On the Impact of Extra Information in Bandit Learning: More is Not Always Merrier

Ramakrishnan Krishnamurthy
Courant Institute of Mathematical Sciences, New York University

Abstract

More data is commonly assumed to improve learning performance, however, we show that this intuition can surprisngly fail in sequential decision-making. Considering a stochastic bandits setup with exogenous extra information, we ask if the free and unbiased extra reward samples are guaranteed to reduce the regret of the learner. We introduce a formal notion of 'non-degradation under extra information' and demonstrate that the widely used Upper Confidence Bound algorithm (UCB) violates this property, i.e., there exist benign extra-information under which UCB incurs strictly higher regret than when run without extra information. We contrast this behavior with Thompson Sampling and Explore-Then-Commit, which appear to benefit from the same extra information. Constructively, we propose a meta-algorithm that enforces non-degradation for a broad class of bandit algorithms. We also run empirical simulations to study our meta-algorithm.

1 Introduction

In classical supervised learning with i.i.d. data, more data almost always helps performance [Vapnik, 1999, Shalev-Shwartz and Ben-David, 2014]. Learning curves formalize this principle by quantifying how risk, accuracy, or error rates improve as the number of samples increases. Under standard assumptions, additional unbiased data monotonically improves estimation and generalization [Hastie, 2009].

Sequential decision-making problems such as reinforcement learning and multi-armed bandits depart fundamentally from this paradigm. The learner does not passively observe data, but must *act* to generate observations [Sutton et al., 1998]. Actions that are suboptimal incur immediate regret, creating an exploration-vs-exploitation trade-off. As a result, algorithmic performance is measured not only by the quality of the final model, but by cumulative reward obtained throughout the learning process [Lai and Robbins, 1985].

This distinction raises a basic and surprisingly underexplored question:

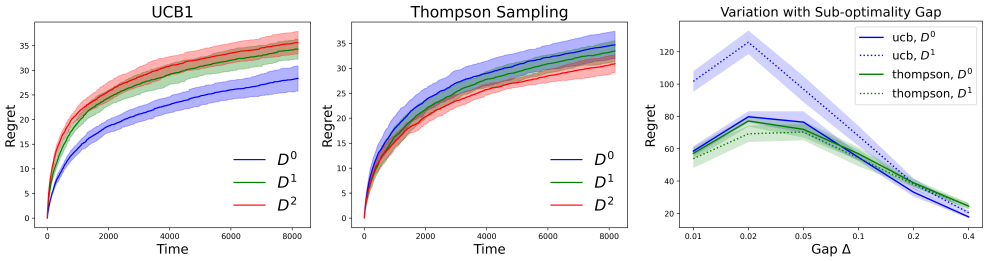
Does extra data always improve performance in sequential decision-making?

In this paper, we show that the answer is no. Even when additional data consists of unbiased, non-adversarial samples from the environment, and is free (incurrs no regret), providing it naively can *increase* overall regret. Unlike in supervised learning, where more data uniformly sharpens estimates, in online learning, additional information can distort exploration incentives and degrade performance.

We show that widely used optimism-based algorithms, such as UCB [Auer et al., 2002], can react pathologically to extra information. Extra samples—particularly that of 'good' actions—can suppress optimism and induce excessive exploration of inferior actions. By contrast, algorithms such as Thompson Sampling [Thompson, 1933] convert the same extra information into improved decisions and lower regret. Figure 1 illustrates this phenomenon on a simple two-armed stochastic bandit:

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

providing increasing extra information about the optimal action consistently *increases* the regret of UCB, while *decreasing* the regret of Thompson Sampling.



(a) UCB with extra info. (b) TS with extra info. (c) Effect of varying the gap Δ.

Figure 1: We compare the Upper Confidence Bound algorithm (UCB) and the Thompson Sampling algorithm (TS) on a two-armed instance with $\mu = \left(\frac{1}{2} + \Delta, \frac{1}{2}\right)$ and $\Delta = 0.1$ under three extra-information designs, where D^i provides i additional unbiased, independent samples of the optimal action at each time. As shown in Figure 1a, increasing extra information consistently increases the regret of UCB, while Figure 1b shows monotone regret reduction for Thompson Sampling. Figure 1c demonstrates that this qualitative contrast persists across different gaps Δ, with larger degradation for UCB on harder (smaller-gap) instances.

1.1 Our Contributions

We study the impact of exogenous extra information on stochastic bandit algorithms and identify a fundamental contrast between regret optimality and robustness to additional data. We formalize notions of non-degradation and monotonicity under extra information (Definitions 1 to 3), that capture the desirable requirement that providing free and unbiased data should not increase the expected regret. Our main contributions are:

- **UCB can degrade.** We show that the canonical UCB algorithm violates non-degradation: there exist benign extra-information designs under which UCB incurs strictly higher regret than with no extra information (Theorem 1), and quantify this degradation (Theorem 2) in a simplified setup isolating the environment noise.
- **Algorithms that benefit from extra information.** We discuss how extra information improves Thompson Sampling (Lemma 1), and show that Explore-Then-Commit algorithm benefits from extra information (Theorem 3).
- **Enforcing non-degradation.** We propose a meta-algorithm that enforces regret 'monotonicity' (i.e., lesser regret under more extra information) for a broad class of bandit algorithms (Theorem 4).

1.2 Related Work

For general introductions to multi-armed bandits algorithms, see Lattimore and Szepesvári [2020], Slivkins et al. [2019].

Side Information. A line of work studies bandits in which the learner observes additional feedback beyond the reward of the played action. Early formulations model this through *feedback graphs* [Mannor and Shamir, 2011, Alon et al., 2015, Buccapatnam et al., 2014], where playing an action reveals rewards of neighboring actions. Regret bounds in this literature depend on structural graph properties, such as independence or clique cover numbers.

Subsequent work generalizes this model to allow side information of varying quality, often assuming a known feedback structure that can be exploited via estimation or optimization techniques [Atsidakou et al., 2022, Kocák et al., 2016, Wu et al., 2015]. Other extensions consider time-varying action sets, switching costs, heavy-tailed rewards [Cortes et al., 2019, Arora et al., 2019, Gou et al., 2023], or unknown and evolving feedback structures [Cohen et al., 2016, Lykouris et al., 2020], thereby reducing the learner's ability to anticipate future side information. All of the above approaches impose

explicit structural assumptions on how side information is generated. In contrast, our framework models side information through a *extra-information design*, specifying an arbitrary sequence of additional unbiased samples provided to the learner. The learner has no knowledge of, nor control over, this design. This abstraction strictly generalizes feedback-graph models and allows side information to be adversarially timed while remaining stochastically benign.

Related work also considers adversarial or hybrid bandit settings [Arora et al., 2019, Chen et al., 2021, Kong et al., 2022]. In contrast, we study i.i.d. rewards with unbiased side information and show that degradation can still occur. Most feedback-graph models limit observation to one additional sample per action per round. Our setup allows an arbitrary number of samples for any action at any time.

Multi-agent and group-based learning. Prior work on multi-agent and group-based bandits often focuses on minimizing *social regret* or fairness objectives under centralized control [Raghavan et al., 2018, Baek and Farias, 2021]. These works show that UCB-style algorithms can induce unfairness by forcing some users to explore on behalf of others.

We adopt a complementary decentralized perspective, where each group is an independent agent running its own bandit algorithm and statistics from other agents appear as side information that does not directly incur regret for the receiving agent. We show that even in this benign setting, naively injecting shared data into UCB can increase individual regret.

Decentralized multi-agent bandit models consider settings in which multiple agents interact with the environment over the full learning horizon and share information to improve estimation. Typically, each agent runs its own instance of a bandit algorithm, while communication is used to aggregate statistics across agents.

A growing body of work studies decentralized bandits under communication constraints or structured communication graphs [Wang et al., 2019, Agarwal et al., 2022, Zhu and Liu, 2021], as well as extensions with richer assumptions, including joint reward models, heavy-tailed rewards, and robustness to Byzantine agents [Chang et al., 2022, Dubey et al., 2020, Zhu et al., 2024, Dubey and Pentland, 2020]. Across these settings, multiple learning agents collaborate by sharing statitics, variants/adaptations of the UCB paradigm are used alongside alternatives such as arm-elimination methods [Zhang et al., 2025]. Beyond MAB, Optimism-based approaches are also prevalent in online and multi-agent reinforcement learning [Qiu et al., 2022, Lee et al., 2022]. In contrast to these multi-agent setups, our work can be interpreted in a per-agent perspective where statistics from other agents are received as side-information.

2 Setting and Preliminaries

Consider a learner facing a sequential decision-making problem with a finite action set $\mathcal{A} = \{1, 2, \ldots, K\}$ over a horizon of length T. Playing an action (or arm) $i \in \mathcal{A}$ yields an independent reward outcome from the environment (or nature) drawn from an unknown sub-Gaussian distribution r_i with unknown expected value $\mu_i \in [0, 1]$. At each time-step (or round) $t \in [T]$, the learner plays an action $a(t) \in \mathcal{A}$ and observes a reward $y(t) \sim r_{a(t)}$. Let

$$H(t) = \big\{\big(a(u), y(u)\big) : u < t\big\} \tag{1}$$

denote the *interaction **h**istory* up to the beginning of round t, consisting of all observed rewards and the actions played by the learner thus far. Let $\mathcal{H}$ denote the collection of all possible interaction histories $H(t)$s for all $t \in [T]$.

Extra information. At the beginning of each round $t \in [T]$, before playing any action, the learner may additionally be given some extra sample outcomes by the environment generated faithfully from the true distributions of the actions. The learner can use these extra samples to improve his action reward estimates. This extra information is 'free' in the sense that the learner does not incur any regret for observing these sample outcomes of the actions. The quantity of extra information is denoted by a **d**esign vector

$$\mathrm{D}(t) = \big(\mathrm{D}_1(t), \mathrm{D}_2(t), \ldots, \mathrm{D}_K(t)\big) \in \mathbb{Z}^K_{\geq 0},$$

where $\mathrm{D}_i(t)$ denotes the number of outcome samples of action i given by the environment at round t. Concretely, at the beginning of round t, the environment reveals extra-information set

$$m_t = \left\{ \left(i, \{y_{i,c}\}_{c=1}^{\mathrm{D}_i(t)} \right) : i \in \mathcal{A} \right\}, \tag{2}$$

where for each action i, the samples $\{y_{i,c}\}_{c=1}^{\mathrm{D}_i(t)}$ are all drawn from the true arm distribution r_i independently across actions and rounds.

The full-horizon extra information design $\mathrm{D} = \big(\mathrm{D}(1), \ldots, \mathrm{D}(T)\big)$ is exogenous and fixed in advance: it does not depend on (and is not adaptive to) the learner's actions, observations, or any internal randomness; the learner cannot seek (or influence or control) which actions are sampled for extra information. Further, this design D is not known to the learner upfront, and is only noticeable in an online fashion when the actual extra-information samples m_t are revealed at round t. Let $M_t = \bigcup_{u \le t} m_u$ denote the *extra-information history* consisting of all extra-information samples revealed so far to the learner upto the beginning of round t. Let $\mathcal{M}$ denote the collection of all possible extra-information histories M_t for all $t \in [T]$.

Algorithm and Regret. The learner employs an algorithm ALG to select an action to play at each round $t \in [T]$, based on the interaction history H_t and the extra-information history M_t available at the beginning of round t. Formally, the algorithm is a (possibly randomized) mapping from the set of histories to the set of actions $\mathrm{ALG} : \mathcal{H} \times \mathcal{M} \to \Delta(\mathcal{A})$, where $\Delta(\mathcal{A})$ denotes the probability simplex over the action set.

The objective of the learner is to design an algorithm ALG that minimizes the (expected) ***regret***, defined as

$$R(\mathrm{ALG}, (\mathcal{A}, \mathrm{D}), T) \;=\; \sum_{t=1}^{T} \mu_\star \;-\; \mathbb{E}_{\mathcal{A}, \mathrm{ALG}, \mathrm{D}} \left[\mu_{a(t)} \right], \tag{3}$$

where $\mu_\star = \max_{i \in \mathcal{A}} \mu_i$ is the optimal expected reward. The regret benchmarks the learner's expected cumulative reward against that of an oracle that always plays the optimal action. We again note that receiving extra-information samples does not make the learner incur regret, since they are observed without playing an action.

The expectation in Equation (3) is taken over all sources of randomness: (i) any internal randomness of the algorithm ALG, and (ii) the stochasticity of the environment, which governs both the rewards observed from played actions and the samples revealed as extra information under the design D.

The action set and the extra-information design profile together specify the sequential decision-making problem faced by the learner. By a slight abuse of notation, we write $\mathcal{A} = (r_1, \ldots, r_K)$ to denote the collection of reward distributions associated with the actions. With that, we define the problem of *bandit learning with extra information* by the tuple $(\mathcal{A}, \mathrm{D}, T)$, with the learner's goal being to minimize the regret in Equation (3).

When the design satisfies $\mathrm{D}_i(t) = 0$ for all $i \in \mathcal{A}$ and all $t \in [T]$ (denoted compactly by design $\mathrm{D} = \mathbf{0}$), the model reduces to the classical multi-armed bandit (MAB) problem. Thus, our setting strictly generalizes the standard MAB framework.

