Prudent-Banker: No Extra Fees for Baseline Safety in Adversarial Bandits With and Without Delays

Ting Hu[*]
Department of Economics[†]
ting.hu@wisc.edu

Luanda Cai[*]
Department of Finance[†]
luanda.cai@wisc.edu

Emmanouil V. Vlatakis-Gkaragkounis
Department of Computer Sciences[†]
vlatakis@wisc.edu

Abstract

We study adversarial multi-armed bandits with and without delayed feedback under a *safety-aware* goal: achieving minimax-optimal worst-case regret while keeping nearly *constant* regret relative to a designated "safe" baseline policy. Existing approaches can balance this trade-off with immediate feedback for smooth comparators, but arbitrary delays can mistime transitions between *conservatism* and *exploration*, endangering the safety guarantee. To bridge this gap, we propose PRUDENT-BANKER, a novel algorithm that combines a delay-adapted variant of Online Mirror Descent with a modified phased-aggression mechanism. Its key technical contribution is a delay-calibrated restart threshold that rigorously accounts for the worst-case distortion induced by unobserved feedback and reliably detects comparator suboptimality. We also establish new lower bounds for safety-constrained adversarial delayed bandits, showing that the regret guarantees of PRUDENT-BANKER are unimprovable, up to logarithmic factors, under the baseline-safety requirement. To the best of our knowledge, PRUDENT-BANKER is the first algorithm to achieve the optimal safety–robustness trade-off: pseudo-regret $\widetilde{O}(\sqrt{T} + \sqrt{D})$ together with $\widetilde{O}(1)$ regret against the safe comparator, both with and without delays. Experiments across diverse delay distributions show that, unlike standard delay-robust baselines, PRUDENT-BANKER effectively balances safety and learning.

1 Introduction

Sequential decision-making is shaped by a recurring dilemma: *explore* to acquire new information, or *exploit* what is already known to perform well. The multi-armed bandit (MAB) framework, introduced by Robbins [53], has been the canonical mathematical model for this trade-off and has become a foundational tool across Operations Research, Economics, and Computer Science.

In this setting, a learner repeatedly chooses one action, an "arm", from a fixed set. After choosing its arm, the learner incurs a loss (or receives a reward) that depends on the environment; The goal is to learn from this feedback over time so as to minimize *regret*, i.e., the gap between the learner's cumulative loss and the cumulative loss of the best single arm in hindsight.

However, classical MAB theory often assumes immediate feedback, but modern systems routinely violate this premise. In web applications, latency and user inactivity delay feedback loops [52]; in video streaming, QoE signals (e.g., buffering and engagement) are observed only after delivery [12]; in robotics, actuation and sensing lag behind commands [44]; and in multi-agent settings, communication is inherently asynchronous [13]. In high-stakes domains such as online advertising [41] and clinical decision-making [60], delays are intrinsic: transactions or patient outcomes may appear long after an action is taken. These realities motivate bandit algorithms whose performance guarantees remain meaningful even when feedback is substantially delayed.

[*]Equal contribution. [†]University of Wisconsin–Madison.

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

While a growing body of work addresses adversarial bandits with delayed feedback [15, 27, 28, 30, 33, 64], the majority of existing guarantees focus on a single metric: *minimax regret minimization*. While this worst-case lens is theoretically robust, it is insufficient for high-stakes deployments. In practice, a learner rarely enters an environment *tabula rasa*; they are often equipped with a vetted baseline policy—such as a standard medical treatment, a compliant control rule, or a conservative investment strategy—that is trusted to avoid catastrophic outcomes. In light of this discussion, we arrive at the main question driving this work:

Can we be worst-case robust while paying essentially no extra cost over a trusted baseline, even with arbitrary feedback delays? $(\star)$

In such settings, delayed feedback significantly amplifies the risk of experimentation. If an exploratory action is harmful, the evidence of failure may only surface after a long lag. During this blind spot, the learner incurs a "hidden debt", repeatedly committing to a suboptimal strategy before the feedback arrives to trigger a correction. This danger motivates exactly the following dual mandate:

Safe Online Learning in Delays (SOLID)

Goal #1: Global Robustness (Minimax Optimality).
The algorithm must ensure regret $\tilde{\mathcal{O}}(\sqrt{T} + \sqrt{D})$ against the optimal arm in hindsight, even if the environment is fully adversarial and delays are arbitrary.

Goal #2: Risk-Averse Behavior at No Extra Cost.
The algorithm must incur nearly constant regret $\tilde{\mathcal{O}}(1)$ relative to the designated safe policy. This ensures that the "price of learning" is bounded, and the method effectively reverts to the baseline when experimentation is risky.

1.1 How Delay Breaks Safety: Three Roadblocks

Achieving the above goals have been already subtle in bandits. We next highlight three obstacles that make this twofold objective fundamentally harder under delays, and guide our algorithmic solution:

A) The Price of Low Comparator Regret in Bandits. Achieving this "best-of-both-worlds"[2] trade-off is non-trivial even under immediate bandit feedback. Specifically, comparator-adaptive learning with bandit feedback is subject to a sharp impossibility result: Lattimore [39] proved that any algorithm guaranteeing $O(1)$ regret relative to a specific deterministic arm must necessarily suffer linear worst-case regret, $\Omega(T)$, against other arms.

This lower bound stems from an inherent *exploration–safety conflict*: to maintain constant regret against a deterministic strategy, i.e., a specific arm, an algorithm is forced to select that action almost exclusively, thereby starving itself of the information required to detect if another arm has become superior. Consequently, strong safety relative to a deterministic comparator is incompatible with sublinear adversarial regret. To circumvent this barrier in a minimal way, Müller et al. [48] relax the requirement on the *safe comparator*: we assume the safe baseline is an *exploratory policy* that assigns strictly positive probability to every action.

B) Why Banker, and Why Safety Is Harder. Lags in the environment impose additional theoretical burdens. Standard minimax-optimal algorithms for online learning and bandits—including *Follow-the-Regularized-Leader* (FTRL) and *Online Mirror Descent* (OMD) [6, 55, 59] and their bandit instantiations such as EXP3 [4] and ZO [1, 22]—rely on potential-based analyses whose core mechanism is a telescoping cancellation across consecutive rounds. Delays disrupt this real-time structure: the learner updates using stale information, while the losses needed for the next cancellation step may still be hidden. This mismatch has historically led to delay-specific techniques and ad-hoc corrections [21, 27, 34, 56, 64].

To address the safety objective on top of delays, a natural first attempt would be to combine such a delay-adapted method with an existing mechanism for baseline safety. This integration is non-trivial. As discussed in Müller et al. [48, Appendix A], standard safety approaches—including shifted variants [54] of PROD-MWU [10, 23], originating from the seminal work of Even-Dar et al. [20], as well as parameter-free adaptations [50, 57]—do not directly yield strong baseline-safety guarantees in the bandit setting. The remaining viable strategy in bandit feedback is *phased aggression* recently explored by [48] in the non-delay safe online bandit learning.

[2] In the vast bandit literature, "*best-of-both-worlds* (BOBW)" most commonly denotes algorithms that attain instance-optimal $\tilde{O}(\log T)$ regret in stochastic environments while maintaining $\tilde{O}(\sqrt{T})$ regret in adversarial ones. Here, we use *BOBW* in a different sense: simultaneous guarantees (i) against a designated comparator and (ii) in the worst case. To our knowledge, this regret-based interpretation was first explicitly termed as "best-of-both-worlds" by recent work of Müller et al. [48].

This paradigm maintains a controlled mixture between the safe baseline and an exploratory no-regret learner, becoming more aggressive only after the data explicitly certifies that the baseline is suboptimal. Yet such technique does not align naturally with generic FTRL frameworks for delayed bandits, whose updates are driven by implicit regularization rather than explicit exploration. The construction of Müller et al. [48] suggests the right path forward—an OMD-based approach, in which explicit exploration can be grafted onto the learning dynamics in a principled way. Thus, to make phased aggression compatible with delays, we need a delay-robust OMD backbone whose modularity exposes and controls exploration. `Banker-OMD` [30] of Huang et. al is precisely such a substrate.

By restoring potential-based reasoning under arbitrary and unknown delays, it cleanly separates no-regret dynamics, bandit feedback, and delay handling, directly addressing our first requirement of near-minimax adversarial regret under delays. What it does *not* address is our second objective—near-zero regret against a safe baseline. The missing ingredient is not delay robustness alone, but delay-robust *safe switching*: under delayed feedback, the hypothesis tests driving phased aggression rely on stale observations, and unseen losses create a hidden debt that can make exploration appear safer, or the baseline less suboptimal, than it truly is. This leads to the central technical question:

How can we safely switch from conservatism to exploration when delays create a hidden debt that can mask the safe policy's suboptimality?

C) Late Switches Between Conservatism and Aggression. The core mechanism of phased aggression relies on a restart condition that detects when the safe comparator x^c performs poorly. In the no-delay setting, Even-Dar et al. [20], Müller et al. [48], achieved this by comparing a *phase-gap statistic* to a threshold of $\mathfrak{R}$: exceeding this value certifies strictly negative regret against x^c within the phase, enabling geometric cancellation and fixed comparator regret. With delayed feedback, this certification becomes treacherous. At time T, the algorithm can only compute the phase gap from feedback that has already arrived:

$$G_T = \max_{x \in \text{Strategies}} \left\langle \sum_{t \in \text{observed by } T} \text{LossEstimation}[t], x^c - x \right\rangle .$$

Crucially, in the phase that triggers a restart, the losses whose feedback has not yet arrived contribute an unknown correction term. This creates a "hidden debt": the baseline may appear less suboptimal than it truly is, or past exploration may appear cheaper than it was. In our analysis, this unobserved distortion can be as large as $O(\sqrt{D}/\delta)$ (δ lower-bounds baseline arm probabilities), potentially offsetting the evidence certified by the standard threshold $\mathfrak{R}$. Thus, relying on the vanilla restart test risks mistimed transitions between conservatism and aggression, and can break the baseline-safety guarantee.

1.2 Our Contributions

The aforementioned discussion suggests that a simplistic "plug-and-play" approach—simply attaching a safety wrapper to a delay-tolerant bandit algorithm—is insufficient. The interaction between delayed feedback and safety test creates an *information deficit* that necessitates a new synergistic design.

To address this, one might first ask: *How can we adapt the restart threshold $\mathfrak{R}$ to account for missing information?* A key theoretical observation driving our work is that augmenting the standard threshold with a delay-dependent slack $\xi \approx O(\sqrt{D})$ effectively neutralizes the risk of "ghost losses" from unobserved feedback. However, a critical obstacle arises: the total delay D is unknown in advance. Relying on a worst-case upper bound (where feedback might arrive only at the very end, implying $D \approx T^2$) would necessitate a prohibitive slack of $\xi \approx O(T)$. While safe, this pessimistic parameterization would yield a vacuous linear regret bound $O(T)$, forfeiting the benefits of learning in any realistic environment where delays are moderate (i.e., $D \ll T^2$).

To resolve this dilemma, we introduce PRUDENT-BANKER, an algorithm that adapts to the unknown delay landscape without sacrificing safety or performance. We prove matching upper and lower bounds for the safety-constrained adversarial bandit problem, showing that the resulting trade-off is optimal up to logarithmic factors, both with and without delayed feedback. To the best of our knowledge, this gives the first affirmative resolution of SOLID in the adversarial delayed-feedback setting, improving the immediate-feedback framework of Müller et al. [48].

Theorem 1.1 (Informal Main Result)**.** PRUDENT-BANKER *is* SOLID *and achieves the optimal safety–robustness instance-dependent constants up to logarithmic factors.*

2 Preliminaries & Problem Setup

Basic Notation. Let $[A] \coloneqq \{1, 2, \ldots, A\}$ denote the set of arms. We denote by Δ_A the probability simplex over $[A]$. For vectors $x, y \in \mathbb{R}^A$, we denote the inner product by $\langle x, y \rangle$. We use $\mathbf{e}_a$ for the a-th standard basis vector and $\tilde{O}(\cdot)$ to hide polylogarithmic factors in T, D, and A. For the stated rates in T, D, the comparator margin $\delta > 0$ is fixed; hidden constants may depend on δ, C_1, C_2.

The Delayed Adversarial Bandit Protocol. We consider adversarial bandits with delayed feedback over horizon T against an oblivious adversary. At each round $t \in [T]$, the adversary chooses $\ell_t \in [0,1]^A$ and delay $d_t \in \mathbb{N}_0$; the learner chooses $x_t \in \Delta_A$, samples $A_t \sim x_t$, incurs unobserved loss ℓ_{t,A_t}, and observes (A_t, ℓ_{t,A_t}) at the end of round $t + d_t$.

Feedback Sets and Delay Counters. Let $\mathcal{O}_t \coloneqq \{\tau \le t : \tau + d_\tau = t\}$ be the feedback arriving at the end of round t, and let $D \coloneqq \sum_{t=1}^T d_t$ be the *total delay*. Following Huang et al. [30], let $\mathfrak{d}_t \coloneqq \sum_{\tau<t} \mathbf{1}\{t \le \tau + d_\tau\}$ be the number of outstanding observations at t and $\mathfrak{D}_t \coloneqq \sum_{r \le t} \mathfrak{d}_r$, with $\mathfrak{D}_T \le D$. The stage counter $\mathfrak{D}_t^{(s)}$ accumulates $\mathfrak{d}_t$ and is reset only at a hard restart. Outstanding-feedback records are retained across both kinds of restart. At the end of round t, after processing arrivals, $\mathfrak{d}_{t+1}$ is the observable number of feedback items still outstanding. For the k-th phase, denoted $k(s)$, within stage s of PRUDENT-BANKER with start time $\mathrm{start}_{k(s)}$, we define the phase-local delay parameters up to time t as $D_t^{k(s)} \coloneqq \sum_{r=\mathrm{start}_{k(s)}}^t d_r$, $\mathfrak{d}_t^{k(s)} \coloneqq \sum_{\tau=\mathrm{start}_{k(s)}}^{t-1} \mathbf{1}\{t \le \tau + d_\tau\}$, and $\mathfrak{D}_t^{k(s)} \coloneqq \sum_{u=\mathrm{start}_{k(s)}}^t \mathfrak{d}_u^{k(s)}$.

Regularization and Banker-OMD Setup. Our active learner uses OMD with a strictly convex, continuously differentiable Legendre regularizer $\Psi : \mathcal{D} \to \mathbb{R}$ on an open convex set $\mathcal{D} \supseteq \mathrm{ri}(\Delta_A)$, with continuous extension to the simplex boundary. We write $\bar{\Psi}$ for its restriction to Δ_A by setting $\bar{\Psi}(x) = +\infty$ for $x \notin \Delta_A$, s.t. $\bar{\Psi}^*(y) \coloneqq \sup_{x \in \mathbb{R}^A}\{\langle y, x\rangle - \bar{\Psi}(x)\}$ implicitly enforces the simplex constraint. The unprojected point in local stability is $\tilde{z}_t = \nabla\Psi^*(\nabla\Psi(\hat{x}_t) - \hat{\ell}_t/\sigma_t)$. Finally, we denote the associated *Bregman divergence* as:

$$D_\Psi(x, y) \coloneqq \Psi(x) - \Psi(y) - \langle \nabla\Psi(y), x - y\rangle.$$

We characterize the geometry ("width" and "curvature") of the regularizer via the following constants:

Definition 2.1 ((C_1, C_2)-Regularity)**.** The regularizer Ψ and initial point x_0 satisfy:

⋆ **Bounded Diameter:** $D_\Psi(y, x_0) \le C_1$ for all $y \in \Delta_A$.
⋆ **Local Norm Stability:** $\mathbb{E}\left[\sigma_t D_\Psi\left(\hat{x}_t, \tilde{z}_t\right)\right] \le C_2/\sigma_t$ for any $\hat{x}_t \in \Delta_A$ and $\sigma_t > 0$.

The expectation is over the current arm draw for each fixed iterate and scale; $\mathcal{F}_{t-1}$ denotes the history before that draw. *Remark. For negative entropy,* $(C_1, C_2) = (\ln A, 1/\delta)$*; for* $1/2$*-Tsallis entropy,* $(C_1, C_2) = (2(\sqrt{A} - 1), 2/\delta)$*. Details are defered in the appendix.*

Performance Metrics. We measure our performance using standard adversarial *pseudo-regret*:

$$\mathcal{R}_T \coloneqq \max_{x \in \Delta_A} \mathbb{E}\left[\sum_{t=1}^T \langle \ell_t, x_t - x\rangle\right] \qquad \text{(Adversarial Regret)}$$

The learner is also equipped with a designated *safe baseline policy* $x^c \in \Delta_A$. In light of the impossibility result of Lattimore [39] and the relaxation advocated by Müller et al. [48], we focus on *smoothed* comparators and assume that the baseline is *uniformly exploratory*, i.e., it assigns nonzero probability to every action:

Assumption 2.1 (Exploratory Baseline)**.** There exists a constant $\delta \in (0, 1/A]$ such that $x^c(a) \ge \delta$ for all arms $a \in [A]$.

A baseline b can be smoothed as $x^c = (1 - \varepsilon)b + \varepsilon\mathbf{1}/A$, giving $\delta = \varepsilon/A$; transferring safety back to b costs at most εT. Our secondary objective is to minimize the *comparator regret* with respect to this baseline, as in Müller et al. [48]; the baseline-to-best-arm gap instead determines restarts:

$$\mathcal{R}(x^c) \;\coloneqq\; \mathbb{E}\left[\sum_{t=1}^T \langle \ell_t, x_t - x^c\rangle\right] \qquad \text{(Comparator Regret)}$$

3 The PRUDENT-BANKER Algorithm

We show that PRUDENT-BANKER is to guarantee $\mathcal{R}_T = \tilde{O}(\sqrt{T} + \sqrt{D})$ worst-case, while ensuring $\mathcal{R}(x^c) = O(\log D)$ relative to the safe baseline. At a high level, PRUDENT-BANKER runs in *epochs* and combines three interlocking components: (i) an *active no-regret learner* based on `Banker-OMD`; (ii) a *mixing schedule* implementing phased aggression; and (iii) a *safety guardrail* based on delay-calibrated restarts.

