Near-Optimal Stochastic Linear Bandits with Delay

Ofir Schlisselberg[*] **Mengxiao Zhang**[†] **Yishay Mansour**[‡]

Abstract

We study stochastic linear bandits with delayed feedback under several delay models and establish near-optimal regret guarantees. Our results identify when delayed linear bandits exhibit the same qualitative behavior as multi-armed bandits (MAB), and when the linear structure creates fundamentally new challenges. Specifically, (1) for *loss-independent delays*, where the delay does not depend on the realized loss (but potentially depends on the arm), we show that delays incur only an additive regret penalty. Under stochastic delays, this penalty scales with the expected delay, while under adversarial delays, it scales with the maximum number of outstanding observations. Notably, both delay penalties are dimension-free, improving upon the state-of-the-art results; (2) for *loss-dependent delays*, we show that linear bandits are substantially harder than MAB: unlike in MAB, we prove matching (up to log factors) upper and lower bounds in linear bandits, whose delay penalty depends on the square root of the dimension. (3) for the *delay-as-payoff model*, a special case of loss-dependent delay, we show that the optimal MAB guarantee, which depends only on the delay of the optimal arm, is also unattainable in linear bandits. Together, these results provide a sharp characterization of how delayed feedback interacts with linear generalization.

1 Introduction

Sequential decision-making under uncertainty is a central problem in machine learning, with applications ranging from recommendation systems to clinical trials. In many real-world scenarios, however, feedback is not observed immediately: actions are taken at the current round, while their outcomes are revealed only after a delay. Such delayed feedback creates a fundamental challenge, since the learner must continue making decisions without access to the most recent observations.

The stochastic multi-armed bandit (MAB) model has served as a primary framework for understanding delayed feedback. Early works considered constant delays [9], and Joulani et al. [16] later provided a general reduction for both stochastic and adversarial delays to the case without delay. However, these reductions showed that the regret incurred in the case with delayed feedback will have a delay-dependent term that is also dependent on the number of arms K: the delay penalty scales as $K\mathbb{E}[d]$ under stochastic delays, where $\mathbb{E}[d]$ is the expected delay, and as $K\sigma_{\max}$ under adversarial delays, where $\sigma_{\max}$ is the maximum number of missing observations over the horizon. Subsequent works showed that this dependence on K is in fact not intrinsic. In particular, Lancewicki et al. [18] obtained optimal stochastic-delay bounds without the extra K factor in the delay-dependent term, and Schlisselberg et al. [26] extended this improvement to adversarial delays. Thus, for standard delayed MAB, the effect of delay is now well understood: delay contributes an additive term that matches the intrinsic delay complexity of the problem.

More involved delay structures are studied for MAB subsequently. Gael et al. [13] first studied the case where the delays are arm-dependent, whose results are later improved by Lancewicki et al.

[*]Tel Aviv University; ofirs4@mail.tau.ac.il
[†]University of Iowa; mengxiao-zhang@uiowa.edu
[‡]Tel Aviv University and Google Research; mansour.yishay@gmail.com

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

[18]. The latter work further considered delays that depend on the realized loss, showing that the regret overhead due to delay remains bounded by $\max_a \mathbb{E}[d(a)]$.[4] A particularly important special case is the *delay-as-payoff* setting, introduced by Tang et al. [28], in which the delay is proportional to the loss. In this setting, Schlisselberg et al. [25] showed that the optimal additive delay term is the mean delay of the best arm, reflecting the fact that once the best arm is identified, the learner does not need to wait for feedback from suboptimal arms.

In contrast, delayed feedback in *stochastic linear bandits* remains much less understood. Linear bandits model a richer decision space in which actions are represented by an n-dimensional feature vectors and feedback from one action can inform the estimates of many others through an unknown linear parameter. However, unlike in MAB, where each arm can be learned essentially independently, delayed observations in linear bandits affect the estimation of a shared parameter. Early works on delayed linear bandits obtained regret bounds scaling with the *sum* of delays over the horizon [36, 4], which can be prohibitively large. The first result avoiding this dependence is Howson et al. [15], which obtained an additive delay term of order $n^{3/2}\mathbb{E}[d]$. While this improves over the sum-of-delay bounds, it still leaves unresolved whether the dimension dependence is an artifact of the analysis or an inherent feature of delayed linear bandits. More generally, the following fundamental question is open:

When can the regret of delayed linear bandits match the delay dependence of MAB?

In this paper, we study delayed feedback in linear bandits under both loss-independent and loss-dependent delay models, and provide a unified view of when dimension plays a role.

- **Loss-independent delays.** We consider arm-dependent delays that are independent of the realized loss. For stochastic delays, we obtain a regret bound of $O\big(\sqrt{nT\log(KT)} + \max_a \mathbb{E}[d(a)]\big)$, where T is the time horizon and $\mathbb{E}[d(a)]$ is the expected delay of arm a, matching the optimal dependence known in MAB. For adversarial delays, we prove a regret bound of $O\big(\sqrt{nT\log(KT)} + \sigma_{\max}\log(\sigma_{\max})\big)$, which is tight up to logarithmic factors and, to the best of our knowledge, is the first such result even in the MAB setting for arm-dependent delays.
- **Loss-dependent delays.** We demonstrate that linear bandits are *fundamentally harder* in this regime. While MAB regret overhead scales only with $\max_a \mathbb{E}[d(a)]$, we prove that linear bandits inherently incur a dimension-dependent delay penalty of $\sqrt{n}\max_a \mathbb{E}[d(a)]$. We establish this by providing a matching (up to logarithmic factors) upper bound and a lower bound, proving that this dependence on the dimension n is unavoidable.
- **Delay-as-payoff.** In MAB, the delay-as-payoff setting admits an additive delay term of $d^\star$, where $d^\star$ represents the mean delay of the best arm. We show that this is *not achievable* in linear bandits: even when $d^\star = 0$, any algorithm must incur an additive regret of order D, the maximal delay. This matches the upper bound of Zhang et al. [35] and establishes its optimality.

Our results reveal a sharp dichotomy. When delays are independent of the loss, linear bandits behave similarly to MAB, and optimal delay dependence can be achieved without additional dimension factors. In contrast, when delays depend on the loss, the shared structure of linear bandits fundamentally limits the learner's ability to exploit partial feedback. This highlights an intrinsic difference between independent and dependent delay models, and provides a complete picture of delay in linear bandits.

1.1 Additional Related Work

Stochastic Linear Bandits. The stochastic linear bandit framework was first introduced by Abe and Long [2], with foundational theoretical contributions subsequently established by [3, 7, 24, 1]. Our algorithm is built upon the phased elimination approach, a classic technique for this problem that was first adapted to the linear setting by Valko et al. [30]. The specific algorithmic framework employed in this work follow the one presented in Lattimore and Szepesvári [19, Chapter 22].

Additional Work on Delayed Bandits. Several works examine delay under various feedback assumptions. [33, 34] study a model with Bernoulli rewards where zero outcomes are never revealed, even after the delay, which can be interpreted as a particular form of loss-dependent delay. Their results rely on assuming a bounded expected delay and that the delay distribution is fully known. Pike-Burke et al. [23] consider a more challenging scenario in which the learner observes only the

[4] Their result is in fact more involved and depends on the quantiles of the delay distribution. For clarity, we present a simplified form of the bound, which we derive for a slight variant of their algorithm in Appendix E.

Table 1: **Comparison of delay dependence across models.** Arm-independent delays are a special case of arm-dependent delays. We show that linear bandits match MAB guarantees for loss-independent delays, but exhibit fundamentally stronger dependence in loss-dependent settings. n denotes the dimension, $\mathbb{E}[d(a)]$ is the expected delay of arm a (and $\mathbb{E}[d]$ in the arm-independent case), $\sigma_{\max}$ is the maximum number of missing observations, d^* is the delay of the optimal arm, and D is the maximum delay. All bounds are up to absolute constants and log factors.

Delay model		**MAB**	**Linear (prior)**	**Linear (this work)**
Arm-inde.	Stochastic	$\mathbb{E}[d]$ [18]	$n^{3/2}\mathbb{E}[d]$ [15]	$\mathbb{E}[d]$
	Adversarial	$\sigma_{\max}$ [26]	—	$\sigma_{\max}$
Arm-dep.	Stochastic	$\max_a \mathbb{E}[d(a)]$ [18]	—	$\max_a \mathbb{E}[d(a)]$
	Adversarial	—		$\sigma_{\max}$
Loss-dep.	Stochastic	$\max_a \mathbb{E}[d(a)]$ [18]	—	$\sqrt{n}\max_a \mathbb{E}[d(a)]$
Delay-as-payoff		d^* [25]	D (upper bound) [35]	D (lower bound)

total reward arriving at each round, without knowing which actions generated it. They assume that the expected delay is known and derive guarantees comparable to those of Joulani et al. [16].

Another line of work considers adversarial losses with delay [6, 29, 14, 31, 32], with [32] extending these ideas to linear bandits. Finally, [21, 22, 26] study delayed best-of-both-worlds guarantees, where a single algorithm adapts to both stochastic and adversarial losses.

2 Preliminaries

Learning model. In this paper, we consider *stochastic delayed linear bandits* over a finite action set $\mathcal{A}$ with $|\mathcal{A}| = K$ and an unknown loss vector $\theta^\star$.[5] Each action $a \in \mathcal{A}$ is associated with a zero-mean noise distribution $\mathcal{P}(a)$ such that for any $\eta \sim \mathcal{P}(a)$, the bound $\left|a^\top \theta^\star + \eta\right| \le 1$ holds almost surely. In the case of *delay-as-payoff*, following Zhang et al. [35], we further require $a^\top \theta^\star + \eta \ge 0$ to ensure that all realized delays are non-negative.

The interaction protocol proceeds as follows: at each round $t \in [T] \triangleq \{1, 2, \ldots, T\}$:

- The learner selects an action $a_t \in \mathcal{A}$.
- The environment samples random noise $\eta_t \sim \mathcal{P}(a_t)$ and generates a loss $\ell_t = a_t^\top \theta^\star + \eta_t$.
- The environment determines a delay $d_t(a_t)$ according to the specific delay model.
- The learner receives the set of all losses whose feedback arrives at time t: $\{(t', \ell_{t'}) : t' + d_{t'}(a_{t'}) = t\}$.

The learner's objective is to minimize their (pseudo) regret, defined as the difference between the expected loss of the best fixed action and the learner's cumulative loss. Formally, we define regret over a time horizon T as:

$$\mathcal{R}_T \triangleq \mathbb{E}\left[\sum_{t=1}^{T} (a_t - a^\star)^\top \theta^\star\right],$$

where $a^\star \in \arg\min_{a \in \mathcal{A}} a^\top \theta^\star$.

Delay model. In the following, we introduce several types of delay models that we consider throughout this paper. Specifically, we consider

- **Loss-independent delay**: In this setting, the delay is independent of the realized loss. We consider two primary variants:
  - *Stochastic delay*, where delay $d_t(a)$ is drawn i.i.d. from an unknown distribution $\mathcal{D}(a)$.
  - *Adversarial delay*, where the entire delay sequence $\{d_t(a)\}_{t \in [T], a \in \mathcal{A}}$ is determined arbitrarily by an adaptive adversary.

  For both variants, we also investigate the **arm-independent** special case, where $\mathcal{D}(a) = \mathcal{D}$ in the stochastic setting and $d_t(a) = d_t$ for all $a \in \mathcal{A}$ in the adversarial setting.

[5] The analysis extends to continuous action sets via standard discretization arguments. Our bounds scale as $\sqrt{nT\log(K)}$, which recovers the usual $n\sqrt{T}$ dependence in the continuous setting up to logarithmic factors.

Algorithm 1 Phased Elimination w/ Loss-Independent Stochastic Delays (Formally in Algorithm 4)

Initialize active set $\mathcal{A}_1 \leftarrow \mathcal{A}$, accuracy $\epsilon_1 \leftarrow \frac{1}{2}$
for phases $m = 1, 2, \ldots$ **do**
  Compute a design π_m over $\mathcal{A}_m$
  Set target samples $N_m = \tilde{O}\left(\frac{n}{\epsilon_m^2}\right)$
  while some arm a has fewer than $N_m(a) = N_m \pi_m(a)$ observed samples **do**
    Sample $a_t \sim \pi_m$
    Collect any feedback that arrives
  Estimate $\widehat{\theta}_m$ from the collected samples
  $\mathcal{A}_{m+1} \leftarrow \left\{ a \in \mathcal{A}_m : \langle \widehat{\theta}_m \,,\, a - b \rangle \le 2\epsilon_m \quad \forall b \in \mathcal{A}_m \right\}$
  Set $\epsilon_{m+1} \leftarrow \epsilon_m / 2$

- **Loss-dependent stochastic delay**: In this regime, the delay $d_t(a)$ is stochastically coupled with the loss $\ell_t(a)$. Specifically, the pair $(\ell_t(a), d_t(a))$ is sampled i.i.d. from an arm-specific joint distribution $\mathcal{D}^{\text{joint}}(a)$. This setting encapsulates real-world scenarios where the feedback latency is inherently tied to the outcome magnitude. For example, in conversion rate optimization, verifying a high-value transaction often incurs a significantly longer latency than registering a simple click.
- **Delay-as-payoff**: As a special instance of loss-dependent delay, we investigate a setting where the latency is a deterministic function of the loss [26, 35]. Formally, we assume $d_t(a) = D \cdot \ell_t(a)$ for some fixed scaling constant $D > 0$. This is motivated by applications such as network routing, where the goal is to minimize the latency, so the latency is both the cost and the delay in observing the information.

G-Optimal design. We use optimal experimental design to estimate $\theta^\star$ efficiently. A design π is a distribution over $\mathcal{A}$, and we define $V(\pi) = \sum_{a \in \mathcal{A}} \pi(a) a a^\top$ and $g(\pi) = \max_{a \in \mathcal{A}} \|a\|_{V(\pi)^{-1}}$. A G-optimal design [19, Chapter 21] minimizes $g(\pi)$. The General Equivalence Theorem [17] states that the optimum satisfies $g(\pi^\star) = n$ and admits a solution with support size at most $n(n+1)/2$. Moreover, approximate designs with support $O(n \log \log n)$ and $g(\pi) \le 2n$ can be computed via Frank–Wolfe-type methods [12, 11]. Such designs control the variance of least-squares estimators and are central to achieving optimal regret in linear bandits [5, 19]. Geometrically, they correspond to the minimum-volume enclosing ellipsoid of $\mathrm{conv}(\mathcal{A})$, with support points as contact points on its boundary [27].