Non-degradation under extra information. A basic desideratum for learning algorithms—be it with offline or online training—is that giving additional data should not worsen performance. In our setting of bandit learning with extra information, it means consuming the free and unbiased extra-information samples should not increase the learner's expected regret. We formalize this requirement as follows:

Definition 1 (Non-degradation under extra information)**.** An algorithm ALG is said to be *non-degrading under extra information* if providing extra information never increases its expected regret. Formally, for all bandit action sets $\mathcal{A}$, and all extra-information design profiles D, for some threshold time T_0, it holds that

$$R(\mathrm{ALG}, (\mathcal{A}, \mathrm{D}), T) \;\le\; R(\mathrm{ALG}, (\mathcal{A}, \mathbf{0}), T), \tag{4}$$

for all $T > T_0$. Here, $\mathbf{0} = 0^{T \times K}$ denotes the design with no extra information.

We simply say "ALG can degrade" if ALG fails to satisfy Definition 1. We next state our first major result: The UCB algorithm can degrade.

3 UCB can degrade with extra information

A canonical example of a provably efficient stochastic bandit algorithm is the UCB algorithm of Auer et al. [2002] (formally, the UCB1 algorithm) which attains the optimal regret rate. It has become one of the most widely studied and deployed algorithmic paradigms for stochastic bandits.

At a high level, UCB augments empirical mean estimates with optimism-based confidence bonuses and selects the action with the largest upper confidence bound. This balances exploration and exploitation and guarantees sufficient exploration of suboptimal actions in the absence of side information.

Given this design, it is natural to expect UCB to benefit from additional data in the form of extra-information samples. Extra samples sharpen empirical estimates and appear to facilitate faster identification of the optimal action. However, this intuition overlooks a subtle but fundamental feature of UCB: the confidence bonus shrinks with the number of samples, scaling as $\simeq \sqrt{1/n}$. As a result, extra samples can *lower* an action's upper confidence bound—even when those samples come from the optimal action, making it less likely to be played, thereby increasing regret. We have

Theorem 1. UCB is *not* non-degrading under side information.

Defering the proof to Section B, we give an outline here. The result is established by constructing a problem instance where UCB degrades, i.e., incurs more regret with extra information than without it. Consider a two-action stochastic bandit instance with action 1 being uniquely optimal, and a extra-information design D that provides some $s > 0$ samples of only the optimal action at the initial round, and no samples for either actions thereafter:

$$\mathrm{D}(1) = (s, 0), \qquad \mathrm{D}(t) = (0, 0) \quad \text{for all } t \geq 2. \tag{5}$$

The intuition is as follows. By receiving samples of the optimal action upfront, UCB behaves as if it had already explored this optimal action sufficiently. This immediately shrinks the optimal action's confidence interval (or bonus term) and lowers its upper confidence bound. Consequently, UCB algorithm, which plays actions with the largest upper confidence bound, has less incentive to pull the optimal action and instead deterministically reallocates effort toward playing the suboptimal action in order to 'catch up' and reduce its uncertainty about the sub-optimal action. Each such pull incurs regret, even though the algorithm could have safely played the optimal action instead.

In our result, importantly, the side information is unbiased and informative: it consists entirely of samples drawn from the true reward distribution of the optimal action, yet the regret increases. The degradation arises solely from UCB's deterministic optimism-based exploration strategy. By front-loading information about the optimal action, the algorithm suppresses its own optimism for that action and over-explores the suboptimal actions, resulting in strictly higher regret.

3.1 Zero-noise thought experiment

The degradation of UCB is intrinsic to the algorithm, not an artifact of reward noise. To isolate the mechanism, consider a deterministic instance in which each action i yields reward μ_i with probability one, and every extra-information sample for action i is also exactly μ_i. We run UCB and Thompson Sampling faithfully, without informing them that the instance is noiseless.

Effect on UCB. Empirical means equal true means, so the upper confidence bound $B_i(t) = \mu_i + \sqrt{2 \log t / n_i(t)}$, where $n_i(t)$ counts both interaction and extra-information samples. When extra information for action i is given, $n_i(t)$ grows and $B_i(t)$ consequently decreases; indices of other actions are unchanged. Thus, samples of the optimal action (say, action 1) only make it *less* attractive to play: the gap $B_1(\cdot) - B_i(\cdot)$ lowers for all sub-optimal actions $i \neq 1$, and UCB deterministically reallocates pulls to the suboptimal action to reduce its uncertainty. Free samples of the optimal action increase the number of suboptimal pulls, and hence the regret.

We introduce some notation. For a given D, write $s_i := \sum_{t=1}^{T} \mathrm{D}_i(t)$ to be the total extra-information sample count of action i over the course of the time horizon T. We shall quantify the degradation lower bound as a function of s_i total counts.

Theorem 2. Let $\mathcal{A}$ be any deterministic K-armed bandit instance, and D be any extra-information design D. the degradation of UCB is lower bounded as follows:

$$R(\text{ALG}, (\mathcal{A}, \text{D}), T) - R(\text{ALG}, (\mathcal{A}, \mathbf{0}), T) \geq \sum_{i=2}^{K} \frac{2(\alpha \ln T)^{1.5}}{\Delta_i^2} \left(\frac{1}{\sqrt{T}} - \frac{1}{\sqrt{T + s_1}} \right) - s_i \Delta_i - o(1). \tag{6}$$

where $\Delta_i := \mu_\star - \mu_i$ is the sub-optimality gap of action $i \in \mathcal{A}$.

There are three interpretational take-aways from the Theorem.

First, the s_1 samples of optimal action, act as a global spoiler. Because the upper confidence bound of the optimal action is a benchmark that every suboptimal action is compared against, lowering that benchmark unleashes unnecessary exploration across all $K - 1$ suboptimal actions. Thus, the degradation scales linearly with the size of the action set as indicated by the summation.

Second, the degradation is proportional to $1/\Delta_i^2$ which is worse than the standard $1/\Delta$ instance-dependent regret Auer et al. [2002], Lai and Robbins [1985]. This indicates that for actions near optimal, providing extra-information of optimal action will cause more degradation, as the algorithm will play near-optimal actions more, as it is unable to use the optimal action's uncertainty (as extra-information s_1 suppresses it) to filter them out. This is also seen empirically in Figure 1c with higher degradation (the difference between the blue dotted and blue solid curves) observed for lower gaps Δ.

Third, extra infomation of the sub-optimal actions gives a direct linear $s_i \Delta_i$ relief from the degradation, where the algorithm benefits from not having to explore by itself those sub-optimal actions.

Aside, the $o(1)$ slack is independent of time horizon T and extra-information sample counts $(s_i)_i$, and is merely an artifact of our analysis that uses Taylor approximations.

Effect on Thompson Sampling. For ease of exposition, we consider bandit instances $\mathcal{A}$ with gaussian reward distributions. Thompson Sampling, then, maintains a Gaussian prior and posterior distribution about each action, $P_i(t) = \mathcal{N}(\mu_i, 1/n_i(t))$. For $\mu_i > \mu_j$, the sampled difference $\tilde{\mu}_i(t) - \tilde{\mu}_i(t)$ is Gaussian with positive mean $\mu_i - \mu_j$ and variance $1/n_i(t) + 1/n_j(t)$. Extra information about either action strictly reduces this variance while preserving the positive mean, thereby increasing $\mathbb{P}\{\tilde{\mu}_t(i) > \tilde{\mu}_t(j)\}$ and hence the probability of selecting the optimal action.

In short, the same extra information suppresses UCB's optimism for the optimal action while concentrating Thompson Sampling's posterior toward it. We make this contrast formal: there is a minimal instance on which extra information has opposite effects on the two algorithms; the proof is in Section D.

Lemma 1. There exists an instance $(\mathcal{A}, \text{D}, T)$ of the bandit learning with extra information problem such that

$$R(\text{UCB}, (\mathcal{A}, \mathbf{0}), T) < R(\text{UCB}, (\mathcal{A}, \text{D}), T) \quad \text{and}$$
$$R(\text{TS}, (\mathcal{A}, \mathbf{0}), T) > R(\text{TS}, (\mathcal{A}, \text{D}), T).$$

4 Data-monotone algorithms

We strengthen the notion 'non-degrading' algorithm further and show algorithms that satisfy this stronger property. Towards that, we formalize the idea that some extra-information designs are intrinsically more informative than others. Intuitively, receiving additional samples of an action cannot reduce the information available to the learner, and receiving samples earlier in the horizon is at least as informative as receiving them later. Indeed, a learner that is given more samples, or receives them sooner, can always ignore some samples or postpone their use in order to emulate a less informative design. This intuition induces a natural partial order over extra-information designs:

Definition 2 (Dominance of extra-information designs)**.** A extra-information design E is said to *dominate* another design D, written $\text{E} \succeq \text{D}$, if E provides at least as many samples of every action as D does by every point in time. Formally, $\text{E} \succeq \text{D}$ if for all actions $i \in \mathcal{A}$ and all time steps $t \in [T]$, it holds that

$$\sum_{u=1}^{t} \text{D}_u(i) \;\leq\; \sum_{u=1}^{t} \text{E}_u(i). \tag{7}$$

This dominance relation captures both the quantity and the timing of extra-information samples. We can therefore ask whether an algorithm's performance improves monotonically as the available side information becomes more informative:

Definition 3 (Data-monotone)**.** An algorithm ALG is said to be *data-monotone* if for any action set $\mathcal{A}$ and any two extra-information designs $\mathrm{E} \succeq \mathrm{D}$,

$$R(\text{ALG}, (\mathcal{A}, \mathrm{E}), T) \;\leq\; R(\text{ALG}, (\mathcal{A}, \mathrm{D}), T), \tag{8}$$

for all time-horizons $T > T_0$ beyond a threshold T_0.

Note that the design with no extra information, $\mathbf{0} = (\mathrm{D}_t(i) = 0 : \forall i \in \mathcal{A},\ t \in [T])$, is dominated by every extra-information design. Consequently, non-degradation under side information (Definition 1) is recovered as a special case of monotonicity under dominating side information (Definition 3). In particular, any algorithm that is monotone under dominating side information is necessarily non-degrading under side information. We next show that there exist algorithms that satisfy this stronger property next.

4.1 Explore-Then-Commit is monotone

We have seen that UCB can degrade under side information, whereas Thompson Sampling uses extra-information to its benefit in those same problem instances.

This raises a natural question: Is internal randomization necessary for an algorithm to be non-degrading under side information? The answer is no. In particular, simple deterministic algorithms can also satisfy monotonicity.

A canonical example is the Explore-Then-Commit (ETC) algorithm. Recall that ETC explores by pulling each action in a round-robin fashion for a fixed number of rounds $T^{ex} < T$, and then commits to the empirically best action for the remainder of the horizon. With an appropriate choice of T', this strategy achieves regret $\widetilde{O}\left(T^{2/3}\right)$. We show that this algorithm is monotone:

Theorem 3. *The* ETC *algorithm is monotone under dominating side information.*

The proof is deferred to Section E. At a high level, the argument is straightforward: providing dominating extra-information samples—earlier or in greater quantity— can only reduce the variance of the corresponding empirical estimate. Since ETC's decision is made via a single comparison at the end of the exploration phase, reduced variance can only decrease the probability of committing to the suboptimal action.

This behavior contrasts sharply with UCB. For ETC, side information about the optimal action strictly increases the probability of committing to it. For UCB, the same information can instead suppress optimism for the optimal action and induce additional exploration of suboptimal ones. This comparison highlights that degradation is not an inherent consequence of determinism or optimal regret rates, but of how exploration pressure is coupled to accumulated data.

5 Algorithm: Enforcing Non-Degradation

The negative result for UCB algorithm shows that naively incorporating extra information can increase regret. We now ask a more constructive question: Can extra information be incorporated in a way that provably avoids degradation? Our answer is affirmative. We present a meta-algorithm that transforms any reasonable bandit algorithm into one that is non-degrading under extra information for any arbitrary set of actions.

The key observation is that extra information differs from action play in that it provides samples without incurring regret; however, if such samples are fed to a learning algorithm in an unbalanced manner, they may distort its exploration behavior and lead to worse cumulative performance. Our approach is therefore based on two principles: (1) Deferred consumption: extra information is not consumed immediately upon arrival; (2) Balanced consumption: extra information is consumed evenly across actions. This idea shall ensure that extra information acts as 'free exploration' that cannot result in more regret than exploring by playing that action.

Our approach works for all learning algorithms that satisfy:

Algorithm 1 META: Deferred and balanced consumption of extra information

Input: Black-box learning algorithm ALG, action set $\mathcal{A}$, horizon T
Online input: Extra-information samples set s_t as in Equation (2).