3.1 Algorithm Description

The pseudocode for PRUDENT-BANKER is provided in Algorithm 1. For clarity, we present the core logic here; the formal definition, including the precise configuration of the `Banker-OMD` step-size schedules and constants, is deferred to the supplement (See Appendix B).

The algorithm orchestrates three hierarchical levels of adaptation:(1) *Stages* to adapt to the unknown total delay D, (2) a *Banker-OMD* core to handle local feedback latency, and (3) *Phases* to adapt the aggression α for safety. More analytically, the following observations are in order.

Algorithm 1 PRUDENT-BANKER

1: **Instantiate Banker:** $\mathcal{B} \leftarrow$ `Banker-OMD`(Ψ, S)
2: {regularizer Ψ and step-size routine S}
3: **Input:** Initial & comparator strategies (x_0, x^c), $x_0 = \mathbf{1}/A$, horizon T, comparator margin $\delta \in (0, 1)$.
4: **Parameters:** Constants $C_1, C_2 > 0$ for thresholding.
5: **Initialize:** $\mathfrak{C} \leftarrow 10\sqrt{C_1^{\Psi} C_2^{\Psi}}$; global time $t \leftarrow 1$.
6: **Delay-Stage** $s \leftarrow 1$, Delay Estimate $\widehat{D}_s \leftarrow 1$.
7: **Aggression-Phase** $k \leftarrow 1$, Start of instance $\tau_s \leftarrow 1$.
8: **Define functions:**
9: $\widehat{R}(D) \leftarrow \mathfrak{C}\big(\sqrt{T} + \sqrt{D\log(D+1)}\big)$
10: $\widehat{\xi}(D) \leftarrow \frac{\sqrt{8D+1}-1}{\delta}$, $B(D) \leftarrow 2 \cdot 2^{\lceil \log_2 \widehat{R}(D) \rceil} + \widehat{\xi}(D)$
11: Aggression $\alpha_k \leftarrow \min\{1, 1/2^{\lceil \log_2 \widehat{R}(\widehat{D}_s) \rceil}\}$.
12: Initialize `Banker-OMD` state $x_{\tau_s} = x_0$, $\mathfrak{D}_0^{(1)} \leftarrow 0$.
13: **while** $t \le T$ **do**
14: {Step 1}
15: {Adaptive Delay Check Stage Logic}
16: Observe outstanding feedback; $\mathfrak{D}_t^{(s)} \leftarrow \mathfrak{D}_{t-1}^{(s)} + \mathfrak{d}_t$.
17: **if** $\mathfrak{D}_t^{(s)} > \widehat{D}_s$ **then**
18: $\widehat{D}_{s+1} \leftarrow 2^{\lceil \log_2 \mathfrak{D}_t^{(s)} \rceil}$, $s \leftarrow s+1$
19: {Doubling Trick on Delay}
20: **Hard Restart:** $\tau_s \leftarrow t$, $k \leftarrow 1$, Reset $\mathcal{B}$ to x_0.
21: Recalculate α_k with new $\widehat{D}_s$; $\mathfrak{D}_t^{(s)} \leftarrow 0$.
22: **end if**
23: {Step 2: Base Learner (Banker-OMD)}
24: Query learner $\mathcal{B}$ for base prediction $\hat{x}_t$.
25: *(Internal):* $\mathcal{B}$ computes step-size σ_t based on S and available feedback credits.
26: {Step 3: Safe Mixture}
27: Sample $A_t \sim x_t \leftarrow \alpha_k \hat{x}_t + (1-\alpha_k)x^c$.
28: Incur loss $\ell_t(x_t)$ (unobserved), receive feedback for past rounds $u \in \mathcal{O}_t$.
29: Update $\mathcal{B}$ with pairs $\{(u, x_u(A_u), \ell_u)\}$ for all $u \in \mathcal{O}_t$ and construct the importance-weighted estimator
30: $\hat{\ell}_u \leftarrow \frac{\ell_u(A_u)}{x_u(A_u)} e_{A_u}$.
31: {Step 4: Phased Safety Check}
32: Compute empirical gap against baseline on observed data:
33: $\widehat{\Delta}_k(t) = \max_{x \in \Delta_A} \sum_{u=\tau_s}^{t} \mathbb{I}\{u \in \text{ARRIVED}\}\langle \hat{\ell}_u, x^c - x\rangle$
34: **if** $\widehat{\Delta}_k(t) > B(\widehat{D}_s)$ **and** $\alpha_k < 1$ **and** $\mathfrak{D}_t^{(s)} + \mathfrak{d}_{t+1} \le \widehat{D}_s$ **then**
35: $k \leftarrow k+1$
36: {Enter new Aggression Phase}
37: Increase aggression: $\alpha_k \leftarrow \min\{2^{k-1}/2^{\lceil \log_2 \widehat{R}(\widehat{D}_s) \rceil}, 1\}$
38: **Soft Restart:** $\tau_s \leftarrow t+1$, Reset $\mathcal{B}$ to $\mathbf{1}/A$.
39: **end if**
40: $t \leftarrow t+1$
41: **end while**

1. Delay-Stage Logic (Unknown D). The outermost layer handles the unknown total delay. We define our threshold function $B(D)$ based on the regret $\widehat{R}(D)$ and a slack function $\widehat{\xi}(D)$ (Line 10). The algorithm maintains a delay estimate $\widehat{D}_s$, initially optimistic. If the accumulated delay within a stage exceeds this estimate (Line 17), a *Stage* transition is triggered ($s \to s+1$). This involves doubling the estimate $\widehat{D}_{s+1} < 2\mathfrak{D}_t^{(s)}$ and performing a **Hard Restart** (Line 20), effectively resetting the instance to adapt to the newly discovered delay scale.

2. Banker-OMD Core (Feedback Latency). At the heart of the process is the `Banker-OMD` learner $\mathcal{B}$. It manages local latency by maintaining an account of step-size credits, computing gradients using importance-weighted estimators $\hat{\ell}_u$ constructed only upon feedback arrival (Lines 28–29). This ensures that the base strategy $\hat{x}_t$ maintains $\tilde{O}(\sqrt{T} + \sqrt{D})$ regret regardless of the arrival order.

3. Aggression-Phase Logic (Safety). The safety mechanism is controlled by the aggression parameter α_k. We continuously monitor the empirical gap $\widehat{\Delta}_k(t)$ between the baseline x^c and the best fixed comparator on the *observed* data (Lines 32-34). Crucially, the restart threshold $B(\widehat{D}_s) = 2 \cdot 2^{\lceil \log_2 \widehat{R}(\widehat{D}_s) \rceil} + \widehat{\xi}(\widehat{D}_s)$ includes a delay-dependent slack $\widehat{\xi}(\widehat{D}_s)$ scaling with $\sqrt{\widehat{D}_s/\delta}$.

This term acts as a buffer against the hidden performance gap caused by missing feedback. If the gap exceeds this threshold (Line 34) and the outstanding-feedback guard holds, we conclude that the baseline is suboptimal. The algorithm then enters a new *Phase* ($k \to k+1$), increases aggression to $\alpha_k \propto 2^{k-1}/\widehat{R}$ (Line 37), and performs a **Soft Restart**, resetting the learner to the uniform distribution to lock in safety gains while exploiting the optimal arm.

Remark 3.1 (The roles of the two restarts). *A **Soft Restart** implements phase aggression: once the observed data certifies that the safe comparator is suboptimal, the algorithm starts a new phase with a larger weight on the active learner. A **Hard Restart** implements delay adaptation: ideally,* PRUDENT-BANKER *would be calibrated using the Banker-OMD regret budget, but this budget depends on the unknown total delay D, including future unseen feedback. Since this quantity is not available online, the algorithm maintains a delay guess $\widehat{D}_s$ and restarts whenever the certified delay scale exceeds it. Thus, **Soft Restart** adapts the baseline–exploration mixture, whereas **Hard Restart** adapt the regret budget to the unknown delay landscape. Without soft restarts, aggression cannot increase; a known valid delay bound permits omitting hard restarts.*

4 Main Results and Analysis

4.1 Proof Sketch: Safety under Latency

We now present our main theoretical guarantees. The analysis relies on a hierarchical decomposition of the regret, treating delay adaptation and safety constraints as coupled objectives. We begin with a high-level roadmap of the proof architecture. Our proof strategy unfolds in three logical steps:

1. Decomposition via Stages and Phases. We first bound the number of delay-adaptive stages using Lemma 4.1, ensuring that the doubling trick on $\widehat{D}_s$ incurs only logarithmic overhead. Within each stage, the regret is analyzed through the lens of *phases*. We distinguish between *Normal Phases* (where the algorithm maintains the baseline mixture) and *Extinction Phases* (where the algorithm detects suboptimality and restarts).

2. The Latency-Robust Base Learner. A core component is establishing that the base learner remains robust to arbitrary delays. Lemma 4.3 proves that `Banker-OMD`, by managing a virtual account of step-sizes, achieves the optimal $\tilde{O}(\sqrt{T}+\sqrt{D})$ regret bound regardless of the arrival order of feedback.

3. Bridging the Gap: The Cost of Missing Information. The most delicate aspect of the proof—distinguishing our work from classical phased aggression—is handling the *pending feedback*. In standard settings, the empirical gap is exact. Here, the safety check is performed on stale data. We control this via Lemma 4.4, which bounds the potential deviation caused by unobserved losses. This bound motivates the slack term $\widehat{\xi}(\widehat{D}_s)$ in our threshold. Finally, Lemma 4.5 and Lemma 4.6 bound the phase cost and the negative restart credit, retaining the realized Banker-regret remainder. Summing these contributions in expectation gives the geometric cancellation of exploration costs, yielding the result of Theorem 4.1.

4.2 Our Algorithmic Guarantee

Our first result shows that PRUDENT-BANKER simultaneously achieves near-minimax adversarial robustness and bounded regret against the safe baseline, even under arbitrary delays.

Theorem 4.1 (Global Safety and Robustness)**.** *Under Definition 2.1 and Assumption 2.1, for any delay sequence with total delay D, let $\{x_t\}_{t=1}^T$ be the strategies generated by* PRUDENT-BANKER *with step-size schedule*

$$\sigma_t = \sqrt{\frac{C_2}{C_1}}\left(\frac{1}{\sqrt{t-\mathrm{start}_{k(s)}+1}} + \mathfrak{d}_t^{k(s)}\sqrt{\frac{\ln(\mathfrak{D}_t^{k(s)}+1)}{\mathfrak{D}_t^{k(s)}}}\right)^{-1}.$$

Then $\mathcal{R}_T \le \widetilde{O}\Big(\sqrt{T}+\sqrt{D}\Big)$ *and* $\mathcal{R}_T(x^c) \le O(1+\log(1+D)) = \widetilde{O}(1)$.

Expanding $\widetilde{O}\Big(\sqrt{T}+\sqrt{D}\Big)$ gives $\tilde{O}\Big(\sqrt{C_1C_2}(\sqrt{T}+\sqrt{D})+\sqrt{D}/\delta\Big)$. As an immediate consequence, in the non-delayed case our bound closes the gap between the safety-constrained upper bound $O(\delta^{-1}\sqrt{T})$) and lower bound $\Omega(\sqrt{\delta^{-1}T})$) left by Müller et al. [48].

Corollary 4.1 (Optimal non-delayed safety-constrained regret)**.** *In the non-delayed setting $D = 0$, instantiating* PRUDENT-BANKER *with negative-entropy regularization where $C_1C_2 = \log A/\delta$ gives*

$$\mathcal{R}_T \le \widetilde{O}\Big(\sqrt{\delta^{-1}T\log A}\Big), \qquad \mathcal{R}_T(x^c) \le O(1).$$

Consequently, PRUDENT-BANKER *matches the safety-constrained lower bound of order $\Omega(\sqrt{T/\delta})$ up to logarithmic factors.*

Deciphering the arm dependence. Without the safety constraint, the minimax adversarial regret is bounded below by $\Omega(\sqrt{AT}+\sqrt{D\log A})$. It is therefore natural to ask how our guarantee compares to this unconstrained benchmark. If we take the safe comparator to be uniform (yielding $\delta = 1/A$) and instantiate PRUDENT-BANKER with negative entropy, Theorem 4.1 gives an upper bound of $\widetilde{O}(\sqrt{AT\log A}+A\sqrt{D})$. Thus, relative to the unconstrained benchmark, our dependence on T is optimal up to logarithmic factors, while the dependence on D incurs an extra A penalty. This dependence arises from the missing-feedback slack $\sqrt{D}/\delta$ when $\delta = 1/A$, whereas the Banker-OMD component contributes $\widetilde{O}(\sqrt{AD})$; the unconstrained benchmark alone does not identify the cost of safety. Another backbone would need a comparator-uniform stopped-phase bound for estimates using the played mixture.

However, this apparent inflation in regret is not an artifact of our analysis, but rather the fundamental price of safety. As we discuss in Section 4.4, our novel lower bounds prove that this resulting dependence is optimal for the safety-constrained problem, up to logarithmic factors.

4.3 Key Lemmas

We now detail the technical lemmas that constitute the building blocks of the proof.

Lemma 4.1 (Conservation of Delayed Mass)**.** *Let $\mathfrak{D}_t^{k(s)}$ denote the cumulative number of missing feedback items inside phase $k(s)$ up to time t. Then, for all $start_{k(s)} \le t \le end_{k(s)}$:*

$$\mathfrak{D}_t^{k(s)} \le \mathfrak{D}_{\mathrm{end}_{k(s)}}^{k(s)} \le D_{\mathrm{end}_{k(s)}}^{k(s)}$$

Remark 4.1. This structural result allows us to decouple the instantaneous missing mass from the time index. It implies that the "penalty" for missing information in any regret bound is bounded by the total delay complexity $D_{\mathrm{end}_{k(s)}}^{k(s)}$ inside phase $k(s)$, rather than the length of phase $k(s)$. This validates the usage of $D_{\mathrm{end}_{k(s)}}^{k(s)}$ as the natural complexity measure for the delay-adaptive step-sizes.

Lemma 4.2 (Bound on Estimated Total Delay)**.** *At a hard restart in round t, the total-delay capacity satisfies:*

$$\widehat{D}_s \le \widehat{D}_{s+1} < 2\mathfrak{D}_t^{(s)}$$

Remark 4.2. This ensures that the number of stages remains logarithmic ($S \approx O(\log D)$), allowing us to sum per-stage regrets efficiently without incurring a linear penalty. For each phase $k(s)$ inside stage s, they share the same $\widehat{D}_s$. This indicates we can interpret $\widehat{D}_s = \widehat{D}_{k(s)}$.

Lemma 4.3 (Banker-OMD Guarantee)**.** *Consider the* `Banker-OMD` *iterates $\{\hat{x}_t\}$ using adaptive the step-sizes of Theorem 4.1 Then, the regret against any comparator $x \in \Delta_A$ satisfies:*

$$\max_{x\in\Delta_A} \mathbb{E}\left[\sum_{t=\mathrm{start}_{k(s)}}^{\mathrm{end}_{k(s)}} \langle \hat{\ell}_t, \hat{x}_t - x\rangle\right] \le \tilde{O}\Big(\sqrt{C_1C_2}\,\mathbb{E}[\sqrt{T_s}+\sqrt{D_{\mathrm{end}_{k(s)}}^{k(s)}}]\Big)$$

where $T_s = end_{k(s)} - start_{k(s)} + 1$.

Remark 4.3. Lemma 4.3 establishes the "best-of-both-worlds" capability of the base learner. By dynamically adjusting the learning rate σ_t based on the arrived data density, the algorithm essentially pauses learning during high-latency periods and accelerates when feedback becomes available, ensuring optimal worst-case performance.

Lemma 4.4 (Bound on Missing Feedback)**.** *Let $\mathcal{M}(T_0)$ denote the set of rounds $u < T_0$ whose feedback is missing before round T_0, with $\mathrm{start}_{k(s)} \le T_0 \le \mathrm{end}_{k(s)} + 1$, inside a mixed phase $k(s)$. Then, for any comparator $x \in \Delta_A$:*

$$\sum_{u \in \mathcal{M}(T_0)} \langle \hat{\ell}_u,\, x^c - x \rangle \;\le\; 2\sqrt{2D^{k(s)}_{\mathrm{end}_{k(s)}} + \tfrac{1}{4}} - 1.$$

Remark 4.4. This lemma quantifies the risk of acting on stale information. Intuitively, missing feedback obscures the true performance gap. Lemma 4.4 bounds the worst-case error introduced by these "phantom" losses. This bound is pivotal for the safety certification, as it dictates the magnitude of the slack term ξ required in the restart condition to prevent false positives.

The missing-feedback bound motivates the slack in Algorithm 1. For each delay stage s, define

$$\widehat{R}_s := 2^{\lceil \log_2 \widehat{R}(\widehat{D}_s) \rceil}, \qquad \widehat{\xi}_s := \widehat{\xi}(\widehat{D}_s) = \frac{\sqrt{8\widehat{D}_s + 1} - 1}{\delta}.$$

The power-of-two regret budget $\widehat{R}_s$ and missing-feedback slack $\widehat{\xi}_s$ remain fixed across all phases of stage s. In this notation, the restart threshold is $B(\widehat{D}_s) = 2\widehat{R}_s + \widehat{\xi}_s$ and the aggression schedule is $\alpha_{k(s)} = \min\{2^{k(s)-1}/\widehat{R}_s, 1\}$.