Additional Notation. For any time $t \in [T]$, let $M_t \subseteq [t-1]$ denote the set of rounds where feedback for the chosen action has not yet been received by the start of round t. We further define $M_t(a) \subseteq M_t$ as the subset of rounds where action a was selected, such that $M_t = \bigcup_{a \in \mathcal{A}} M_t(a)$. Let $\sigma_t(a) = |M_t(a)|$ denote the number of outstanding observations for action a, and let $\sigma_t = |M_t| = \sum_{a \in \mathcal{A}} \sigma_t(a)$ represent the total number of missing observations at time t. We define the global maximum number of outstanding observations as $\sigma_{\max} = \max_{t \in [T]} \sigma_t$. When the delay follows a distribution (either loss-dependent or independent), let $\mathcal{Q}(q; a)$ denote the q-quantile of the delay distribution for action a and let $\mathbb{E}[d(a)]$ denote the expected delay of action a. In the arm-independent setting, these notations simplify to $\mathcal{Q}(q)$ and $\mathbb{E}[d]$, respectively.

3 Loss-Independent Stochastic Delay

In this section, we study delayed feedback models in which the delay process is independent of the realized loss. The delay distribution may depend on the action, but conditional on the chosen action, the delay is independent of the loss observation. This setting captures an important class of delayed bandit problems in which the learner may face heterogeneous reporting times across actions, while still avoiding the additional statistical difficulty caused by loss-dependent censoring.

Our algorithm, Algorithm 4, extends the classical phased elimination framework to this delayed setting. At a high level, the algorithm proceeds in phases. In phase m, it maintains an active set $\mathcal{A}_m$ of candidate actions and an accuracy level ϵ_m. The goal of the phase is to collect enough observed feedback to estimate the loss of every active action up to accuracy ϵ_m, eliminate actions that are provably suboptimal, and then move to the next phase with $\epsilon_{m+1} = \epsilon_m / 2$.

The main challenge is that, under stochastic delayed feedback, the number of actions played in a phase can significantly exceed the number of observations received. In the standard non-delayed setting, this is not an issue: one can simply pull each arm a in the design support deterministically to match the target number of pulls exactly. However, stochastic delays introduce a fundamental randomness that cannot be eliminated through deterministic scheduling. Since a phase can only terminate once the required observations have arrived for every action in the design support, the phase duration is governed by the tail behavior of the delays. Consequently, the phase length depends not only on the statistical complexity N_m, but also on the interplay between the design's probability mass and the delay quantiles.

This creates a subtle technical obstacle. Standard optimal designs minimize the estimation error $g(\pi)$, but they may assign extremely small probability mass to certain actions. While this is efficient under immediate feedback, it becomes problematic with delays: if an action is sampled with very low probability, the time needed to collect enough observations can grow significantly, even if the design itself is statistically near-optimal. In this regime, the lower-order concentration terms, which are usually negligible, become the primary bottleneck because the delay process itself functions as an uncontrollable stochastic sampling process.

A contribution of our analysis is demonstrating that this obstacle can be removed without sacrificing statistical efficiency. It can be shown that any design can be converted into a balanced design where support probabilities are uniformly lower bounded, while the design objective $g(\pi)$ increases by only a constant factor; see Lemma A.2. Combined with standard Frank–Wolfe constructions, this yields a design with support size $O(n \log\log n)$, minimum probability at least $1/(n \log\log n)$ on its support, and design value $g(\pi) = O(n)$; see Lemma A.3. This balancing step is the primary technical ingredient that allows phased elimination to remain efficient under stochastic delays.

With this balanced design in place, the regret analysis separates into two parts. We first control the duration of each phase. Since every supported action is sampled with a non-negligible probability, after the algorithm has played the phase for approximately $(N_m + n\log(T)\log\log(n))/q$ rounds, each action has been selected approximately $qN_m(a)$ times with high probability. Consequently, after waiting up to the corresponding q-quantile of its delay distribution, a sufficient number of feedback observations has arrived. This intuition is formalized by the following phase-length bound.

Lemma 3.1 *Let T_m be the length of phase m of Algorithm 4. Under stochastic arm-dependent delays, with probability at least $1 - \frac{1}{T}$,*

$$T_m = O\left(\min_q \left\{ \frac{N_m + n\log(T)\log\log(n)}{q} + \max_a \mathcal{Q}(q;a) \right\}\right).$$

The proof is deferred to Appendix B. The term $(N_m + n\log(T)\log\log(n))/q$ is the number of plays needed to ensure that enough samples are generated and observed with high probability, while $\max_a \mathcal{Q}(q;a)$ accounts for the additional waiting time induced by the delay distribution. The minimum over q captures the optimal tradeoff between playing longer to compensate for delayed arrivals and waiting for a larger delay quantile.

The second part of the analysis is the standard elimination argument, adapted to the delayed setting. At the end of phase m, the algorithm has received $N_m(a)$ observations from every action a in the support of the design. By the choice of N_m and the guarantee that $g(\pi_m) = \max_{a\in\mathcal{A}} \|a\|_{V(\pi_m)^{-1}} = O(n)$, the least squares estimator $\widehat{\theta}_m$ is ϵ_m-accurate uniformly over all actions in $\mathcal{A}_m$ with high probability. Consequently, the elimination rule in Algorithm 1 has two properties: (i) the optimal action is never eliminated, and (ii) every action that remains active after the phase has suboptimality gap at most $8\epsilon_m$; see Lemma B.1 and Lemma B.2. Therefore, once the algorithm reaches a phase with accuracy ϵ_m, all actions with gap larger than $8\epsilon_m$ are removed.

Combining the phase-length bound with this elimination guarantee gives the following regret bound.

Theorem 3.2 *Algorithm 4 guarantees that under stochastic arm-dependent delays,*

$$\mathcal{R}_T = O\left(\min_q \left\{ \sqrt{\frac{nT\log(KT)}{q}} + \frac{n\log(T)\log\log(n)}{q} + \max_a \mathcal{Q}(q;a) \right\}\right).$$

The proof is deferred to Appendix B. Theorem 3.2 shows that, up to logarithmic factors, delayed feedback affects the regret through an additive delay-quantile term and a multiplicative $1/\sqrt{q}$ factor

in the statistical term. In particular, the algorithm recovers the classic $\sqrt{nT}$ regret when the relevant constant quantile of the delay is $O(\sqrt{nT})$.

A direct corollary of Theorem 3.2 is obtained by translating the delay quantiles to mean delays. By Markov's inequality, the median delay of action a is at most $2\mathbb{E}[d(a)]$; see Lemma B.4. Taking $q = 1/2$ in the theorem gives

$$\mathcal{R}_T = O\left(\sqrt{nT\log(KT)} + \max_a \mathbb{E}[d(a)]\right),$$

as stated formally in Corollary B.5. Thus, for loss-independent stochastic delays, the price of delay in linear bandits is additive in the largest mean delay without dimensional dependency, which is the same as the one in MAB shown in [18]. Finally, we note that the same algorithmic idea also extends beyond the stochastic arm-dependent setting. In Appendix B.3, we show that under adversarial but arm-independent delays, the same algorithm also achieves $O(\sqrt{nT\log(KT)} + \sigma_{\max})$ regret. Specifically, the cost of delayed feedback is only an additive regret term $O(\sigma_{\max})$, which again is the same as the one shown in MAB [26].

4 Loss-Independent Adversarial Delay

We now extend our analysis to arm-dependent adversarial delays, which remain independent of the realized losses. While our algorithm still follows the high-level phased elimination structure used in the stochastic setting, the previous sampling strategy is no longer sufficient. In the stochastic regime, sampling arms according to a fixed design ensures that missing observations are recovered after certain periods with high probability. In contrast, when delays are chosen by an adversary, the adversary can concentrate delays on specific arms. If an arm is assigned a small probability mass (e.g., $\Theta(1/n)$), delaying $\sigma_{\max}$ of its observations may force the learner to wait an additional $\Theta(n \cdot \sigma_{\max})$ rounds to collect them. This leads to an undesirable dimension-dependent penalty that we want to avoid.

To resolve this issue, we modify how samples are collected in each phase. Instead of the probabilistic sampling used in the stochastic case, we now use a deterministic approach. By playing each arm a fixed number of times, we ensure that the adversary cannot stall the algorithm by targeting arms with low sampling probabilities. Under this approach, each phase is divided into two parts:

- **First Part** (Line 3 to Line 5 of Algorithm 2): we play every arm $a \in \mathcal{A}_m$ exactly $N_m(a)$ times. Since feedback is delayed, not all observations will be available immediately after these pulls are completed.
- **Second Part** (Line 6 to Line 8 of Algorithm 2): the algorithm iteratively replays only the missing samples needed to complete the phase requirements.

Similar to the stochastic delay case, to bound the regret, it suffices to bound the length of each phase. Specifically, in the following key lemma, we show that when the delay is adversarial, Algorithm 5 guarantees that the m-th phase length is at most $O(N_m + \sigma_{\max}\log(\sigma_{\max}))$.

Lemma 4.1 *Let T_m be the length of phase m in Algorithm 5. Then with probability at least $1 - \frac{1}{T}$, Algorithm 5 guarantees that $T_m = O(N_m + \sigma_{\max}\log\sigma_{\max})$.*

The proof of Lemma 4.1 is based on a technical analysis of the number of times Line 6 is executed and the number of arms played during each execution. The full proof is deferred to Appendix C. Based on this phase-length bound, the regret guarantee follows the same logic as the stochastic case, yielding an additive dependence on $\sigma_{\max}\log(\sigma_{\max})$.

Theorem 4.2 *Algorithm 5 guarantees that $\mathcal{R}_T = O\left(\sqrt{nT\log(KT)} + \sigma_{\max}\log(\sigma_{\max})\right)$.*

Theorem 4.2 shows that in the adversarial delay setting, the price of delay in linear bandits is additive in the maximum number of missing observations but still without dimensional dependency. Finally, we note that in the arm-independent stochastic setting, $\sigma_{\max}$ can be bounded in terms of the expected delay. Consequently, the same algorithm also achieves the desired guarantees for stochastic delays; see Appendix C.3 for details.

Algorithm 2 Phased Elimination w/ Adversarial Loss-Independent Delay (Formally in Algorithm 5)

Initialize active set $\mathcal{A}_1 \leftarrow \mathcal{A}$, accuracy $\epsilon_1 \leftarrow \frac{1}{2}$
for phases $m = 1, 2, \ldots$ **do**
  Compute a design π_m over $\mathcal{A}_m$
  Set target samples $N_m = \tilde{O}\left(\frac{n}{\epsilon_m^2}\right)$
  Play each arm a approximately $N_m(a) = N_m \pi_m(a)$ times
  while some arm a has fewer than $N_m(a)$ observed samples **do**
    Replay the missing samples of each arm
    Collect delayed feedback as it arrives
  Estimate $\widehat{\theta}_m$ from the collected samples
  $\mathcal{A}_{m+1} \leftarrow \left\{ a \in \mathcal{A}_m : \langle \widehat{\theta}_m \,,\, a - b \rangle \le 2\epsilon_m \quad \forall b \in \mathcal{A}_m \right\}$
  Set $\epsilon_{m+1} \leftarrow \epsilon_m / 2$

5 Loss-Dependent Delays

In this section, we switch our focus to the general loss-dependent delay model. In contrast to the loss-independent setting, the delay may depend on the realized loss. This creates a statistical difficulty that is absent from the previous cases: the set of observations that arrive early is no longer an unbiased sample of the losses. For example, if smaller losses tend to be observed faster (i.e. the delay is positively correlated with the loss), then a phase that simply waits until enough feedback has arrived will over-represent such losses, leading to a biased estimate. Hence, the standard delayed phased-elimination strategy cannot be applied directly. In the following, we will first introduce our algorithm obtaining a regret bound of $O(\sqrt{n} \max_{a \in \mathcal{A}} \mathbb{E}[d(a)])$ delay dependency, showing a strictly worse dimensional dependency compared to the MAB case [18] where they do not have the $\sqrt{n}$ dependency. Then, we compliment our upper bound by showing a nearly matching lower bound with delay dependency $\Omega(\frac{\sqrt{n} \max_{a \in \mathcal{A}}[d(a)]}{\log(n \max_{a \in \mathcal{A}}[d(a)])})$, showing that the $\sqrt{n}$ dimensional dependency is intrinsic in linear bandits.

5.1 Upper Bound

Our algorithm (Algorithm 3, formally given in Algorithm 6) still follows a phased elimination framework, but the estimation step has to be modified to account for this censoring bias. In phase m, we first fix the entire collection of samples that will be used for estimation: we compute a design π_m over the active set $\mathcal{A}_m$, set the target number of samples $N_m = \widetilde{O}(\frac{n}{\epsilon_m^2} + \frac{n^{3/2}}{\epsilon_m})$, and play each arm a approximately $N_m(a) = N_m \pi_m(a)$ times.[6] The important point is that the estimation sample is determined before observing which feedback arrives. Once these designated samples have been collected, the learner continues playing arbitrary active arms solely to allow the feedback from the estimation pulls to arrive. The phase terminates once at least a $1 - \epsilon_m/\sqrt{n}$ fraction of the designated samples for every arm in the design has been observed. As we will demonstrate later, this specific fraction is the minimum required to ensure sufficient estimation accuracy for each action.

At the end of the phase, a small fraction of the designated samples may still be missing. Since their delays may depend on their realized losses, simply ignoring them would introduce bias. Instead, we construct directional confidence estimates. For every candidate arm a, we form two arm-dependent estimators: a lower estimator $\widehat{\theta}^-_{m,a}$ and an upper estimator $\widehat{\theta}^+_{m,a}$ for the loss of arm a. We construct these two estimators for each action individually because of the dependency among actions: an estimator that overestimates the loss of one action may not overestimate some other actions.

Concretely, for a missing sample generated by arm b, its contribution to the prediction of arm a enters through $a^\top V_m^{-1} b$, where $V_m \triangleq \sum_{a \in \mathrm{Supp}(\pi_m)} N_m(a) a a^\top$. Therefore, to form the upper estimate for arm a, we complete this missing observation in the direction that maximizes the predicted loss of a, using the sign of $a^\top V_m^{-1} b$. The lower estimate uses the opposite completion. Thus, although the learner cannot form a single unbiased least-squares estimator from the partially observed data, it can

[6] The $O(n^{3/2}/\epsilon_m)$ part is due to the fact that we are interested in the quintile $1 - \epsilon_m/\sqrt{n}$, and to get samples from this quintile one needs $O(\sqrt{n}/\epsilon_m)$ samples per arms, one needs $O(n^{3/2}/\epsilon_m)$ samples overall.