1: For all actions $i \in \mathcal{A}$, initialize buffers $S_i \leftarrow \emptyset$ and counters $c_i \leftarrow 0$
2: **for** time-step $t = 1, 2, \ldots, T$ **do**
3: Receive extra-information samples and put in buffers S_i
4: Query the black-box algorithm for an action recommendation: $i \leftarrow \text{ALG}(H_t)$
5: **while** $S_i \neq \emptyset$ **and** $c_i \leq \min_{j \in \mathcal{A}} c_j$ **do**
6: $s \leftarrow$ Remove one buffer sample from S_i
7: Append $(x = i, y = s)$ to history H_t
8: Increment buffer consumption counter c_i
9: Query the black-box algorithm again: $i \leftarrow \text{ALG}(H_t)$
10: **end while**
11: Play action $x_t = i$ on the environment and observe outcome y_t
12: Append (x_t, y_t) to history H_{t+1}
13: **end for**

Property 4 (Better-than-random algorithm)**.** An algorithm ALG is said to be *better than random* if for any action set $\mathcal{A}$ of size K, there exists a threshold time T' such that for all time horizons $T > T'$ and all window lengths $w \geq 1$, the expected regret of ALG over the window $[T : T + w]$ is upper bounded by the regret of playing all actions uniformly at random over the same window. Precisely,

$$R(\text{ALG}, \mathcal{A}, [T : T + w]) \leq \frac{w}{K} \sum_{i \in \mathcal{A}} \Delta_i,$$

where $\Delta_i = \mu_* - \mu_i$ denotes the sub-optimality gap of action i.

This property is mild and asks that an algorithm learn enough by some suitably large time-step T so as to thereafter play the optimal action more often than each sub-optimal action in expectation, i.e., the algorithm performs better than a policy that plays/explores all actions uniformly at random (or in round robin). It is easily seen that this property is satisfied by standard bandit algorithms such as UCB, Thompson Sampling etc. which all have known asymptotic sub-linear regret upper bounds, whereas performing uniform exploration at all times would incur a linear regret.

Algorithm 1 presents the meta-algorithm META, which controls the manner in which extra information is consumed by a black-box learning algorithm ALG.

Meta-algorithm description. The algorithm maintains a buffer of unused extra information samples for each action, as well as counters that track the number of extra information samples consumed per action. At each round t, the extra information samples revealed at that time are first stored in their corresponding buffers and are not immediately consumed (line 3). Using the current history H_t, the meta-algorithm then queries the black-box learner ALG to obtain an action recommendation (line 4). If a extra information sample for the recommended action is available in the buffer, and the total number of previously consumed extra information samples for that action does not exceed the minimum consumption across all actions, the meta-algorithm consumes one such sample. (line 5).

The sample is appended to the history without interacting with the environment, and the black-box algorithm is queried again (line 9). This procedure repeats until either no extra information is available for the recommended action or the balanced-consumption condition is violated. Once no further extra information can be consumed, the meta-algorithm executes the recommended action in the environment, observes the resulting outcome, (incurs regret,) and appends it to the history for the next time-step (line 12). This process continues for the entire time horizon. Deferring the proof to Section F in the interest of space, we have

Theorem 4. *(Enforcing data-monotonicity.)* When run with any learning algorithm ALG that satisfies Definition 4, META is monotone in dominating extra information.

While our META algorithm ensures that regret weakly decreases when extra information increases, a natural question arises: Does the META algorithm make optimal use of the extra information? Does it achieve the maximum regret reduction possible? First, our META is a wrapper algorithm that intrinsically depends on the decision-making of an internal bandit algorithm ALG that can degrade; so, it is unlikely to achieve the maximum regret reduction, and it is to be stressed that the design goal

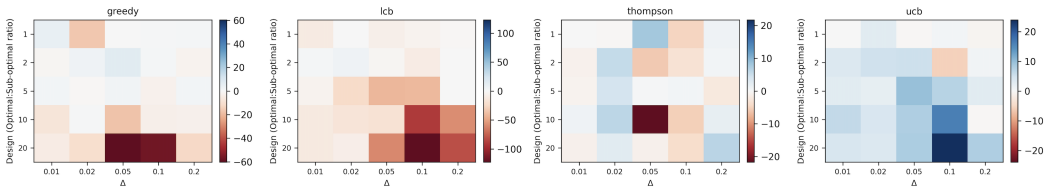


Figure 2: An ablation study of how regret with extra-information is provided, across different algorithms in the different plots. In each, we plot the regret difference between the normal algorithm and that of wrapped in META, plotted across different gaps Δ and different ratios of optimal-to-suboptimal extra-information samples.

of META is towards provably making ALG robust to degradation. Second, quantifying the maximum possible regret reduction with extra information by any possible bandit algorithm is a fundamental information-theoretic question. It is possible that the regret minimality for different structures of extra-information designs are obtained by different bandit algorithms which take advantage of specific structural aspects. We leave this as an interesting direction for future work.

5.1 Ablation study of META.

Denote by META + ALG the execution of our META algorithm with an internal bandit algorithm ALG. Then, for any two extra-information designs D and E s.t. E dominates D (as in Definition 2) and for any bandit problem instance I, for a suitably large T, our results show that $R(\text{META} + \text{ALG}, (I, \text{E}), T) \leq R(\text{META} + \text{ALG}, (I, \text{D}), T)$.

However, there is complementary question that arises: For a specific bandit instance I, and extra-information design D, how is the regret of the learner affected by using the META algorithm on top the original bandit algorithm ALG? Specifically, how does $R(\text{META} + \text{ALG}, (I, \text{D}), T) - R(\text{ALG}, (I, \text{D}), T)$ behave for different I, D, ALG combinations? We plot in Figure 2 an ablation study across these parameters: a plot for each ALG, and designs of extra information across the vertical axis, and sub-optimality gaps Δ across the horizontal axis. These plots help understand how the META algorithm affects the instance-wise regret, while the non-degradation/monotonicity property is guaranteed over dominating extra-information designs.

The META algorithm lowers UCB's regret across 24 out of the 25 cases. The lower rows which correspond to disproportionately high amount of optimal action samples causes the UCB to degrade more, and thus the benefit of using META algorithm is larger, as indicated by the more dark blue-ish mosaic.

For Thompson Sampling, for roughly half the cases, the difference is negative (orange in colour) because Thompson Sampling doesn't inherently degrade with more extra information. So, in those cases, the META algorithm unnecessarily delays the consumption of extra information and increases the regret. A take-away from these plots is that the META algorithm is more suitable for algorithms that are known to degrade, like UCB for example.

For algorithms with no exploration (like Greedy, and the LCB-Lower Confidence Bound algorithms), our meta-algorithm behaves erratically (dark mosaics) as these algorithms violate Definition 4.

6 Conclusion

We show that more data can hurt sequential decision-making. Although UCB achieves optimal regret guarantees, it can incur higher regret with benign side information, unlike Thompson Sampling and Explore-Then-Commit, which benefit in similar settings. We show that the effect of side information depends on how algorithms translate information into exploration. We also propose a meta-algorithm that makes learners monotone in dominating side information. Experiments in collaborative recommendation settings reveal diminishing or negative returns under UCB-style methods, showing that regret optimality alone does not ensure that more data helps.

References

Mridul Agarwal, Vaneet Aggarwal, and Kamyar Azizzadenesheli. Multi-agent multi-armed bandits with limited communication. *Journal of Machine Learning Research*, 23(212):1–24, 2022.

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.

Raman Arora, Teodor Vanislavov Marinov, and Mehryar Mohri. Bandits with feedback graphs and switching costs. *Advances in Neural Information Processing Systems*, 32, 2019.

Alexia Atsidakou, Orestis Papadigenopoulos, Constantine Caramanis, Sujay Sanghavi, and Sanjay Shakkottai. Asymptotically-optimal gaussian bandits with side observations. In *International Conference on Machine Learning*, pages 1057–1077. PMLR, 2022.

Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. Finite-time Analysis of the Multiarmed Bandit Problem. *Machine Learning*, 47(2):235–256, May 2002. ISSN 1573-0565. doi: 10.1023/A:1013689704352. URL https://doi.org/10.1023/A:1013689704352.

Jackie Baek and Vivek Farias. Fair exploration via axiomatic bargaining. *Advances in Neural Information Processing Systems*, 34:22034–22045, 2021.

Swapna Buccapatnam, Atilla Eryilmaz, and Ness B Shroff. Stochastic bandits with side observations on networks. In *The 2014 ACM international conference on Measurement and modeling of computer systems*, pages 289–300, 2014.

William Chang, Mehdi Jafarnia-Jahromi, and Rahul Jain. Online learning for cooperative multi-player multi-armed bandits. In *2022 IEEE 61st Conference on Decision and Control (CDC)*, pages 7248–7253. IEEE, 2022.

Houshuang Chen, Shuai Li, Chihao Zhang, et al. Understanding bandits with graph feedback. *Advances in Neural Information Processing Systems*, 34:24659–24669, 2021.

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.

Corinna Cortes, Giulia DeSalvo, Claudio Gentile, Mehryar Mohri, and Scott Yang. Online learning with sleeping experts and feedback graphs. In *International Conference on Machine Learning*, pages 1370–1378. PMLR, 2019.

Abhimanyu Dubey and Alex Pentland. Private and byzantine-proof cooperative decision-making. In *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems*, pages 357–365, 2020.

Abhimanyu Dubey et al. Cooperative multi-agent bandits with heavy tails. In *International conference on machine learning*, pages 2730–2739. PMLR, 2020.

Yutian Gou, Jinfeng Yi, and Lijun Zhang. Stochastic graphical bandits with heavy-tailed rewards. In *Uncertainty in Artificial Intelligence*, pages 734–744. PMLR, 2023.

Trevor Hastie. The elements of statistical learning: data mining, inference, and prediction, 2009.

Qijia He, Minghan Wang, Xutong Liu, Zhiyong Wang, and Fang Kong. Learning across the gap: Hybrid multi-armed bandits with heterogeneous offline and online data. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Tomáš Kocák, Gergely Neu, and Michal Valko. Online learning with noisy side observations. In *Artificial Intelligence and Statistics*, pages 1186–1194. PMLR, 2016.

Fang Kong, Yichi Zhou, and Shuai Li. Simultaneously learning stochastic and adversarial bandits with general graph feedback. In *International Conference on Machine Learning*, pages 11473–11482. PMLR, 2022.

Tze Leung Lai and Herbert Robbins. Asymptotically efficient adaptive allocation rules. *Advances in applied mathematics*, 6(1):4–22, 1985.

Tor Lattimore and Csaba Szepesvári. *Bandit algorithms*. Cambridge University Press, 2020.

Dongwoo Lee, Yu Zhao, Jun-Bae Seo, and Joohyun Lee. Multi-agent reinforcement learning for a random access game. *IEEE Transactions on Vehicular Technology*, 71(8):9119–9124, 2022.

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.

Shie Mannor and Ohad Shamir. From bandits to experts: On the value of side-observations. *Advances in Neural Information Processing Systems*, 24, 2011.

Bastian Oetomo, R Malinga Perera, Renata Borovica-Gajic, and Benjamin IP Rubinstein. Cutting to the chase with warm-start contextual bandits. *Knowledge and Information Systems*, 65(9): 3533–3565, 2023.

Franco P Preparata and Michael I Shamos. *Computational geometry: an introduction*. Springer Science & Business Media, 2012.

Shuang Qiu, Lingxiao Wang, Chenjia Bai, Zhuoran Yang, and Zhaoran Wang. Contrastive ucb: Provably efficient contrastive self-supervised learning in online reinforcement learning. In *International Conference on Machine Learning*, pages 18168–18210. PMLR, 2022.

Manish Raghavan, Aleksandrs Slivkins, Jennifer Vaughan Wortman, and Zhiwei Steven Wu. The externalities of exploration and how data diversity helps exploitation. In *Conference on Learning Theory*, pages 1724–1738. PMLR, 2018.

Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

Nihal Sharma, Soumya Basu, Karthikeyan Shanmugam, and Sanjay Shakkottai. Warm starting bandits with side information from confounded data. *arXiv preprint arXiv:2002.08405*, 2020.

Nicollas Silva, Thiago Silva, Heitor Werneck, Leonardo Rocha, and Adriano Pereira. User cold-start problem in multi-armed bandits: When the first recommendations guide the user's experience. *ACM Transactions on Recommender Systems*, 1(1):1–24, 2023.

Aleksandrs Slivkins et al. Introduction to multi-armed bandits. *Foundations and Trends® in Machine Learning*, 12(1-2):1–286, 2019.

Richard S Sutton, Andrew G Barto, et al. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

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.

Vladimir N Vapnik. An overview of statistical learning theory. *IEEE transactions on neural networks*, 10(5):988–999, 1999.

Yuanhao Wang, Jiachen Hu, Xiaoyu Chen, and Liwei Wang. Distributed bandit learning: Near-optimal regret with efficient communication. In *International Conference on Learning Representations*, 2019.

Yifan Wu, András György, and Csaba Szepesvári. Online learning with gaussian payoffs and side observations. *Advances in Neural Information Processing Systems*, 28, 2015.

Hong Xie, Haoran Gu, and Zhi Qi. Efficient algorithms for multi-armed bandits with additional feedbacks: Modeling and algorithms. *Information Sciences*, 633:453–468, 2023.