Lemma 4.5 (Normal Phase Analysis)**.** *Consider any mixed phase $k(s)$ with $\alpha_{k(s)} < 1$. All expectations in this lemma are conditional on its starting history. For all $x \in \Delta_A$, the regret is bounded by the stage budget and missing feedback:*

$$\mathbb{E}\left[\sum_{t=\mathrm{start}_{k(s)}}^{\mathrm{end}_{k(s)}} \langle \hat{\ell}_t, x_t - x \rangle\right] \;\le\; 4\widehat{R}_s + \widehat{\xi}_s + 2\sqrt{2\widehat{D}_s + \tfrac{1}{4}} + 1.$$

Furthermore, the regret against the safe baseline satisfies: $\mathbb{E}\left[\sum_{t=\mathrm{start}_{k(s)}}^{\mathrm{end}_{k(s)}} \langle \hat{\ell}_t, x_t - x^c \rangle\right] \;\le\; 2^{k(s)-1}$.

Remark 4.5. Above we ensure that as long as the algorithm remains in a specific phase, the deviation from the baseline is controlled by the current aggression level α_k. The bound reflects that the empirical evidence (the gap statistic) has not yet provided sufficient proof to justify abandoning the current mixture.

Lemma 4.6 (Extinction Phase Analysis)**.** *If phase $k(s)$ is terminated by a soft-restart condition (indicating the baseline is suboptimal), then pathwise:*

$$\sum_{t=\mathrm{start}_{k(s)}}^{\mathrm{end}_{k(s)}} \langle \hat{\ell}_t,\, x_t - x^c \rangle \;\le\; -2^{k(s)-1} + \alpha_{k(s)} \left(\max_{x \in \Delta_A} \sum_{t=\mathrm{start}_{k(s)}}^{\mathrm{end}_{k(s)}} \langle \hat{\ell}_t, \hat{x}_t - x \rangle - \widehat{R}_s \right).$$

Remark 4.6. Lemma 4.6 is the mechanism that funds the exploration. It retains the negative restart credit together with the realized Banker-regret remainder. This restart credit offsets the next phase's doubled budget in expectation, as shown in Theorem C.1.

Summary. *Theorem 4.1 unifies the phase-level analyses to provide a global guarantee. For the pseudo-regret $\mathcal{R}_T$, the bound is dominated by the final aggressive phases where $\alpha \approx 1$, matched by the guarantee of the base learner (Lemma 4.3). Since α_k doubles geometrically, there are only logarithmically many phases, preserving any $\tilde{O}(\cdot)$ rate. Crucially, for the safety constraint $\mathcal{R}_T(x^c)$, the geometric restarts ensure that each next phase's budget is paid for by the preceding restart credit in expectation (Lemma 4.6), resulting in a telescopic cancellation that bounds the deviation by a constant per stage.*

4.4 Lower Bound: The Price of Safety Under Delay

Before concluding with the experiments, we ask the converse question:

What is the best regret any safe algorithm can hope for under delayed bandit feedback?

We prove a lower bound showing that the $\sqrt{\delta^{-1}T} + \sqrt{\delta^{-1}D}$ rate achieved by PRUDENT-BANKER is unimprovable in the safety-constrained regime—matching the standard delayed-bandit lower bounds while exposing that $\delta^{-1/2}$ overhead is the intrinsic cost of the safety constraint.

Our lower-bound provides a novel interplay between the two environments (safe,unsafe) [48] and a construction that follows the blocking methodology used in delayed bandit lower bounds, particularly the batching idea of Zimmert and Seldin [64]. The key observation is that a delayed bandit instance can be organized into blocks so that feedback from actions played inside one block is revealed only after the block has ended. We first prove a safety-constrained lower bound for the corresponding batched bandit game. We then show that any delayed bandit algorithm can be simulated by a batched algorithm on the induced block structure. This reduction transfers the batched safe lower bound back to the delayed setting and yields a delay-side lower bound of order $\delta^{-1/2}\sqrt{D}$. For concision, we defer the proof to Section E.

Theorem 4.2 (Delay-side price of safety; informal)**.** *There exist delay sequences with total delay D and a safe comparator x^c satisfying $\min_i x_i^c \geq \delta$ such that every delayed bandit algorithm satisfies, for some adversarial loss sequence,*

$$\mathcal{R}_T(x^c) \leq \log D \quad \Longrightarrow \quad \mathcal{R}_T \geq \widetilde{\Omega}\Big(\delta^{-1/2}\sqrt{D}\Big)\,.$$

Putting time and delay together. The no-delay lower bound of Müller et al. [48, Theorem 3.2] already shows that baseline safety alone forces regret $\Omega(\sqrt{\delta^{-1}T})$ up to lower-order terms. Combining this obstruction with Theorem 4.2, we obtain delay and loss sequences for which any baseline-safe algorithm must incur

$$\mathcal{R}_T \geq \Omega\Big(\sqrt{\delta^{-1}T} + \sqrt{\delta^{-1}D}\Big) - O\Big(\delta^{-3/4}T^{1/4} + \delta^{-3/4}D^{1/4}\Big)\,.$$

Thus the guarantee of PRUDENT-BANKER in Theorem 4.1 is optimal up to logarithmic factors and lower-order terms, both in the delayed and non-delayed regimes.

5 Experiments

We conclude our work by empirically examining the safety–learning trade-off of PRUDENT-BANKER under delay. We pit PRUDENT-BANKER against two safety-aware baselines run with delayed feedback: CONSERVATIVE-UCB [62] and SAFE-EXP3-IX, which couples the conservative wrapper of Wu et al. [62] with EXP3-IX [36, 49]. All three methods process feedback only upon arrival; the distinguishing feature of PRUDENT-BANKER is its *delay-calibrated* restart rule, while the baselines treat delay implicitly through lazy updates.

Setup. We evaluate on a non-stationary synthetic bandit environment with horizon $T = 50{,}000$ and $A = 100$ arms. Losses are generated in $S = 500$ blocks, with block-specific means and variances so that the best arm changes over time. To stress-test safety, the comparator x^c is a full-support distribution concentrated on the hindsight best fixed arm $i^\star$, assigning mass $1 - (A-1)\delta$ to $i^\star$ and $\delta = 10^{-3}$ to every other arm. This oracle choice is deliberately favorable to the baselines and isolates the safety–learning trade-off. We compare performance under three delay regimes: no delay, geometric delays, and heavy-tailed Pareto delays (1% activation for the delayed conditions), using the same loss tables across conditions.[3]

[3] Full implementation details, parameter choices, and additional ablations are deferred to Section F. Figure 3 reports arm-count and ten-seed comparisons. The plots use the empirical restart-threshold multiplier 10^{-3}; the theorem uses multiplier 1.

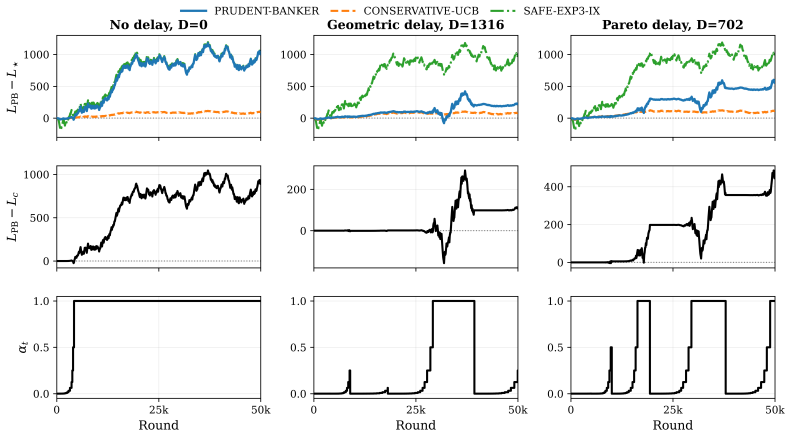


Figure 1: Dynamic performance of PRUDENT-BANKER versus delay-aware safety baselines. Results use random seed 0. *Top:* cumulative pseudo-loss minus that of the hindsight best fixed arm $i^\star$. *Middle:* PRUDENT-BANKER cumulative loss minus comparator loss L_c (negative values mean PRUDENT-BANKER beats x^c). *Bottom:* the PRUDENT-BANKER mixing coefficient α_t. Under delay, PRUDENT-BANKER repeatedly performs conservative restarts and only re-enters aggressive phases after the delay-calibrated threshold is crossed.

Results. Figure 1 illustrates three aspects of the tuned implementation. *(i) Comparator gap.* The middle row has positive terminal gaps in all three settings (about 920, 112, and 471), with a temporary negative interval under geometric delay. By contrast, the safety wrappers built around CONSERVATIVE-UCB and SAFE-EXP3-IX only enforce a multiplicative $(1 - \alpha_{\text{safe}})$ budget and cannot certify comparator-level safety. *(ii) Delay-calibrated phase transitions.* The bottom row visualizes the operational role of the delay-aware restart threshold: in the no-delay column, α_t rises monotonically to 1 and stays there, recovering the immediate-feedback behavior of phased aggression; under the geometric and Pareto delays, α_t oscillates as the observable delay count repeatedly exceeds the current estimate $\widehat{D}_s$, triggering hard restarts and demanding fresh evidence before re-aggression—exactly the mechanism that Lemmas 4.4 and 4.6 were designed to enforce. *(iii) Empirical competitiveness.* In the top row, PRUDENT-BANKER has lower terminal loss than SAFE-EXP3-IX, whose adversarial exploration is wasteful when the comparator is already concentrated. CONSERVATIVE-UCB remains closer to $i^\star$, benefiting from its default arm *being* $i^\star$ in this diagnostic setting; this oracle advantage does not test construction of a safe anchor online.

Takeaway. Together with upper and the lower bounds, our experiments support the central message of the paper: *baseline safety and delay-robust adversarial learning can be combined without losing the optimal safety–robustness trade-off.* Natural next steps include adaptive or learned baselines and extensions to contextual, linear, reinforcement-learning, and multi-agent settings.

Acknowledgments and Disclosure of Funding

We thank Andreas Müller for insightful discussions during the preliminary stages of this work. Emmanouil V. Vlatakis-Gkaragkounis acknowledges support from the faculty startup funds provided by the University of Wisconsin–Madison.

Competing Interests. The authors declare no competing financial interests related to this work.

References

[1] Jacob D Abernethy, Elad Hazan, and Alexander Rakhlin. Competing in the dark: An efficient algorithm for bandit linear optimization. In *Conference on Learning Theory*, number 110, 2009.

[2] Alekh Agarwal and John C Duchi. Distributed delayed stochastic optimization. *Advances in neural information processing systems*, 24, 2011.

[3] Peter Auer and Ronald Ortner. Ucb revisited: Improved regret bounds for the stochastic multi-armed bandit problem. *Periodica Mathematica Hungarica*, 61(1-2):55–65, 2010.

[4] Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E Schapire. The nonstochastic multiarmed bandit problem. *SIAM journal on computing*, 32(1):48–77, 2002.

[5] Ashwinkumar Badanidiyuru, Robert Kleinberg, and Aleksandrs Slivkins. Bandits with knapsacks. *Journal of the ACM (JACM)*, 65(3):1–55, 2018.

[6] Amir Beck and Marc Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, 31(3):167–175, 2003.

[7] Lilian Besson and Emilie Kaufmann. What doubling tricks can and can't do for multi-armed bandits. *arXiv preprint arXiv:1803.06971*, 2018.

[8] Ilai Bistritz, Zhengyuan Zhou, Xi Chen, Nicholas Bambos, and Jose Blanchet. Online exp3 learning in adversarial bandits with delayed feedback. *Advances in neural information processing systems*, 32, 2019.

[9] Ilai Bistritz, Zhengyuan Zhou, Xi Chen, Nicholas Bambos, and Jose Blanchet. No weighted-regret learning in adversarial bandits with delays. *Journal of Machine Learning Research*, 23 (139):1–43, 2022.

[10] Nicolo Cesa-Bianchi, Yishay Mansour, and Gilles Stoltz. Improved second-order bounds for prediction with expert advice. *Machine Learning*, 66(2):321–352, 2007.

[11] Nicolo Cesa-Bianchi, Claudio Gentile, and Yishay Mansour. Delay and cooperation in nonstochastic bandits. *Journal of Machine Learning Research*, 20(17):1–38, 2019.

[12] Nesrine Changuel, Bessem Sayadi, and Michel Kieffer. Online learning for qoe-based video streaming to mobile receivers. In *2012 IEEE Globecom Workshops*, pages 1319–1324. IEEE, 2012.

[13] Baiming Chen, Mengdi Xu, Zuxin Liu, Liang Li, and Ding Zhao. Delay-aware multi-agent reinforcement learning for cooperative and competitive environments. *arXiv preprint arXiv:2005.05441*, 2020.

[14] Ashok Cutkosky and Francesco Orabona. Black-box reductions for parameter-free online learning in banach spaces. In *Conference On Learning Theory*, pages 1493–1529. PMLR, 2018.

[15] Yan Dai, Haipeng Luo, and Liyu Chen. Follow-the-perturbed-leader for adversarial markov decision processes with bandit feedback. *Advances in Neural Information Processing Systems*, 35:11437–11449, 2022.

[16] Steven Damer and Maria Gini. Safely using predictions in general-sum normal form games. In *Proceedings of the 16th conference on autonomous agents and multiagent systems*, pages 924–932, 2017.

[17] Thomas Desautels, Andreas Krause, and Joel W Burdick. Parallelizing exploration-exploitation tradeoffs in gaussian process bandit optimization. *J. Mach. Learn. Res.*, 15(1):3873–3923, 2014.

[18] Miroslav Dudik, Daniel Hsu, Satyen Kale, Nikos Karampatziakis, John Langford, Lev Reyzin, and Tong Zhang. Efficient optimal learning for contextual bandits. *arXiv preprint arXiv:1106.2369*, 2011.

[19] Yonathan Efroni, Shie Mannor, and Matteo Pirotta. Exploration-exploitation in constrained mdps. *arXiv preprint arXiv:2003.02189*, 2020.

[20] Eyal Even-Dar, Michael Kearns, Yishay Mansour, and Jennifer Wortman. Regret to the best vs. regret to the average. *Machine Learning*, 72(1):21–37, 2008.

[21] Genevieve E Flaspohler, Francesco Orabona, Judah Cohen, Soukayna Mouatadid, Miruna Oprescu, Paulo Orenstein, and Lester Mackey. Online learning with optimism and delay. In *International Conference on Machine Learning*, pages 3363–3373. PMLR, 2021.

[22] Abraham D Flaxman, Adam Tauman Kalai, and H Brendan McMahan. Online convex optimization in the bandit setting: gradient descent without a gradient. *arXiv preprint cs/0408007*, 2004.

[23] Pierre Gaillard, Gilles Stoltz, and Tim Van Erven. A second-order bound with excess losses. In *Conference on Learning Theory*, pages 176–196. PMLR, 2014.

[24] Sam Ganzfried and Tuomas Sandholm. Safe opponent exploitation. *ACM Transactions on Economics and Computation (TEAC)*, 3(2):1–28, 2015.

[25] Evrard Garcelon, Mohammad Ghavamzadeh, Alessandro Lazaric, and Matteo Pirotta. Conservative exploration in reinforcement learning. In *International conference on artificial intelligence and statistics*, pages 1431–1441. PMLR, 2020.

[26] Geoffrey J Gordon. Regret bounds for prediction problems. In *Proceedings of the twelfth annual conference on Computational learning theory*, pages 29–40, 1999.

[27] Andras Gyorgy and Pooria Joulani. Adapting to delays and data in adversarial multi-armed bandits. In *International Conference on Machine Learning*, pages 3988–3997. PMLR, 2021.

[28] Benjamin Howson, Ciara Pike-Burke, and Sarah Filippi. Delayed feedback in episodic reinforcement learning. *arXiv preprint arXiv:2111.07615*, 2021.

[29] Jiatai Huang and Longbo Huang. Robust wireless scheduling under arbitrary channel dynamics and feedback delay. In *2021 33th International Teletraffic Congress (ITC-33)*, pages 1–8. IEEE, 2021.

[30] Jiatai Huang, Yan Dai, and Longbo Huang. Banker online mirror descent: A universal approach for delayed online bandit learning. In *International Conference on Machine Learning*, pages 13814–13844. PMLR, 2023.

[31] Marcus Hutter and Jan Poland. Adaptive online prediction by following the perturbed leader. 2005.

[32] Shinji Ito, Daisuke Hatano, Hanna Sumita, Kei Takemura, Takuro Fukunaga, Naonori Kakimura, and Ken-Ichi Kawarabayashi. Delay and cooperation in nonstochastic linear bandits. *Advances in Neural Information Processing Systems*, 33:4872–4883, 2020.

[33] Tiancheng Jin, Tal Lancewicki, Haipeng Luo, Yishay Mansour, and Aviv Rosenberg. Near-optimal regret for adversarial mdp with delayed bandit feedback. *Advances in Neural Information Processing Systems*, 35:33469–33481, 2022.

[34] Pooria Joulani, Andras Gyorgy, and Csaba Szepesvári. Online learning under delayed feedback. In *International conference on machine learning*, pages 1453–1461. PMLR, 2013.

[35] Michael Kapralov and Rina Panigrahy. Prediction strategies without loss. *Advances in Neural Information Processing Systems*, 24, 2011.

[36] Tomáš Kocák, Gergely Neu, Michal Valko, and Rémi Munos. Efficient learning by implicit exploration in bandit problems with side observations. *Advances in Neural Information Processing Systems*, 27, 2014.

[37] Wouter M Koolen. The pareto regret frontier. *Advances in Neural Information Processing Systems*, 26, 2013.

[38] Tal Lancewicki, Aviv Rosenberg, and Yishay Mansour. Learning adversarial markov decision processes with delayed feedback. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 7281–7289, 2022.

[39] Tor Lattimore. The pareto regret frontier for bandits. *Advances in Neural Information Processing Systems*, 28, 2015.

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

[41] Lihong Li, Wei Chu, John Langford, and Robert E Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 661–670, 2010.

[42] Mingyang Liu, Chengjie Wu, Qihan Liu, Yansen Jing, Jun Yang, Pingzhong Tang, and Chongjie Zhang. Safe opponent-exploitation subgame refinement. *Advances in Neural Information Processing Systems*, 35:27610–27622, 2022.