Algorithm 3 Phased Elimination w/ Loss-Dependent Delays (Formally in Algorithm 6)

Initialize active set $\mathcal{A}_1 \leftarrow \mathcal{A}$, accuracy $\epsilon_1 \leftarrow \frac{1}{2}$
for phases $m = 1, 2, \ldots$ **do**
  Compute a design π_m over $\mathcal{A}_m$ and set target samples $N_m = \tilde{O}\left(\frac{n}{\epsilon_m^2} + \frac{n^{3/2}}{\epsilon_m}\right)$
  Play each arm a approximately $N_m(a) = N_m \pi_m(a)$ times
  while some arm a has fewer than $(1 - \frac{\epsilon_m}{\sqrt{n}})N_m(a)$ observed samples **do**
    Play arbitrary $a' \in \mathcal{A}_m$
    Collect delayed feedback from the initial samples
  for each arm $a \in \mathcal{A}_m$ **do**
    Construct optimistic estimator $\widehat{\theta}^+_{m,a}$ and pessimistic estimator $\widehat{\theta}^-_{m,a}$ using Equation (5).
  $\mathcal{A}_{m+1} \leftarrow \left\{a \in \mathcal{A}_m : \langle \widehat{\theta}^-_{m,b}, b\rangle - \langle \widehat{\theta}^+_{m,a}, a\rangle \le 6\epsilon_m \quad \forall b \in \mathcal{A}_m\right\}$
  Set $\epsilon_{m+1} \leftarrow \epsilon_m/2$

still bracket the loss of each arm in the specific direction needed for elimination. (See Equation (5) in Section D.)

The elimination rule is a robust analogue of phased elimination: an arm a is eliminated if there exists another arm b whose pessimistic estimated loss is smaller than the optimistic estimated loss of a.

The idea of optimistic and pessimistic estimation appears in the MAB analysis of Lancewicki et al. [18]. Here, however, the challenge is to define optimism and pessimism in the linear setting, where missing observations affect each arm differently. The following lemma, proved in the appendix, shows that these estimators are both valid and sufficiently accurate

Lemma 5.1 *For every phase m and $a \in \mathcal{A}_m$, $0 \le \langle \widehat{\theta}^+_{m,a} - \widehat{\theta}_m, a\rangle \le 2\epsilon_m$, and $0 \le \langle \widehat{\theta}_m - \widehat{\theta}^-_{m,a}, a\rangle \le 2\epsilon_m$, where $\widehat{\theta}_m$, defined in Definition D.6, is the estimator if we had all N_m samples.*

Thus, $\widehat{\theta}^+_{m,a}$ is optimistic and $\widehat{\theta}^-_{m,a}$ is pessimistic for arm a, and observing only a $1 - \frac{\epsilon_m}{\sqrt{n}}$ fraction of the samples introduces only an additional $O(\epsilon_m)$ error. Since the optimistic and pessimistic errors are of order ϵ_m, the standard phased elimination analysis again implies that every arm surviving phase m is $O(\epsilon_m)$-suboptimal.

It remains to bound the phase length T_m. Since we wait for a $1 - \frac{\epsilon_m}{\sqrt{n}}$ fraction of the feedback to arrive, Lemma D.3 shows that, with high probability, this requires an additional $\mathcal{Q}(1 - \frac{\epsilon_m}{\sqrt{n}})$ rounds. Applying Markov's inequality yields $T_m = O(N_m + \frac{\sqrt{n}\max_a \mathbb{E}[d(a)]}{\epsilon_m})$.

Since each active arm has suboptimality $O(\epsilon_m)$, the regret incurred in phase m is bounded by $O\left(N_m\epsilon_m + \sqrt{n}\max_a \mathbb{E}[d(a)]\right)$. Summing over phases gives the following bound.

Theorem 5.2 *Algorithm 6 guarantees that*

$$\mathcal{R}_T = O\left(\sqrt{nT\log(KT)} + \log(T)^2 \log\log(n)\, n^{3/2} + \sqrt{n}\log(T)\max_a \mathbb{E}[d(a)]\right).$$

5.2 Lower Bound

In this section, we show that the dependence on $\sqrt{n}$ in the delay-dependent term in our upper bound shown in Theorem 5.2 is necessary up to logarithmic factors.

Theorem 5.3 *For any $\bar{d}, n \ge 32\log(\bar{d}n), T \ge \frac{\sqrt{n}\bar{d}}{\sqrt{\log(\bar{d}n)}}$ there exists a distribution over loss-dependent delayed linear bandit instances with expected delay for all arms bounded by $\bar{d}$ such that any algorithm will suffer a regret of $\Omega(\frac{\sqrt{n}\bar{d}}{\sqrt{\log(\bar{d}n)}})$ with a constant probability over the distribution and the randomness of the algorithm.*

The proof of Theorem 5.3 is deferred to Appendix D.2. The argument leverages a standard construction of a "hard" action set, establishing the existence of a collection of K unit vectors in n-dimensional space such that all pairs of distinct actions are nearly orthogonal.

Lemma 5.4 (Lemma 3.1 in Lattimore et al. [20]) *There exists a set of actions* $\{a_1, \dots, a_K\} \subset \mathbb{R}^n$ *such that* $\|a_i\|_2 = 1 \; \forall i$ *and* $|\langle a_i, a_j \rangle| \leq \sqrt{\frac{8\log(K)}{n}} \; \forall i \neq j$.

To provide a proof sketch for Theorem 5.3, let $q = \sqrt{\frac{8\log(n\overline{d})}{n}}$. We utilize the hard action set from Lemma 5.4 with $K = \lceil 2\overline{d}/q \rceil$ and choose $\theta^* = -a_{i^*}$ uniformly at random. Under this construction, arm i^* has a loss of -1, while every other arm $i \neq i^*$ satisfies $|\langle a_i, \theta^* \rangle| \leq q$.

Consequently, the mean losses of all suboptimal arms fall within a q-window of one another. The delay distribution leverages this q-level proximity to obscure the information that distinguishes the suboptimal arms. Specifically, the q-quantile of each suboptimal arm is delayed by d/q, while any feedback received immediately is distributed identically across all suboptimal arms. Therefore, for any $t < \overline{d}/q$, pulling any arm other than a_{i^*} reveals no information about which arm is optimal; all suboptimal arms remain indistinguishable to the learner.

Since i^* is chosen uniformly from $K = 2\overline{d}/q$ arms, any algorithm will sample the optimal arm during the first $\overline{d}/q$ rounds with probability at most $1/2$. Thus, with constant probability, the learner receives no informative feedback before time $\overline{d}/q$ and incurs a regret of $\Omega(\overline{d}/q) = \Omega\left(\frac{\sqrt{n\overline{d}}}{\log(n\overline{d})}\right)$.

6 Delay as Payoff

We now consider the *delay-as-payoff* setting, a special case of delay-dependent delay where the delay is proportional to the loss, i.e., $d_t = D \cdot \ell_t$. This model introduces a strong coupling between losses and delays, and has been studied in the multi-armed bandit setting, where it is known that the regret admits an additive term of order d^*, where d^* is the expected delay of the optimal arm. A follow-up work by [35] provides an upper bound of order D in the linear bandits case.

A natural question is whether this D dependence is unavoidable in linear bandits or it can be improved to d^* similar to the MAB case. In the following theorem, we show that in contrast to MAB, an additive dependence of order d^* is not achievable in linear bandits. In fact, even when $d^* = 0$, an additive $\Omega(D)$ term is unavoidable, showing that the delay dependency obtained in [35] is optimal.

Theorem 6.1 *For any* $n \geq 24, K \leq e^{n/100}, D \leq 10K, T \geq D$, *there exists a distribution over delay-as-payoff linear bandit instances with no noise and* $d^* = 0$ *for which any algorithm suffers regret* $\Omega(D)$ *with constant probability.*

Proof Sketch. The construction ensures that one action has zero delay, while all other actions incur delay at least $\Omega(D)$. Thus, the only way to obtain feedback early in the game is to identify and play this special action. We use the probabilistic method to construct a large set of actions $\{a_1, \dots, a_K\} \subset \mathbb{R}^n$ and $\theta^\star$ so that for a uniformly random index i^*,$\langle a_{i^*}, \theta^\star \rangle = 0$, while $\langle a_i, \theta^\star \rangle \geq c$ for all $i \neq i^*$ for some constant $c > 0$. This implies that the optimal action a_{i^*} has delay $d^* = 0$, whereas all other actions incur delay at least $\Omega(D)$.

As a result, unless the algorithm plays a_{i^*} early on, it receives no feedback for $\Omega(D)$ rounds. Since i^* is chosen uniformly at random and the number of actions is large, with constant probability the optimal action is not played in the first $\Theta(D)$ rounds, leading to regret $\Omega(D)$. □

We remark that the gap between MAB and linear bandits in the delay-as-payoff setting stems from a fundamental difference in how information is acquired. In MAB, the d^* bound arises because after roughly d^* rounds, the learner receives feedback from the optimal arm. Moreover, arms that have accumulated a large amount of missing feedback can safely be inferred to be suboptimal, since each arm is explored independently.

In contrast, linear bandits do not allow for such independent exploration. Learning about a single action requires sampling other actions, as information is shared through the underlying parameter $\theta^\star$. As a result, a large amount of missing feedback associated with a particular action does not imply that the action is suboptimal. It may simply reflect that the actions used to explore it yield large delays. This coupling between actions prevents the learner from ruling out suboptimal directions based on missing feedback alone, and leads to the unavoidable $\Omega(D)$ regret even when $d^* = 0$.

7 Discussion

We studied delayed feedback in linear bandits under several models of delay, identifying a clear separation between settings in which linear bandits behave similarly to MAB and those in which they fundamentally differ. In particular, extending from arm-independent to arm-dependent delays preserves the qualitative dependence on delay and allows MAB-style guarantees without dimension dependence, whereas loss-dependent delays lead to a fundamentally harder problem in which dimension becomes unavoidable. An important direction for future work is to understand adversarial loss-dependent delays, a setting that remains unexplored even in MAB. More broadly, our results highlight that the impact of delay depends critically on how it interacts with the feedback structure, and it would be interesting to investigate whether a similar dichotomy arises in richer settings such as contextual bandits or MDPs, which have a different structure of information.

Acknowledgements

OS and YM are supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (grant agreement No. 882396), by the Israel Science Foundation (1357/24), the Blavatnik foundation at Tel Aviv University and by a grant from the Tel Aviv University Center for AI and Data Science (TAD). OS is also supported by the TAD Excellence Program for Doctoral Students in Artificial Intelligence and Data Science from the Tel Aviv University Center for AI and Data Science (TAD) and from the Israeli Council for Higher Education (CHE) Fellowship for Outstanding PhD Students in Data Science.

References

[1] Yasin Abbasi-yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc., 2011.

[2] Naoki Abe and Philip M. Long. Associative reinforcement learning using linear probabilistic concepts. In *Proceedings of the Sixteenth International Conference on Machine Learning*, ICML '99, page 3–11, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc. ISBN 1558606122.

[3] Peter Auer. Using confidence bounds for exploitation-exploration trade-offs. *J. Mach. Learn. Res.*, 3(null):397–422, March 2003. ISSN 1532-4435.

[4] Jose Blanchet, Renyuan Xu, and Zhengyuan Zhou. Delay-adaptive learning in generalized linear contextual bandits. *Mathematics of Operations Research*, 49(1):326–345, 2024.

[5] Sébastien Bubeck, Nicolò Cesa-Bianchi, and Sham M. Kakade. Towards minimax policies for online linear optimization with bandit feedback, 2012.

[6] Nicol'o Cesa-Bianchi, Claudio Gentile, Yishay Mansour, and Alberto Minora. Delay and cooperation in nonstochastic bandits. In *Conference on Learning Theory*, pages 605–622, 2016.

[7] Varsha Dani, Thomas P Hayes, and Sham M Kakade. Stochastic linear optimization under bandit feedback. In *21st Annual Conference on Learning Theory*, pages 355–366, 2008.

[8] Christoph Dann, Tor Lattimore, and Emma Brunskill. Unifying pac and regret: Uniform pac bounds for episodic reinforcement learning. *Advances in Neural Information Processing Systems*, 30, 2017.

[9] Miroslav Dudik, Daniel Hsu, Satyen Kale, Nikos Karampatziakis, John Langford, Lev Reyzin, and Tong Zhang. Efficient optimal learning for contextual bandits. In *Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence*, pages 169–178, 2011.

[10] Yonathan Efroni, Nadav Merlis, Aadirupa Saha, and Shie Mannor. Confidence-budget matching for sequential budgeted learning. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 2937–2947. PMLR, 18–24 Jul 2021.

[11] Valerii Fedorov. *Theory of Optimal Experiments Designs*. Academic Press, New York, 01 1972.

[12] Marguerite Frank and Philip Wolfe. An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, 3:95–110, 1956.

[13] Manegueu Anne Gael, Claire Vernade, Alexandra Carpentier, and Michal Valko. Stochastic bandits with arm-dependent delays. In *International Conference on Machine Learning*, pages 3348–3356. PMLR, 2020.

[14] Andras Gyorgy and Pooria Joulani. Adapting to delays and data in adversarial multi-armed bandits. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 3988–3997. PMLR, 18–24 Jul 2021.

[15] Benjamin Howson, Ciara Pike-Burke, and Sarah Filippi. Delayed feedback in generalised linear bandits revisited. In *International Conference on Artificial Intelligence and Statistics*, pages 6095–6119. PMLR, 2023.

[16] Pooria Joulani, Andras Gyorgy, and Csaba Szepesvári. Online learning under delayed feedback. In *International Conference on Machine Learning*, pages 1453–1461, 2013.

[17] J. Kiefer and J. Wolfowitz. The equivalence of two extremum problems. *Canadian Journal of Mathematics*, 12:363–366, 1960. doi: 10.4153/CJM-1960-030-4.

[18] Tal Lancewicki, Shahar Segal, Tomer Koren, and Yishay Mansour. Stochastic multi-armed bandits with unrestricted delay distributions. In *International Conference on Machine Learning*, pages 5969–5978. PMLR, 2021.

[19] Tor Lattimore and Csaba Szepesvári. *Bandit Algorithms*. Cambridge University Press, 2020.

[20] Tor Lattimore, Csaba Szepesvari, and Gellert Weisz. Learning with good feature representations in bandits and in rl with a generative model. In *International conference on machine learning*, pages 5662–5670. PMLR, 2020.

[21] Saeed Masoudian, Julian Zimmert, and Yevgeny Seldin. A best-of-both-worlds algorithm for bandits with delayed feedback. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 11752–11762. Curran Associates, Inc., 2022.

[22] Saeed Masoudian, Julian Zimmert, and Yevgeny Seldin. A best-of-both-worlds algorithm for bandits with delayed feedback with robustness to excessive delays. *Advances in Neural Information Processing Systems*, 37:141071–141102, 2024.

[23] Ciara Pike-Burke, Shipra Agrawal, Csaba Szepesvari, and Steffen Grunewalder. Bandits with delayed, aggregated anonymous feedback. In *International Conference on Machine Learning*, pages 4105–4113. PMLR, 2018.

[24] Paat Rusmevichientong and John N Tsitsiklis. Linearly parameterized bandits. *Mathematics of Operations Research*, 35(2):395–411, 2010.