Haoran Zhang, Xuchuang Wang, Hao-Xu Chen, Hao Qiu, Lin Yang, and Yang Gao. Near-optimal regret bounds for federated multi-armed bandits with fully distributed communication. In *The 41st Conference on Uncertainty in Artificial Intelligence*, 2025.

Jingxuan Zhu and Ji Liu. A distributed algorithm for multi-armed bandit with homogeneous rewards over directed graphs. In *2021 American Control Conference (ACC)*, pages 3038–3043. IEEE, 2021.

Jingxuan Zhu, Alec Koppel, Alvaro Velasquez, and Ji Liu. Byzantine-resilient decentralized multi-armed bandits. *Transactions on machine learning research*, 2024.

A Extended Related Work

Due to space constraints in the main paper, in this section, we discuss some additional related work in this problem space of bandit learning with exogenous side information.

Warm-start and Offline data. While not directly related, warm-start bandits are a special case of our setup. From our perspective, these settings can be viewed as special cases of side-information designs in which samples are front-loaded at initialization.

Warm-start and offline-to-online bandits address the use of historical data to mitigate cold-start effects, particularly in recommender systems [Silva et al., 2023, Oetomo et al., 2023, Xie et al., 2023]. Existing approaches typically modify algorithms to explicitly incorporate prior data, for example by adjusting confidence bounds or priors [Sharma et al., 2020, He et al., 2025]. In contrast, we treat the learning algorithm as fixed and ask whether additional unbiased samples are inherently beneficial. Our results show that even in the simplest stochastic setting, this need not hold. Related work also studies offline-to-online bandits under distribution shift [He et al., 2025]; in contrast, we focus on i.i.d. rewards and unbiased side information, and show that degradation can occur even in this benign setting.

B Proof of Theorem 1

Theorem 1. UCB is *not* non-degrading under side information.

Proof. We exhibit an online bandit problem with side information for which UCB violates Equation (4).

Let $\mathcal{A} = \{1, 2\}$ be a two-action problem, where rewards are drawn from distributions $r(1)$ and $r(2)$ with means $\mu(1) > \mu(2)$, so that action 1 is optimal.

Consider a side-information design D that provides $s \geq 1$ samples of the optimal action at the initial round and no further side information thereafter:

$$\mathrm{D}_1 = (s, 0), \qquad \mathrm{D}_t = (0, 0) \quad \text{for all } t \geq 2. \tag{9}$$

To establish the theorem, it suffices to show that UCB incurs strictly larger expected regret under this design than in the absence of side information. This is formalized in the following lemma.

Lemma 2. *For the action set $\mathcal{A}$ and side-information design* D *defined in Equation (9), it holds that*

$$R(\textsc{Ucb}, (\mathcal{A}, \mathrm{D}), T) \;>\; R(\textsc{Ucb}, (\mathcal{A}, \mathbf{0}), T)$$

for all $T > 2$.

Proof. To formalize the argument, we adopt a coupling-based view of the environment. Fix an infinite outcome tape (or film) $F = (F(1), F(2))$, which specifies, for each action $i \in \mathcal{A}$, an infinite sequence of realized rewards $F(i) = (f_1(i), f_2(i), \dots)$ drawn i.i.d. from that action's reward outcome distribution $r(i)$.

A run of UCB—with or without side information—can then be simulated deterministically from the tape F by consuming the next unused tape entry whenever the algorithm either (i) plays an action and receives a reward, or (ii) receives a side-information sample for that action.

Under this construction, all environmental randomness that manifest in the form of stochastic rewards and side-information samples is realized in advance by the tape F. Since UCB itself is deterministic, the entire trajectory of the algorithm is uniquely determined by the tape F, the side-information design D, and the horizon T.

We define the *tape regret* (or film regret) $R^F(\textsc{Ucb}, (\mathcal{A}, \mathrm{D}), T)$ as the pseudo-regret incurred by running UCB for T rounds on instance $(\mathcal{A}, \mathrm{D})$ with action outcomes generated according to the tape F. In particular, for the two-action instance considered here, the tape regret is a nonnegative integer equal to the number of times the suboptimal action is played during this deterministic simulation.

The proof proceeds in two conceptual steps. First, in Lemma 3, we establish a *pointwise* comparison : for every fixed outcome tape F, the regret incurred by UCB without side information is at

most the regret incurred with side-information design D. Second, in Lemma 4, we show that this comparison is *strict* on a set of tapes with strictly positive probability. On these tapes, the early side-information samples of the optimal action are unlucky enough to permanently suppress its UCB index, causing UCB to play the suboptimal action for the entire horizon under design D, while the no-side-information run eventually plays the optimal action. Together, these two claims imply that expected regret is strictly larger under D.

We show that for every fixed tape F, the regret incurred by UCB without side information is no larger than the regret incurred when side information is provided according to the design D. This pointwise comparison is captured by the following claim:

Claim 3 (Tape regret inequality)**.** *For every outcome tape F,*

$$R^F(\text{UCB}, (\mathcal{A}, \mathbf{0}), T) \;\leq\; R^F(\text{UCB}, (\mathcal{A}, \text{D}), T).$$

Proof. In this simulation, the s side-information samples of the optimal action—namely $f_1(1), f_2(1), \ldots, f_s(1)$—are consumed from the tape at the very beginning of the simulation. As a result, by the end of T decision rounds, a total of $T + s$ samples have been consumed from the two tapes $F(1)$ and $F(2)$ combined.

Let $n_2^{\text{D}} \leq T$ denote the total number of samples consumed from the suboptimal action's tape $F(2)$. Because the design D provides no side information for action 2, each such sample must correspond to an actual pull of the suboptimal action. Hence, the tape regret under design D is

$$R^F(\text{UCB}, (\mathcal{A}, \text{D}), T) = n_2^{\text{D}}. \tag{10}$$

We now compare this with a run of UCB for T decision rounds with no side information (design $\mathbf{0}$), using the same tape realization F. We show that, in this run, the suboptimal action can be played at most n_2^{D} times.

We distinguish two cases based on the trajectory induced by the run on $(\mathcal{A}, \text{D})$ over T rounds.

Case 1: The optimal action is never played. In this case, the suboptimal action is played in every round, so $n_2^{\text{D}} = T$. Trivially,

$$R^F(\text{UCB}, (\mathcal{A}, \mathbf{0}), T) \leq T = n_2^{\text{D}}. \tag{11}$$

Case 2: The optimal action is played at least once. Let $n_1^{\text{D}} = T - n_2^{\text{D}} > 0$ denote the number of times the optimal action is played under design D. In this case, the run of UCB has effectively progressed beyond the initial side-information phase, after which the evolution of the algorithm proceeds in a standard manner.

We argue by the desired result contradiction. Suppose that, when UCB is run with no side information on the same tape F, the suboptimal action is played for a $(n_2^{\text{D}} + 1)$-th time at some round $t' \leq T$.

Under design D, the run consumes exactly $s + n_1^{\text{D}} = s + T - n_2^{\text{D}}$ samples from the optimal action's tape $F(1)$, and n_2^{D} samples from the suboptimal action's tape $F(2)$.

Since UCB is deterministic given its observations, we have in the run with design $\mathbf{0}$ as well that before the $n_2^{\text{D}} + 1$-th sample from $F(2)$ is consumed, to match the same state, $s + T - n_2^{\text{D}}$ samples from $F(1)$ must have been consumed.

However, in the absence of side-information, all those samples are consumed exclusively via actual action pulls. Therefore, to consume $n_2^{\text{D}} + 1$ samples from $F(2)$ and $s + T - n_2^{\text{D}}$ samples from $F(1)$ via action pulls alone, the algorithm must have run for at least $(n_2^{\text{D}} + 1) + (s + T - n_2^{\text{D}}) = s + T + 1$ rounds. This contradicts the assumption that $t' \leq T$.

We conclude that, in the run without side information, the suboptimal action is played at most n_2^{D} times.

Consequently, from both the cases above, we have $R^F(\text{UCB}, (\mathcal{A}, 0), T) = n_2^{\text{D}} \leq R^F(\text{UCB}, (\mathcal{A}, \text{D}), T)$, completing the proof of the Claim. □

It remains to show that the regret inequality is strict on a set of tapes with strictly positive probability.

Fix $m := \sqrt{\frac{2\log(T+s)}{s}}$, and define the set of *unlucky tapes*

$$\mathcal{F}^{\varepsilon} := \left\{ F \in \mathcal{F} : \left(\frac{1}{s} \sum_{u=1}^{s} f_u(1) \leq \mu(2) - m \right) \wedge \left(\forall t \in [T], f_t(2) > \mu(2) \right) \right\}. \tag{12}$$

That is, the average of the first s samples of the optimal action are unusually pessimistic, while all the samples of the sub-optimal action exceed its expected value. For gaussian reward distributions, $\mathcal{F}^{\varepsilon}$ has strictly positive probability, $\mathbb{P}_{\mathcal{A}}\{\mathcal{F}^{\varepsilon}\} > 0$.

Claim 4. *For any unlucky tape $F \in \mathcal{F}^{\varepsilon}$, $R^F(\text{UCB}, (\mathcal{A}, \mathbf{0}), T) < R^F(\text{UCB}, (\mathcal{A}, \text{D}), T)$.*

Proof. We first show that, for any unlucky tape $F \in \mathcal{F}^{\varepsilon}$, the run of UCB with side-information design D never selects/plays the optimal action over the horizon T.

We show this by induction over time-step $t \in [T]$.

Hypothesis. Let $H(t)$ denote the event that action 1 has not been selected in the first t decision rounds. We prove that $H(t)$ holds for all $t \in \{0, 1, \ldots, T\}$.

Base Case ($t = 1$). After consuming the s side-information samples of action 1 from the tape F, we have empirical mean $\widehat{\mu}_{t=0}(1) \leq \mu(2) - m$ and and its upper confidence bound $\text{UCB}_{t=0}(1) = \mu(2) - m + \sqrt{\frac{2\log(0+s)}{s}} \leq \mu(2)$.

Action 2 has not yet been sampled, so its upper confidence bound index is strictly larger, and UCB selects action 2 at the first round. Hence $H(1)$ holds.

Inductive step. Assume $H(t)$ is true for some $t < T$, i.e., action 1 has not been selected in the first t rounds. Then the empirical mean and sample count of action 1 remain unchanged: $\widehat{\mu}_t(1) = \widehat{\mu}_0(1) \leq \mu(2) - m$ and its upper confidence bound is

$$\text{UCB}_t(1) = \mu(2) - m + \sqrt{\frac{2\log(t+s)}{s}} \leq \mu(2), \tag{13}$$

where the last inequality follows from $t < T$ and the definition of m.

By contrast, action 2 has been selected in each of the first t rounds. Since $f_u(2) > \mu(2)$ for all u on tapes in $\mathcal{F}^{\varepsilon}$, we have

$$\text{UCB}_t(2) = \widehat{\mu}_t(2) + \sqrt{\frac{2\log(t+s)}{t}} \overset{\text{(a)}}{>} \mu(2), \tag{14}$$

where (a) is due to the tape values obeying $f_u(2) > \mu(2)$ for all $u \in [T]$ as in Equation (12).

Combining Equations (13) and (14), we conclude that $\text{UCB}_t(2) > \text{UCB}_t(1)$, so action 2 is selected at round $t + 1$. Thus $H(t + 1)$ holds.

By mathematical induction, action 1 is never selected over the horizon T, and hence $R^F(\text{UCB}, (\mathcal{A}, \text{D}), T) = T$.

No-side-information run. In contrast, when UCB is run without side-information, the algorithm must sample each action at least once to initialize its estimates. Therefore, action 1 is selected at least once, implying $R^F(\text{UCB}, (\mathcal{A}, \mathbf{0}), T) < T$.

Combining the two conclusions yields the claim. □

Taking expectation over the random tape realization F, drawn according to the product distribution, say $P_{\mathcal{A}}$, induced by the reward distributions $r(1)$ and $r(2)$, we obtain

$$R(\text{UCB}, (\mathcal{A}, \text{D}), T) = \mathbb{E}\left[R^F(\text{UCB}, (\mathcal{A}, \text{D}), T)\right] \overset{\text{(b)}}{>} \mathbb{E}\left[R^F(\text{UCB}, (\mathcal{A}, \mathbf{0}), T)\right] = R(\text{UCB}, (\mathcal{A}, \mathbf{0}), T), \tag{15}$$

where (b) follows from the tape-wise inequality in Lemma 3 together with the strict inequality on a set of tapes of strictly positive probability established in Lemma 4. Finally, Equation (15) shows the Lemma. □

This completes the proof of the theorem. While the argument is demonstrated using a specific counter-example of a two-action instance and a particular side-information design D, similar phenomena can arise in settings with larger action sets or with alternative designs in which side-information is revealed gradually over time. □

C Proof of Theorem 2

For a given problem instance $(\mathcal{A}, \mathrm{D}, T)$, recall that the UCB algorithm maintains at time $t \in [T]$ an upper confidence **b**ound/index/envelope) $B_i^{\mathrm{D}}(t)$ for each action $i \in \mathcal{A}$ and plays the action $a(t) = \arg\max_{i \in \mathcal{A}} B_i^{\mathrm{D}}(t)$.