[43] Tao Liu, Ruida Zhou, Dileep Kalathil, Panganamala Kumar, and Chao Tian. Learning policies with zero or bounded constraint violation for constrained mdps. *Advances in Neural Information Processing Systems*, 34:17183–17193, 2021.

[44] A Rupam Mahmood, Dmytro Korenkevych, Brent J Komer, and James Bergstra. Setting up a reinforcement learning task with a real-world robot. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4635–4640. IEEE, 2018.

[45] Arnab Maiti, Kevin Jamieson, and Lillian J Ratliff. Logarithmic regret for matrix games against an adversary with noisy bandit feedback. *arXiv preprint arXiv:2306.13233*, 2023.

[46] Saeed Masoudian, Julian Zimmert, and Yevgeny Seldin. A best-of-both-worlds algorithm for bandits with delayed feedback. *Advances in Neural Information Processing Systems*, 35: 11752–11762, 2022.

[47] Chris Mesterharm. On-line learning with delayed label feedback. In *International Conference on Algorithmic Learning Theory*, pages 399–413. Springer, 2005.

[48] Adrian Müller, Jon Schneider, Stratis Skoulakis, Luca Viano, and Volkan Cevher. Best of both worlds: Regret minimization versus minimax play. *arXiv preprint arXiv:2502.11673*, 2025.

[49] Gergely Neu. Explore no more: Improved high-probability regret bounds for non-stochastic bandits. *Advances in Neural Information Processing Systems*, 28, 2015.

[50] Francesco Orabona. A modern introduction to online learning. *arXiv preprint arXiv:1912.13213*, 2019.

[51] Francesco Orabona and Dávid Pál. Coin betting and parameter-free online learning. *Advances in Neural Information Processing Systems*, 29, 2016.

[52] Parvaneh Parvin. Just-in-time adaptive anomaly detection and personalized health feedback. 2020.

[53] Herbert Robbins. Some aspects of the sequential design of experiments. 1952.

[54] Amir Sani, Gergely Neu, and Alessandro Lazaric. Exploiting easy data in online optimization. *Advances in Neural Information Processing Systems*, 27, 2014.

[55] Shai Shalev-Shwartz. Online learning and online convex optimization. *Foundations and Trends® in Machine Learning*, 4(2):107–194, 2025.

[56] Tobias Sommer Thune, Nicolò Cesa-Bianchi, and Yevgeny Seldin. Nonstochastic multiarmed bandits with unrestricted delays. *Advances in Neural Information Processing Systems*, 32, 2019.

[57] Dirk van der Hoeven, Ashok Cutkosky, and Haipeng Luo. Comparator-adaptive convex bandits. *Advances in Neural Information Processing Systems*, 33:19795–19804, 2020.

[58] 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.

[59] Manfred K Warmuth. Continuous and discrete time nonlinear gradient descent: relative loss bounds and convergence. In *Electronic Proceedings of Fifth International Symposium on Artificial Intelligence and Mathematics, 1998*, 1998.

[60] James MS Wason and Lorenzo Trippa. A comparison of bayesian adaptive randomization and multi-stage designs for multi-arm clinical trials. *Statistics in medicine*, 33(13):2206–2221, 2014.

[61] Chen-Yu Wei and Haipeng Luo. More adaptive algorithms for adversarial bandits. In *Conference On Learning Theory*, pages 1263–1291. PMLR, 2018.

[62] Yifan Wu, Roshan Shariff, Tor Lattimore, and Csaba Szepesvári. Conservative bandits. In *International Conference on Machine Learning*, pages 1254–1262. PMLR, 2016.

[63] Zhengyuan Zhou, Renyuan Xu, and Jose Blanchet. Learning in generalized linear contextual bandits with stochastic delays. *Advances in Neural Information Processing Systems*, 32, 2019.

[64] Julian Zimmert and Yevgeny Seldin. An optimal algorithm for adversarial bandits with arbitrary delays. In *International Conference on Artificial Intelligence and Statistics*, pages 3285–3294. PMLR, 2020.

Contents

1 Introduction **1**

1.1 How Delay Breaks Safety: Three Roadblocks 2

1.2 Our Contributions 3

2 Preliminaries & Problem Setup **4**

3 The PRUDENT-BANKER Algorithm **5**

3.1 Algorithm Description 5

4 Main Results and Analysis **6**

4.1 Proof Sketch: Safety under Latency 6

4.2 Our Algorithmic Guarantee 6

4.3 Key Lemmas 7

4.4 Lower Bound: The Price of Safety Under Delay 9

5 Experiments **9**

A Related Work **17**

A.1 Exploitation vs. Safety Trade-offs. 17

A.2 Comparator Adaptivity in Full Information. 17

A.3 Hardness of Safe Bandit Learning. 17

A.4 Conservative Bandits and Safe RL. 17

A.5 Adversarial bandits and mirror-descent viewpoints. 17

A.6 Online learning with delayed feedback. 18

B Formal Description of PRUDENT-BANKER **19**

B.1 Further Insights in Prudent-Banker 20

B.2 Further Insights in Banker-OMD 20

B.3 Proof Roadmap 21

C Deferred Proofs **22**

C.1 Proof of Lemma C.1 (Bound on Cumulative Outstanding Delay) 22

C.2 Proof of Lemma C.2 (Bound on Estimated Total Delay) 23

C.3 Proof of Lemma C.3 (Regret Guarantee for Banker-OMD) 24

C.4 Proof of Lemma C.4 (Bound on Missing Feedback Contribution) 27

C.5 Proof of Lemma C.5 (Regret Bound for Normal Phase) 28

C.6 Proof of Lemma C.6 (Negative Regret Credit in Extinction Phase) 31

C.7 Proof of Theorem C.1(Global Regret and Safety Guarantee) 33

D Examples of Different Regularizers **36**

D.1 1/2-Tsallis 36

D.2 negative entropy 37

E Lower Bound **38**

E.1 Proof of Lemma E.1 (Greedy bucket properties) 38

E.2 Proof of Lemma E.2 (Safe lower bound for the batched bandit game) 39

E.3 Proof of Lemma E.3 (Uniform simulation of a delayed bandit algorithm by a batched bandit algorithm) 44

E.4 Proof of Theorem E.1(Delayed safety lower bound) 46

F Experimental details **50**

F.1 Synthetic non-stationary loss process 50

F.2 Safe comparator 50

F.3 Delay models 51

F.4 Evaluation metrics 52

F.5 Algorithms 52

F.6 Interpretation of Figure 2 54

A Related Work

A.1 Exploitation vs. Safety Trade-offs.

Approaches to safe exploitation typically deviate from minimax strategies to target sub-optimal opponents, limiting potential losses to the minimax value [24]. These methods often interpolate between conservative and aggressive strategies using tunable confidence parameters [16, 42]. However, this introduces a fundamental trade-off: such algorithms either risk $\Omega(T)$ regret against strong adversaries [16, 42] or lack theoretical guarantees for exploitation [24]. While recent advances establish logarithmic regret for small-scale games, they rely on restrictive assumptions like pairwise distinct payoffs or full information feedback [45]. It is interesting to understand if we can achieve similar best-of-both-worlds results under adversarial setting with delayed feedback.

A.2 Comparator Adaptivity in Full Information.

Safety, defined as ensuring low regret relative to a pre-specified baseline policy x^c, is a cornerstone of "safe AI." In the full-information setting, it is established that algorithms can achieve constant regret against a specific comparator strategy while maintaining optimal worst-case regret guarantees. Notable examples include [20, 31, 35, 37, 54]. Similarly, parameter-free methods adapt to unknown comparators [14, 50, 51]. These works successfully resolve the "Best-of-Both-Worlds" challenge regarding safety and adversarial robustness. PRUDENT-BANKER extends the Phased Aggression framework [20] to the bandit setting. While algorithms above apply to immediate feedback setting well, our algorithm aims to achieve safety comparator guarantee on delayed, partial feedback.

A.3 Hardness of Safe Bandit Learning.

Under bandit feedback, achieving constant regret against a deterministic comparator generally implies linear worst-case regret against other actions [39]. This impossibility result necessitates relaxing assumptions on the comparator to make $\tilde{\mathcal{O}}(1)$ safety feasible. Recent work by Muller et al. [48] demonstrates that if the safe baseline is required to be *full-support*, one can indeed achieve constant safety regret alongside near-optimal minimax regret using a phased aggression mechanism. We adopt this full-support assumption to render the safety goal achievable. However, the standard phased aggression mechanism in [48] relies on immediate feedback. PRUDENT-BANKER extends the framework of [48] to delayed feedback setting.

A.4 Conservative Bandits and Safe RL.

Our framework's definition of safety is distinct from the paradigms often studied in conservative bandits [62] and conservative reinforcement learning (RL) [25]. In those contexts, algorithms are designed to satisfy a constraint—often defined as securing at least a $(1-\alpha)$-fraction of a baseline's return. While this ensures performance relative to a baseline, it allows for linear regret $\mathcal{O}(\alpha T)$ relative to the optimal policy in the worst case. Furthermore, in constrained settings such as Constrained MDPs [5, 19], achieving constant regret against a safe baseline is frequently unobtainable under adversarial environemnt [43]. In contrast, PRUDENT-BANKER aims for constant $\tilde{\mathcal{O}}(1)$ regret relative to the baseline while simultaneously minimizing regret against the best strategy in hindsight, even in the presence of arbitrary delays.

A.5 Adversarial bandits and mirror-descent viewpoints.

Many techniques have been applied to adversarial multi-armed bandits (MAB) setting. For example, exponential-weights methods EXP3 [4], Online Mirror Descent (OMD) and Follow-the-Regularized-Leader (FTRL) frameworks [26, 40, 59]. Some new developments employ different regularizer, including Tsallis-entropy and related OMD variants [61, 64]. Our work follows the OMD paradigm but targets a more complicated regime where (i) feedback is delayed, and (ii)regret is constant relative to a designated safe comparator.

A.6 Online learning with delayed feedback.

Online learning with delayed feedback has been studied across several settings, including multi-armed bandits (MABs), linear bandits, and more general sequential decision-making problems. For MABs, early work considered stochastic settings with i.i.d. delays [34], while adversarial formulations established minimax lower bounds under uniform delays [11]; subsequent results achieved near-optimal regret depending on both the horizon T and total delay D via OMD-based methods with moving negative-entropy regularization [8, 56, 64]. Among which the `Banker-OMD` framework [30] provides a unified delay-handling mechanism that decouples delays from algorithm design and extends delayed bandit guarantees to scale-free loss settings. Delayed feedback has also been investigated in linear bandits, including stochastic models with i.i.d. or partially observable delays and adversarial models with uniform delays [32, 58, 63]. Beyond bandits, delays have been studied in stochastic and adversarial Markov Decision Processes, general online optimization with full-information or bandit feedback, and wireless network optimization [2, 15, 17, 18, 28, 29, 33, 38, 47].

B Formal Description of PRUDENT-BANKER

Algorithm 2 PRUDENT-BANKER

Input: number of arms A, horizon T, comparator x^c, comparator margin $\delta \in (0, 1/A]$, constants (C_1, C_2), mirror map (Ψ, x_0), $x_0 = \mathbf{1}/A$, step-size subroutine S

Initialize: stage $s \leftarrow 1$, $\widehat{D}_s \leftarrow 1$, $k \leftarrow 1$, start $\leftarrow 1$, $\hat{x}_1 \leftarrow x_0$, $B_0 \leftarrow 0$, $\mathfrak{D}_0^{(1)} \leftarrow 0$

Define

$$\widehat{R}(\widehat{D}) = \sqrt{C_1 C_2}\left(3\sqrt{T} + 7\sqrt{2\widehat{D}\ln(\widehat{D}+1)}\right), \qquad \widehat{\xi}(\widehat{D}) = \frac{\sqrt{8\widehat{D}+1}-1}{\delta}$$

$\widehat{R}_s \leftarrow 2^{\lceil \log_2 \widehat{R}(\widehat{D}_s)\rceil}$, $\widehat{\xi}_s \leftarrow \widehat{\xi}(\widehat{D}_s)$, $\alpha \leftarrow \min\{1/\widehat{R}_s, 1\}$

for $t = 1$ to T **do**

  $\mathfrak{D}_t^{(s)} \leftarrow \mathfrak{D}_{t-1}^{(s)} + \mathfrak{d}_t$

  if $\mathfrak{D}_t^{(s)} > \widehat{D}_s$ **then**

    $\widehat{D}_{s+1} \leftarrow 2^{\lceil \log_2 \mathfrak{D}_t^{(s)}\rceil}$, $s \leftarrow s+1$, $k \leftarrow 1$

    start $\leftarrow t$, $\widehat{R}_s \leftarrow 2^{\lceil \log_2 \widehat{R}(\widehat{D}_s)\rceil}$, $\widehat{\xi}_s \leftarrow \widehat{\xi}(\widehat{D}_s)$

    $\alpha \leftarrow \min\{1/\widehat{R}_s, 1\}$, $\hat{x}_t \leftarrow x_0$

    $\mathfrak{D}_t^{(s)} \leftarrow 0$, $B_{t-1} \leftarrow 0$

  end if

$$\sigma_t \leftarrow S\Big(t - \text{start} + 1, \{(A_u, \text{status}_u, \hat{\ell}_u, \sigma_u, v_u)\}_{u<t}\Big)$$

  Banker initialization: $v_t \leftarrow \sigma_t$, $b_t \leftarrow \sigma_t$, $\sigma_{t,u} \leftarrow 0$ for start $\le u < t$

  for all $u = \text{start}, \dots, t-1$ with $\text{status}_u = \text{ARRIVED}$ **do**

    $\sigma_{t,u} \leftarrow \min\{v_u, b_t\}$

    $v_u \leftarrow v_u - \sigma_{t,u}$, $b_t \leftarrow b_t - \sigma_{t,u}$

  end for

  $B_t \leftarrow B_{t-1} + b_t$

$$\hat{x}_t \leftarrow \nabla\bar{\Psi}^*\left(\frac{1}{\sigma_t}\sum_{\text{start}\le u\le t-1}\sigma_{t,u}\nabla\Psi(z_u) + \frac{b_t}{\sigma_t}\nabla\Psi(x_0)\right)$$

  $x_t \leftarrow \alpha\hat{x}_t + (1-\alpha)x^c$

  Sample $A_t \sim x_t$, incur loss $\ell_t(A_t)$; $\text{status}_t \leftarrow \text{MISSING}$

  Upon receiving each new feedback $(u, \ell_u(A_u))$:

    $\text{status}_u \leftarrow \text{ARRIVED}$

    $\hat{\ell}_u \leftarrow \dfrac{\ell_u(A_u)}{x_u(A_u)} e_{A_u}$

    $z_u \leftarrow \nabla\bar{\Psi}^*\left(\nabla\Psi(\hat{x}_u) - \dfrac{1}{\sigma_u}\hat{\ell}_u\right)$

  $g_t \leftarrow \sum_{u=\text{start}}^{t} \mathbf{1}\{\text{status}_u = \text{ARRIVED}\}\hat{\ell}_u$

  if $\max_{x\in\Delta_A}\langle g_t, x^c - x\rangle > 2\widehat{R}_s + \widehat{\xi}_s$ **and** $\alpha < 1$ **and** $\mathfrak{D}_t^{(s)} + \mathfrak{d}_{t+1} \le \widehat{D}_s$ **then**

    $k \leftarrow k+1$, start $\leftarrow t+1$, $\hat{x}_{t+1} \leftarrow \mathbf{1}/A$, $B_t \leftarrow 0$

    $\alpha \leftarrow \min\{2^{k-1}/\widehat{R}_s, 1\}$

  end if

end for

B.1 Further Insights in Prudent-Banker

From technical perspectives, our contributions are threefold:

1. Eliminating the safety certification gap. We derive a *delay-calibrated restart condition* that accounts for the information deficit induced by delayed feedback. In standard bandits, verifying that the baseline/comparator is suboptimal reduces to a relatively direct hypothesis test. We show that under delays, this test must be relaxed by an explicit slack term

$$\xi_s \;=\; \Theta\!\left(\frac{\sqrt{D_s}}{\delta}\right),$$

where D_s denotes the cumulative delay mass within stage s. This slack acts as a buffer against *phantom suboptimality* arising from unobserved losses, preventing spurious restarts and thereby preserving an $O(1)$ safety guarantee per stage.

2. Cost-free adaptation to unknown delays. A central challenge is adapting to an unknown total delay D without compromising safety. Standard "doubling trick" approaches [3, 4] typically treat each restart as a fresh instance. While such restarts incur only mild overhead for conventional regret, they can be catastrophic for safety: naively resetting the mechanism forfeits the *negative regret credit* accumulated against x^c, potentially forcing the algorithm to pay an additive penalty proportional to the instance scale (e.g., $\sqrt{T}$ or $\sqrt{D}$) repeatedly.

Indeed, the difficulty of learning under an *unknown* delay budget is well-documented in delayed bandits [8] and further refined in subsequent work [9], where the typical adaptation cost scales on the order of $\sqrt{TK + D\log K}$. Furthermore, as analyzed by Besson and Kaufmann [7], simple "oblivious" geometric restarts generally fail to preserve instance-dependent guarantees because periodic resets destroy the accumulated statistical evidence required for fast rates. In contrast, our coupled schedule restarts only when warranted by the observed delay structure and safety diagnostics, insuring only a negligible logarithmic overhead. Consequently, delay adaptation becomes essentially "free" at the level of the safety bounds.

3. Generalizing `Banker-OMD` to Safe Learning. Finally, we answer the challenge of extending safety guarantees beyond standard immediate-feedback environments. While Müller et al. [48] established the foundations of the (Safe) online linear minimization framework, the feasibility of such guarantees under partial monitoring remained a question.
We bridge this gap by providing a unified analysis that integrates `Banker-OMD` with safety constraints. We demonstrate that by decomposing the regret analysis into "arrived" and "pending" components, It is possible the potential-function credits to be used for delay tolerance can be *dually leveraged* to certify safety, effectively broadening the applicability of "best-of-both-worlds" mechanisms to the asynchronous reality of deployed systems.