[25] Ofir Schlisselberg, Ido Cohen, Tal Lancewicki, and Yishay Mansour. Delay as payoff in mab. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(19):20310–20317, Apr. 2025. doi: 10.1609/aaai.v39i19.34237.

[26] Ofir Schlisselberg, Tal Lancewicki, Peter Auer, and Yishay Mansour. Improved best-of-both-worlds regret for bandits with delayed feedback. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[27] SD Silvey and B Sibson. Discussion of dr. wynn's and of dr. laycock's papers. *Journal of Royal Statistical Society (B)*, 34(174-175):271, 1972.

[28] Yifu Tang, Yingfei Wang, and Zeyu Zheng. Stochastic multi-armed bandits with strongly reward-dependent delays. In Sanjoy Dasgupta, Stephan Mandt, and Yingzhen Li, editors, *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*, volume 238 of *Proceedings of Machine Learning Research*, pages 3043–3051. PMLR, 02–04 May 2024.

[29] Tobias Sommer Thune, Nicolò Cesa-Bianchi, and Yevgeny Seldin. Nonstochastic multiarmed bandits with unrestricted delays. In *Advances in Neural Information Processing Systems*, pages 6541–6550, 2019.

[30] Michal Valko, Rémi Munos, Branislav Kveton, and Tomáš Kocák. Spectral bandits for smooth graph functions. In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, ICML'14, page II–46–II–54. JMLR.org, 2014.

[31] Dirk Van Der Hoeven and Nicolò Cesa-Bianchi. Nonstochastic bandits and experts with arm-dependent delays. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 2022.

[32] Dirk Van Der Hoeven, Lukas Zierahn, Tal Lancewicki, Aviv Rosenberg, and Nicolò Cesa-Bianchi. A unified analysis of nonstochastic delayed feedback for combinatorial semi-bandits, linear bandits, and mdps. In *The Thirty Sixth Annual Conference on Learning Theory*, pages 1285–1321. PMLR, 2023.

[33] Claire Vernade, Olivier Cappé, and Vianney Perchet. Stochastic bandit models for delayed conversions. In *Conference on Uncertainty in Artificial Intelligence*, 2017.

[34] Claire Vernade, Alexandra Carpentier, Tor Lattimore, Giovanni Zappella, Beyza Ermis, and Michael Brueckner. Linear bandits with stochastic delayed feedback. In *International Conference on Machine Learning*, pages 9712–9721. PMLR, 2020.

[35] Mengxiao Zhang, Yingfei Wang, and Haipeng Luo. Contextual linear bandits with delay as payoff. *International Conference on Machine Learning*, 2025.

[36] Zhengyuan Zhou, Renyuan Xu, and Jose Blanchet. Learning in generalized linear contextual bandits with stochastic delays. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

A General Phased Elimination Lemmas

We slightly abuse notation and write $(\cdot)_m$ to indicate this quantity at the end of phase m, e.g σ_m is the number of outstanding observation at the end of phase m and $\sigma_m(a)$ is the number of outstanding observation for action a at the end of phase m.

This section is for general lemmas for Algorithms 4 to 6. All those algorithms are based on Phased Elimination with some common notations: N_m for the needed samples, $N_m(a) \triangleq \lceil N_m \pi_m(a) \rceil$, $V_m \triangleq \sum_{a \in \mathcal{A}} N_m(a) a a^\top$, and S_m for the set of observed samples in the end of the phase m.

The following lemma shows that similar designs provide similar g values .

Lemma A.1 *Let π, π' be two distributions over $\mathcal{A}$ such that for every $a \in \mathcal{A}$:*

$$\pi(a) \leq 2\pi'(a)$$

Recall that $g(\pi) = \max_{a \in \mathcal{A}} \|a\|_{V(\pi)^{-1}}$, with $V(\pi) = \sum_{a \in \mathcal{A}} \pi(a) a a^\top$. Thus:

$$g(\pi') \leq 2g(\pi).$$

Proof:

$$2V(\pi') = \sum_a 2\pi'(a) a a^T \succeq \sum_a \pi(a) a a^T = V(\pi) \implies V(\pi)^{-1} \succeq \left(2V(\pi')\right)^{-1}$$

Thus, for every $a \in \mathcal{A}$:

$$\begin{aligned} &\|a\|_{V(\pi)^{-1}} \geq \|a\|_{(2V(\pi'))^{-1}} = \frac{1}{2} \|a\|_{V(\pi')^{-1}} \\ &\implies g(\pi') = \max_a \|a\|_{V(\pi')^{-1}} \leq 2 \max_a \|a\|_{V(\pi)^{-1}} = 2g(\pi) \end{aligned}$$

■

Lemma A.2 *Let π be a design with $|\mathrm{Supp}(\pi)| = \psi$, then there is a design π' with $\min_{a:\pi(a)>0} \pi'(a) \geq 1/2\psi$ and $g(\pi') \leq 2g(\pi)$.*

Proof: Let f be the following mapping:

$$f(a) = \begin{cases} \frac{1}{\psi} & 0 < \pi(a) < \frac{1}{\psi}, \\ \pi(a) & else. \end{cases}$$

Now denote π' to be the normalized version of f:

$$\pi(a) = \frac{f(a)}{\sum_a f(a)}.$$

Since the support of π is bounded by ψ:

$$\sum_a f(a) \leq \sum_a \pi(a) + \psi \frac{1}{\psi} = 2.$$

Since $\min_{a:f(a)>0} f(a) \geq \psi$, we have the same for $1/2\psi$ for π'. Additionally, we have that for every a, $\pi(a) \leq 2\pi'(a)$, and thus Lemma A.1 concludes the proof. ■

Lemma A.3 *There is a design π with $\min_{a:\pi(a)>0} \pi(a) \geq \frac{1}{n \log\log(n)}$ and $g(\pi) \leq 4n$.*

Proof: We use the design π' from Note 21.2.3 in [19], which has a support of $n \log\log(n)$ and $g(\pi') \leq 2n$. Lemma A.2 concludes the construction. ■

The following is general good event that will be used in the analysis of Algorithms 4 to 6.

Definition A.4 *Let $\mathcal{E}$ be the event that, for every phase m and action a, at the end of phase m:*

$$\left| \sum_{t \in S_m} a^T V_m^{-1} a_t \eta_t \right| \leq \sqrt{4 \|a\|^2_{V_m^{-1}} \log(TK)}. \tag{1}$$

Next, we prove that this event happens with high probability.

Lemma A.5

$$\Pr[\mathcal{E}] \geq 1 - \frac{1}{T}.$$

Proof: Fix phase m and arm a. Notice that the sum in LHS of Equation (1) is a sum of 1-subgaussian r.v. According to Hoeffding' inequality, with probability at least $1 - \frac{1}{T^2 K}$, we have

$$\left| \sum_{t \in S_m} a^T V_m^{-1} a_t \eta_t \right| \leq \sqrt{4 \sum_{t \in S_m} \left(a^T V_m^{-1} a_t\right)^2 \log(TK)}. \tag{2}$$

We note that the design of phase m and the number of times each $a \in \text{Supp}(\pi_m)$ is present in the above summation is in the filtration of the beginning of phase m.

Taking a union bound over all $a \in \mathcal{A}_m$ and $m \in [T]$, Equation (2) is true with probability $1 - \frac{1}{T}$ for all $a \in \mathcal{A}_m$ and $m \in [T]$.

To bound the inner part we need:

$$\begin{aligned}
\sum_{t \in S_m} \left(a^T V_m^{-1} a_t\right)^2 &= \sum_{t \in S_m} a^T V_m^{-1} a_t a_t^T V_m^{-1} a \\
&= a^T V_m^{-1} \left(\sum_{t \in S_m} a_t a_t^T \right) V_m^{-1} a \\
&= a^T V_m^{-1} \left(N_m(a) a_t a_t^T \right) V_m^{-1} a \\
&= a^T V_m^{-1} a \\
&= \|a\|^2_{V_m^{-1}},
\end{aligned} \tag{3}$$

where we used the fact that in the end of the phase, $|S_m(a)| = N_m(a)$.

Incorporating Equation (3) in Equation (2) we get:

$$\left| \sum_{t \in S_m} a^T V_m^{-1} a_t \eta_t \right| \leq \sqrt{4 \|a\|^2_{V_m^{-1}} \log(TK)}.$$

■

The following lemma shows that when we have enough samples, in the direction of the design, we can safely bound the g function.

Lemma A.6 *Assume that there is some N such that $N_m(b) \geq N \cdot \pi_m(b)$ for all $b \in \text{Supp}(\pi_m)$. Then, for every $a \in \mathcal{A}_m \subseteq \mathbb{R}^n$:*

$$\|a\|^2_{V_m^{-1}} \leq \frac{4n}{N}.$$

Proof:

$$\begin{aligned}
V_m = \sum_a N_m(a) a a^T &\succeq N \sum_a \pi_m(a) a a^T = N V(\pi) \\
\implies V_m^{-1} &\preceq \frac{1}{N} V(\pi_m)^{-1} \\
\implies \|a\|_{V_m^{-1}} &\leq \frac{1}{N} \|a\|_{V(\pi_m)^{-1}} \leq \frac{4n}{N}.
\end{aligned}$$

■

This is a core lemma in the analysis of Algorithms 4 to 6. We show that if we have enough samples in the direction of the design, we can safely bound the estimation error. In Algorithms 4 and 5 the algorithm actually have the needed number of samples and thus $\hat{\theta}_m$ is known to the algorithm. In Algorithm 6 doesn't have enough samples, so this quantity is unknown to the algorithm.

Lemma A.7 *Assume $\mathcal{E}$ and phase m, and let*

$$\hat{\theta}_m \triangleq V_m^{-1} \sum_{a \in \mathrm{Supp}(\pi_m)} a \sum_{t \in S_m(a)} \ell_t,$$

where $S_m(a)$ is a set of time steps with i.i.d losses such that $|S_m(a)| \geq \frac{16n\log(TK)\pi_m(a)}{\epsilon_m^2}$. Then for every action $a \in \mathcal{A}_m$, at the end of phase m, we have

$$\left|\langle \widehat{\theta}_m - \theta^\star,\, a\rangle\right| \leq \epsilon_m.$$

Proof: We short $V \coloneqq V_m$, $\widehat{\theta} \coloneqq \widehat{\theta}_m$. Then direct calculation shows that

$$\begin{aligned}
\langle \widehat{\theta} - \theta^\star,\, a\rangle &= \sum_{t \in S_m(a)} a^T \left(V^{-1} a_t \ell_t - \theta^\star\right) \\
&= \sum_{t \in S_m(a)} a^T \left(V^{-1} a_t \left(a_t^T \theta^\star + \eta_t\right) - \theta^\star\right) \\
&= \sum_{t \in S_m(a)} a^T \left(V^{-1} a_t a_t^T \theta^\star - \theta^\star + V^{-1}\eta_t\right) \\
&= a^T \left(V^{-1}\left(\sum_{t \in S_m(a)} a_t a_t^T\right)\theta^\star - \theta^\star + \sum_{t \in S_m(a)} V^{-1} a_t \eta_t\right) \\
&= \sum_{t \in S_m(a)} a^T V^{-1} a_t \eta_t.
\end{aligned}$$

From $\mathcal{E}$ and Lemma A.6, we know that

$$\begin{aligned}
\left|\langle \widehat{\theta} - \theta^\star,\, a\rangle\right| &\leq \sqrt{4\left\|a\right\|_{V^{-1}}^2 \log\left(TK\right)} \\
&\leq \sqrt{\frac{16n\log\left(TK\right)\epsilon_m^2}{16n\log\left(KT\right)}} \\
&= \epsilon_m.
\end{aligned}$$

■

The following lemma shows that N_m is big enough so that the fact that $N_m(a)$ can be $N_m\pi_m(a)+1$ does affect asymptotically.

Lemma A.8 *For every phase m, assume $N_m > n\log(K)$, then we have*

$$\sum_a N_m(a) \leq 2N_m.$$

Proof:

$$\begin{aligned}
\sum_a N_m(a) &\leq \sum_{a\,:\,\pi_m(a)>0} N_m\pi_m(a) + 1 \\
&= \mathrm{Supp}(\pi_m) + \sum_a N_m\pi_m(a) \\
&\leq n\log\log(n) + N_m \\
&< 2N_m.
\end{aligned}$$

The last follows from the fact that $N_m > n\log(K) > n\log(n) > n\log\log(n)$. ■

B Omitted Details in Section 3

B.1 Algorithm

In this section, we provide Algorithm 4, the formal version of Algorithm 2 shown in the main text.

Algorithm 4 Phased Elimination for Delayed Linear Bandits

Require: Action set $\mathcal{A} \subset \mathbb{R}^d$

Initialize $\mathcal{A}_1 \leftarrow \mathcal{A}$, $\epsilon_1 \leftarrow \frac{1}{2}$
for $m = 1, 2, \ldots$ **do**
  Compute a design π_m over $\mathcal{A}_m$ from Lemma A.3
  $N_m = \frac{16n \log(KT)}{\epsilon_m^2}$
  $N_m(a) \leftarrow \lceil N_m \pi_m(a) \rceil$
  $S_m(a) \leftarrow \emptyset$ for all $a \in \mathcal{A}_m$ ▷ *observations collected in phase m for action a*
  while there exists $a \in \mathcal{A}_m$ such that $|S_m(a)| < N_m(a)$ **do**
    Sample $a_t \sim \pi_m$
    Receive all losses that arrive at round t
    for each newly arrived loss generated in phase m from action $a \in \mathcal{A}_m$ **do**
      Add this loss to $S_m(a)$
  Set $V_m \triangleq \sum_{a \in \mathrm{Supp}(\pi_m)} N_m(a) a a^T$
  $\widehat{\theta}_m \leftarrow V_m^{-1} \sum_{a \in \mathrm{Supp}(\pi_m)} a \sum_{t \in S_m(a)} \ell_t$
  Compute estimator $\widehat{\theta}_m$ using data from phase m
  $\mathcal{A}_{m+1} \leftarrow \left\{ a \in \mathcal{A}_m : \langle \widehat{\theta}_m, a - b \rangle \le 2\epsilon_m \quad \forall b \in \mathcal{A}_m \right\}$
  $\epsilon_{m+1} \leftarrow \epsilon_m / 2$

B.2 Analysis

We first show that, as long as $\mathcal{E}$ holds (Definition A.4), the optimal arm isn't eliminated. This is critical when we'll analyze below the elimination time of suboptimal arms.