Here, the upper confidence bound is given by $B_i^{\mathrm{D}}(t) := \hat{\mu_i}(t) + \sqrt{\frac{\alpha \ln T}{n_i^{\mathrm{D}}(t) + s_i^{\mathrm{D}}(t)}}$, where $n_i^{\mathrm{D}}(t)$ is the number of times UCB plays action i upto time t when side-information is provided as per design D, and $s_i^{\mathrm{D}}(t)$ is the number of side-information samples of action i provided upto time t as per D. When there is no side-information, i.e., design is $\mathbf{0}$, we have $s_i^{\mathbf{0}}(t) = 0$ for all t, i.

Theorem 2. Let $\mathcal{A}$ be any deterministic K-armed bandit instance, and D be any extra-information design D. the degradation of UCB is lower bounded as follows:

$$R(\textsc{Alg}, (\mathcal{A}, \mathrm{D}), T) - R(\textsc{Alg}, (\mathcal{A}, \mathbf{0}), T) \geq \sum_{i=2}^{K} \frac{2(\alpha \ln T)^{1.5}}{\Delta_i^2} \left(\frac{1}{\sqrt{T}} - \frac{1}{\sqrt{T + s_1}} \right) - s_i \Delta_i - o(1). \tag{6}$$

where $\Delta_i := \mu_\star - \mu_i$ is the sub-optimality gap of action $i \in \mathcal{A}$.

Proof. To quantify the degradation of UCB because of side-information D and show the inequality in Equation (6), we need to upper bound the regret with side-information D and lower bound the regret when there is no side-information. For this, we lower bound the number of times a sub-optimal action is played with D in Lemma 5, and upper bound the number of times a sub-optimal action is played with no side-information $\mathbf{0}$ in Lemma 6. Finally, we use these to show Equation (6).

Claim 5 (Lower bound of sub-optimal action play in D)**.** *When* UCB *is run on* $(\mathcal{A}, \mathrm{D}, T)$ *for a suitable* T*, the number of times a sub-optimal action* i *is played is lower bounded as follows:*

$$n_i^{\mathrm{D}}(T) \geq \frac{\alpha \ln T}{\Delta_i^2} \left(1 - \frac{2}{\Delta_i} \sqrt{\frac{\alpha \ln T}{s_1 + n_1^{\mathrm{D}}(T)}} \right) - s_i. \tag{16}$$

Proof. Let T be a time-step in which optimal action 1 is played. Then, its UCB is higher than that of all the other actions. Thus, for any sub-optimal action i, we have

$$\begin{aligned}
& \mu_i + \sqrt{\frac{\alpha \ln T}{s_i + n_i^{\mathrm{D}}(T)}} \leq \mu_1 + \sqrt{\frac{\alpha \ln T}{s_1 + n_1^{\mathrm{D}}(T)}} \\
\iff \quad & \frac{\alpha \ln T}{s_i + n_i^{\mathrm{D}}(T)} \leq \left(\Delta_i + \sqrt{\frac{\alpha \ln T}{s_1 + n_1^{\mathrm{D}}(T)}} \right)^2 \\
\iff \quad & n_i^{\mathrm{D}}(T) \geq \frac{\alpha \ln T}{\left(\Delta_i + \sqrt{\frac{\alpha \ln T}{s_1 + n_1^{\mathrm{D}}(T)}} \right)^2} - s_i
\end{aligned} \tag{17}$$

Further simplifying Equation (17), we have

$$n_i^{\mathrm{D}}(T) \geq \frac{\alpha \ln T}{\Delta_i^2 \left(1 + \frac{1}{\Delta_i}\sqrt{\frac{\alpha \ln T}{s_1 + n_1^{\mathrm{D}}(T)}}\right)^2} - s_i$$
$$\overset{\text{(a)}}{\geq} \frac{\alpha \ln T}{\Delta_i^2}\left(1 - \frac{2}{\Delta_i}\sqrt{\frac{\alpha \ln T}{s_1 + n_1^{\mathrm{D}}(T)}}\right) - s_i,$$

where (a) uses a first-order Taylor approximation $(1+x)^{-2} \geq 1 - 2x$ for $x > 0$ with $x = \frac{1}{\Delta_i}\sqrt{\frac{\alpha \ln T}{s_1 + n_1^{\mathrm{D}}(T)}}$. □

Claim 6 (Upper bound of sub-optimal action play in $\mathbf{0}$)**.** *When* UCB *is run on* $(\mathcal{A}, \mathbf{0}, T)$*, i.e., without any side-information, the number of times a sub-optimal action* i *is played is upper bounded as*

$$n_i^{\mathbf{0}}(T) \leq \frac{\alpha \ln T}{\Delta_i^2}\left(1 - \frac{2}{\Delta_i}\sqrt{\frac{\alpha \ln T}{n_1^{\mathrm{D}}(T)}} + \frac{3}{\Delta_i^2}\frac{\alpha \ln T}{n_1^{\mathbf{0}}(T)}\right). \tag{18}$$

Proof. Let T_i be the last time-step at which the sub-optimal action i is played. Then, its upper confidence bound is higher than that of the optimal action 1. Thus, for any sub-optimal action i, we have

$$\mu_i + \sqrt{\frac{\alpha \ln T}{n_i^{\mathbf{0}}(T_i)}} \geq \mu_1 + \sqrt{\frac{\alpha \ln T}{n_1^{\mathbf{0}}(T_i)}}$$
$$\iff \frac{\alpha \ln T}{n_i^{\mathbf{0}}(T_i)} \geq \left(\Delta_i + \sqrt{\frac{\alpha \ln T}{n_1^{\mathbf{0}}(T_i)}}\right)^2$$
$$\iff n_i^{\mathbf{0}}(T_i) \leq \frac{\alpha \ln T}{\left(\Delta_i + \sqrt{\frac{\alpha \ln T}{n_1^{\mathbf{0}}(T_i)}}\right)^2}$$
$$\overset{\text{(b)}}{\implies} n_i^{\mathbf{0}}(T) \leq \frac{\alpha \ln T}{\left(\Delta_i + \sqrt{\frac{\alpha \ln T}{n_1^{\mathbf{0}}(T)}}\right)^2}, \tag{19}$$

where (b) uses the facts $n_i^{\mathbf{0}}(T) = n_i^{\mathbf{0}}(T_i)$ which holds by definition of T_i, and $n_1^{\mathbf{0}}(T_i) \leq n_1^{\mathbf{0}}(T)$ which holds by monotonicity of the play count. Further simplifying Equation (19), we have

$$n_i^{\mathrm{D}}(T) \leq \frac{\alpha \ln T}{\Delta_i^2\left(1 + \frac{1}{\Delta_i}\sqrt{\frac{\alpha \ln T}{n_1^{\mathbf{0}}(T)}}\right)^2}$$
$$\overset{\text{(c)}}{\leq} \frac{\alpha \ln T}{\Delta_i^2}\left(1 - \frac{2}{\Delta_i}\sqrt{\frac{\alpha \ln T}{n_1^{\mathbf{0}}(T)}} + \frac{3}{\Delta_i^2}\frac{\alpha \ln T}{n_1^{\mathbf{0}}(T)}\right),$$

where (c) uses a second-order Taylor approximation $(1+x)^{-2} \leq 1 - 2x + 3x^2$ for $x > 0$ with $x = \frac{1}{\Delta_i}\sqrt{\frac{\alpha \ln T}{n_1^{\mathbf{0}}(T)}}$. □

Decomposing the regret of the algorithm as the sum of the number of times each sub-optimal action is played, and using its lower and upper bounds in Lemmas 5 and 6, we quantify the degradation of UCB as

$$R(\text{UCB}, (\mathcal{A}, \mathrm{D}), T) - R(\text{UCB}, (\mathcal{A}, \mathbf{0}), T) = \sum_{i=2}^{K} \Delta_i \times \left(n_i^{\mathrm{D}}(T) - n_i^{\mathbf{0}}(T)\right)$$
$$\overset{\text{(d)}}{\geq} \sum_{i=2}^{K} \Delta_i \times \left(\frac{\alpha \ln T}{\Delta_i^2}\left(\frac{2}{\Delta_i}\left[\sqrt{\frac{\alpha \ln T}{n_1^{\mathbf{0}}(T)}} - \sqrt{\frac{\alpha \ln T}{s_1 + n_1^{\mathrm{D}}(T)}}\right] - \frac{3}{\Delta_i^2}\frac{\alpha \ln T}{n_1^{\mathbf{0}}(T)}\right) - s_i\right)$$
$$= \sum_{i=2}^{K} \frac{2(\alpha \ln T)^{1.5}}{\Delta_i^2}\left[\frac{1}{\sqrt{n_1^{\mathbf{0}}(T)}} - \frac{1}{\sqrt{s_1 + n_1^{\mathrm{D}}(T)}}\right] - \Delta_i\, s_i - \frac{3(\alpha \ln T)^2}{\Delta_i^3\, n_1^{\mathbf{0}}(T)}.$$

Here, (d) uses Lemmas 5 and 6. □

D One-shot comparison of UCB versus Thompson Sampling.

This section presents a minimal example illustrating a qualitative difference between UCB and Thompson Sampling in how they respond to side-information. In this example, we show that providing a single extra sample of the optimal action decreases UCB's probability of selecting that optimal action in the final round, while the same side-information increases this probability for Thompson Sampling. The example serves to contrast the distinct ways in which these algorithms incorporate side-information.

Lemma 1. There exists an instance $(\mathcal{A}, \mathrm{D}, T)$ of the bandit learning with extra information problem such that

$$R(\textsc{Ucb}, (\mathcal{A}, \mathbf{0}), T) < R(\textsc{Ucb}, (\mathcal{A}, \mathrm{D}), T) \quad \text{and}$$
$$R(\textsc{Ts}, (\mathcal{A}, \mathbf{0}), T) > R(\textsc{Ts}, (\mathcal{A}, \mathrm{D}), T).$$

The instance setup and the following Lemma 7 complete the proof of the Theorem.

Instance setup. Let time horizon $T = 3$ and two actions $\mathcal{A} = \{1, 2\}$. The rewards are Gaussian with unit variance and means

$$\mu = \left(\tfrac{1}{2} + \Delta,\ \tfrac{1}{2}\right),$$

for some $\Delta \in (0, \frac{1}{2}]$ so that action 1 is optimal. We consider the side-information design D defined by

$$\mathrm{D}_t = (0, 0) \text{ for } t = 1, 2, \qquad \mathrm{D}_3 = (1, 0),$$

so that no side-information is provided initially, and a single additional sample of the optimal action is revealed at the beginning of the final time-step before the final action is taken.

Thompson Sampling (in short, TS) is initialized with an improper flat prior belief $P_0(i) = \mathcal{N}(0, \infty)$ for $i \in \mathcal{A}$. Both algorithms pull each action once in the first two rounds in order to initialize their estimates and proper beliefs.

Lemma 7. *When the* UCB *and Thompson Sampling (*TS*) algorithms run on with action set $\mathcal{A}$ with and without side-information* D *for horizon $T = 3$, their expected regrets obey:*

$$R(\textsc{Ucb}, (\mathcal{A}, \mathbf{0}), T) \leq R(\textsc{Ucb}, (\mathcal{A}, \mathrm{D}), T) \quad \textit{and}$$
$$R(\textsc{Ts}, (\mathcal{A}, \mathbf{0}), T) \geq R(\textsc{Ts}, (\mathcal{A}, \mathrm{D}), T).$$

Proof. The Lemma is shown from the conclusions of Sections D.1 and D.2 by explicitly working out the probabilities. □

We write $\mathbb{P}_{\mathrm{D}}\{\cdot\}$ for probabilities induced by running the algorithm under design D, and $\mathbb{P}_{\mathbf{0}}\{\cdot\}$ for the no-side-information design. We write $\mathcal{N}(\rho, \sigma)$ to denote the guassian distribution with mean ρ and variance σ, and write $\eta(\rho, \sigma)$ to denote a random variable that is distributed as $\mathcal{N}(\rho, \sigma)$.

After two rounds, both algorithms have observed exactly one sample from each action. Thus,

$$\widehat{\mu}_2(1) \sim \mathcal{N}(\mu(1), 1), \qquad \widehat{\mu}_2(2) \sim \mathcal{N}(\mu(2), 1), \tag{20}$$

and the joint distribution of observations up to $t = 2$ is identical under D and $\mathbf{0}$. (The side-informations only differ at $t = 3$.)

D.1 UCB trajectory.

Denote $r(N, n) = \sqrt{\frac{2 \ln N}{n}}$ be the confidence radius of an action where where n is the number of samples of an action and N is the total number of samples observed so far over all actions.