B.2 Further Insights in Banker-OMD

As we discussed in the main draft, *Prudent-Banker* combines two mechanisms—one to *survive delayed bandit feedback* and one to *stay safe relative to a baseline comparator*. Our delayed bandit feedback mechanism follows banker OMD idea. The intuition is to carefully account of how much past information is allowed to influence each decision.

At the beginning of round t, the algorithm selects a target action scale σ_t, which specifies the total amount of update weight to be used when forming the next decision. Due to delayed feedback, only a subset of past rounds have revealed their losses by time t, and therefore only those rounds can contribute update information. In particular, when feedback from round $s < t$ arrives, we can compute the corresponding mirror-descent update:

$$z_s \;=\; \nabla\bar{\Psi}^*\!\left(\nabla\Psi(\hat{x}_s) - \frac{1}{\sigma_s}\hat{\ell}_s\right).$$

To construct the decision, the algorithm draws update weight from previously arrived rounds in a greedy manner. Each past round $s < t$ maintains a remaining budget v_s, representing how much of

its feedback can still influence future decisions. For every arrived round s, the algorithm allocates an amount

$$\sigma_{t,s} = \min\{v_s,\ b_t\},$$

where b_t denotes the remaining portion of σ_t that has not yet been covered. The allocated amount $\sigma_{t,s}$ is deducted both from v_s and from b_t, and this process continues until either the full budget σ_t is reached or no further arrived feedback is available.

If the available feedback is insufficient to cover σ_t, the remaining portion b_t is filled using a fixed default distribution x_0. The resulting decision is defined as a mirror-map average

$$\tilde{x}_t = \nabla\bar{\Psi}^*\left(\frac{1}{\sigma_t}\sum_{s<t}\sigma_{t,s}\nabla\Psi(z_s)\ +\ \frac{b_t}{\sigma_t}\nabla\Psi(x_0)\right),$$

which combines all available mirror-descent updates with the default one, in proportions determined exactly by the amount of missing feedback. This ensures that each decision relies only on information that has actually arrived, while reducing to a default when feedback is delayed.

This construction guarantees that each unit of learning-rate budget is used at most once, restores a telescoping structure in the regret analysis, and ensures that the additional regret incurred due to delays depends only on the cumulative missing budget, which can be bounded in terms of the total delay.

B.3 Proof Roadmap

Stage Partitioning via Doubling Trick. Since the total delay D is unknown, the algorithm proceeds in stages $s = 1, 2, \ldots, S$. We maintain a delay estimator $\widehat{D}_s$, initialized at $\widehat{D}_1 = 1$. At any time step t, if the realized cumulative delay exceeds the current estimator, we update $\widehat{D}_{s+1} \geq 2\widehat{D}_s$ and initiate a new stage. This doubling scheme ensures that the total number of stages is bounded by $S \leq \lceil \log_2(1+D) \rceil + 1$ (Lemma 4.2), allowing the algorithm to adapt to the delay scale without prior knowledge.

Phased Aggression Framework. Within each stage s, the time horizon is further partitioned into a sequence of phases $k = 1, 2, \ldots, K(s)$. In each phase, the algorithm plays a mixed strategy $x_t = \alpha\hat{x}_t + (1-\alpha)x^c$, where $\hat{x}_t$ is the Banker-OMD strategy and x^c is a safe comparator. The aggression parameter α determines the influence of the comparator and is scheduled as:

$$\alpha = \min\{\frac{2^{k-1}}{\widehat{R}_s}, 1\} \tag{1}$$

where $\widehat{R}_s$ denotes the stage-specific regret upper bound. A phase transition $k \to k+1$ is triggered whenever the cumulative loss of the safe comparator exceeds a threshold, signifying "poor" performance. Upon transitioning, we increase the weight on the OMD learner (increase α), effectively "tuning out" the poorly performing comparator.

Regret Analysis and Complexity. This nested architecture facilitates a dual-layered regret guarantee:

- **Comparator Regret:** By maintaining a phased mixture, the algorithm ensures that the regret relative to the safe comparator remains $O(1)$ within each stage (Lemma 4.6, Lemma 4.5, Lemma 4.4).
- **Worst-case Regret:** We prove the regret upper bound for Banker-OMD using Lemma 4.1. Then decompose the regret into Banker-OMD regret (Lemma 4.3) and regret of comparator relative to strategy x. Combining Lemma 4.2, each phase incurs a worst-case regret bounded by $4\widehat{R}_s + \widehat{\xi}_s + 2\sqrt{2\widehat{D}_s + \frac{1}{4}} + 1$ (Lemma 4.5). Since α grows exponentially, the number of phases per stage is bounded by $K(s) = O(\log \widehat{R}_s)$.
- **Total Regret:** Summing the regret over $K(s)$ phases and S stages yields a global regret bound that accounts for both the estimation of D and the adaptive weighting of the comparator.

C Deferred Proofs

C.1 Proof of Lemma C.1 (Bound on Cumulative Outstanding Delay)

Intuition. Following lemma relates two different measurements of delay. The quantity $\mathfrak{d}_t^{k(s)}$ *counts how many feedback items within phase* $k(s)$ *are still missing at time* t*, and* $\mathfrak{D}_t^{k(s)} = \sum_{u=\text{start}}^{t} \mathfrak{d}_u^{k(s)}$ *is the* cumulative delayed feedback items *up to* t*. The lemma shows* $\mathfrak{D}_t^{k(s)}$ *is monotone in* t *and that* $\mathfrak{D}_{\text{end}_{k(s)}}^{k(s)}$ *is bounded by the total delay lengths* $D_{\text{end}_{k(s)}}^{k(s)}$*. Intuitively, each round* r *with delay* d_r *can contribute at most* d_r *time-steps of "being missing," so the total missing-mass over the horizon cannot exceed the sum of delay lengths. Since the regret upper bound of Algorithm 1 relies on* σ_t *constructed by* $\mathfrak{D}_t^{k(s)}$*, we need this lemma to ensure our regret bound is parametrized by* $D_{\text{end}_{k(s)}}^{k(s)}$*. The proof works by double-counting:* $\mathfrak{D}_{\text{end}_{k(s)}}^{k(s)}$ *counts the number of pairs* (r, u) *such that the feedback of* r *is still missing at time* u*, which equals* $\sum_{r=\text{start}}^{end_{k(s)}} \min\{d_r, \text{end}_{k(s)} - r\}$ *and is therefore at most* $\sum_{r=\text{start}_{k(s)}}^{end_{k(s)}} d_r = D_{\text{end}_{k(s)}}^{k(s)}$*.*

Lemma C.1: Bound on Cumulative Outstanding Delay. The cumulative number of delayed feedback items up to time t, is at most the cumulative delayed feedback items over phase $k(s)$ inside stage s, which in turn is bounded by the sum of the delay lengths. Formally,

$$\mathfrak{D}_t^{k(s)} \le \mathfrak{D}_{\text{end}_{k(s)}}^{k(s)} \le D_{\text{end}_{k(s)}}^{k(s)}.$$

Proof. Since $\mathfrak{D}_t^{k(s)} = \sum_{u=\text{start}_{k(s)}}^{t} \mathfrak{d}_u^{k(s)}$ with $\mathfrak{d}_u^{k(s)} = \sum_{r=\text{start}_{k(s)}}^{u-1} \mathbf{1}\{u \le r + d_r\} \ge 0$, it follows immediately that $\mathfrak{D}_t^{k(s)}$ is non-decreasing in t. Thus, the first inequality $\mathfrak{D}_t^{k(s)} \le \mathfrak{D}_{\text{end}_{k(s)}}^{k(s)}$ holds for all $t \le \text{end}_{k(s)}$.

For the second inequality, we first show that $\mathfrak{D}_{\text{end}_{k(s)}}^{k(s)} = \sum_{u=\text{start}_{k(s)}}^{\text{end}_{k(s)}} \min\{d_u, \text{end}_{k(s)} - u\}$. By definition of delay complexity,

$$\mathfrak{D}_{\text{end}_{k(s)}}^{k(s)} = \sum_{u=\text{start}_{k(s)}}^{\text{end}_{k(s)}} \mathfrak{d}_u^{k(s)} = \sum_{u=\text{start}_{k(s)}}^{\text{end}_{k(s)}} \sum_{r=\text{start}_{k(s)}}^{u-1} \mathbf{1}\{u \le r + d_r\}.$$

We swap the order of summation (counting pairs of (r, u) such that $r < u \le r + d_r$): $\mathfrak{D}_{\text{end}_{k(s)}}^{k(s)} = \sum_{r=\text{start}_{k(s)}}^{\text{end}_{k(s)}} \sum_{u=r+1}^{\text{end}_{k(s)}} \mathbf{1}\{u \le r + d_r\}$. Fix s. We are counting the number of integers u such that $u \in \{r+1, \dots, \text{end}_{k(s)}\}$ and $u \le r + d_r$. Thus, u ranges from $r+1$ up to $\min\{\text{end}_{k(s)}, r + d_r\}$. The total number of such integers is:

$$\begin{aligned}\max\left(0, \min\left\{\text{end}_{k(s)}, r + d_r\right\} - (r+1) + 1\right) &= \max\left(0, \min\left\{\text{end}_{k(s)}, r + d_r\right\} - r\right) \\ &= \min\left\{d_r, \text{end}_{k(s)} - r\right\},\end{aligned}$$

where the last step follows because delays are non-negative ($d_r \ge 0$). Plugging this back gives:

$$\mathfrak{D}_{\text{end}_{k(s)}}^{k(s)} = \sum_{r=\text{start}_{k(s)}}^{\text{end}_{k(s)}} \min\left\{d_r, \text{end}_{k(s)} - r\right\}.$$

Finally, we bound this sum by $D_{\text{end}_{k(s)}}^{k(s)}$:

1. If all feedback arrives by end of phase $k(s)$ (i.e., $r + d_r \le \text{end}_{k(s)}$ for all r), then $\min\{d_r, \text{end}_{k(s)} - r\} = d_r$, and $\mathfrak{D}_{\text{end}_{k(s)}}^{k(s)} = \sum d_r = D_{\text{end}_{k(s)}}^{k(s)}$.
2. If some feedback arrives after $\text{end}_{k(s)}$ (i.e., $r + d_r > \text{end}_{k(s)}$ for some r), then $\min\{d_r, \text{end}_{k(s)} - r\} < d_r$ for those r, and thus $\mathfrak{D}_{\text{end}_{k(s)}}^{k(s)} < D_{\text{end}_{k(s)}}^{k(s)}$.

In all cases, $\mathfrak{D}_{\text{end}_{k(s)}}^{k(s)} \le D_{\text{end}_{k(s)}}^{(s)}$. □

C.2 Proof of Lemma C.2 (Bound on Estimated Total Delay)

Intuition. This lemma establishes that the updated budget $\widehat{D}_{s+1}$ provides a tight upper bound on the realized total delay. Specifically, we show that $\widehat{D}_{s+1} < 2\mathfrak{D}_t^{(s)}$, ensuring that the estimated capacity does not overestimate the actual total delay by more than a factor of two. The analysis proceeds in two steps: (i) we first demonstrate the monotonicity of the estimator ($\widehat{D}_{s+1} \geq \widehat{D}_s$); (ii) we then derive the upper bound using the doubling update rule defined in Algorithm 1. This inequality is critical for subsequent lemmas, as it allows us to bound the number of phases and the total delay accumulated within them by relating the budget back to the realized total delay.

Lemma C.2: Bound on Estimated Total Delay. At a hard restart in round t, the total-delay capacity satisfies:

$$\widehat{D}_s \leq \widehat{D}_{s+1} < 2\mathfrak{D}_t^{(s)}$$

Proof.

- **Monotonicity ($\widehat{D}_s \leq \widehat{D}_{s+1}$):** The inequality holds by construction. In Algorithm 1, the budget is updated via $\widehat{D}_{s+1} \leftarrow \widehat{D}_s \cdot 2^m$ for some integer $m \geq 1$ when a reset occurs, or remains constant otherwise. Thus, $\widehat{D}_{s+1} \geq \widehat{D}_s$.

- **Tightness ($\widehat{D}_{s+1} < 2\mathfrak{D}_t^{(s)}$):** A stage reset is triggered only when $\mathfrak{D}_t^{(s)} > \widehat{D}_s$. The algorithm finds the smallest integer $m \in \mathbb{N}_{\geq 1}$ such that the new budget covers the observed delay count:

  $$2^m \widehat{D}_s \geq \mathfrak{D}_t^{(s)}.$$

  The update rule sets $\widehat{D}_{s+1} = 2^m \widehat{D}_s$. Due to the minimality of m, the exponent $m-1$ would fail to cover the observed delay count, implying:

  $$2^{m-1} \widehat{D}_s < \mathfrak{D}_t^{(s)} \leq 2^m \widehat{D}_s = \widehat{D}_{s+1}. \tag{2}$$

  To obtain the upper bound, we rearrange the left-hand side of (2):

  $$\frac{1}{2}(2^m \widehat{D}_s) < \mathfrak{D}_t^{(s)} \implies \frac{1}{2}\widehat{D}_{s+1} < \mathfrak{D}_t^{(s)} \implies \widehat{D}_{s+1} < 2\mathfrak{D}_t^{(s)}.$$

  Combining the lower and upper bounds, we conclude that $\mathfrak{D}_t^{(s)} \leq \widehat{D}_{s+1} < 2\mathfrak{D}_t^{(s)}$.

Before the reset, $\mathfrak{D}_t^{(s)} \leq \mathfrak{D}_T \leq D$. Hence $\widehat{D}_s \leq \max\{1, 2D\}$ and $S \leq 1 + \lceil \log_2(1+D) \rceil$. □

C.3 Proof of Lemma C.3 (Regret Guarantee for Banker-OMD)

Intuition. This lemma establishes the regret guarantee for the Banker-OMD strategy running with action scales σ_t, achieving a bound of roughly $\tilde{O}(\sqrt{T}+\sqrt{D})$. We need this because Algorithm 1 is built by mixing the base Banker-OMD iterate $\hat{x}_t$ with the safe comparator x^c. The proof follows the standard Banker-OMD decomposition: the regret is bounded by the sum of total borrowing $B_{end_{k(s)}}$ *(before reset) and total* immediate cost $\sum_t \sigma_t^{-1}$. *To control $B_{end_{k(s)}}$, we consider the last round T_0 when borrowing occurred; the amount borrowed is the current bill plus bills from rounds with outstanding feedback. To control $\sum_t \sigma_t^{-1}$, we use the specific structure of σ_t to separate a $1/\sqrt{t}$ term and a delay-dependent term, applying the inequality $\sum_{t=1}^{T} \frac{x_t}{\sqrt{\sum_{s\le t} x_s}} \le 2\sqrt{\sum_t x_t}$ along with the bound $\mathfrak{D}_{end_{k(s)}}^{k(s)} \le D_{\mathrm{end}_{k(s)}}^{k(s)}$ from Lemma C.1.*

Lemma C.3: Regret Guarantee for Banker-OMD. Let $\tilde{t} = t - \mathrm{start}_{k(s)} + 1$. Set $\ln(1+z)/z = 1$ at $z = 0$.

Setting $\sigma_t = \left(\frac{1}{\sqrt{\tilde{t}}} + \mathfrak{d}_t^{k(s)} \sqrt{\frac{\ln\left(\mathfrak{D}_t^{k(s)}+1\right)}{\mathfrak{D}_t^{k(s)}}} \right)^{-1}$, we have:

$$\max_{x\in\Delta_A} \mathbb{E}\left[\sum_{t=\mathrm{start}_{k(s)}}^{\mathrm{end}_{k(s)}} \langle \hat{\ell}_t, \hat{x}_t - x\rangle\right] \le \mathbb{E}\left[(C_1+2C_2)\sqrt{\mathrm{end}_{k(s)}} + (3C_1+2C_2)\sqrt{D_{\mathrm{end}_{k(s)}}^{k(s)} \ln(D_{\mathrm{end}_{k(s)}}^{k(s)}+1)}\right].$$

Setting $\sigma_t = \sqrt{\frac{C_2}{C_1}} \left(\frac{1}{\sqrt{\tilde{t}}} + \mathfrak{d}_t^{k(s)} \sqrt{\frac{\ln\left(\mathfrak{D}_t^{k(s)}+1\right)}{\mathfrak{D}_t^{k(s)}}} \right)^{-1}$, we have:

$$\max_{x\in\Delta_A} \mathbb{E}\left[\sum_{t=\mathrm{start}_{k(s)}}^{\mathrm{end}_{k(s)}} \langle \hat{\ell}_t, \hat{x}_t - x\rangle\right] \le \sqrt{C_1 C_2}\,\mathbb{E}\left[3\sqrt{\mathrm{end}_{k(s)}} + 7\sqrt{2D_{\mathrm{end}_{k(s)}}^{k(s)} \ln(D_{\mathrm{end}_{k(s)}}^{k(s)}+1)}\right].$$

Proof. Since $\hat{\ell}_t$ is an unbiased estimator, $\mathbb{E}\left[\sum_{t=\mathrm{start}_{k(s)}}^{\mathrm{end}_{k(s)}} \langle \hat{\ell}_t, \hat{x}_t - x\rangle\right] = \mathbb{E}\left[\sum_{t=\mathrm{start}_{k(s)}}^{\mathrm{end}_{k(s)}} \langle \ell_t, \hat{x}_t - x\rangle\right]$. Taking the maximum over $x \in \Delta_A$:

$$\mathcal{R}_{k(s)} := \max_{x\in\Delta_A} \mathbb{E}\left[\sum_{t=\mathrm{start}_{k(s)}}^{\mathrm{end}_{k(s)}} \langle \hat{\ell}_t, \hat{x}_t - x\rangle\right] = \max_{x\in\Delta_A} \mathbb{E}\left[\sum_{t=\mathrm{start}_{k(s)}}^{\mathrm{end}_{k(s)}} \langle \ell_t, \hat{x}_t - x\rangle\right].$$

Since (Ψ, x_0) is (C_1, C_2)-regular, we can apply the uniform bound in Theorem 4.2 and Corollary 4.3 of Huang et al. [30]:

$$\begin{aligned} \mathcal{R}_{k(s)} &\le \mathbb{E}\left[\max_{x\in\Delta_A} \sum_{t=\mathrm{start}_{k(s)}}^{\mathrm{end}_{k(s)}} \langle \hat{\ell}_t, \hat{x}_t - x\rangle\right] \\ &\le C_1 \cdot \mathbb{E}\left[B_{k(s)}\right] + C_2 \cdot \mathbb{E}\left[\sum_{t=\mathrm{start}_{k(s)}}^{\mathrm{end}_{k(s)}} \sigma_t^{-1}\right]. \end{aligned} \tag{3}$$

Apply single-draw stability at each fixed history, then use iterated expectation for the phase-entry weighted estimate.