Lemma B.1 *Assume $\mathcal{E}$ (Definition A.4), then we have for every m, $a^* \in \mathcal{A}_m$.*

Proof: We prove by induction. The base holds trivially since $\mathcal{A}_1 = \mathcal{A}$. Assume by contradiction that $a^* \in \mathcal{A}_m$ but $a^* \notin \mathcal{A}_{m+1}$, namely it was eliminated in phase m. This means there is an action $b \in \mathcal{A}$ such that:

$$\langle \widehat{\theta}_m, a - b \rangle > 2\epsilon_m.$$

From Lemma A.7:

$$\begin{aligned} 2\epsilon_m &< \langle \widehat{\theta}_m, -b \rangle \\ &= \langle \widehat{\theta}_m - \theta^\star, a^* \rangle + \langle \theta^\star, a^* - b \rangle + \langle \theta^\star - \widehat{\theta}_m, b \rangle \\ &\le \langle \theta^\star, a^* - b \rangle + 2\epsilon_m \\ \implies 0 &< \langle \theta^\star, a^* - b \rangle, \end{aligned}$$

which contradicts the fact that a^* is the optimal arm. ■

Here we use the fact that the optimal arm is never eliminated to bound when an arm is eliminated.

Lemma B.2 *Assume $\mathcal{E}$ (Definition A.4) and action a such that $a \in \mathcal{A}_m$. Then, $\Delta(a) \le 8\epsilon_m$.*

Proof: From Lemma B.1 also $a^* \in \mathcal{A}_m$. If $a \in \mathcal{A}_m$, we know that:

$$\begin{aligned} 2\epsilon_{m-1} &\ge \langle \widehat{\theta}_{m-1}, a - a^* \rangle \\ &= \langle \widehat{\theta}_{m-1} - \theta^\star, a \rangle + \langle \theta^\star, a - a^* \rangle + \langle \theta^\star - \widehat{\theta}_{m-1}, a^* \rangle \\ &\ge \langle \theta^\star, a - a^* \rangle - 2\epsilon_{m-1} \\ \implies \Delta(a) &\le 4\epsilon_{m-1} \le 8\epsilon_m, \end{aligned}$$

where the second inequality is from Lemma A.7. ■

We now construct a general lemma to bound the regret given a bound to each phase. We will use this lemma multiple times in the followings.

Lemma B.3 *Assume that there is some $\alpha, \beta > 0$ such that, with probability at least $1 - \frac{1}{T}$, $T_m \leq \alpha N_m + \beta$, then:*

$$\mathcal{R}_T = O\Big(\sqrt{\alpha n T \log(KT)} + \beta\Big).$$

Proof: Assume event $\mathcal{E}$ holds, which occurs with probability at least $1 - \frac{1}{T}$. From Lemma B.2, we can assume that in the m-th phase, all actions incur a bounded regret hit of $8\epsilon_m$. Let $\tilde{T}_m \triangleq \alpha N_m + \beta$ denote the maximal phase length. If the phase lengths were exactly $T_m = \tilde{T}_m$ and the regret hit in every step was ϵ_m, this would only strictly increase our upper bound on the total regret. We proceed under this worst-case assumption to bound the realized regret.

Notice that a phase with $T_m = T$ will be the last. Thus:

$$\begin{aligned} T &> \frac{16\alpha n \log(KT)}{\epsilon_m^2} = 16\alpha n 2^{2m} \log(KT) \\ \implies m &< \frac{1}{2} \log\left(\frac{T}{16\alpha n \log(KT)}\right). \end{aligned}$$

Denote the final phase as m_{end}.

Based on Lemma B.2 and our definition of T_m, we can bound the regret conditioned on $\mathcal{E}$ holding as follows:

$$\begin{aligned} &\sum_{m=1}^{m_{end}-1} 8 T_m \epsilon_m \\ &\leq \sum_{m=1}^{m_{end}-1} 8(\alpha N_m + \beta)\epsilon_m \\ &\leq \sum_{m=1}^{\infty} 8\beta 2^{-m} + \sum_{m=1}^{m_{end}-1} \alpha \frac{128 n \log(KT)}{\epsilon_m} \\ &\leq 8\beta + 128\alpha n \log(KT) \sum_{m=1}^{m_{end}-1} 2^m \\ &< 8\beta + 128\alpha n \log(KT) 2^{m_{end}} \\ &= 8\beta + 128\alpha n \log(KT) \sqrt{\frac{T}{16\alpha n \log(KT)}} \\ &= 8\beta + 32\sqrt{\alpha n T \log(KT)}. \end{aligned}$$

Finally, we compute the expected regret $\mathcal{R}_T$. The probability that either event $\mathcal{E}$ fails or the phase bounds do not hold is at most $\frac{2}{T}$. In the worst-case failure scenario, the maximum possible cumulative regret is bounded by T. Therefore, the failure case contributes at most $T \times \frac{2}{T} = 2$ to the expected regret. This yields:

$$\mathcal{R}_T \leq 8\beta + 32\sqrt{\alpha n T \log(KT)} + 2$$

This adds only a constant term to the expected regret, concluding the proof.

■

In the following, we prove Lemma 3.1. For convenience, we restate the lemma as follows.

Lemma 3.1 *Let T_m be the length of phase m of Algorithm 4. Under stochastic arm-dependent delays, with probability at least $1 - \frac{1}{T}$,*

$$T_m = O\left(\min_q \left\{ \frac{N_m + n \log(T) \log\log(n)}{q} + \max_a \mathcal{Q}(q; a) \right\}\right).$$

Proof: Fix some $q \in (0, 1]$. After T_m steps in this phase we have for every $a \in \mathcal{A}$:

$$\begin{aligned}
|S_m(a)| &= \sum_{t=1}^{T_m} \mathbb{1}[a_t = a \wedge t + d_t \leq T_m] \\
\implies \mathbb{E}[|S_m(a)|] &= \sum_{t=1}^{T_m} \Pr\left[a_t = a \wedge t + d_t \leq T_m\right] \\
&= \sum_{t=1}^{T_m} \Pr\left[a_t = a\right] \Pr\left[t + d_t \leq T_m \mid a_t = a\right] \\
&\geq \sum_{t=1}^{T_m - \mathcal{Q}(q;a)} \Pr\left[a_t = a\right] \Pr\left[d_t \leq \mathcal{Q}(q;a)\right] \\
&= (T_m - \mathcal{Q}(q;a))\pi_m(a)q.
\end{aligned}$$

From Lemma G.1, we know that with probability at least $1 - \frac{1}{T}$,

$$|S_m(a)| \geq \frac{1}{2}(T_m - \mathcal{Q}(q;a))\pi_m(a)q - \log(T).$$

Thus, on $T_m = \frac{2}{q}(2N_m + n\log\log(n)(\log(T) + 1)) + \max_a \mathcal{Q}(q;a)$, for all $a \in \mathcal{A}$:

$$|S_m(a)| \geq N_m\pi_m(a) + n\log\log(n)\pi_m(a)(\log(T) + 1)\pi_m(a) - \log(T) \geq N_m\pi_m(a) + 1 \geq N_m(a), \tag{4}$$

where the second inequality is from Lemma A.3. Equation (4) is the condition to end the phase, and thus on $T_m = \frac{2}{q}(2N_m + n\log\log(n)(\log(T) + 1)) + \max_a \mathcal{Q}(q;a)$ the phase is over. ■

Now we are ready to bound the total regret.

Theorem 3.2 *Algorithm 4 guarantees that under stochastic arm-dependent delays,*

$$\mathcal{R}_T = O\left(\min_q \left\{\sqrt{\frac{nT\log(KT)}{q}} + \frac{n\log(T)\log\log(n)}{q} + \max_a \mathcal{Q}(q;a)\right\}\right).$$

Proof: The proof is done by directly combining Lemmas B.3 and 3.1. ■

We now show that a simple Markov argument can give us the desired $\mathbb{E}[d(a)]$ dependent bound.

Lemma B.4 *For any random variable d with its corresponding qunatile $\mathcal{Q}$,*

$$\mathcal{Q}\left(\frac{1}{2}\right) \leq 2\mathbb{E}[d].$$

Proof: From Markov inequality:

$$\begin{aligned}
\frac{1}{2} &= \Pr\left[d \geq \mathcal{Q}\left(\frac{1}{2}\right)\right] \leq \frac{\mathbb{E}[d]}{\mathcal{Q}\left(\frac{1}{2}\right)} \\
\implies \mathcal{Q}\left(\frac{1}{2}\right) &\leq 2\mathbb{E}[d].
\end{aligned}$$

■

Corollary B.5 *The regret of Algorithm 4 is bounded by,*

$$\mathcal{R}_T = O\left(\sqrt{nT\log(KT)} + \max_a \mathbb{E}[d(a)]\right).$$

Proof: From Theorem 3.2:

$$\begin{aligned}
\mathcal{R}_T &= O\left(min_q\left\{\sqrt{\frac{nT\log(KT)}{q}} + \max_a \mathcal{Q}(q;a)\right\}\right) \\
&\leq \sqrt{2nT\log(KT)} + \max_a \mathcal{Q}(\frac{1}{2};a) \\
&\leq \sqrt{2nT\log(KT)} + \max_a \mathbb{E}[d(a)],
\end{aligned}$$

where the last is from Lemma B.4. ■

B.3 Arm Independent Adversarial Delay

In this section we show that the same algorithm (Algorithm 4) can achieve an improved regret in the arm-independent adversarial delay case.

Lemma B.6 *Algorithm 4 guarantees that with probability at least* $1 - \frac{1}{T}$*, for all* m*,*

$$T_m = O(N_m + n \log(T) \log\log(n) + \sigma_{\max}).$$

Proof: After T_m steps, notice for $S_m = \bigcup_a S_m(a)$, $|S_m| \geq T_m - \sigma_{\max}$. We have:

$$\mathbb{E}[|S_m(a)|] = \mathbb{E}\left[\sum_{t \in S_m} Pr[a_t = a]\right] \geq (T_m - \sigma_{\max})\pi_m(a).$$

From Lemma G.1, we know that with probability at least $1 - \frac{1}{T}$,

$$|S_m(a)| \geq \frac{1}{2}(T_m - \sigma_{\max})\pi_m(a) - \log(T).$$

Thus, on $T_m = 2(N_m + n \log\log(n) \log(T)) + \sigma_{\max}$, we have:

$$|S_m(a)| \geq N_m \pi_m(a) + n \log\log(n) \log(T) \pi_m(a) - \log(T) \geq N_m \pi_m(a) = N_m(a),$$

which ends the phase. ■

Theorem B.7

$$\mathcal{R}_T = O\left(\sqrt{nT \log(KT)} + \sigma_{\max}\right)$$

Proof: The proof is done by directly combining Lemmas B.3 and B.6. ■

C Omitted Detailed in Section 4

C.1 Algorithm

In this section, we show Algorithm 5, the formal version of Algorithm 2.

Algorithm 5 Phased Elimination for Loss Independent Delay

Require: Action set $\mathcal{A} \subset \mathbb{R}^d$

1: Initialize $\mathcal{A}_1 \leftarrow \mathcal{A}$, $\epsilon_1 \leftarrow \frac{1}{2}$
2: **for** $m = 1, 2, \ldots$ **do**
3: Compute a design π_m over $\mathcal{A}_m$ from Lemma A.3
4: $N_m = \frac{16n \log(KT)}{\epsilon_m^2}$
5: $N_m(a) \leftarrow \lceil N_m \pi(a) \rceil$
6: $S_m(a) \leftarrow \emptyset$ for all $a \in \mathcal{A}_m$ ▷ *observations collected in phase m for action a*
7: Play $N_m(a)$ times each $a \in \mathcal{A}$
8: **while** there exists $a \in \mathcal{A}_m$ such that $|S_m(a)| < N_m(a)$ **do**
9: Play $N_m(a) - |S_m(a)|$ times each a
10: Update $S_m(a)$ with newly arrived loss
11: $V_m = \sum_{a \in \text{Supp}(\pi_m)} N_m(a) a a^T$
12: $\widehat{\theta}_m \leftarrow V_m^{-1} \sum_{a \in \text{Supp}(\pi_m)} a \sum_{t \in S_m(a)} \ell_t$
13: $\mathcal{A}_{m+1} \leftarrow \left\{a \in \mathcal{A}_m : \langle \widehat{\theta}_m, a - b \rangle \leq 2\epsilon_m \quad \forall b \in \mathcal{A}_m\right\}$
14: $\epsilon_{m+1} \leftarrow \epsilon_m / 2$

C.2 Analysis

For some phase m, we denote the start times of the while-loop of Line 8 with $t_m^1, t_m^2, \ldots$. Namely, the time before the i-th iteration of the while-loop is t_m^i. Additionally, we denote $U_t(a)$ to be $N_m(a) - |S_m(a)|$ at time t.

Our goal in the next few lemmas is two bound the total length of the while loop in Line 8. The following Lemma is the main ingredient in the upcoming lemmas.

Lemma C.1 *Fix phase m, We have:*

$$i \cdot U_{t_m^i}(a) \leq \sigma_{t_m^i}(a).$$

Proof: We'll prove by induction on i. The base is $i = 1$ (before all iterations) - every sample in $N_m(a)$ that is not in $S_m(a)$ is a missing observation so $U_{t_m^1}(a) \leq \sigma_{t_m^1}(a)$.

Assume true for some i, in the next iteration each a was sampled $U_{t_m^i}(a)$ times. Assume samples of a was returned ν times. Notice that $U_{t_m^{i+1}}(a) = \max\left\{0, U_{t_m^i}(a) - \nu\right\}$. If $U_{t_m^{i+1}}(a) = 0$, it is trivial. Else:

$$\begin{aligned}
\sigma_{t_m^{i+1}}(a) &= \sigma_{t_m^i}(a) + U_{t_m^i}(a) - \nu \\
&\geq iU_{t_m^i}(a) + U_{t_m^i}(a) - \nu \\
&= iU_{t_m^i}(a) + U_{t_m^{i+1}}(a) \\
&\geq (i+1)U_{t_m^i}(a),
\end{aligned}$$

where the last inequality is because U only decreases. ■

We can now bound the total number of iterations of Line 8 as a simple consequence of Lemma C.1.

Lemma C.2 *The number of iterations of Line 8 in Algorithm 5 is bounded by $\sigma_{\max}$.*

Proof: Let ν be the number of iterations. It means that before the νth iteration, there was some a such that $U_{t_m^\nu}(a) \geq 1$. Thus:

$$\sum_a U_{t_m^\nu}(a) \geq 1.$$

From Lemma C.1:

$$\begin{aligned}
\nu &\leq \nu \sum_a U_{t_m^\nu}(a) \\
&\leq \sum_a \sigma_{t_m^\nu}(a) \\
&\leq \sigma_{\max}.
\end{aligned}$$

■

Another simple consequence of Lemma C.1 is the length of each iteration.

Lemma C.3 *The length of iteration i of Line 8 in Algorithm 5 is bounded by $\frac{\sigma_{\max}}{i}$.*

Proof: Since the length of the iteration is $\sum_a U_i(a)$, from Lemma C.1:

$$\begin{aligned}
U_i(a) &\leq \frac{\sigma_{i-1}(a)}{i} \\
\implies \sum_a U_i(a) &\leq \frac{\sigma_{\max}}{i}.
\end{aligned}$$

■

Now that we have a bound for both the number of iterations and the length of each one, we can bound the total length using a harmonic series bound.