No side-information. At the final step $t = 3$, both actions have one sample. The probability of playing the optimal action 1 is

$$\begin{aligned}\mathbb{P}_{\mathbf{0}}\{x_3 = 1\} &= \mathbb{P}_{\mathbf{0}}\{\mathrm{UCB}_3(1) > \mathrm{UCB}_3(2)\} = \mathbb{P}_{\mathbf{0}}\{\widehat{\mu}_2(1) + r(2,1) > \widehat{\mu}_2(2) + r(2,1)\} \\ &= \mathbb{P}_{\mathbf{0}}\{\widehat{\mu}_2(1) > \widehat{\mu}_2(2)\} = \mathbb{P}\{\eta(\mu(1),1) - \eta(\mu(2),1)\} \overset{(a)}{=} \mathbb{P}\{\eta(\Delta,2) > 0\} \\ &\overset{(b)}{=} \Phi(\Delta/\sqrt{2}) \geq \Phi(0.7 \times \Delta), \end{aligned} \tag{21}$$

where (a) uses the independence of action rewards, and thus the estimates, and (b) uses Φ as the cumulative density function of the standard normal distribution.

With side-information. Under design D, action 1 receives an additional sample before the final round. Thus, its empirical mean deviates from Equation (20) to become

$$\widehat{\mu}_2(1) \sim \mathcal{N}(\mu(2), 1/2)$$

having a tigher variance. Then, at $t = 3$, we have $n(1) = 2$ and $n(2) = 1$ yielding

$$\begin{aligned}\mathbb{P}_{\mathrm{D}}\{x_3 = 1\} =& \mathbb{P}_{\mathrm{D}}\{\mathrm{UCB}_3(1) > \mathrm{UCB}_3(2)\} = \mathbb{P}_{\mathrm{D}}\{\widehat{\mu}_2(1) + r(3,2) > \widehat{\mu}_2(2) + r(3,1)\} \\ =& \mathbb{P}_{\mathrm{D}}\{\widehat{\mu}_2(1) - \widehat{\mu}_2(2) > r(3,1) - r(3,2)\} = \mathbb{P}\left\{\eta(\mu(1),1/2) - \eta(\mu(1),1) > \sqrt{2\ln 3}\left(1 - \tfrac{1}{\sqrt{2}}\right)\right\} \\ \leq& \mathbb{P}\{\eta(\Delta, 3/2) > 0.44\} \leq \Phi\left((\Delta - 0.35)/\sqrt{3/2}\right) \leq \Phi(0.82(\Delta - 0.35)). \end{aligned} \tag{22}$$

Combining Equations (21) and (22), we have that

$$\mathbb{P}_{\mathrm{D}}\{x_3 = 1\} \leq \Phi(0.82(\Delta - 0.35)) \overset{(c)}{\leq} \Phi(0.7\Delta) \leq \mathbb{P}_{\mathbf{0}}\{x_3 = 1\}, \tag{23}$$

where (c) holds for all $\Delta < 2.92$. So, providing side-information decreases the probability that UCB selects the optimal action at the final round.

D.2 Thompson Sampling trajectory

Thompson Sampling maintains Gaussian posteriors

$$P_t(i) = \mathcal{N}(\widehat{\mu}_t(i), 1/n_t(i)),$$

where $n_t(i)$ counts all observations of action i, including side-information. At each round t, it independently samples $\tilde{\mu}_t(i) \sim P_{t-1}(i)$ for every i, and plays the action with the largest sample, $\arg\max_{i \in \mathcal{A}} \tilde{\mu}_t(i)$.

After two rounds of action play, both actions have one observation, so

$$P_2(1) = \mathcal{N}(\widehat{\mu}_2(1), 1) \qquad \text{and } P_2(2) = \mathcal{N}(\widehat{\mu}_2(2), 1)$$

Marginally, using Equation (20),

$$P_2(1) \sim \mathcal{N}(\mu(1), 2) \qquad \text{and } P_2(2) \sim \mathcal{N}(\mu(2), 2). \tag{24}$$

No side-information. The probability of playing the optimal action at the final step $t = 3$ is

$$\begin{aligned}\mathbb{P}_{\mathbf{0}}\{x_3 = 1\} = \mathbb{P}_{\mathbf{0}}\{\tilde{\mu}_t(1) > \tilde{\mu}_t(2)\} &\overset{(d)}{=} \mathbb{P}\{\eta(\mu(1),2) > \eta(\mu(2),2)\} \\ &\overset{(e)}{=} \mathbb{P}\{\eta(\mu(1) - \mu(2), 4) > 0\} = \mathbb{P}\{\eta(\Delta, 4) > 0\} = \Phi(\Delta/2). \end{aligned} \tag{25}$$

Here, (d) uses Equation (24), and (e) uses the independence of the posterior samples $\tilde{\mu}_\cdot(\cdot)$. Here, Φ denotes the c.d.f. of the standard normal distribution.

With side-information. Under design D, action 1 has two observations while action 2 has one. Thus,

$$P_2(1) = \mathcal{N}\left(\widehat{\mu}_2(1), 1/2\right) \qquad \text{and } P_2(2) = \mathcal{N}\left(\widehat{\mu}_2(2), 1\right).$$

With the empirical means themselves distributed as $\widehat{\mu}_2(1) = \mathcal{N}\left(\mu(1), 1/2\right)$ (with one sample from side-information, and one from action play) and $\widehat{\mu}_2(2) = \mathcal{N}\left(\mu(2), 1\right)$ (with one sample from action play), the distribution about the posterior belief distributions obey

$$P_2(1) \sim \mathcal{N}\left(\mu(1), 1\right) \qquad \text{and } P_2(2) \sim \mathcal{N}\left(\mu(2), 2\right). \tag{26}$$

Then, the probability of playing the optimal action at the final step $t = 3$ is

$$\begin{aligned}\mathbb{P}_{\mathrm{D}}\left\{x_3 = 1\right\} = \mathbb{P}_{\mathrm{D}}\left\{\tilde{\mu}_t(1) > \tilde{\mu}_t(2)\right\} &\overset{\text{(f)}}{=} \mathbb{P}\left\{\eta(\mu(1), 1) > \eta(\mu(2), 2)\right\} \\ &\overset{\text{(g)}}{=} \mathbb{P}\left\{\eta(\mu(1) - \mu(2), 3) > 0\right\} = \mathbb{P}\left\{\eta(\Delta, 3) > 0\right\} = \Phi(\Delta/\sqrt{3}).\end{aligned} \tag{27}$$

Here, (f) uses Equation (26), and (g) uses the independence of the posterior samples $\tilde{\mu}_t(\cdot)$.

Thus, from Equations (25) and (27), we have that $\mathbb{P}_{\mathrm{D}}\left\{x_3 = 1\right\} > \mathbb{P}_{\mathbf{0}}\left\{x_3 = 1\right\}$. we see that side-information increases the probability that Thompson Sampling selects the optimal action in the final round.

E Proof of Theorem 3

Algorithmic Description and Regret Decomposition. The Explore-Then-Commit (ETC) algorithm proceeds in two phases: an exploration phase of length $T' < T$ where all actions are uniformly played in a round-robin fashion. Let $\widehat{\mu}(i)$ be the empirical reward mean of action i at the end of this exploration phase from both the action play outcomes and side-information samples. Then, the algorithm chooses (or 'commits' to) the empirical best action $\widehat{i_\star} = \arg\max_{i \in \mathcal{A}} \widehat{\mu}(i)$ and plays this $\widehat{i_\star}$ for all rounds $T' + 1, \ldots, T$ thereafter.

The regret of ETC naturally decomposes into that of these two phases: the regret during the exploration phase and the regret during commit phase after that.

During the exploration phase, ETC deterministically plays all actions in a round-robin fashion for the first $T' < T$ rounds. This phase is not adaptive, i.e., the choice of actions played does not depend on the action outcomes/rewards and side-information sample values received. Hence the regrets incurred in the exploration phase under instances $(\mathcal{A}, \mathrm{D})$ and $(\mathcal{A}, \mathrm{E})$ are identical.

E.1 Regret inequality in Commit Phase

To show the Theorem, what remains to be shown is that the regrets in the commit phase $[T' + 1, T]$ obey

$$R(\textsc{Etc}, (\mathcal{A}, \mathrm{E}), [T' + 1, T]) \leq R(\textsc{Etc}, (\mathcal{A}, \mathrm{D}), [T' + 1, T]), \tag{28}$$

where side-information E dominates D (i.e., $\mathrm{E} \succeq \mathrm{D}$ as in Definition 2).

We shall show this Equation (28) in two steps:

1. First, we show in Lemma 8 that the action reward mean estimates in instance $(\mathcal{A}, \mathrm{E})$ with richer/dominating side-information 'more accurate' than the estimate in instance $(\mathcal{A}, \mathrm{D})$.
2. Second, with the estimates behaving as such, we show in Lemma 9 that the probability of committing to a sub-optimal action in the commit phase is 'lesser' when the learner has richer side-information.

We setup some notations. Write $\mu = (\mu(1), \mu(2), \ldots, \mu(K)) \in \mathbb{R}^K$ to be the list/vector of reward means of all actions in $\mathcal{A}$, and analogously define $\widehat{\mu} = (\widehat{\mu}(1), \ldots, \widehat{\mu}(K))$ to be the vector of empirical reward means. We characterize the statistical properties of $\widehat{\mu}$ under the two problem instances corresponding to the two side-information designs D, E.

We assume, as elsewhere in the paper, unit-variance Gaussian reward distributions, so that all reward and side-information samples of action i are i.i.d. draws from $\mathcal{N}(\mu(i), 1)$. Let $n(i)$ denote the number of times ETC plays action i during the (round-robin) exploration phase; since this phase is deterministic and unaffected by side information, $n(i)$ is the same under D and E. Let $s_{\mathrm{D}}(i) := \sum_{u=1}^{T'} \mathrm{D}_u(i)$ and $s_{\mathrm{E}}(i) := \sum_{u=1}^{T'} \mathrm{E}_u(i)$ denote the number of side-information samples of action i revealed by the end of the exploration phase under designs D and E respectively; dominance $\mathrm{E} \succeq \mathrm{D}$ (Equation (7)) gives $s_{\mathrm{E}}(i) \geq s_{\mathrm{D}}(i)$ for every action i.

Claim 8 (Dominant side-information $\Rightarrow$ Tighter estimate)**.** *The estimate $\widehat{\mu}$ obeys the gaussian distribution with expectation and variance as follows:*

$$\widehat{\mu} \sim_{\mathrm{D}} \mathcal{N}(\mu, \Sigma_{\mathrm{D}}) \text{ and } \widehat{\mu} \sim_{\mathrm{E}} \mathcal{N}(\mu, \Sigma_{\mathrm{E}}), \text{ with } \Sigma_{\mathrm{E}} \preceq \Sigma_{\mathrm{D}},$$

where $\Sigma_{\mathrm{D}} = \mathrm{diag}\left(\frac{1}{n(1)+s_{\mathrm{D}}(1)}, \ldots, \frac{1}{n(K)+s_{\mathrm{D}}(K)}\right)$*, and similarly for* Σ_{E}.

Proof. Since the $n(i) + s_{\mathrm{D}}(i)$ (resp. $n(i) + s_{\mathrm{E}}(i)$) samples of action i pooled by the commit time are i.i.d. draws from $\mathcal{N}(\mu(i), 1)$, and are independent across actions, the empirical mean vector $\widehat{\mu}$ is gaussian with mean μ and the stated diagonal covariance under each design. Since $s_{\mathrm{E}}(i) \geq s_{\mathrm{D}}(i)$ for every action i, we have $n(i) + s_{\mathrm{E}}(i) \geq n(i) + s_{\mathrm{D}}(i)$, so each diagonal entry of Σ_{E} is at most the corresponding entry of Σ_{D}; as both covariances are diagonal, this gives $\Sigma_{\mathrm{E}} \preceq \Sigma_{\mathrm{D}}$. □

Without loss of generality, assume that actions in $\mathcal{A}$ are numbered in decreasing optimality, i.e., $\mu(1) > \mu(2) > \cdots > \mu(K)$.

Let $\mathbb{P}_{\mathrm{D}}\{\cdot\}$ and $\mathbb{P}_{\mathrm{E}}\{\cdot\}$ be the probability measures of the algorithmic trajectory (or history) induced by running ETC with on problem instances $(\mathcal{A}, \mathrm{D})$ and $(\mathcal{A}, \mathrm{E})$ respectively.

Lemma 9 (Sub-optimal action play probabilities)**.** *For any action $1 \leq i \leq K$, the probability of* ETC *committing to action at least as sub-optimal as i with richer side-information* E *is at most the corresponding probability with side-information* D*. That is,*

$$\mathbb{P}_{\mathrm{E}}\left\{\widehat{i_\star} \in \{i, i+1, \ldots, K\}\right\} \leq \mathbb{P}_{\mathrm{D}}\left\{\widehat{i_\star} \in \{i, i+1, \ldots, K\}\right\}.$$

Proof. Towards proving this, we develop some constructs using a generic side-information design F and shall later instantiate them using appropriate side-information designs D and E.

We introduce the following collection of sets $\{C_{A,i,t}\}_{i \in [K]}$, where we define

$$C_i = \{\mu' \in \mathbb{R}^K : \mu'(i) \geq \mu'(j) \text{ for all } j \neq i.\} \tag{29}$$

to be the set/region of values that estimate vector $\widehat{\mu}$ should belong to for action i to be committed to by ETC. Precisely,

$$\widehat{\mu} \in C_i \iff \widehat{i_\star} = i, \tag{30}$$

We shall study the probability that ETC commits to a certain action i by studying the probability that the computed estimate vector $\widehat{\mu}$ falls in C_i.