Step 1: Bounding the Borrowing Term $B_{k(s)}$. Consider the last round T_0 when B_t increases within phase $k(s)$. Define the phase-local index $\tilde{T}_0 = T_0 - \mathrm{start}_{k(s)} + 1$. Assume there are $m = \mathfrak{d}_{T_0}^{k(s)}$ feedback items missing at the beginning of T_0, corresponding to rounds $t_1 < \cdots < t_m$ within phase $k(s)$. Define phase-local indices $\tilde{t}_i = t_i - \mathrm{start}_{k(s)} + 1$. Using the definition $\sigma_t^{-1} =$

$\frac{1}{\sqrt{\tilde{t}}} + \mathfrak{d}_t^{k(s)} \sqrt{\frac{\ln\left(\mathfrak{D}_t^{k(s)}+1\right)}{\mathfrak{D}_t^{k(s)}}}$, we have $\sigma_t^{-1} \geq \frac{1}{\sqrt{\tilde{t}}}$ and $\sigma_t^{-1} \geq \mathfrak{d}_t^{k(s)} \sqrt{\frac{\ln\left(\mathfrak{D}_t^{k(s)}+1\right)}{\mathfrak{D}_t^{k(s)}}}$. By Lemma 4.5 of Huang et al. [30], we have:

$$B_{\text{end}_{k(s)}} = B_{T_0} = \sigma_{T_0} + \sum_{t=\text{start}_{k(s)}}^{T_0-1} \mathbf{1}\left\{t + d_t \geq T_0\right\} \sigma_t.$$

If $m = 0$, $B_{\text{end}_{k(s)}} = \sigma_{T_0} \leq \sqrt{\widetilde{\text{end}}_{k(s)}}$; below assume $m \geq 1$. Since there are $m = \mathfrak{d}_{T_0}^{k(s)}$ feedback items pending, the sum covers exactly the indices t_i. Thus:

$$B_{\text{end}_{k(s)}} = \sigma_{T_0} + \sum_{i=1}^{m} \sigma_{t_i}.$$

Using the bound $\sigma_t \leq \frac{1}{\mathfrak{d}_t^{k(s)}} \sqrt{\frac{\mathfrak{D}_t^{k(s)}}{\ln(\mathfrak{D}_t^{k(s)}+1)}}$ and noting that $x \mapsto \sqrt{\frac{x}{\ln(x+1)}}$ is increasing:

$$\begin{aligned} B_{\text{end}_{k(s)}} &\leq \frac{1}{\mathfrak{d}_{T_0}^{k(s)}} \sqrt{\frac{\mathfrak{D}_{T_0}^{k(s)}}{\ln\left(\mathfrak{D}_{T_0}^{k(s)}+1\right)}} + \sigma_{t_1} + \sum_{i=2}^{m} \frac{1}{\mathfrak{d}_{t_i}^{k(s)}} \sqrt{\frac{\mathfrak{D}_{t_i}^{k(s)}}{\ln\left(\mathfrak{D}_{t_i}^{k(s)}+1\right)}} \\ &\leq \frac{1}{m} \sqrt{\frac{D_{\text{end}_{k(s)}}^{k(s)}}{\ln\left(D_{\text{end}_{k(s)}}^{k(s)}+1\right)}} + \sqrt{\tilde{t}_1} + \sqrt{\frac{D_{\text{end}_{k(s)}}^{k(s)}}{\ln\left(D_{\text{end}_{k(s)}}^{k(s)}+1\right)}} \sum_{i=2}^{m} \frac{1}{\mathfrak{d}_{t_i}^{k(s)}}. \end{aligned}$$

Since $\tilde{t}_1 \leq \tilde{\text{end}}_{k(s)}$ and at time t_i there are at least $i-1$ other missing feedback items (from $t_1, \ldots, t_{i-1}$), we have $\mathfrak{d}_{t_i}^{k(s)} \geq i - 1$. Thus:

$$\begin{aligned} B_{\text{end}_{k(s)}} &\leq \sqrt{\tilde{\text{end}}_{k(s)}} + \sqrt{\frac{D_{\text{end}_{k(s)}}^{k(s)}}{\ln(D_{\text{end}_{k(s)}}^{k(s)}+1)}} \left(\frac{1}{m} + \sum_{i=2}^{m} \frac{1}{i-1}\right) \\ &= \sqrt{\tilde{\text{end}}_{k(s)}} + \sqrt{\frac{D_{\text{end}_{k(s)}}^{k(s)}}{\ln(D_{\text{end}_{k(s)}}^{k(s)}+1)}} \left(\frac{1}{m} + \sum_{j=1}^{m-1} \frac{1}{j}\right). \end{aligned}$$

For $m \geq 2$, using the harmonic series bound $\sum_{j=1}^{k} \frac{1}{j} \leq \ln k + \gamma + \frac{1}{2k}$ (where $\gamma \approx 0.577$):

$$\begin{aligned} B_{\text{end}_{k(s)}} &\leq \sqrt{\tilde{\text{end}}_{k(s)}} + \sqrt{\frac{D_{\text{end}_{k(s)}}^{k(s)}}{\ln(D_{\text{end}_{k(s)}}^{k(s)}+1)}} \left(\frac{1}{m} + \ln(m-1) + \gamma + \frac{1}{2(m-1)}\right) \\ &\leq \sqrt{\tilde{\text{end}}_{k(s)}} + \sqrt{\frac{D_{\text{end}_{k(s)}}^{k(s)}}{\ln(D_{\text{end}_{k(s)}}^{k(s)}+1)}} (\ln m + 2). \end{aligned}$$

Using the property that the maximum concurrency m satisfies $m(m+1)/2 \leq \mathfrak{D}_{\text{end}_{k(s)}}^{k(s)} \leq D_{\text{end}_{k(s)}}^{k(s)}$, hence $m \leq \sqrt{2D_{\text{end}_{k(s)}}^{k(s)}}$, we substitute back:

$$B_{\text{end}_{k(s)}} \leq \sqrt{\tilde{\text{end}}_{k(s)}} + \frac{1}{2} \sqrt{\frac{D_{\text{end}_{k(s)}}^{k(s)}}{\ln(D_{\text{end}_{k(s)}}^{k(s)}+1)}} \ln\left(2D_{\text{end}_{k(s)}}^{k(s)} + \frac{1}{4}\right) + 2\sqrt{\frac{D_{\text{end}_{k(s)}}^{k(s)}}{\ln(D_{\text{end}_{k(s)}}^{k(s)}+1)}}. \quad (4)$$

For $m = 1$, use the empty sum; for $m \geq 2$, $D_{\text{end}_{k(s)}}^{k(s)} \geq 3$. Both give a delay term at most $3\sqrt{D_{\text{end}_{k(s)}}^{k(s)} \ln(D_{\text{end}_{k(s)}}^{k(s)}+1)}$.

Step 2: Bounding the Immediate Cost $\sum \sigma_t^{-1}$.

$$\begin{aligned}
\sum_{t=\text{start}_{k(s)}}^{\text{end}_{k(s)}} \sigma_t^{-1} &= \sum_{t=\text{start}_{k(s)}}^{\text{end}_{k(s)}} \left(\frac{1}{\sqrt{\tilde{t}}} + \mathfrak{d}_t^{k(s)} \sqrt{\frac{\ln\left(\mathfrak{D}_t^{k(s)} + 1\right)}{\mathfrak{D}_t^{k(s)}}} \right) \\
&\leq \sum_{t=\text{start}_{k(s)}}^{\text{end}_{k(s)}} \frac{1}{\sqrt{\tilde{t}}} + \sqrt{\ln(D_{\text{end}_{k(s)}}^{k(s)} + 1)} \sum_{t=\text{start}_{k(s)}}^{\text{end}_{k(s)}} \frac{\mathfrak{d}_t^{k(s)}}{\sqrt{\mathfrak{D}_t^{k(s)}}} \\
&\leq 2\sqrt{\tilde{\text{end}}_{k(s)}} + \sqrt{\ln(D_{\text{end}_{k(s)}}^{k(s)} + 1)} \cdot 2\sqrt{\mathfrak{D}_{\text{end}_{k(s)}}^{k(s)}} \\
&\leq 2\sqrt{\tilde{\text{end}}_{k(s)}} + 2\sqrt{D_{\text{end}_{k(s)}}^{k(s)} \ln(D_{\text{end}_{k(s)}}^{k(s)} + 1)}. \qquad (5)
\end{aligned}$$

Here we used $\sum_{t=1}^{T} \frac{x_t}{\sqrt{\sum_{s=1}^{t} x_s}} \leq 2\sqrt{\sum_{t=1}^{T} x_t}$ and $\mathfrak{D}_{\text{end}_{k(s)}}^{k(s)} \leq D_{\text{end}_{k(s)}}^{k(s)}$(Lemma C.1).

Step 3: Combining Terms. Plugging (4) and (5) into (3):

$$\mathcal{R}_{k(s)} \leq \mathbb{E}\big[(C_1 + 2C_2)\sqrt{\tilde{\text{end}}_{k(s)}} + (3C_1 + 2C_2)\sqrt{D_{\text{end}_{k(s)}}^{k(s)} \ln(D_{\text{end}_{k(s)}}^{k(s)} + 1)}\big].$$

Finally, for the tuned learning rate $\sigma_t \leftarrow \sqrt{\frac{C_2}{C_1}}\sigma_t$, the terms scale as $C_1\sqrt{\frac{C_2}{C_1}}B_{\text{end}_{k(s)}}$ and $C_2\sqrt{\frac{C_1}{C_2}}\sum \sigma_t^{-1}$, effectively multiplying the final bound by $\sqrt{C_1 C_2}$ (and adjusting constants). Specifically:

$$\begin{aligned}
\mathcal{R}_{k(s)} &\leq \sqrt{C_1 C_2}\,\mathbb{E}\big[3\sqrt{\tilde{\text{end}}_{k(s)}} + 5\sqrt{D_{\text{end}_{k(s)}}^{k(s)} \ln(D_{\text{end}_{k(s)}}^{k(s)} + 1)}\big] \\
&\leq \sqrt{C_1 C_2}\,\mathbb{E}\big[3\sqrt{\tilde{\text{end}}_{k(s)}} + 7\sqrt{2D_{\text{end}_{k(s)}}^{k(s)} \ln(D_{\text{end}_{k(s)}}^{k(s)} + 1)}\big].
\end{aligned}$$

The bounds also hold with $D_{\text{end}_{k(s)}}^{k(s)}$ replaced by $\mathfrak{D}_{\text{end}_{k(s)}}^{k(s)}$. Using the bounds $\widehat{D}_s$ and T on the clock and length, (3) yields

$$\mathbb{E}\left[\max_{x \in \Delta_A} \sum_{t=\text{start}_{k(s)}}^{\text{end}_{k(s)}} \langle \hat{\ell}_t, \hat{x}_t - x \rangle \,\middle|\, \mathcal{F}_{\text{start}_{k(s)}-1}\right] \leq \widehat{R}_s.$$

□

C.4 Proof of Lemma C.4 (Bound on Missing Feedback Contribution)

Intuition. Following lemma upper bounds the contribution of missing *feedback items before round T_0: it controls $\sum_{u\in\mathcal{M}(T_0)}\langle\hat{\ell}_u, x^c - x\rangle$ where $\mathcal{M}(T_0)$ is the set of rounds whose feedback has not arrived before round T_0. Intuitively, missing feedback is dangerous because it prevents the phase-gap statistic from "seeing" some losses, so we must separately bound how bad the unseen (missing) terms can be. We need this lemma in both phase analyses (normal and extinction): it is the only place where we pay explicitly for delayed/missing information, and it is exactly why the restart threshold includes the additive slack $\hat{\xi}_s$. The proof has two parts. First we bound the* number *of missing items: if feedback from s is still missing before round T_0 then necessarily $d_s \geq T_0 - s$, so summing these lower bounds over the missing indices yields a triangular-number constraint $D^{k(s)}_{\mathrm{end}_{k(s)}} \geq 1 + 2 + \cdots + m = m(m+1)/2$ and hence $m \lesssim \sqrt{D^{k(s)}_{\mathrm{end}_{k(s)}}}$. Second we bound the* per-item *impact of a missing estimator using the mixture structure $x_t = \alpha\hat{x}_t + (1-\alpha)x^c$.*

Lemma C.4: Bound on Missing Feedback Contribution. Fix a phase $k(s)$ of Algorithm 1 starting at $\mathrm{start}_{k(s)}$ with $\alpha_{k(s)} < 1$. Let $\mathrm{start}_{k(s)} \leq T_0 \leq \mathrm{end}_{k(s)} + 1$ index the feedback available through round $T_0 - 1$. Let

$$\mathcal{M}(T_0) \;\coloneqq\; \{\, u \in \{\mathrm{start}_{k(s)}, \ldots, T_0{-}1\} : \mathrm{status}_u(T_0{-}1) = \textsc{missing} \,\}, \qquad m \;\coloneqq\; |\mathcal{M}(T_0)|$$

be the set and number of rounds within phase $k(s)$ whose feedback has not arrived before round T_0 . Then for any $x \in \Delta_A$:

$$\sum_{u\in\mathcal{M}(T_0)} \langle\hat{\ell}_u,\, x^c - x\rangle \;\leq\; 2\sqrt{2D^{k(s)}_{\mathrm{end}_{k(s)}} + \tfrac{1}{4}} - 1.$$

Proof. **Step 1: Bounding the number of missing feedbacks.** For any $r \in \mathcal{M}(T_0)$, the feedback for round u has not arrived before round T_0, so its delay satisfies $d_u \geq T_0 - u$. Summing over all missing items:

$$D^{k(s)}_{\mathrm{end}_{k(s)}} \;=\; \sum_{t=\mathrm{start}_k(s)}^{\mathrm{end}_{k(s)}} d_t \;\geq\; \sum_{u\in\mathcal{M}(T_0)} d_u \;\geq\; \sum_{u\in\mathcal{M}(T_0)} (T_0 - u).$$

Let the missing indices be ordered as $u_1 < u_2 < \cdots < u_m$. Since the latest possible missing round is $T_0 - 1$ (feedback is checked at end of round), we have:

$$\begin{aligned} T_0 - u_m &\geq 1, \\ T_0 - u_{m-1} &\geq 2, \\ &\vdots \\ T_0 - u_1 &\geq m. \end{aligned}$$

Summing these lower bounds yields a triangular number:

$$\sum_{i=1}^{m}(T_0 - u_i) \;\geq\; \sum_{j=1}^{m} j \;=\; \frac{m(m+1)}{2}.$$

Combining this with the above lower bound on $D^{k(s)}_{\mathrm{end}_{k(s)}}$, we obtain

$$\frac{m(m+1)}{2} \;\leq\; D^{k(s)}_{\mathrm{end}_{k(s)}}.$$

Solving the quadratic for m:

$$m^2 + m - 2D^{k(s)}_{\mathrm{end}_{k(s)}} \leq 0 \quad\Longrightarrow\quad m \;\leq\; \frac{-1 + \sqrt{1 + 8D^{k(s)}_{\mathrm{end}_{k(s)}}}}{2} \;=\; \sqrt{2D^{k(s)}_{\mathrm{end}_{k(s)}} + \tfrac{1}{4}} - \tfrac{1}{2}.$$

The same counting gives $m(m+1)/2 \leq \mathfrak{D}^{k(s)}_{T_0}$, so the lemma also holds with $D^{k(s)}_{\mathrm{end}_{k(s)}}$ replaced by $\mathfrak{D}^{k(s)}_{T_0}$. **Step 2: Bounding the per-round inner product.** When the feedback for round u arrives, the

algorithm constructs the importance-weighted estimator $\hat{\ell}_u(a) = \frac{\ell_u(a)}{x_u(a)}\mathbf{1}\{A_u = a\}$. We analyze the inner product $\langle \hat{\ell}_u, x^c - x\rangle$:

$$\langle \hat{\ell}_u, x^c - x\rangle = \langle \hat{\ell}_u, x^c\rangle - \underbrace{\langle \hat{\ell}_u, x\rangle}_{\geq 0} \leq \langle \hat{\ell}_u, x^c\rangle.$$

Evaluating the term $\langle \hat{\ell}_u, x^c\rangle$:

$$\langle \hat{\ell}_u, x^c\rangle = \sum_a \frac{\ell_u(a)\mathbf{1}\{A_u = a\}}{x_u(A_u)} x^c(a) = \frac{\ell_u(A_u)x^c(A_u)}{x_u(A_u)}.$$

In any mixed phase, the played policy is a mixture $x_u = \alpha\hat{x}_u + (1-\alpha)x^c$. By the explicit rounding in Algorithm 2, $\hat{R} = 2^r$ is a power of 2. Since $\alpha = \frac{2^{k-1}}{\hat{R}} < 1$ implies $k < \log_2 \hat{R} + 1 = r + 1$, then $\alpha \leq 2^{r-1}/2^r = 1/2$. We have $1 - \alpha \geq 1/2$, and thus:

$$x_u(A_u) \geq (1-\alpha)x^c(A_u) \geq \tfrac{1}{2}x^c(A_u).$$

Using $\ell_u(A_u) \leq 1$, we get:

$$\langle \hat{\ell}_u, x^c\rangle \leq \frac{1 \cdot x^c(A_u)}{\frac{1}{2}x^c(A_u)} = 2.$$

Therefore, for every missing round u, $\langle \hat{\ell}_u, x^c - x\rangle \leq 2$.