Lemma 4.1 *Let T_m be the length of phase m in Algorithm 5. Then with probability at least $1 - \frac{1}{T}$, Algorithm 5 guarantees that $T_m = O(N_m + \sigma_{\max} \log \sigma_{\max})$.*

Proof: In this proof we call the part of the algorithm that samples $N_m(a)$ times "first part" and the iterations afterwards "second part".

The total length of the first part is given by Lemma A.8. The second part length, from Lemmas C.2 and C.3:

$$\sum_{i=1}^{\sigma_{\max}} \frac{\sigma_{\max}}{i} = O(\sigma_{\max} \log(\sigma_{\max})).$$

■

Finally, we prove our regret guarantee of Algorithm 5 in the arm-dependent adversarial delay case, which is to combine Lemmas B.3 and 4.1.

Theorem 4.2 *Algorithm 5 guarantees that* $\mathcal{R}_T = O\left(\sqrt{nT\log(KT)} + \sigma_{\max}\log(\sigma_{\max})\right)$.

Proof: We use the same analysis as for Algorithm 5 since, beside the phase length, it is the same algorithm. Thus, this is proven directly from Lemmas B.3 and 4.1. ■

C.3 Arm Independent Stochastic Delay

In this subsection we show that the same Algorithm 5 can have a regret guarantee in the stochastic delay case, under the assumption that the delay is arm-independent.

Lemma C.4 *If the delays are stochastic and arm-independent, with probability* $1 - \frac{1}{T}$*:*

$$\sigma_{\max} \leq 2\mathbb{E}[d] + 8\log T$$

Proof: For every $t \leq T$:

$$\begin{aligned}
\mathbb{E}[\sigma(t)] &= \sum_{s=1}^{t} \Pr[d > t - s] \\
&= \sum_{s=1}^{t} \sum_{i=t-s}^{\infty} \Pr[d = i] \\
&\leq \sum_{i=1}^{\infty} i \Pr[d = i] \\
&= \mathbb{E}[d].
\end{aligned}$$

Fix some $t \leq T$. From Lemma G.2, w.p $1 - \frac{1}{T^2}$:

$$\sigma(t) \leq 2\mathbb{E}[d] + 8\log T.$$

Taking a union bound over all $t \in [T]$ concludes the proof. ■

Corollary C.5 *If the delays are stochastic and arm-independent, Algorithm 5 promises a regret of,*

$$\mathcal{R}_T = O\Big(\sqrt{nT\log(KT)} + \mathbb{E}[d]\log(\mathbb{E}[d])\Big).$$

Proof: From Lemma C.4 and theorem 4.2:

$$\mathcal{R}_T = O\Big(\sqrt{nT\log(KT)} + \mathbb{E}[d]\log(\mathbb{E}[d]) + \log(T)\log\log(T)\Big) = O\Big(\sqrt{nT\log(KT)} + \mathbb{E}[d]\log(\mathbb{E}[d])\Big).$$

Where the second equality is since $\log(T)\log\log(T) = O(\sqrt{T})$. ■

D Omitted Details in Section 5

D.1 Omitted Details in Section 5.1

D.1.1 Algorithm

In this section, we provide Algorithm 6, the formal version of Algorithm 3. Before that, we first introduce some notations. For simplicity, we treat $\sigma_m(a)$ as the outstanding observations from the first $N_m(a)$ pulls (Line 7), i.e not including the arbitrary pulls afterwards.

Definition D.1 *For every phase m with design π_m, let $S_m(a)$ be the set of observed pulls of arm a. We define the optimistic and pessimistic estimators as,*

$$\begin{aligned}\widehat{\theta}^{+}_{m,a} &:= V_m^{-1} \sum_{b \in \mathrm{Supp}(\pi_m)} b \left(\sigma_m(b) Sign\big(a^T V_m^{-1} b\big) + \sum_{t \in S_m(b)} \ell_t \right), \\ \widehat{\theta}^{-}_{m,a} &:= V_m^{-1} \sum_{b \in \mathrm{Supp}(\pi_m)} b \left(-\sigma_m(b) Sign\big(a^T V_m^{-1} b\big) + \sum_{t \in S_m(b)} \ell_t \right)\end{aligned} \tag{5}$$

where we recall that $V_m \triangleq \sum_{a \in \mathrm{Supp}(\pi_m)} N_m(a) a a^\top$.

Algorithm 6 Phased Elimination for Dependent Delayed Linear Bandits

Require: Action set $\mathcal{A} \subset \mathbb{R}^d$

Initialize $\mathcal{A}_1 \leftarrow \mathcal{A}$, $\epsilon_1 \leftarrow \frac{1}{2}$
for $m = 1, 2, \ldots$ **do**
  Compute a design π_m over $\mathcal{A}_m$ from Lemma A.3
  $N_m = \max\left\{ \frac{48 \log(T) \log\log(n) n^{3/2}}{\epsilon_m}, \frac{16 n \log(KT)}{\epsilon_m^2} \right\}$
  $N_m(a) \leftarrow \lceil N_m \pi_\ell(a) \rceil$
  $S_m(a) \leftarrow \emptyset$ for all $a \in \mathcal{A}_m$ ▷ *observations collected in phase m for action a*
  Play $N_m(a)$ times each $a \in \mathcal{A}$
  while there exists $a \in \mathcal{A}_m$ such that $|S_m(a)| < \left(1 - \frac{\epsilon_m}{\sqrt{n}}\right) N_m(a)$ **do**
    Play arbitrary $a \in \mathcal{A}_m$
    Receive all losses that arrive at round t from the first N_m steps
    for each newly arrived loss generated in Line 7 from action $a \in \mathcal{A}_m$ **do**
      Add this loss to $S_m(a)$
  Set $V_m = \sum_{a \in \mathrm{Supp}(\pi_m)} N_m(a) a a^T$
  for $a \in \mathcal{A}_m$ **do**
    $\widehat{\theta}^{+}_{m,a} = V_m^{-1} \sum_{b \in \mathrm{Supp}(\pi_m)} b \Big(\sigma_m(b) Sign\big(a^T V_m^{-1} b\big) + \sum_{t \in S_m(b)} \ell_t \Big)$
    $\widehat{\theta}^{-}_{m,a} = V_m^{-1} \sum_{b \in \mathrm{Supp}(\pi_m)} b \Big(-\sigma_m(b) Sign\big(a^T V_m^{-1} b\big) + \sum_{t \in S_m(b)} \ell_t \Big)$
  $\mathcal{A}_{m+1} \leftarrow \left\{ a \in \mathcal{A}_m : \langle \widehat{\theta}^{-}_{m,a}, a \rangle - \langle \widehat{\theta}^{+}_{m,b}, b \rangle \le 6\epsilon_m \quad \forall b \in \mathcal{A}_m \right\}$
  $\epsilon_{m+1} \leftarrow \epsilon_m / 2$

D.1.2 Analysis

In this section, we still use the event $\mathcal{E}$ as a good event (Definition A.4), like in the previous sections. However, we need another good event defined as follows.

Definition D.2 *Let $\mathcal{E}'$ be the event that for every m,*

$$T_m \le N_m + \max_{a \in \mathcal{A}_m} \mathcal{Q}\left(1 - \frac{\epsilon_m}{3\sqrt{n}}; a\right).$$

Lemma D.3 *$\mathcal{E}'$ holds with probability at least $1 - \frac{1}{T}$.*

Proof: Fix some $a \in \mathrm{Supp}(\pi_m)$. When $T_m = N_m + \max_{a \in \mathcal{A}_m} \mathcal{Q}\left(1 - \frac{\epsilon_m}{3\sqrt{n}}; a\right)$, all $N_m(a)$ pulls of it were at least $\max_{a' \in \mathcal{A}_m} \mathcal{Q}\left(1 - \frac{\epsilon_m}{3\sqrt{n}}; a'\right) \ge \mathcal{Q}\left(1 - \frac{\epsilon_m}{3\sqrt{n}}; a\right)$ step ago, we have that each of those pulls has returned w.p at least $\left(1 - \frac{\epsilon_m}{3\sqrt{n}}\right)$. Thus, the number of missing pulls in the first N_m pulls is a sum of Bernoulli random variables, each w.p at least $\frac{\epsilon_m}{3\sqrt{n}}$.

From Lemma G.2, w.p $1-\frac{1}{T^2}$, the number of missing is bounded by $\frac{2\epsilon_m}{3\sqrt{n}}N_m(a)+8\log(T)$. Since $\pi_m(a)\geq\frac{1}{2n\log\log(n)}$, we have:

$$\begin{aligned} N_m &\geq \frac{48\log(T)\log\log(n)n^{3/2}}{\epsilon_m} \\ N_m(a) &\geq \frac{24\log(T)\sqrt{n}}{\epsilon_m} \\ \implies \frac{\epsilon_m}{3\sqrt{n}}N_m(a) &\geq 8\log(T). \end{aligned}$$

Thus, the number of missing pulls is bounded by $\frac{\epsilon_m}{\sqrt{n}}N_m(a)$. Taking a union bound over all m concludes the proof w.p $1-\frac{1}{T}$. ■

We define $\widehat{\theta}_m$ as a helper to bound the optimistic and pessimistic estimators. The idea is to first bound the difference between those estimators and $\widehat{\theta}_m$, and then use the already-constructed difference between $\widehat{\theta}_m$ and $\theta^\star$.

Definition D.4 *For any phase m and arm $a\in\mathrm{Supp}(\pi_m)$, denote $\tilde{S}_m(a)$ to be the set of time steps in which the first $N_m(a)$ pulls of arm a.*

Definition D.5 *Let $\tilde{\mathcal{E}}$ be the same event as $\mathcal{E}$ (Definition A.4) but w.r.t $\tilde{S}$ instead of S.*

Definition D.6 *Let $\widehat{\theta}_m$ be the estimator of $\theta^\star$ in phase m. Formally,*

$$\widehat{\theta}_m \triangleq V_m^{-1}\sum_{a\in\mathrm{Supp}(\pi_m)} a\sum_{t\in\tilde{S}_m(a)}\ell_t.$$

Lemma 5.1 *For every phase m and $a\in\mathcal{A}_m$, $0\leq\langle\widehat{\theta}^+_{m,a}-\widehat{\theta}_m,\,a\rangle\leq 2\epsilon_m$, and $0\leq\langle\widehat{\theta}_m-\widehat{\theta}^-_{m,a},\,a\rangle\leq 2\epsilon_m$, where $\widehat{\theta}_m$, defined in Definition D.6, is the estimator if we had all N_m samples.*

Proof: We short $V\coloneqq V_m$, $\widehat{\theta}\coloneqq\widehat{\theta}_m$. We use M_m here to denote the missing samples from the first N_m. That is because afterwards we play arbitrary arms and the samples is discarded anyway.

For each $b\in\mathrm{Supp}(\pi_m)$,

$$\langle\hat{\theta}^+_a-\hat{\theta},\,a\rangle=\sum_b a^TV^{-1}b\left(\sum_{t\in M_m(b)}Sign\big(a^TV^{-1}b\big)-\ell_t\right).$$

In the same way,

$$\langle\hat{\theta}-\hat{\theta}^-_a,\,a\rangle=\sum_b a^TV^{-1}b\left(\sum_{t\in M_m(b)}\ell_t-Sign\big(a^TV^{-1}b\big)\right).$$

We first note that since $-1\leq\ell_t\leq 1$, both expressions has,

$$\begin{aligned} 0\leq\langle\hat{\theta}^+_a-\hat{\theta},\,a\rangle&\leq 2\sum_b\sigma_m(b)\big|a^TV^{-1}b\big|, \\ 0\leq\langle\hat{\theta}-\hat{\theta}^-_a,\,a\rangle&\leq 2\sum_b\sigma_m(b)\big|a^TV^{-1}b\big|. \end{aligned}$$

To finish the proof of both expressions,

$$\begin{aligned}
\sum_b \sigma_m(b)\left|a^T V^{-1} b\right| &\leq \frac{\epsilon_m}{\sqrt{n}} \sum_b N_m(b)\left|a^T V^{-1} b\right| \\
&= \frac{\epsilon_m}{\sqrt{n}} \sum_b N_m(b)\sqrt{a^T V^{-1} b b^T V^{-1} a} \\
&= \frac{\epsilon_m}{\sqrt{n}} \sum_{t=1}^{N_m} \sqrt{a^T V^{-1} b_t b_t^T V^{-1} a} \\
&\leq \frac{\epsilon_m}{\sqrt{n}} \sqrt{N_m a^T V^{-1} \left(\sum_{t=1}^{N_m} b_t b_t^T\right) V^{-1} a} \\
&= \frac{\epsilon_m}{\sqrt{n}} \sqrt{N_m a^T V^{-1} V V^{-1} a} \\
&= \frac{\epsilon_m}{\sqrt{n}} \sqrt{N_m \left\|a\right\|_{V^{-1}}^2} \\
&\leq \frac{\epsilon_m}{\sqrt{n}} \sqrt{n} \\
&= \epsilon_m,
\end{aligned}$$

where the second inequality is Cauchy-Shwartz, and third inequality is Lemma A.6. ■

Corollary D.7 *Assume $\tilde{\mathcal{E}}$, For each phase m and arm $a \in \mathcal{A}_m$:*

$$\begin{aligned}
-\epsilon_m \leq \langle \widehat{\theta}^+_{m,a} - \theta^\star,\, a\rangle \leq 3\epsilon_m, \\
-\epsilon_m \leq \langle \theta^\star - \widehat{\theta}^-_{m,a},\, a\rangle \leq 3\epsilon_m.
\end{aligned}$$

Proof: Directly from Lemmas A.7 and 5.1. We note that since $\widehat{\theta}_m$ is an estimator constructed from N_m i.i.d samples, Lemma A.7 holds for it under $\tilde{\mathcal{E}}$. ■

We now have the desired property - our estimation is $O(\epsilon_m)$ accurate. We can safely prove the two important Lemmas for phased elimination: (1) the optimal arm isn't eliminated with high probability and (2) an active arm at phase m has a suboptimality of $O(\epsilon_m)$.