Claim 10 (Convex cone)**.** *For any action $i \in [K]$, the set C_i is a convex cone.*

Proof. The Claim follows since $C_i = \bigcap_{j \neq i} \{\mu' \in \mathbb{R}^K : \mu'(i) - \mu'(j) \geq 0\}$ is an intersection of homogeneous half-spaces, each linear in μ' and hence itself a convex cone.

First, it is a cone since for all $\mu' \in C_i$, we have $c\mu' \in C_i$ for any positive constant $c > 0$, as

$$\mu'(i) = \max_{j \in [K]} \mu'(j) \iff c\mu'(i) = \max_{j \in [K]} c\mu'(j).$$

Second, it is convex since for any $\mu', \mu'' \in C_i$, we have that $c_1\mu' + c_2\mu'' \in C_i$ for positive constants $c_1, c_2 \geq 0$, as

$$\begin{aligned}(\mu'(i) = \max_j \mu'(j)) \wedge (\mu''(i) = \max_j \mu''(j)) &\implies (c_1.\mu'(i) + c_2.\mu''(i) = c_1.\max_j \mu'(j) + c_2.\max_j \mu''(j)) \\ &\implies c_1.\mu'(i) + c_2.\mu''(i) = \max_j c_1.\mu'(j) + c_2.\mu''(j)).\end{aligned}$$

□

Geometrically, the collection $\{C_i\}_{i\in[K]}$ partitions $\mathbb{R}^K$ into pie slices with apex/centre at origin.

We recall the following notion, relaxing convexity, along with a fact about Gaussian probabilities over such sets.

Definition 5 (Star-shaped set)**.** Let V be a Euclidean space. A set $S \subseteq V$ is said to be *star-shaped* if there exists some point/element $x \in S$ such that for every $y \in S$, the line segment $\overline{xy}$ lies within S, i.e., $\forall \lambda \in [0,1] : \lambda x + (1-\lambda)y \in S$.

The set of all such x is called the *kernel* of set S.

A convex set is also star-shaped, and it can be seen that the entire convex set is its own kernel.

These star-shaped sets in the two-dimensional Euclidean plane are known as star-shaped polygons and appear to be well-studied in the computational geometry literature (see e.g., the 1985 textbook of Preparata and Shamos [2012]).

Claim 11 (Kernel-centred Gaussian distributions over star-shaped polytopes)**.** *Consider a star-shaped polytope S with μ in its kernel. Let $A \sim \mathcal{N}(\mu, \Sigma_A)$ and $B \sim \mathcal{N}(\mu, \Sigma_B)$ be Gaussian random variables with $\Sigma_A \preceq \Sigma_B$, i.e., A is more tightly centred around μ than B is, in every direction.*

Then, $\mathbb{P}\{A \in S\} \geq \mathbb{P}\{B \in S\}$.

Proof. Write f_A (and f_B) to be the p.d.f of $\mathcal{N}(\mu, \Sigma_A)$ (sim. $\mathcal{N}(\mu, \Sigma_B)$) in d dimensions. We show the Claim by comparing the corresponding probability integrals in the polar system. We start with the L.H.S.

$$\begin{aligned}
\mathbb{P}\{A \in S\} &= \int_{s\in S} f_A(s).ds = \int_{s\in S} (2\pi)^{d/2} \det(\Sigma_A)^{-1/2} \exp\left\{\frac{-1}{2}\|s-\mu\|_{\Sigma_A^{-1}}\right\}.ds \\
&\overset{(a)}{=} \int_{\phi}\int_{r=0}^{r_\phi} (2\pi)^{d/2} \det(\Sigma_A)^{-1/2} \exp\left\{\frac{-1}{2}\|r\|_{\Sigma_A^{-1}}\right\}.dr.d\phi \\
&\overset{(b)}{\geq} \int_{\phi}\int_{r=0}^{r_\phi} (2\pi)^{d/2} \det(\Sigma_B)^{-1/2} \exp\left\{\frac{-1}{2}\|r\|_{\Sigma_B^{-1}}\right\}.dr.d\phi \\
&\overset{(c)}{=} \int_{s\in S} f_B(s).ds = \mathbb{P}\{B \in S\},
\end{aligned}$$

where (b) is due to $\Sigma_A \preceq \Sigma_B$. Further, crucially, the conversion to and from the polar systems in (a) and (c) over a continuous $[0, r_\phi]$ is possible because S is star-shaped with μ in its kernel, i.e., in any direction ϕ, a ray originating from μ exits the polytope S at most once (and never enters again), and we call r_ϕ to be the distance of this exit point from μ (which is ∞ if it doesn't exit). □

To show the Lemma, as observed in Equation (30), we shall show that, for all $1 \leq i \leq k$, $\widehat{\mu}$ falls in the corresponding union of sets with a higher probability for richer side-information:

Claim 12. $\mathbb{P}_{\mathrm{E}}\left\{\widehat{\mu} \in \bigcup_{j=1}^{i} C_j\right\} \geq \mathbb{P}_{\mathrm{D}}\left\{\widehat{\mu} \in \bigcup_{j=1}^{i} C_j\right\}$ *for all* $1 \leq i \leq K$.

Proof. With the gaussian nature of distribution of $\widehat{\mu}$ as in Lemma 8, we are argue due to Lemma 11, it is sufficient to show that the union of regions $\bigcup_{j=1}^{i} C_i$ is star-shaped polytope (Definition 5) with its kernel containing the true μ, the original reward means vector.

To start, for $i = 1$, C_i is a convex set (Lemma 10) and is thus star-shaped, so the entire set is its own kernel. Further $\mu \in C_1$ as if estimate $\widehat{\mu} = \mu$ is accurate, then the algorithm commits to the optimal action $i = 1$. Thus, μ belongs to the kernel of C_1.

For $i \in [2, k]$, we show this using proof by contradiction. Assume for some i that $\bigcup_{j=1}^{i} C_i$ is not a star-shaped polytope w.r.t kernel point μ. Then, there exists some ray originating at μ that traverses μ^ℓ and μ^g (in that order) such that $\mu^\ell \in C_\ell$ and $\mu^g \in C_g$ with $\ell > i \geq g$, i.e., action g is more optimal than action ℓ. In other words, using observation Equation (30), there is some direction in which as the estimate $\widehat{\mu}$ moves away from the true μ, the committing action of ETC shall ceases to be the optimal action 1 and changes to the sub-optimal action ell as the estimate gets to μ^ℓ, and then changes again to a relatively optimal action g as the estimate moves further away to μ^g.

As $\mu^\ell \in C_\ell$, by its definition Equation (29),

$$\begin{aligned}
&\mu^\ell(\ell) \geq \mu^\ell(g) \\
\overset{\text{(d)}}{\implies} &(c\mu(\ell) + (1-c)\mu^g(\ell)) \geq (c\mu(g) + (1-c)\mu^g(g)) \\
\implies &c\mu(\ell) \geq c\mu(g) + (1-c)\left(\mu^g(g) - \mu^g(\ell)\right) \\
\overset{\text{(e)}}{\implies} &c\mu(\ell) \geq c\mu(g)
\end{aligned} \tag{31}$$

where (d) is by linear interpolation for some $0 < c < 1$, (e) uses that $\mu^g(g) - \mu^g(\ell) > 0$.

Finally, Equation (31) implies that action ℓ is more optimal than action g, which is a contradiction. This completes the proof of the Claim. □

The above Claim, in conjunction with Equation (30), completes the proof of the Lemma. □

Finally, to complete the proof of the Theorem, we show in Lemma 13, the instantaneous regret in the commit phase is lower with side-information E than it is with D.

Claim 13 (Instantaneous regret inequality)**.** *The per-time regret during the commit phase obeys:*

$$\sum_{j=2}^{K} \mathbb{P}_{\mathrm{E}}\left\{\widehat{i_\star} = j\right\} \Delta(j) \leq \sum_{j=2}^{K} \mathbb{P}_{\mathrm{D}}\left\{\widehat{i_\star} = j\right\} \Delta(j),$$

where $\Delta(j) := \mu(1) - \mu(j)$ is the sub-optimality 'gap' of action $j \in [2, K]$.

Proof. Write short-hands $p_i^{\mathrm{E}} = \mathbb{P}_{\mathrm{E}}\left\{\widehat{i_\star} = i\right\}$, $p_{i,K}^{\mathrm{E}} = \mathbb{P}_{\mathrm{E}}\left\{\widehat{i_\star} \in \{i, i+1, \ldots, K\}\right\}$ and similarly define $p_i^{\mathrm{D}}, p_{i,K}^{\mathrm{D}}$.

We show the Claim by induction on action index i.

Hypothesis.

$$H(i) : \sum_{j=i}^{k} \left(p_j^{\mathrm{E}} - p_j^{\mathrm{D}}\right) \Delta_j \leq \left(p_{i,K}^{\mathrm{E}} - p_{i,K}^{\mathrm{D}}\right) \Delta_i. \tag{32}$$

Base Case. Statement $H(k)$ holds as

$$p_K^{\mathrm{E}} \cdot \Delta_k - p_K^{\mathrm{D}} \cdot \Delta_k = \left(p_{K,K}^{\mathrm{D}} - p_{K,K}^{\mathrm{E}}\right) \Delta_K.$$

Induction Step. Let $H(i+1)$ be true for some $i + 1 \leq K$. We show $H(i)$ is true.

$$\begin{aligned}
\sum_{j=i}^{K} \left(p_j^{\mathrm{E}} - p_j^{\mathrm{D}}\right) \Delta_j &= \left(p_i^{\mathrm{E}} - p_i^{\mathrm{D}}\right) \Delta_i + \sum_{j=i+1}^{K} \left(p_j^{\mathrm{E}} - p_j^{\mathrm{D}}\right) \Delta_j \\
&\overset{\text{(f)}}{\leq} \left(p_i^{\mathrm{E}} - p_i^{\mathrm{D}}\right) \Delta_i + \left(p_{i+1,K}^{\mathrm{E}} - p_{i+1,K}^{\mathrm{D}}\right) \Delta_{i+1} \\
&\overset{\text{(g)}}{\leq} \left(p_i^{\mathrm{E}} - p_i^{\mathrm{D}}\right) \Delta_i + \left(p_{i+1,K}^{\mathrm{E}} - p_{i+1,K}^{S}\right) \Delta_i = \left(p_{i,K}^{\mathrm{E}} - p_{i,K}^{\mathrm{D}}\right) \Delta_i.
\end{aligned} \tag{33}$$

Here, (f) upper bounds the second term by using the induction assumption $H(i+1)$, then (g) is due to $p_{i+1,K}^{\mathrm{E}} - p_{i+1,K}^{\mathrm{D}} \leq 0$ by Lemma 9 and $\Delta_i \leq \Delta_{i+1}$. And Equation (33) shows $H(i)$. By mathematical induction, we have that $H(2)$ holds.

Along with the observation that $p_{i,K}^{\mathrm{E}} - p_{i,K}^{\mathrm{D}} \leq 0$ from Lemma 9, statement $H(2)$ implies the Claim. □

F Proof of Theorem 4

We give a short Proof sketch, before restating the Theorem and giving the formal complete proof.

Proof sketch. Consider two side-information designs D and E such that $\mathrm{E} \succeq \mathrm{D}$. The proof proceeds by coupling the executions of the meta-algorithm META under the two designs.

The key observation is that the internal trajectory of the black-box algorithm ALG is identical in both runs. Indeed, META only supplies side-information to ALG at times when ALG itself requests to play an action; from ALG's perspective, receiving a reward sample from the environment or from the side-information buffer is indistinguishable. Consequently, the sequence of actions recommended by ALG is the same under D and E.

Under this coupling, the only difference between the two runs is the availability and consumption of side-information. By Lemma 14, dominance of E over D implies that, for every action i, the number of side-information samples consumed under E is at least that under D, and hence the total number of consumed samples satisfies $\tau^{\mathrm{E}} \geq \tau^{\mathrm{D}}$.

The regret of META admits a decomposition into two parts: (i) the regret incurred by running the black-box algorithm ALG for $T + \tau$ effective steps on the problem instance without side-information, and (ii) a correction term that subtracts the regret saved by replacing environment pulls with side-information samples. Comparing the two designs, the difference in regret reduces to the effect of extending the black-box run from $T + \tau^{\mathrm{D}}$ to $T + \tau^{\mathrm{E}}$ and the additional side-information consumed.

Finally, since ALG satisfies Definition 4, the regret accumulated by ALG over the additional window $[T + \tau^{\mathrm{D}},\, T + \tau^{\mathrm{E}}]$ is at most proportional to the number of additional side-information samples consumed. By the balanced consumption property of META, this increase is exactly offset by the extra regret saved through those samples Therefore,

$$R(\textsc{Meta}, (\mathcal{A}, \mathrm{E}), T) \leq R(\textsc{Meta}, (\mathcal{A}, \mathrm{D}), T),$$

establishing monotonicity of META with respect to dominating side-information.