Conclusion. Summing over the set of missing rounds $\mathcal{M}(T_0)$:

$$\sum_{u\in\mathcal{M}(T_0)} \langle \hat{\ell}_u, x^c - x\rangle \leq \sum_{u\in\mathcal{M}(T_0)} 2 = 2m \leq 2\left(\sqrt{2D^{k(s)}_{\mathrm{end}_{k(s)}} + \tfrac{1}{4}} - \tfrac{1}{2}\right) = 2\sqrt{2D^{k(s)}_{\mathrm{end}_{k(s)}} + \tfrac{1}{4}} - 1.$$

□

C.5 Proof of Lemma C.5 (Regret Bound for Normal Phase)

Intuition. Before the final checkpoint of a mixed phase, the phase-gap statistic does not exceed the threshold $2\widehat{R}_s + \widehat{\xi}_s$. This lemma shows two things: 1. The regret against any strategy x is bounded by a controlled quantity (roughly $2\widehat{R}_s + \widehat{\xi}_s$ plus a bounded missing-feedback contribution). 2. The regret against the safe comparator x^c is small (at most 2^{k-1}). We need this because the algorithm either keeps running in a normal phase, or it restarts due to a substantially suboptimal comparator; this lemma controls the regret up to either endpoint. Proof intuition: based on $x_t = \alpha_k\hat{x}_t + (1-\alpha_k)x^c$, we decompose regret into (i) regret of the base Banker-OMD sequence $\hat{x}_t$ (bounded by Lemma C.3) and (ii) a comparator-gap term. For the comparator-gap term, we split into "arrived" feedback (controlled by the phase-gap definition since restart didn't trigger) and "missing" feedback (controlled by Lemma C.4).

Lemma C.5: Regret Bound for Normal Phase. Consider a phase $k(s)$ of Algorithm 1 that starts at index $\mathrm{start}_{k(s)}$ and ends at the last round $\mathrm{end}_{k(s)}$ before the next soft or hard restart (or at T). Let $\alpha_{k(s)} < 1$ denote the value of α during this phase, and recall that for all t in this phase the played distribution is $x_t = \alpha_{k(s)}\hat{x}_t + (1-\alpha_{k(s)})x^c$. Let $\widehat{\xi}_s := \frac{\sqrt{8\widehat{D}_s+1}-1}{\delta}$. All expectations below are conditional on the phase-start history. Then for any comparator $x \in \Delta_A$:

$$\mathbb{E}\left[\sum_{t=\mathrm{start}_{k(s)}}^{\mathrm{end}_{k(s)}} \langle \hat{\ell}_t, x_t - x\rangle\right] \leq 4\widehat{R}_s + \widehat{\xi}_s + 2\sqrt{2\widehat{D}_s + \tfrac{1}{4}} + 1.$$

Moreover, for the comparator x^c:

$$\mathbb{E}\left[\sum_{t=\mathrm{start}_{k(s)}}^{\mathrm{end}_{k(s)}} \langle \hat{\ell}_t, x_t - x^c\rangle\right] \leq 2^{k(s)-1}.$$

Proof. Fix a phase $k(s)$ inside stage s. For simplicity of notation, we use k instead.

Fix a phase k. Since $x_t = \alpha_k \hat{x}_t + (1-\alpha_k)x^c$ throughout the phase, we can decompose the regret for any $x \in \Delta_A$:

$$\sum_{t=\text{start}_k}^{\text{end}_k} \langle \hat{\ell}_t, x_t - x \rangle = \sum_{t=\text{start}_k}^{\text{end}_k} \Big\langle \hat{\ell}_t, \alpha_k(\hat{x}_t - x) + (1-\alpha_k)(x^c - x) \Big\rangle$$
$$= \alpha_k \sum_{t=\text{start}_k}^{\text{end}_k} \langle \hat{\ell}_t, \hat{x}_t - x \rangle \; + \; (1-\alpha_k) \sum_{t=\text{start}_k}^{\text{end}_k} \langle \hat{\ell}_t, x^c - x \rangle.$$

Step 1: Bounding the Base Regret. For all phases in stage $s = 1$ and all phases in stages $s \geq 2$, Lemma C.3 applied to the Banker-OMD base sequence $\{\hat{x}_t\}$ indicates $\mathbb{E}[\sum_{t=\text{start}_k}^{\text{end}_k} \langle \hat{\ell}_t, \hat{x}_t - x \rangle] \leq \widehat{R}_s$. The observable phase clock is bounded by $\widehat{D}_s$, and each restart initializes the learner before its first action. Thus:

$$\alpha_k \, \mathbb{E}\left[\sum_{t=\text{start}_k}^{\text{end}_k} \langle \hat{\ell}_t, \hat{x}_t - x \rangle \right] \leq \alpha_k \widehat{R}_s. \tag{6}$$

Step 2: Bounding the Comparator Gap (Arrived Feedback). Recall the definition of the cumulative gap statistics:

$$g_t := \sum_{j=\text{start}_k}^{t} \hat{\ell}_j \, \mathbf{1}\{\text{status}_j(t) = \textsc{arrived}\}, \qquad \text{phase_gap}_t := \max_{u \in \Delta_A} \langle g_t, x^c - u \rangle.$$

Since the phase does *not* restart before time end_k, the restart condition must have failed for all check-points $t \in \{\text{start}_k, \ldots, \text{end}_k - 1\}$. At each such checkpoint the delay condition in the soft-restart test holds; otherwise a hard restart at $t+1$ would already end the phase. For a one-round phase, both pre-final-round sums are empty. Therefore:

$$\text{phase_gap}_t \leq 2\widehat{R}_s + \widehat{\xi}_s \qquad \text{for all } t \leq \text{end}_k - 1.$$

Evaluating this at the last check-point $t = \text{end}_k - 1$ for our specific comparator x:

$$\sum_{j=\text{start}_k}^{\text{end}_k - 1} \langle \hat{\ell}_j \mathbf{1}_{\{\text{status}_j(\text{end}_k - 1) = \textsc{arrived}\}}, x^c - x \rangle \leq \text{phase_gap}_{\text{end}_k - 1} \leq 2\widehat{R}_s + \widehat{\xi}_s. \tag{7}$$

Step 3: Bounding Missing and Last-Round Terms. We decompose the total comparator gap sum as:

$$\sum_{t=\text{start}_k}^{\text{end}_k} \langle \hat{\ell}_t, x^c - x \rangle = \underbrace{\sum_{t=\text{start}_k}^{\text{end}_k - 1} \langle \hat{\ell}_t \mathbf{1}_{\{\text{arrived}\}}, x^c - x \rangle}_{\text{Bounded by } 2\widehat{R}_s + \widehat{\xi}_s \text{ (Eq. 7)}} + \underbrace{\sum_{t=\text{start}_k}^{\text{end}_k - 1} \langle \hat{\ell}_t \mathbf{1}_{\{\text{missing}\}}, x^c - x \rangle}_{\text{Missing Feedback}}$$
$$+ \underbrace{\langle \hat{\ell}_{\text{end}_k}, x^c - x \rangle}_{\text{Last Round}}.$$

For the **Missing Feedback** term, we apply Lemma C.4 at $T_0 = \text{end}_k$, where $\mathfrak{D}_{\text{end}_k}^{k(s)} \leq \widehat{D}_s$:

$$\sum_{t=\text{start}_k}^{\text{end}_k - 1} \langle \hat{\ell}_t \mathbf{1}_{\{\text{status}_t = \textsc{missing}\}}, x^c - x \rangle \leq 2\sqrt{2\widehat{D}_s + \tfrac{1}{4}} - 1.$$

For the **Last Round** term, note that $\hat{\ell}_u = \frac{\ell_u(A_u)}{x_u(A_u)} e_{A_u}$. Thus:

$$\langle \hat{\ell}_{\text{end}_k}, x^c - x \rangle \leq \langle \hat{\ell}_{\text{end}_k}, x^c \rangle = \frac{\ell_{\text{end}_k}(A_{\text{end}_k}) x^c(A_{\text{end}_k})}{x_{\text{end}_k}(A_{\text{end}_k})} \leq \frac{x^c(A_{\text{end}_k})}{(1-\alpha_k)x^c(A_{\text{end}_k})} = \frac{1}{1-\alpha_k}.$$

In a normal phase, $\alpha_k \leq 1/2$ (by the explicit dyadic rounding), so $\frac{1}{1-\alpha_k} \leq 2$. Thus, the last round contributes at most 2.

Step 4: Conclusion. Combining all terms, taking expectation, and using $(1-\alpha_k) \le 1$:

$$\begin{aligned}
\mathbb{E}\Big[\sum_{t=\mathrm{start}_k}^{\mathrm{end}_k} \langle \hat{\ell}_t, x_t - x\rangle\Big] &\le \alpha_k \widehat{R}_s + (1-\alpha_k)\Big((2\widehat{R}_s + \widehat{\xi}_s) + (2\sqrt{2\widehat{D}_s + \tfrac{1}{4}} - 1) + 2\Big) \\
&\le \alpha_k \widehat{R}_s + (1-\alpha_k)\Big((4\widehat{R}_s + \widehat{\xi}_s) + (2\sqrt{2\widehat{D}_s + \tfrac{1}{4}} - 1) + 2\Big) \\
&\le 4\widehat{R}_s + \widehat{\xi}_s + 2\sqrt{2\widehat{D}_s + \tfrac{1}{4}} + 1.
\end{aligned}$$

This proves the first claim. For the second claim, set $x = x^c$. The comparator gap term vanishes entirely, leaving only the base regret term:

$$\mathbb{E}\left[\sum_{t=\mathrm{start}_k}^{\mathrm{end}_k} \langle \hat{\ell}_t, x_t - x^c\rangle\right] = \alpha_k \, \mathbb{E}\left[\sum_{t=\mathrm{start}_k}^{\mathrm{end}_k} \langle \hat{\ell}_t, \hat{x}_t - x^c\rangle\right] \le \alpha_k \widehat{R}_s \le 2^{k-1}.$$

□

C.6 Proof of Lemma C.6 (Negative Regret Credit in Extinction Phase)

Intuition. The following lemma captures the negative comparator-regret credit $-2^{k(s)-1}$ in an extinction phase, together with the realized Banker-regret remainder. We need this lemma for the per-stage $O(1)$ guarantee against the safe comparator x^c: each time we restart, we obtain geometric credit relative to x^c, which pays for the next phase's doubled budget in expectation. Proof intuition: rewrite $x_t - x^c = \alpha_{k(s)}(\hat{x}_t - x^c)$ and then insert an optimizer x^* of the phase-gap at the exit time, which makes the arrived part contribute at most $-(2\widehat{R}_s + \widehat{\xi}_s)$. The base Banker-OMD regret has phase-start conditional expectation at most $\widehat{R}_s$, and the missing part is bounded by $O(\sqrt{D^{k(s)}_{\mathrm{end}_{k(s)}}})$. Parameter $\widehat{\xi}_s$ is chosen so that it dominates the missing part, leaving a pathwise credit $-2\widehat{R}_s$ before the base-regret term. Centering at $\widehat{R}_s$ gives the certificate; Theorem C.1 sums in expectation.

Lemma C.6: Negative Regret Credit in Extinction Phase. Consider a phase $k(s)$ of Algorithm 1 that starts at index $\mathrm{start}_{k(s)}$ and ends at $\mathrm{end}_{k(s)} := \mathrm{start}_{k(s)+1} - 1$ when the soft-restart condition is triggered. Then pathwise:

$$\sum_{t=\mathrm{start}_{k(s)}}^{\mathrm{end}_{k(s)}} \langle \hat{\ell}_t,\, x_t - x^c \rangle \le -2^{k(s)-1} + \alpha_{k(s)} \left(\max_{x \in \Delta_A} \sum_{t=\mathrm{start}_{k(s)}}^{\mathrm{end}_{k(s)}} \langle \hat{\ell}_t, \hat{x}_t - x \rangle - \widehat{R}_s \right).$$

Proof. For simplicity of notation, use k to replace $k(s)$ in the following.

Step 1: Decomposition. During phase k, the mixing parameter is constant $\alpha := \alpha_k = \frac{2^{k-1}}{\widehat{R}_s} < 1$, and the played distribution is $x_t = \alpha \hat{x}_t + (1-\alpha)x^c$. Hence, $x_t - x^c = \alpha(\hat{x}_t - x^c)$, which allows us to scale the regret:

$$\sum_{t=\mathrm{start}_k}^{\mathrm{end}_k} \langle \hat{\ell}_t, x_t - x^c \rangle = \alpha \sum_{t=\mathrm{start}_k}^{\mathrm{end}_k} \langle \hat{\ell}_t, \hat{x}_t - x^c \rangle. \tag{8}$$

Step 2: Exit Condition Analysis. At time end_k, the algorithm computes the cumulative gap statistics using only arrived feedback:

$$g_{\mathrm{end}_k} := \sum_{j=\mathrm{start}_k}^{\mathrm{end}_k} \hat{\ell}_j \, \mathbf{1}\{\mathrm{status}_j(\mathrm{end}_k) = \textsc{arrived}\}.$$

Since end_k is the last round before soft-restarting, the restart condition must hold:

$$\mathrm{phase_gap}_{\mathrm{end}_k} = \max_{x \in \Delta_A} \langle g_{\mathrm{end}_k},\, x^c - x \rangle > 2\widehat{R}_s + \widehat{\xi}_s.$$

Let $x^* \in \arg\max_{x \in \Delta_A} \langle g_{\mathrm{end}_k}, x^c - x \rangle$. Then:

$$\langle g_{\mathrm{end}_k}, x^c - x^* \rangle > 2\widehat{R}_s + \widehat{\xi}_s \quad \Longleftrightarrow \quad \langle g_{\mathrm{end}_k}, x^* - x^c \rangle < -(2\widehat{R}_s + \widehat{\xi}_s). \tag{9}$$

Expanding g_{end_k}, this implies:

$$\sum_{t=\mathrm{start}_k}^{\mathrm{end}_k} \left\langle \hat{\ell}_t \, \mathbf{1}\{\mathrm{status}_t(\mathrm{end}_k) = \textsc{arrived}\},\, x^* - x^c \right\rangle < -(2\widehat{R}_s + \widehat{\xi}_s). \tag{10}$$

Step 3: Term-by-Term Bounding. We decompose the sum in (8) by introducing x^* and splitting arrived/missing terms:

$$\begin{aligned}\sum_{t=\text{start}_k}^{\text{end}_k} \langle \hat{\ell}_t, \hat{x}_t - x^c \rangle = &\sum_{t=\text{start}_k}^{\text{end}_k} \langle \hat{\ell}_t, \hat{x}_t - x^* \rangle \\ &+ \sum_{t=\text{start}_k}^{\text{end}_k} \left\langle \hat{\ell}_t \, \mathbf{1}\{\text{status}_t(\text{end}_k) = \textsc{arrived}\}, x^* - x^c \right\rangle \\ &+ \sum_{t=\text{start}_k}^{\text{end}_k} \left\langle \hat{\ell}_t \, \mathbf{1}\{\text{status}_t(\text{end}_k) = \textsc{missing}\}, x^* - x^c \right\rangle. \end{aligned} \tag{11}$$

First bound the terms pathwise:

(i) Banker-OMD Regret Term: The first term is at most

$$\max_{x \in \Delta_A} \sum_{t=\text{start}_k}^{\text{end}_k} \langle \hat{\ell}_t, \hat{x}_t - x \rangle.$$

By Lemma C.3, its expectation conditional on the phase's starting history is at most $\widehat{R}_s$; this estimate is used in Theorem C.1.

(ii) Arrived Feedback Term: From (10), this term is at most $-(2\widehat{R}_s + \widehat{\xi}_s)$.

(iii) Missing Feedback Term: Let m be the number of missing items at end_k. Since $\ell_t(A_t) \le 1$ and $x^c(a) \ge \delta$, we bound the inner product:

$$\langle \hat{\ell}_t, x^* - x^c \rangle \le \langle \hat{\ell}_t, x^* \rangle = \frac{\ell_t(A_t) x^*(A_t)}{x_t(A_t)} \le \frac{1}{x_t(A_t)}.$$

Since $x_t(a) \ge (1-\alpha)x^c(a) \ge (1-\alpha)\delta$ and dyadic rounding gives $\alpha \le 1/2$, $x_t(a) \ge \delta/2$. Thus, $\langle \hat{\ell}_t, x^* - x^c \rangle \le 2/\delta$. Summing over m missing items:

$$\sum_{\text{missing}} \langle \hat{\ell}_t, x^* - x^c \rangle \le \frac{2m}{\delta}. \tag{12}$$

Step 4: Conclusion. Plugging the bounds back into (11) and multiplying by α:

$$\sum_{t=\text{start}_k}^{\text{end}_k} \langle \hat{\ell}_t, x_t - x^c \rangle \le \alpha \left(\max_{x \in \Delta_A} \sum_{t=\text{start}_k}^{\text{end}_k} \langle \hat{\ell}_t, \hat{x}_t - x \rangle - (2\widehat{R}_s + \widehat{\xi}_s) + \frac{2m}{\delta} \right).$$

Recall we set $\widehat{\xi}_s = \frac{\sqrt{8\widehat{D}_s + 1} - 1}{\delta}$. The soft-restart delay check and Lemma C.4 give $m(m+1)/2 \le \mathfrak{D}^{k(s)}_{\text{end}_k + 1} \le \widehat{D}_s$, so $\frac{2m}{\delta} \le \widehat{\xi}_s$. The missing-feedback term drops out (or helps), leaving:

$$\sum_{t=\text{start}_k}^{\text{end}_k} \langle \hat{\ell}_t, x_t - x^c \rangle \le -\alpha \widehat{R}_s + \alpha \left(\max_{x \in \Delta_A} \sum_{t=\text{start}_k}^{\text{end}_k} \langle \hat{\ell}_t, \hat{x}_t - x \rangle - \widehat{R}_s \right).$$

Using $\alpha = 2^{k-1}/\widehat{R}_s$, we get $-\alpha \widehat{R}_s = -2^{k-1}$. □

C.7 Proof of Theorem C.1(Global Regret and Safety Guarantee)

Intuition. This theorem combines the per-phase analyses to obtain two global statements: (1) standard pseudo-regret against any strategy $x \in \Delta_A$ is at most $\tilde{O}(\sqrt{T} + \sqrt{D})$, and (2) the regret against the special safe comparator x^c is $O(1 + \log(1 + D))$. Intuitively, the algorithm alternates between cautious behavior (mixing with x^c) and increasingly aggressive behavior (doubling α); either it eventually becomes fully aggressive ($\alpha = 1$) and behaves like Banker-OMD, or it keeps restarting, but each restart certifies x^c behaves poorly. We need this theorem to justify both objectives of phased aggression: near-optimal regret scaling with delays, and stability control via the bounded deviation from x^c. Proof intuition: partition time into stages, then further decompose stage into phases. The maximum number of stages is bounded by the number of times $\widehat{D}_1$ can be doubled before exceeding D, which is $O(\log D)$. Therefore, we only need to focus on the stage-level regret. For the worst-case regret bound, apply Lemma C.5 to every phase with $\alpha < 1$ and Lemma C.3 to the final phase if α reaches 1; the number of $\alpha < 1$ phases is only $O(\log \widehat{R}_s)$ since α doubles each restart, which produces the $\tilde{O}$ bound. For the x^c-comparator guarantee, sum the phase-wise bounds: all exiting phases contribute credits $-\sum 2^{k(s)-1}$ by Lemma C.6, while the final phase contributes a budget of at most $+2^{K(s)-1}$ by Lemma C.5, so the total forms a geometric cancellation that leaves only a constant per stage after averaging the centered Banker terms.