Lemma D.8 *Assume $\mathcal{E}$, for every m, $a^* \in \mathcal{A}_m$.*

Proof: We prove this by induction. The base case is proven by the fact that every $\mathcal{A}_1 = \mathcal{A}$. Assume by contradiction that $a^* \in \mathcal{A}_m$ but $a^* \notin \mathcal{A}_{m+1}$, namely it was eliminated in phase m. This means there is an action $b \in \mathcal{A}$ such that:

$$\langle \widehat{\theta}^-_{m,a^*}\,,\, a^*\rangle - \langle \widehat{\theta}^+_{m,b}\,,\, b\rangle > 6\epsilon_m.$$

From Corollary D.7:

$$\begin{aligned}
6\epsilon_m &< \langle \widehat{\theta}^-_{m,a^*}\,,\, a^*\rangle - \langle \widehat{\theta}^+_{m,b}\,,\, b\rangle \\
&= \langle \widehat{\theta}^-_{m,a^*} - \theta^\star,\, a^*\rangle + \langle \theta^\star,\, a^* - b\rangle + \langle \theta^\star - \widehat{\theta}^+_{m,b},\, b\rangle \\
&\leq \langle \theta^\star,\, a^* - b\rangle + 2\epsilon_m \\
\implies 4\epsilon_m &< \langle \theta^\star,\, a^* - b\rangle,
\end{aligned}$$

which contradicts the fact that a^* is the optimal arm. ■

Lemma D.9 *Assume $\mathcal{E}$ and action a such that $a \in \mathcal{A}_m$. Then, $\Delta(a) \leq 24\epsilon_m$*

Proof: From Lemma D.8 also $a^* \in \mathcal{A}_m$. If $a \in \mathcal{A}_m$ it means that,

$$\begin{aligned}
6\epsilon_{m-1} &\geq \langle \widehat{\theta}^-_{m,a}\,,\, a\rangle - \langle \widehat{\theta}^+_{m,a^*}\,,\, a^*\rangle \\
&= \langle \widehat{\theta}^-_{m,a} - \theta^\star,\, a\rangle + \langle \theta^\star,\, a - a^*\rangle + \langle \theta^\star - \widehat{\theta}^+_{m,a^*},\, a^*\rangle \\
&\geq \langle \theta^\star,\, a - a^*\rangle - 6\epsilon_{m-1} \\
\implies \Delta(a) &\leq 12\epsilon_{m-1} \leq 24\epsilon_m,
\end{aligned}$$

the second inequality is from Corollary D.7. ■

Finally, we prove our main result Theorem 5.2 for the loss-dependent delay case.

Theorem 5.2 *Algorithm 6 guarantees that*

$$\mathcal{R}_T = O\Big(\sqrt{nT\log(KT)} + \log(T)^2\log\log(n)\,n^{3/2} + \sqrt{n}\log(T)\max_a \mathbb{E}[d(a)]\Big)\,.$$

Proof: Assume $\tilde{\mathcal{E}}$ (Definition A.4) and $\mathcal{E}'$.

Notice that a phase with $T_m = T$ will be the last. Thus:

$$T \geq T_m \geq N_m \geq \frac{16n\log(KT)}{\epsilon_m^2}$$
$$\implies m < \frac{1}{2}\log\left(\frac{T}{16n\log(KT)}\right),$$

and denote the last phase as m_{end}.

From Markov inequality,

$$\mathcal{Q}\left(1 - \frac{\epsilon_m}{3\sqrt{n}}; a\right) \leq \frac{3\sqrt{n}\mathbb{E}[d]}{\epsilon_m}.$$

Thus, $\mathcal{E}'$ becomes,

$$T_m \leq N_m + \max_{a\in\mathcal{A}_m}\frac{3\sqrt{n}\mathbb{E}[d(a)]}{\epsilon_m}.$$

From Lemma D.9,

$$\begin{aligned}
\mathcal{R}_T &\leq \sum_{m=1}^{m_{end}} 24T_m\epsilon_m \\
&\leq \sum_{m=1}^{m_{end}} 24\epsilon_m\left(N_m + \max_{a\in\mathcal{A}_m}\frac{3\sqrt{n}\mathbb{E}[d(a)]}{\epsilon_m}\right) \\
&\leq \sum_{m=1}^{m_{end}} 24\epsilon_m\left(\underbrace{\frac{48\log(T)\log\log(n)n^{3/2} + \max_{a\in\mathcal{A}_m}3\sqrt{n}\mathbb{E}[d(a)]}{\epsilon_m}}_{(i)} + \underbrace{\frac{16n\log(KT)}{\epsilon_m^2}}_{(ii)}\right).
\end{aligned}$$

We bound each term separately. To bound (i), direct calculation shows that

$$\begin{aligned}
(i) &= O\left(\sum_{m=1}^{m_{end}}\epsilon_m\frac{\log(T)\log\log(n)n^{3/2} + \max_{a\in\mathcal{A}_m}\sqrt{n}\mathbb{E}[d(a)]}{\epsilon_m}\right) \\
&\leq O\left(\log(T)^2\log\log(n)n^{3/2} + \max_{a\in\mathcal{A}_m}\log(T)\sqrt{n}\mathbb{E}[d(a)]\right),
\end{aligned}$$

where we use the fact that $m_{end} \leq \log(T)$.

To bound (ii),

$$(ii) = \sum_{m=1}^{m_{end}} O\big(n\log(KT)2^{-m}\big) \leq O\left(n\log(KT)\sqrt{\frac{T}{n\log(KT)}}\right) = O\Big(\sqrt{nT\log(KT)}\Big).$$

Alternatively, we can write,

$$\begin{aligned}
\mathcal{R}_T &\leq \sum_{m=1}^{m_{end}} 24 T_m \epsilon_m \\
&\leq \sum_{m=1}^{m_{end}} 24\epsilon_m \left(N_m + \max_{a \in \mathcal{A}_m} \mathcal{Q}\left(1 - \frac{\epsilon_m}{3\sqrt{n}}\right); a \right) \\
&\leq \sum_{m=1}^{m_{end}} 24\epsilon_m \left(\underbrace{\frac{48 \log(T) \log\log(n) n^{3/2}}{\epsilon_m}}_{(i)} + \underbrace{\frac{16 n \log(KT)}{\epsilon_m^2}}_{(ii)} + \underbrace{\max_{a \in \mathcal{A}_m} \mathcal{Q}\left(1 - \frac{\epsilon_m}{3\sqrt{n}}; a\right)}_{(iii)} \right).
\end{aligned}$$

The bound of (i) and (ii) is similar to above. To bound (iii), from Lemma D.9 and the monotonicity of the CDF for every m and $a \in \mathcal{A}_m$, $a \neq a^*$

$$\mathcal{Q}\left(1 - \frac{\epsilon_m}{3\sqrt{n}}; a\right) \leq \mathcal{Q}\left(1 - \frac{\Delta(a)}{72\sqrt{n}}; a\right).$$

If $|\mathcal{A}_m| = 1$, there is no regret from this phase from Lemma D.8. If $|\mathcal{A}_m| \geq 2$, it means that $\epsilon_m \geq 16\Delta_{\min}$, thus,

$$\mathcal{Q}\left(1 - \frac{\epsilon_m}{3\sqrt{n}}; a^*\right) \leq \mathcal{Q}\left(1 - \frac{\Delta_{\min}}{72\sqrt{n}}; a^*\right),$$

which means:

$$\max_{a \in \mathcal{A}_m} \mathcal{Q}\left(1 - \frac{\epsilon_m}{3\sqrt{n}}; a^*\right) \leq \max_{a \neq a^*} \mathcal{Q}\left(1 - \frac{\Delta(a)}{72\sqrt{n}}; a\right) + \mathcal{Q}\left(1 - \frac{\Delta_{\min}}{72\sqrt{n}}; a^*\right).$$

We now bound (iii) as follows

$$\begin{aligned}
(iii) &= \sum_{m=1}^{m_{end}} O\left(\epsilon_m \max_{a \in \mathcal{A}_m} \mathcal{Q}\left(1 - \frac{\epsilon_m}{3\sqrt{n}}\right); a \right) \\
&\leq \sum_{m=1}^{m_{end}} O\left(2^{-m} \max_{a \neq a^*} \mathcal{Q}\left(1 - \frac{\Delta(a)}{72\sqrt{n}}; a\right) + \mathcal{Q}\left(1 - \frac{\Delta_{\min}}{72\sqrt{n}}; a^*\right) \right) \\
&\leq \sum_{m=1}^{\infty} O\left(2^{-m} \max_{a \neq a^*} \mathcal{Q}\left(1 - \frac{\Delta(a)}{72\sqrt{n}}; a\right) + \mathcal{Q}\left(1 - \frac{\Delta_{\min}}{72\sqrt{n}}; a^*\right) \right) \\
&= O\left(\max_{a \neq a^*} \mathcal{Q}\left(1 - \frac{\Delta(a)}{72\sqrt{n}}; a\right) + \mathcal{Q}\left(1 - \frac{\Delta_{\min}}{72\sqrt{n}}; a^*\right) \right),
\end{aligned}$$

which finishes the proof.

■

D.2 Omitted Details in Section 5.2

In this section, we provide the omitted proof for the $\tilde{\Omega}(\sqrt{n}d)$ lower bound in the general loss dependent delay case.

Theorem 5.3 *For any $\bar{d}, n \geq 32 \log(\bar{d}n), T \geq \frac{\sqrt{n}\bar{d}}{\sqrt{\log(\bar{d}n)}}$ there exists a distribution over loss-dependent delayed linear bandit instances with expected delay for all arms bounded by $\bar{d}$ such that any algorithm will suffer a regret of $\Omega(\frac{\sqrt{n}\bar{d}}{\sqrt{\log(\bar{d}n)}})$ with a constant probability over the distribution and the randomness of the algorithm.*

Proof: Denote $q = \sqrt{\frac{8\log(\bar{d}n)}{n}}$. Since $n \geq 32\log(\bar{d}n)$, $q \in (0, \frac{1}{2}]$.

We use the action set $\mathcal{A}$ in Lemma G.3 of size $K = \lceil 2\bar{d}/q \rceil$. Notice that:

$$\log(K) \leq \log\left(2\bar{d}\sqrt{\frac{n}{8\log(\bar{d}n)}} + 1\right) \leq \log(\bar{d}n).$$

The parameter $\theta^\star$ is a drawn uniformly from $\mathcal{A}$ and multiplied by -1. Thus, we have in the i^*th instance:

$$\langle a_{i^*}, \theta^\star \rangle = -1,$$
$$|\langle a_i, \theta^\star \rangle| \leq \sqrt{\frac{8\log(K)}{n}} \leq q \qquad \forall i \neq i^*.$$

Denote $\mu_a \coloneqq \langle a_i, \theta^\star \rangle$.

The delay as loss distribution of a_{i^*} is deterministic $\ell_t = -1$, $d_t = 0$.

The delay and loss distribution for any $a \neq a_{i^*}$ is:

$$\Pr[\ell_t = 1, d_t = 0] = \frac{1-q}{2},$$
$$\Pr[\ell_t = -1, d_t = 0] = \frac{1-q}{2},$$
$$\Pr[\ell_t = 1, d_t = \bar{d}/q] = \frac{q+\mu_a}{2},$$
$$\Pr[\ell_t = -1, d_t = \bar{d}/q] = \frac{q-\mu_a}{2}.$$

First notice that since $|\mu_a| \leq q < 1$, all probabilities are legal.

Second, we make sure the expectation is correct. Indeed, for loss expectation we have:

$$1\left(\frac{1-q}{2} + \frac{q+\mu_a}{2}\right) - 1\left(\frac{1-q}{2} + \frac{q-\mu_a}{2}\right) = \frac{1+\mu_a}{2} - \frac{1-\mu_a}{2} = \mu_a,$$

and for delay expectation we have:

$$\frac{\bar{d}}{q}\left(\frac{q+\mu_a}{2} + \frac{q-\mu_a}{2}\right) + 0\left(\frac{1-q}{2} + \frac{1-q}{2}\right) = \bar{d}.$$

Crucially, conditioned on $d_t = 0$, for every $i \neq i^*$ the loss is 1 w.p $1/2$ and -1 w.p $1/2$. Additionally, the delay has probability exactly q. Thus, no matter which arm was played as long as it is not i^*, the results is the same until $\bar{d}/q$. Therefore, until $t = \bar{d}/q$, we can think that the only received feedback is that $a_t \neq a_{i^*}$.

Fix an algorithm and its randomness, and denote $\mathcal{A}'$ to be the first $\bar{d}/q$ actions that are played if there is no feedback. If i^* isn't there, there regret is at least $\Omega(\bar{d}/q)$. Since there are more than $2\bar{d}/q$ instances this is true w.p $\geq 1/2$. Since this is true for every randomness of the algorithm, it is true unconditionally. ■

E MAB with Loss Dependent Delay

In this appendix we create an algorithm, with the same structure as Algorithm 6, but specifically for multi armed bandit. The idea is that we use the geometry of the problem so that it suffices to have $1 - \epsilon_m$ quantile of the information.

The regret bound of Lancewicki et al. [18] depends on the $(1-\Delta)$ qunatile of the delay. Since this quantile isn't multiplied by Δ, we can't use Markov inequaility to bound it with $\mathbb{E}[d]$. However, we show here that if you (1) use Phased Elimination instead of Successive Elimination and (2) do

the Markov inequality in an earlier stage (not after having the final bound), we can indeed have an additive term that depends only on the expected delay.

We denote the mean loss of each arm $\mu(a)$, $\Delta(a) = \mu(a) - \mu(a^*)$.

Algorithm 7 Phased Elimination for Dependent Delayed MAB

Require: Action set $\mathcal{A}$

Initialize $\mathcal{A}_1 \leftarrow \mathcal{A}$, $\epsilon_1 \leftarrow \frac{1}{2}$
for $m = 1, 2, \ldots$ **do**
  $N_m = \frac{24 \log(KT)}{\epsilon_m^2}$
  $S_m(a) \leftarrow \emptyset$ for all $a \in \mathcal{A}_m$ ▷ *observations collected in phase m for action a*
  Play N_m times each $a \in \mathcal{A}$
  while there exists $a \in \mathcal{A}_m$ such that $|S_m(a)| < (1 - \epsilon_m) N_m$ **do**
    Play arbitrary $a \in \mathcal{A}_m$
    Receive all losses that arrive at round t from the first N_m steps
    for each newly arrived loss generated in phase m from action $a \in \mathcal{A}_m$ **do**
      Add this loss to $S_m(a)$
  for $a \in \mathcal{A}_m$ **do**
    $\widehat{\mu}^+(a)_m = \frac{1}{N_m}\left(\sigma_m(a) + \sum_{t \in S_m} \ell_t\right)$
    $\widehat{\mu}^-(a)_m = \frac{1}{N_m(a)}\left(-\sigma_m(a) + \sum_{t \in S_m} \ell_t\right)$
  $\mathcal{A}_{m+1} \leftarrow \left\{a \in \mathcal{A}_m : \widehat{\mu}^-(a) - \widehat{\mu}^+(b) \le 4\epsilon_m \quad \forall b \in \mathcal{A}_m\right\}$
  $\epsilon_{m+1} \leftarrow \epsilon_m / 2$

We start by defining two good events - the first controls the length of the phase, namely how much time we should wait for that $1 - \epsilon_m$ quantile of information, and the second controls the estimation error of each phase m.