We now restate the Theorem and then formally prove it.

Theorem 4. *(Enforcing data-monotonicity.)* When run with any learning algorithm ALG that satisfies Definition 4, META is monotone in dominating extra information.

Proof. Consider the action set $\mathcal{A}$ and side-information designs D and E that obey $\mathrm{E} \succeq \mathrm{D}$ (E dominates D).

We shall refer to the problem run $(\textsc{Meta}, (\mathcal{A}, \mathrm{D}), T)$ (similarly $(\textsc{Meta}, (\mathcal{A}, \mathrm{E}), T)$) loosely as run D (similarly, run E). Write $c^{\mathrm{D}}(i)$ to be the $c(i)$ counter of the side-information samples of action i consumed in problem run D. At the end of time T, denote $\tau^{\mathrm{D}} = \sum_i c^{\mathrm{D}}(i)$ to be the total number of side-information samples consumed in scenario D. The quantities are analogous defined for scenario E. In general we shall use the superscripts D, E to differentiate the algorithmic quantities w.r.t. side-information designs, and use subscripts (usually, t, u, ρ) to qualify the temporal information of the algorithmic quantities.

Lemma 14. *The number of side-information samples consumed in scenario* E *it at least as many as consumed in scenario* D*. That is,* $c^{\mathrm{E}}(i) \geq c^{\mathrm{D}}(i)$ *throughout the time-horizon. Consequently,* $\tau^{\mathrm{E}} > \tau^{\mathrm{D}}$ *at the end of the time-horizon.*

Proof. The black-box algorithm ALG's trajectory is not affected by the side-information design. This is because the meta-algorithm META delays the consumption of side-information such that it is fed to ALG only when ALG naturally requires to play that action. From ALG's point of view, it is identical to get the action outcome by playing the action in the environment and or to get it from the side-information buffer.

Let ρ be used as a running variable to count the number of times the black-box ALG has been queried. Thus, when queried the ρ-th time, if the black-box ALG recommends action i under scenario D, then it shall also recommend action i under scenario E. Write $c^{\mathrm{E}}_{\rho}(i), c^{\mathrm{D}}_{\rho}(i)$ to be the side-information consumption counter values at the end of black-box algorithm's step ρ in scenario E and D respectively.

We show the required result by induction over the black-box algorithm steps ρ and the actions $i \in \mathcal{A}$.

Hypothesis. For any step ρ, consider the hypothesis $H(\rho, i) : c^{\mathrm{E}}_{\rho}(i) \geq c^{\mathrm{D}}_{\rho}(i)$.

Base case. Trivially, $c_0^{\mathrm{E}}(i) = 0 \geq 0 = c_0^{\mathrm{D}}(i)$ as no side-information is consumed before the meta-algorithm begins. Thus, $H(0, i)$ is true for all i. Induction step. Assume for some $\rho \in [T + \tau^{\mathrm{D}}]$ that $H(\rho, i)$ is true for all i.

If $c_\rho^{\mathrm{D}}(i) < c_\rho^{\mathrm{E}}(i)$, then, as the couters are whole numbers, and as the counters can increase by at most one in a single step, it follows that $c_{\rho+1}^{\mathrm{D}}(i) \leq c_{\rho+1}^{\mathrm{E}}(i)$, thus $H(\rho, i)$ holds.

When $c_\rho^{\mathrm{D}}(i) = c_\rho^{\mathrm{E}}(i)$, we show $H(\rho + 1, i)$ for each i by arguing that when the black-box algorithm is queried for the $\rho + 1$-th time, if the meta-algorithm feeds it an outcome from the side-information buffer $S(i)$ in scenario D, then it does so in scenario E as well, i.e., if $c^{\mathrm{D}}(i)$ increases by one, then $C^{\mathrm{E}}(i)$ shall also increase by one.

We show this by arguing about the two conditions in Line 5 that if it is satisfied in scenario D, it is also satisfied in scenario E.

1. Let $t' \in [T]$ be the most recent environment time-step when the black-box algorithm is in internally in step $\rho + 1$. Always, all side information received by the meta-algorithm so far is either in the buffer or has been consumed. Thus, writing $S_\rho^{\mathrm{E}}(i) = |S(i)|$ to be the cardinality of the buffer after step ρ, we have $S_\rho^{\mathrm{E}}(i) + c_\rho^{\mathrm{E}}(i) = \sum_{u=1}^{t'} \mathrm{E}_u(i) \geq \sum_{u=1}^{t'} \mathrm{E}_u(i) = S_\rho^{\mathrm{D}}(i) + c_\rho^{\mathrm{D}}(i)$. As $c_\rho^{\mathrm{E}}(i) = c_\rho^{\mathrm{D}}(i)$, we have that $S_\rho^{\mathrm{E}}(i) \geq S_\rho^{\mathrm{D}}(i)$, i.e., the buffer has in scenario E at least as many samples as there is in scenario D. Thus, if the buffer is note empty $S(i) \neq \emptyset$ in scenario D, then it is not empty in scenario E also.

2. If $c_\rho^{\mathrm{D}}(i) \leq \min_{j \in \mathcal{A}} c_\rho^{\mathrm{D}}(j)$, then,

$$c_\rho^{\mathrm{E}}(i) = c_\rho^{\mathrm{D}}(i) \leq \min_{j \in \mathcal{A}} c_\rho^{\mathrm{D}}(j) \overset{\text{(a)}}{\leq} \min_{j \in \mathcal{A}} c_\rho^{\mathrm{E}}(j),$$

   where (a) uses the induction assumption $H(\rho, i)$ for all $i \in \mathcal{A}$.

The above two implications imply that if the meta-algorithm consumes an outcome from the side-information buffer $S(i)$ in scenario D, then it does so in scenario E as well, i.e., if $c^{\mathrm{D}}(i)$ increases by one, then $C^{\mathrm{E}}(i)$ shall also increase by one.

By mathematical induction, we have that $c_{T+\tau^{\mathrm{D}}}^{\mathrm{E}}(i) \geq c_{T+\tau^{\mathrm{D}}}^{\mathrm{D}}(i)$ for all $i \in \mathcal{A}$, and thus $\tau^{\mathrm{E}} \geq \tau^{\mathrm{D}}$, Further, as the counter value can only increase with the steps, $c_{T+\tau^{\mathrm{E}}}^{\mathrm{E}}(i) \geq c_{T+\tau^{\mathrm{D}}}^{\mathrm{E}}(i)$. This completes the proof of the Lemma. □

Recall τ^{D} is the total number of side-information samples consumed. Thus, the black box algorithm has been queried for a total of $T + \tau^{\mathrm{D}}$ times. Owing to the identically distributed nature of the rewards and side-information samples of an action, in effect, the black box algorithm has been run on problem instance without side-informaton, i.e., $(\mathcal{A}, 0)$ for a period of $T + \tau^{\mathrm{D}}$ time-steps. Write the regret of this black-box algorithm here to be $R(\textsc{Alg}, (\mathcal{A}, 0), T + \tau^{\mathrm{D}})$.

The meta-algorithm has incurred the regret of the black-box algorithm minus whatever samples it used from the side-information buffers to feed the black-box algorithm. Note that whenever it couldn't use samples from the side-information buffers, it played the action in the environment and incurred regret. Thus, we have

$$R(\textsc{Meta}, (\mathcal{A}, \mathrm{D}), T) = R(\textsc{Alg}, (\mathcal{A}, 0), T + \tau^{\mathrm{D}}) - \sum_{i \in \mathcal{A}} c^{\mathrm{D}}(i)\Delta_i. \tag{34}$$

$$R(\textsc{Meta}, (\mathcal{A}, \mathrm{E}), T) = R(\textsc{Alg}, (\mathcal{A}, 0), T + \tau^{\mathrm{E}}) - \sum_{i \in \mathcal{A}} c^{\mathrm{E}}(i)\Delta_i. \tag{35}$$

Let $T > T'$ be suitably large where the consumptions counts $c^{\mathrm{D}}(i)$s are equal as per the balanced consumption of side-information.

We have

$$
\begin{aligned}
& R(\textsc{Alg}, (\mathcal{A}, 0), T + \tau^{\mathrm{E}}) - R(\textsc{Alg}, (\mathcal{A}, 0), T + \tau^{\mathrm{D}}) \\
\overset{\text{(b)}}{=} & R(\textsc{Alg}, (\mathcal{A}, 0), [T + \tau^{\mathrm{D}}, T + \tau^{\mathrm{E}}]) \\
\overset{\text{(c)}}{\leq} & \frac{\tau^{\mathrm{E}} - \tau^{\mathrm{D}}}{K} \sum_{i \in \mathcal{A}} \Delta_i \\
\overset{\text{(d)}}{=} & \sum_{i \in \mathcal{A}} c^{\mathrm{E}}(i) - c^{\mathrm{D}}(i)
\end{aligned} \tag{36}
$$

Here, (b) uses Lemma 14 to have $[T + \tau^{\mathrm{D}}, T + \tau^{\mathrm{E}}]$ is a valid positive-sized time interval. Then, (c) uses that ALG obeys Definition 4, and (d) follows from the meta algorithm's balanced-consumption of side-information. Now, substituting Equation (36) into the equation obtained by subtracting Equation (34) from Equation (35), we have

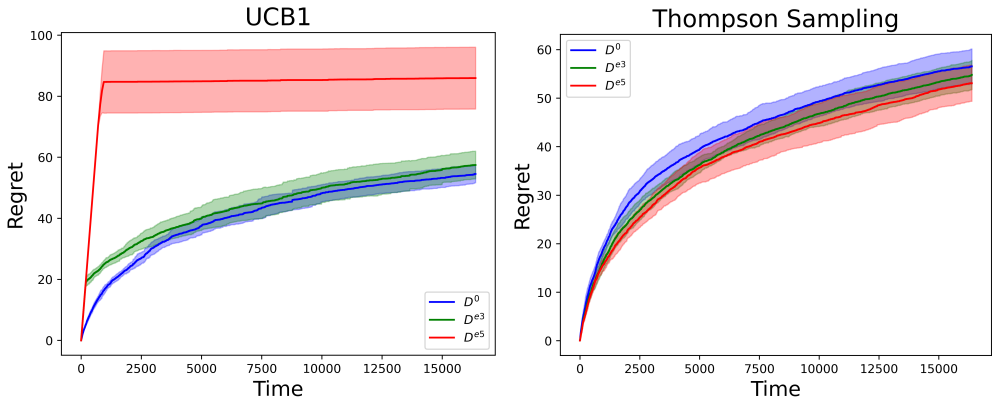
$$
\begin{aligned}
& R(\textsc{Meta}, (\mathcal{A}, \mathrm{E}), T) - R(\textsc{Meta}, (\mathcal{A}, \mathrm{D}), T) \\
= & R(\textsc{Alg}, (\mathcal{A}, 0), T + \tau^{\mathrm{E}}) - R(\textsc{Alg}, (\mathcal{A}, 0), T + \tau^{\mathrm{D}}) - \left(\sum_{i \in \mathcal{A}} c^{\mathrm{E}}(i) \Delta_i - \sum_{i \in \mathcal{A}} c^{\mathrm{D}}(i) \Delta_i \right) \\
\overset{\text{(e)}}{\leq} & 0,
\end{aligned} \tag{37}
$$

where (e) is due to Equation (36). This Equation (37) shows the Theorem. □

G Additional Simulation Experiments

G.1 The argument about front-loading optimal action samples.

In our theoretical results (especially Theorem 1), we employed instances where the optimal action's outcomes are provided at the begining as extra information. We argued that this leads to UCB incurs more regret, whereas Thompson Sampling doesn't. We provide experimental backing to that in Figure 3. The 'catch-up' effect described in the main paper is evident. Upon receiving huge amounts of optimal samples, UCB ends up exploring the sub-optimal action to a commensurate level by continuously playing the sub-optimal action at the beginning. Whereas, Thompson Sampling doesn't suffer that. It starts playing the sub-optimal action less frequently right from the outset given the better estimate of the optimal action with more side-information.



(a) Impact of front-loading side-information on UCB (b) Impact of different agents on agent $g = 6$.

Figure 3: For an instance of $\mu(\frac{1}{2}, \frac{1}{2} + \Delta)$, we show the regrets over three different side-information designs. D^0 contains no side-information. D^{e3} and D^{e5} provide 1,000 and 100,000 samples of the optimal action at time $t = 0$.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes], [No], or [N/A].
- [N/A] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for [N/A]).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will also be asked to include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While [Yes] is generally preferable to [No], it is perfectly acceptable to answer [No] provided a proper justification is given (e.g., error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering [No] or [N/A] is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [Yes]

Justification: [N/A]

Guidelines:

- The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [Yes]

Justification: [N/A]

Guidelines:

- The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: [N/A]

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.

- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: [Yes]

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: [N/A]

   Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: [N/A]

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: [N/A]

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.

- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: [N/A]

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: [N/A]

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: [N/A]

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.

- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: [N/A]

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [N/A]

    Justification: [N/A]

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: [N/A]

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: [N/A]

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: [N/A]

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: [N/A]

    Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.