Proceedings of the forty-second ACM symposium on Theory of computing*, pages 715–724, 2010.

[16] Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3-4):211–407, 2014.

[17] Eyal Even-Dar, Shie Mannor, and Yishay Mansour. PAC bounds for multi-armed bandit and markov decision processes. In *International Conference on Computational Learning Theory*, pages 255–270. Springer, 2002.

[18] Yoav Freund and Robert E Schapire. A decision-theoretic generalization of online learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139, 1997.

[19] Aurélien Garivier and Olivier Cappé. The KL-UCB algorithm for bounded stochastic bandits and beyond. In *Proceedings of the 24th annual conference on learning theory*, pages 359–376. JMLR Workshop and Conference Proceedings, 2011.

[20] Abhradeep Guha Thakurta and Adam Smith. (Nearly) optimal algorithms for private online learning in full-information and bandit settings. *Advances in Neural Information Processing Systems*, 26:2733–2741, 2013.

[21] Bingshan Hu, Zhiming Huang, and Nishant A Mehta. Optimal algorithms for private online learning in a stochastic environment. *arXiv preprint arXiv:2102.07929*, 2021.

[22] Bingshan Hu, Nishant A Mehta, and Jianping Pan. Problem-dependent regret bounds for online learning with feedback graphs. In *Uncertainty in Artificial Intelligence*, pages 852–861. PMLR, 2020.

[23] Prateek Jain, Pravesh Kothari, and Abhradeep Thakurta. Differentially private online learning. In *Conference on Learning Theory*, pages 24–1. JMLR Workshop and Conference Proceedings, 2012.

[24] Emilie Kaufmann, Olivier Cappé, and Aurélien Garivier. On Bayesian upper confidence bounds for bandit problems. In *Artificial intelligence and statistics*, pages 592–600. PMLR, 2012.

[25] Emilie Kaufmann, Nathaniel Korda, and Rémi Munos. Thompson sampling: An asymptotically optimal finite-time analysis. In *International conference on algorithmic learning theory*, pages 199–213. Springer, 2012.

[26] Tze Leung Lai and Herbert Robbins. Asymptotically efficient adaptive allocation rules. *Advances in applied mathematics*, 6(1):4–22, 1985.

[27] Tan Li, Linqi Song, and Christina Fragouli. Federated recommendation system via differential privacy. *2020 IEEE International Symposium on Information Theory (ISIT)*, 2020.

[28] Fang Liu, Swapna Buccapatnam, and Ness Shroff. Information directed sampling for stochastic bandits with graph feedback. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[29] Fang Liu, Zizhan Zheng, and Ness Shroff. Analysis of Thompson sampling for graphical bandits without the graphs. *Uncertainty in Artificial Intelligence (UAI)*, 2018.

[30] Thodoris Lykouris, Eva Tardos, and Drishti Wali. Feedback graph regret bounds for Thompson sampling and UCB. In *Algorithmic Learning Theory*, pages 592–614. PMLR, 2020.

[31] Shie Mannor and Ohad Shamir. From bandits to experts: On the value of side-observations. *Advances in Neural Information Processing Systems*, 24:684–692, 2011.

[32] Nikita Mishra and Abhradeep Thakurta. (Nearly) optimal differentially private stochastic multi-arm bandits. In *Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence*, pages 592–601, 2015.

[33] Touqir Sajed and Or Sheffet. An optimal private stochastic-MAB algorithm based on optimal private stopping rule. In *International Conference on Machine Learning*, pages 5579–5588. PMLR, 2019.

[34] Roshan Shariff and Or Sheffet. Differentially private contextual linear bandits. In *Advances in Neural Information Processing Systems*, pages 4296–4306, 2018.

[35] Cem Tekin and Mihaela Van Der Schaar. Distributed online learning via cooperative contextual bandits. *IEEE Transactions on Signal Processing*, 63(14):3700–3714, 2015.

[36] William R Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285–294, 1933.

[37] Aristide Tossou and Christos Dimitrakakis. Algorithms for differentially private multi-armed bandits. *arXiv preprint arXiv:1511.08681*, 2015.

[38] Aristide Tossou, Christos Dimitrakakis, and Devdatt Dubhashi. Thompson sampling for stochastic bandits with graph feedback. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.

[39] Alexandre B Tsybakov. *Introduction to nonparametric estimation*. Springer Science & Business Media, 2008.

[40] Tim Van Erven, Wojciech Kotłowski, and Manfred K Warmuth. Follow the leader with dropout perturbations. In *Conference on Learning Theory*, pages 949–974. PMLR, 2014.

[41] Vladimir Vovk. A game of prediction with expert advice. *Journal of Computer and System Sciences*, 56(2):153–173, 1998.

[42] Yuhong Yang and Andrew R Barron. An asymptotic property of model selection criteria. *IEEE Transactions on Information Theory*, 44(1):95–116, 1998.