Definition E.1 *Let $\mathcal{E}'_{MAB}$ be the event that for every m:*

$$T_m \le nN_m + \max_{a \in \mathcal{A}_m} \mathcal{Q}\Big(1 - \frac{\epsilon_m}{3}; a\Big).$$

Lemma E.2 *$\mathcal{E}'_{MAB}$ is true with probability at least $1 - \frac{1}{T}$.*

Proof: Fix some $a \in \mathcal{A}_m$. Since all N_m pulls of it were at least $\max_{a' \in \mathcal{A}_m} \mathcal{Q}\big(1 - \frac{\epsilon_m}{3}; a'\big) \ge \mathcal{Q}\big(1 - \frac{\epsilon_m}{3}; a\big)$ step ago, we have that each of those pulls has returned w.p at least $\big(1 - \frac{\epsilon_m}{3}\big)$. Thus, the number of missing pulls is a sum of Bernoulli random variables, each w.p at least $\frac{\epsilon_m}{3}$.

From Lemma G.2, w.p $1 - \frac{1}{T^2}$, the number of missing is bounded by $\frac{2\epsilon_m}{3} N_m + 8 \log(T)$. To show the number of missing is bounded by $\epsilon_m N_m$ we need to show that $8 \log(T) \le \frac{\epsilon_m}{3} N_m$. Indeed:

$$\frac{\epsilon_m}{3} N_m = \frac{\epsilon_m}{3} \frac{24 \log(KT)}{\epsilon_m^2} \ge 8 \log(T).$$

Union bound for all m concludes the proof with probability at least $1 - \frac{1}{T}$. ■

Definition E.3 *Let $\mathcal{E}_{MAB}$ be the event that, for every phase m and $a \in \mathcal{A}_m$,*

$$-\epsilon_m \le \widehat{\mu}_m^+(a) - \mu(a) \le 3\epsilon_m,$$
$$-\epsilon_m \le \mu(a) - \widehat{\mu}_m^-(a) \le 3\epsilon_m.$$

Lemma E.4 *$\mathcal{E}_{MAB}$ is true with probability at least $1 - \frac{1}{T}$.*

Proof: Fix $a \in \mathcal{A}_m$. Denote $\widehat{\mu}_m(a)$ be the average loss of each arm in the first N_m pulls of arm a.

Since $-1 \le \ell_t \le 1$, both expressions above are bounded from below by 0. For the other side:

$$\widehat{\mu}_m^+(a) - \widehat{\mu}_m(a) = \frac{1}{N_m} \sum_{t \in M_m(a)} 1 - \ell_t \le \frac{2}{N_m} \sigma_m(a)$$

$$\widehat{\mu}_m(a) - \widehat{\mu}_m^-(a) = \frac{1}{N_m} \sum_{t \in M_m(a)} \ell_t + 1 \leq \frac{2}{N_m} \sigma_m(a).$$

By the definition of the phase $\frac{\sigma_m(a)}{N_m} \leq \epsilon_m$.

Thus, we only need to bound $|\widehat{\mu}_m - \mu|$ at the end of phase m. Using Hoeffding, w.p $1 - \frac{1}{T^2 K}$,

$$|\widehat{\mu}_m - \mu| \leq \epsilon_m.$$

Union bound on all arms and phases concludes the proof. ■

We now do that two classical Lemmas of Successive Elimination - optimal arm isn't eliminated with high probability and non-eliminated suboptimal arms have a bounded suboptimality.

Lemma E.5 *Assume $\mathcal{E}_{\texttt{MAB}}$, for every m, $a^* \in \mathcal{A}_m$.*

Proof: We prove by induction on the phases. The base is just the fact that $a^* \in \mathcal{A}$. Assume $a^* \in \mathcal{A}_m$ and assume by contradiction that $a^* \notin \mathcal{A}_{m+1}$. Thus, there is an action $a \in \mathcal{A}_m$ such that:

$$\widehat{\mu}_m^-(a^*) - \widehat{\mu}_m^+(a) > 4\epsilon_m.$$

From $\mathcal{E}_{\texttt{MAB}}$:

$$\begin{aligned} 4\epsilon_m &< \widehat{\mu}_m^-(a^*) - \widehat{\mu}_m^+(a) \\ &= (\widehat{\mu}_m^-(a^*) - \mu(a^*)) + (\mu(a^*) - \mu(a)) + (\mu(a) - \widehat{\mu}_m^+(a)) \\ &\leq \mu(a^*) - \mu(a) + 4\epsilon_m \\ \implies 0 &< \mu(a^*) - \mu(a), \end{aligned}$$

which contradicts the fact that a^* is optimal. ■

Lemma E.6 *Assume $\mathcal{E}_{\texttt{MAB}}$ and arm a such that $a \in \mathcal{A}_m$. Then, $\Delta(a) \leq 16\epsilon_m$.*

Proof: From Lemma E.5, also $a^* \in \mathcal{A}_m$. From the elimination rule:

$$\begin{aligned} 4\epsilon_{m-1} &\geq \widehat{\mu}_m^-(a) - \widehat{\mu}_m^+(a^*) \\ &= (\widehat{\mu}_m^-(a) - \mu(a)) + (\mu(a) - \mu(a^*)) + (\mu(a^*) - \widehat{\mu}_m^+(a^*)) \\ &\geq \mu(a) - \mu(a^*) - 4\epsilon_{m-1} \\ \implies \Delta(a) &\leq 8\epsilon_{m-1} \leq 16\epsilon_m. \end{aligned}$$

The second inequality is from the definition of $\mathcal{E}_{\texttt{MAB}}$. ■

Finally, we show that with the above Lemmas we can bound the regret.

Theorem E.7 *The regret of Algorithm 7 is bounded by.*

$$\mathcal{R}_T = O\Big(\sqrt{nT\log(KT)} + \log(T) \max_a \mathbb{E}[d(a)]\Big).$$

Proof: Assume $\mathcal{E}_{\texttt{MAB}}$ and $\mathcal{E}'_{\texttt{MAB}}$.

Notice that a phase with $T_m = T$ will be the last. Thus:

$$\begin{aligned} T \geq T_m \geq N_m &\geq \frac{16n\log(KT)}{\epsilon_m^2} \\ \implies m &< \frac{1}{2}\log\left(\frac{T}{16n\log(KT)}\right) \end{aligned}$$

Denote the last phase as m_{end}.

From Markov inequality:

$$\mathcal{Q}\Big(1 - \frac{\epsilon_m}{3}; a\Big) \leq \frac{\mathbb{E}[d]}{\epsilon_m}$$

Thus, $\mathcal{E}'_{\texttt{MAB}}$ becomes:

$$T_m \leq N_m + \max_{a \in \mathcal{A}_m} \frac{\mathbb{E}[d(a)]}{\epsilon_m}$$

From Lemma E.6:

$$\begin{aligned}
\mathcal{R}_T &\leq \sum_{m=1}^{m_{end}} 24 T_m \epsilon_m \\
&\leq \sum_{m=1}^{m_{end}} 24 \epsilon_m \left(N_m + \max_{a \in \mathcal{A}_m} \frac{\mathbb{E}[d(a)]}{\epsilon_m} \right) \\
&\leq \sum_{m=1}^{m_{end}} 24 \epsilon_m \left(\underbrace{\frac{\max_{a \in \mathcal{A}_m} \mathbb{E}[d(a)]}{\epsilon_m}}_{(i)} + \underbrace{\frac{16 n \log(KT)}{\epsilon_m^2}}_{(ii)} \right)
\end{aligned}$$

We finish the proof by bounding (i) and (ii) respectively:

$$(i) = O\left(\sum_{m=1}^{m_{end}} \epsilon_m \frac{\max_{a \in \mathcal{A}_m} \mathbb{E}[d(a)]}{\epsilon_m} \right) \leq O\left(\max_{a \in \mathcal{A}_m} \log(T) \mathbb{E}[d(a)] \right),$$

$$(ii) = \sum_{m=1}^{m_{end}} O\big(n \log(KT) 2^{-m}\big) \leq O\left(n \log(KT) \sqrt{\frac{T}{n \log(KT)}} \right) = O\Big(\sqrt{nT \log(KT)} \Big).$$

The fact that $\mathcal{E}_{\texttt{MAB}}$ and $\mathcal{E}'_{\texttt{MAB}}$ are true w.p $1 - \frac{2}{T}$ concludes the proof. ■

F Omitted Details in Section 6

In this section, we provide the omitted proof for the $\Omega(D)$ lower bound in delay-as-payoff.

Theorem 6.1 *For any $n \geq 24, K \leq e^{n/100}, D \leq 10K, T \geq D$, there exists a distribution over delay-as-payoff linear bandit instances with no noise and $d^* = 0$ for which any algorithm suffers regret $\Omega(D)$ with constant probability.*

Proof: For any subset $S \subset [n]$ with $|S| = n/2$, define the action

$$a(S) \;:=\; \frac{1}{\sqrt{n}} \mathbf{1}_S \in \mathbb{R}^n.$$

We first show that there exists a deterministic collection of K sets $S_1, \ldots, S_K \subset [n]$ with $|S_i| = n/2$ such that for all ordered pairs $i \neq j$,

$$|S_i \setminus S_j| \;\geq\; \frac{n}{20}. \tag{6}$$

To see this, sample $S_1, \ldots, S_K$ i.i.d. uniformly from $\{S \subset [n] : |S| = n/2\}$. Fix any ordered pair (i, j) with $i \neq j$ and define the random variable $X_{ij} := |S_i \setminus S_j|$. Conditioned on S_j, X_{ij} is hypergeometric with mean $\mu = \mathbb{E}[X_{ij} \mid S_j] = n/4$. A standard multiplicative Chernoff bound for hypergeometric random variables gives that for any $\delta \in (0, 1)$,

$$\Pr[X_{ij} \leq (1 - \delta)\mu \mid S_j] \;\leq\; \exp\left(-\frac{\delta^2 \mu}{2} \right).$$

Taking $\delta = 4/5$ so that $(1 - \delta)\mu = (1/5) \cdot (n/4) = n/20$, we obtain

$$\Pr\left[X_{ij} \leq \frac{n}{20} \right] \;\leq\; \exp\left(-\frac{(4/5)^2 \cdot (n/4)}{2} \right) \;=\; \exp\left(-\frac{2n}{25} \right).$$

Union bounding over all $K(K-1) \le K^2$ ordered pairs,

$$\Pr\Big[\exists\, i \neq j :\ |S_i \setminus S_j| \le \frac{n}{20}\Big] \ \le\ K^2 \exp\!\left(-\frac{2n}{25}\right).$$

Using $K \le \exp(n/100)$, we have $K^2 \le \exp(n/50)$, hence

$$K^2 \exp\!\left(-\frac{2n}{25}\right) \ \le\ \exp\!\left(\frac{n}{50} - \frac{2n}{25}\right) \ =\ \exp\!\left(-\frac{3n}{50}\right) \ \le\ \frac{1}{4},$$

where the last inequality uses $n \ge 24$. Therefore, with probability at least $3/4$, the sampled collection satisfies (6). By the probabilistic method, there exists a deterministic choice of $S_1, \ldots, S_K$ that satisfies (6). Fix such a choice and define $\mathcal{A} = \{a(S_1), \ldots, a(S_K)\}$.

The parameter $\theta^\star$ is chosen as $a\big(\overline{S_{i^*}}\big)$, where i^* is drawn uniformly from $[K]$ and $\overline{S_{i^*}}$ is the complementary of S_{i^*}. One can see that $\langle a_{i^*},\, \theta^\star\rangle = 0$ which means that indeed d^*. For any other $i \neq i^*$, we have from (6) that $\langle a_i,\, \theta^\star\rangle \ge 1/20$ which means that its delay is at least $D/20$. Thus, unless a_{i^*} is played in the first $D/20$ actions of the algorithm, the algorithm gets no feedback.

Fix an algorithm and all its randomness, and denote $\mathcal{A}'$ to be the first $D/20$ actions that are played if there is no feedback. If i^* does not belong to $\mathcal{A}'$, then the regret is at least $\Omega(D)$ since the optimal action is not chosen for $\Omega(D)$ rounds and the suboptimality gap is $\Omega(1)$. Since, there are $K \ge D/10$ actions, the above event holds with probability at least $1/2$. Since this is true for every randomness of the algorithm, it is true unconditionally. ■

G Auxiliary Lemmas

Lemma G.1 (Lemma F.4 in [8]) *Let $\{X_t\}_{t=1}^T$ be a sequence of Bernoulli random and a filtration $\mathcal{F}_1 \subseteq \mathcal{F}_2 \subseteq ...\mathcal{F}_T$ with $\mathbb{P}(X_t = 1 \mid \mathcal{F}_t) = P_t$, P_t is $\mathcal{F}_t$-measurable and X_t is $\mathcal{F}_{t+1}$-measurable. Then, for all $t \in [T]$ simultaneously, with probability $1-\delta$,*

$$\sum_{k=1}^{t} X_k \ge \frac{1}{2}\sum_{k=1}^{t} P_k - \log\frac{1}{\delta}.$$

Lemma G.2 (Consequence of Freedman's Inequality, e.g., Lemma 27 in Efroni et al. [10]) *Let $\{X_t\}_{t\ge 1}$ be a sequence of random variables, supported in $[-R, R]$, and adapted to a filtration $\mathcal{F}_1 \subseteq \mathcal{F}_2 \subseteq ...\mathcal{F}_T$. For any T, with probability $1-\delta$,*

$$\sum_{t=1}^{T} X_t \le 2\sum_{t=1}^{T} \mathbb{E}[X_t \mid \mathcal{F}_t] + 4R\log\frac{1}{\delta}.$$

Lemma G.3 (Lemma 3.1 in [20]) *There exists a set of actions $\{a_1, a_2, \ldots, a_K\} \subset \mathbb{R}^n$ such that:*

$$\|a_i\|_2 = 1, \quad \forall i \in [K],$$
$$|\langle a_i, a_j\rangle| \le \sqrt{\frac{8\log(K)}{n}}, \quad \forall j \neq i.$$

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The main claims made in the abstract and introduction accurately reflect the paper's contributions and scope.

   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: The paper discuss the limitations.

   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: The paper provides the full set of assumptions and complete (and correct) proofs.

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: [N/A]

   Justification: The paper does not include experiments.

   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: [N/A]

Justification: The paper does not include experiments.

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: [N/A]

Justification: The paper does not include experiments.

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: [N/A]

Justification: The paper does not include experiments.

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: [N/A]

   Justification: The paper does not include experiments.

   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: The research conducted in the paper conforms, in every respect, with the NeurIPS Code of Ethics.

   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: This paper is theoretical and does not have societal impacts beyond those of general ML/AI research.

    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: The paper poses no such risks.

    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: The paper does not use existing assets.

    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: the paper does not release new assets.

    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: The paper does not involve crowdsourcing nor research with human subjects.

    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: The paper does not involve crowdsourcing nor research with human subjects.

    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: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

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.