Theorem C.1: Global Regret and Safety Guarantee. Under Definition 2.1 and Assumption 2.1, let $\delta \in (0, 1/A]$. Let $\{x_t\}_{t=1}^T$ be the sequence produced by Algorithm 1, and let $\hat{\ell}_t \in \mathbb{R}^A$ be the importance-weighted estimator defined in the algorithm. Then:

$$\max_{x \in \Delta_A} \mathbb{E}\left[\sum_{t=1}^{T} \langle \hat{\ell}_t,\ x_t - x \rangle\right] \ \le\ \tilde{O}\left(\sqrt{T} + \sqrt{D}\right),$$

and moreover:

$$\mathbb{E}\left[\sum_{t=1}^{T} \langle \hat{\ell}_t,\ x_t - x^c \rangle\right] \le O(1 + \log_2(1 + D)).$$

Proof. Partition the time horizon into stages indexed by $s = 1, 2, \ldots, S$. Further partition each stage s into phases indexed by $k(s) = 1, 2, \ldots, K(s)$. Phase $k(s)$ runs from $\text{start}_{k(s)}$ to $\text{end}_{k(s)}$. Let $\alpha_{k(s)}$ denote the constant mixing parameter for phase $k(s)$. Recall that $\alpha_{k(s)} = \min\{2^{k(s)-1}/\widehat{R}_s, 1\}$.

By construction, the delay budget $\widehat{D}_s$ doubles after each stage reset (specifically $\hat{D}_{s+1} \ge 2\hat{D}_s$). Each triggering observable counter is at most D, so the updated power-of-two budget is at most $2^{\lceil \log_2 D \rceil}$ when $D \ge 1$; when $D = 0$, no reset occurs. It follows that:

$$S \le \lceil \log_2(1 + D) \rceil + 1 = O(1 + \log(D + 1)).$$

Part 1: Standard Regret Bound (Against any x). Let $k_\star(s) := \min\{k(s) : \alpha_{k(s)} = 1\}$ (set $k_\star(s) = +\infty$ if $\alpha_{k(s)}$ never reaches 1). Since $\alpha_{k(s)} < 1$ implies $2^{k(s)-1} < \widehat{R}_s$, the number of phases with $\alpha_{k(s)} < 1$ is at most $\lceil \log_2 \max\{1, \widehat{R}_s\} \rceil$. For every such phase k, Lemma C.5 guarantees that for any $x \in \Delta_A$:

$$\mathbb{E}\left[\sum_{t=\text{start}_{k(s)}}^{\text{end}_{k(s)}} \langle \hat{\ell}_t,\ x_t - x \rangle\right] \le 4\widehat{R}_s + \widehat{\xi}_s + 2\sqrt{2\widehat{D}_s + \tfrac{1}{4}} + 1.$$

If a final phase occurs with $\alpha_{k_\star(s)} = 1$, the algorithm reduces to Banker-OMD for that duration. By Lemma C.3:

$$\mathbb{E}\left[\sum_{t=\text{start}_{k_\star(s)}}^{\text{end}_{k_\star(s)}} \langle \hat{\ell}_t,\ x_t - x \rangle\right] \le \widehat{R}_s.$$

Summing over all phases within a stage, conditional on its starting history, apply iterated expectation on each phase's entry event and bound its probability by 1.

$$
\begin{aligned}
&\mathbb{E}\left[\sum_{k(s)=1}^{K(s)}\sum_{t=\text{start}_{k(s)}}^{\text{end}_{k(s)}}\langle\hat{\ell}_t,\ x_t - x\rangle\right] \\
&\le \lceil\log_2\max\{1,\widehat{R}_s\}\rceil\left(4\widehat{R}_s + \widehat{\xi}_s + 2\sqrt{2\widehat{D}_s + \tfrac{1}{4}} + 1\right) \\
&+\ \widehat{R}_s \\
&\le \lceil 1 + \log_2\max\{1,\widehat{R}(2D)\}\rceil\left(8\widehat{R}(2D) + \frac{\sqrt{16D+1}-1}{\delta} + 2\sqrt{4D+\tfrac{1}{4}} + 1\right) \\
&+\ 2\widehat{R}(2D) \\
&= \tilde{O}\left(\widehat{R}(D) + \sqrt{D}/\delta\right).
\end{aligned}
$$

For $D \ge 1$, use $\widehat{D}_s \le 2D$ and $\widehat{R}_s \le 2\widehat{R}(\widehat{D}_s)$. For $D = 0$, one stage with $\widehat{D}_1 = 1$ gives the same rate. Substituting $\widehat{R}(D) = \tilde{O}\left(\sqrt{C_1C_2}(\sqrt{T} + \sqrt{D})\right)$, we obtain the stated bound $\tilde{O}\left(\sqrt{T} + \sqrt{D}\right)$.

Then sum over all stages,

$$
\mathbb{E}\left[\sum_{s=1}^{S}\sum_{k(s)=1}^{K(s)}\sum_{t=\text{start}_{k(s)}}^{\text{end}_{k(s)}}\langle\hat{\ell}_t,\ x_t - x\rangle\right] \le (\lceil\log_2(1+D)\rceil + 1)\tilde{O}\left(\sqrt{T} + \sqrt{D}\right).
$$

Part 2: Safety Guarantee (Regret against x^c). We analyze the sum $\sum_k \text{Regret}_k(x^c)$ by considering two cases. Condition on the start of stage s. Applying Lemma C.6 to the earlier phases and the mixture identity to the final phase gives, pathwise,

$$
\begin{aligned}
&\sum_{k=1}^{K(s)}\sum_{t=\text{start}_{k(s)}}^{\text{end}_{k(s)}}\langle\hat{\ell}_t, x_t - x^c\rangle \\
&\quad\le \min\{2^{K(s)-1}, \widehat{R}_s\} - \sum_{k=1}^{K(s)-1} 2^{k-1} \\
&\qquad + \sum_{k=1}^{K(s)}\alpha_{k(s)}\left(\max_{x\in\Delta_A}\sum_{t=\text{start}_{k(s)}}^{\text{end}_{k(s)}}\langle\hat{\ell}_t, \hat{x}_t - x\rangle - \widehat{R}_s\right).
\end{aligned}
$$

Case A: The algorithm never reaches $\alpha_{k(s)} = 1$. with in a stage s. The final phase $K(s)$ is a normal phase. Its budget contribution is $2^{K(s)-1}$. Every earlier phase $k(s) < K(s)$ ended because the restart condition was triggered. By Lemma C.6, such phases contribute the credits $-2^{k(s)-1}$. Summing these terms yields a telescoping geometric series:

$$
2^{K(s)-1} + \sum_{k(s)=1}^{K(s)-1}(-2^{k(s)-1}) = 2^{K(s)-1} - (2^{K(s)-1} - 1) = 1.
$$

Case B: The algorithm reaches $\alpha_{k(s)} = 1$ in phase $k_\star(s)$. Phases $1, \dots, k_\star(s) - 1$ are exiting phases. By Lemma C.6, their total credit against x^c is:

$$
\sum_{k(s)=1}^{k_\star(s)-1}(-2^{k(s)-1}) = -(2^{k_\star(s)-1} - 1).
$$

The final phase $k_\star(s)$ uses $\alpha_{k(s)} = 1$ (pure Banker-OMD). Its budget contribution is $\hat{R}_s$. The geometric term is thus bounded by $\hat{R}_s - (2^{k_\star(s)-1} - 1)$. Since $\alpha_{k_\star(s)} = 1$ implies $2^{k_\star(s)-1}/\hat{R}_s \ge 1 \iff 2^{k_\star(s)-1} \ge \hat{R}_s$, we have:

$$
\hat{R}_s - (2^{k_\star - 1} - 1) \le \hat{R}_s - (\hat{R}_s - 1) = 1.
$$

Both cases bound the geometric term by 1. The retained centered sum has nonpositive conditional expectation by Lemma C.3: sum over all possible phase-entry events, use $\alpha_{k(s)}$ fixed at phase start, and apply iterated expectation. Taking expectation after combining both cases gives a stage regret at most 1. In both cases, the total regret incurred in each stage s against x^c is $O(1)$.

Then sum over all stages,

$$\mathbb{E}\left[\sum_{s=1}^{S}\sum_{k(s)=1}^{K(s)}\sum_{t=\mathrm{start}_{k(s)}}^{\mathrm{end}_{k(s)}}\langle\hat{\ell}_t,\ x_t - x^c\rangle\right] \le (\lceil\log_2(1+D)\rceil + 1) = O(1+\log(1+D)).$$

□

D Examples of Different Regularizers

D.1 1/2-Tsallis

Take $x_0 = \left(\frac{1}{A}, \ldots, \frac{1}{A}\right)$. For $\Psi(x) = -2\sum_i \sqrt{x_i}$, $\nabla\Psi_i(x) = -\frac{1}{\sqrt{x_i}}$. Hence, $\nabla\Psi(x_0) = (-\sqrt{A}, \ldots, -\sqrt{A})$. For any $y \in \Delta_A$,

$$\begin{aligned} D_\Psi(y, x_0) &= \Psi(y) - \Psi(x_0) - \langle \nabla\Psi(x_0), y - x_0 \rangle \\ &= -2\sum_{i=1}^{A} \sqrt{y_i} + 2\sqrt{A} - \left\langle -\sqrt{A}\mathbf{1}, y - x_0 \right\rangle. \end{aligned}$$

Since $y, x_0 \in \Delta_A$, we have $\langle \mathbf{1}, y - x_0 \rangle = 0$, so

$$D_\Psi(y, x_0) = 2\left(\sqrt{A} - \sum_{i=1}^{A} \sqrt{y_i}\right).$$

Also $0 \le y_i \le 1$, so $\sqrt{y_i} \ge y_{i,}$ and therefore

$$\sum_{i=1}^{A} \sqrt{y_i} \ge \sum_{i=1}^{A} y_i = 1.$$

Thus

$$D_\Psi(y, x_0) \le 2(\sqrt{A} - 1).$$

Start from the same second-order expansion used in Lemma B.5 in Huang et al. [30]:

$$\sigma_t D_\Psi(\hat{x}_t, \tilde{z}_t) = \frac{1}{2\sigma_t} \left\| \hat{\ell}_t \right\|^2_{\nabla^2\Psi^*(\theta_t)},$$

for some θ_t on the segment joining $\nabla\Psi(\hat{x}_t) - \hat{\ell}_t/\sigma_t$ and $\nabla\Psi(\hat{x}_t)$. Now compute the Hessians for 1/2-Tsallis:

$$\nabla^2\Psi(x) = \operatorname{diag}\left(\frac{1}{2x_i^{3/2}}\right).$$

For the conjugate,

$$\nabla^2\Psi^*(\nabla\Psi(x)) = \left(\nabla^2\Psi(x)\right)^{-1} = \operatorname{diag}\left(2x_i^{3/2}\right).$$

Also, coordinatewise,

$$\nabla\Psi_i(\hat{x}_t) - \frac{\hat{\ell}_{t,i}}{\sigma_t} \le \nabla\Psi_i(\hat{x}_t),$$

because $\hat{\ell}_t \ge 0$ for the standard importance estimator with nonnegative losses. For this regularizer, each diagonal entry of $\nabla^2\Psi^*(\cdot)$ is increasing on the relevant negative domain, so

$$\nabla^2\Psi^*(\theta_t) \preceq \nabla^2\Psi^*(\nabla\Psi(\hat{x}_t)) = \operatorname{diag}\left(2\hat{x}_{t,i}^{3/2}\right).$$

Therefore,

$$\sigma_t D_\Psi(\hat{x}_t, \tilde{z}_t) \le \frac{1}{2\sigma_t}\sum_{i=1}^{A} \hat{l}_{t,i}^2 \cdot 2\hat{x}_{t,i}^{3/2} = \frac{1}{\sigma_t}\sum_{i=1}^{A} \hat{l}_{t,i}^2 \hat{x}_{t,i}^{3/2} = \frac{1}{\sigma_t}\left(\frac{l_{t,A_t}}{x_{t,A_t}}\right)^2 \hat{x}_{t,A_t}^{3/2} \le \frac{2}{\sigma_t\delta}\frac{\hat{x}_{t,A_t}^{3/2}}{x_{t,A_t}}.$$

For $\alpha < 1$, notice $l_{t,A_t} \in [0,1]$ and $x_{t,A_t} = \alpha\hat{x}_{t,A_t} + (1-\alpha)x^c \ge \delta/2$, then $\frac{1}{x_{t,A_t}} \le 2/\delta$.

$$\mathbb{E}\left[\sigma_t D_\Psi(\hat{x}_t, \tilde{z}_t) \mid \mathcal{F}_{t-1}\right] \le \frac{2}{\sigma_t\delta}\sum_{i=1}^{A} x_{t,i}\frac{\hat{x}_{t,i}^{3/2}}{x_{t,i}} = \frac{2}{\sigma_t\delta}\sum_{i=1}^{A} \hat{x}_{t,i}^{3/2} \le \frac{2}{\sigma_t\delta}$$

For $\alpha = 1$, the same quadratic bound gives conditional cost at most $\sqrt{A}/\sigma_t \le 2/(\delta\sigma_t)$.

D.2 negative entropy

Take $\Psi(x) = \sum_{i=1}^{A} x_i \log x_i$. Then $\nabla\Psi_i(x) = 1 + \log x_i, \nabla^2\Psi(x) = \operatorname{diag}(1/x_i)$. With uniform x_0,

$$D_\Psi(y, x_0) = \sum_{i=1}^{A} y_i \log \frac{y_i}{1/A} = \sum_{i=1}^{A} y_i \log(Ay_i).$$

This is exactly the KL divergence $D_{\mathrm{KL}}(y\|x_0)$, and over the simplex its maximum is attained at a vertex, giving

$$D_\Psi(y, x_0) \le \log A.$$

So we take $C_1 = \log A$.

Start from the same second-order expansion used in Lemma B.5 in Huang et al. [30]:

$$\sigma_t D_\Psi(\hat{x}_t, \tilde{z}_t) = \frac{1}{2\sigma_t} \left\| \hat{\ell}_t \right\|^2_{\nabla^2\Psi^*(\theta_t)},$$

for some θ_t on the segment joining $\nabla\Psi(\hat{x}_t) - \hat{\ell}_t/\sigma_t$ and $\nabla\Psi(\hat{x}_t)$. For negative entropy, the conjugate is $\Psi^*(\theta) = \sum_{i=1}^{A} e^{\theta_i - 1}$, so

$$\nabla^2\Psi^*(\theta) = \operatorname{diag}\left(e^{\theta_i - 1}\right).$$

Because the losses are nonnegative in the delayed adversarial MAB setup and the importance estimator has nonnegative coordinates, we have

$$\nabla\Psi(\hat{x}_t) - \frac{\hat{\ell}_t}{\sigma_t} \le \nabla\Psi(\hat{x}_t) \qquad \text{coordinatewise.}$$

Since $e^{\theta_i - 1}$ is increasing in θ_i,

$$\nabla^2\Psi^*(\theta_t) \preceq \nabla^2\Psi^*(\nabla\Psi(\hat{x}_t)) = \nabla^2\Psi(\hat{x}_t)^{-1} = \operatorname{diag}(\hat{x}_{t,i}).$$

Hence, for $\alpha < 1$,

$$\sigma_t D_\Psi(\hat{x}_t, \tilde{z}_t) \le \frac{1}{2\sigma_t} \left\| \hat{\ell}_t \right\|^2_{\operatorname{diag}(\hat{x}_t)} = \frac{1}{2\sigma_t} \sum_{i=1}^{A} \hat{x}_{t,i} \hat{\ell}_{t,i}^2 = \frac{1}{2\sigma_t} \left(\frac{l_{t,A_t}}{x_{t,A_t}} \right)^2 \hat{x}_{t,A_t} \le \frac{1}{\sigma_t \delta} \frac{\hat{x}_{t,A_t}}{x_{t,A_t}}$$

Then

$$\mathbb{E}\left[\sigma_t D_\Psi(\hat{x}_t, \tilde{z}_t) \mid \mathcal{F}_{t-1}\right] \le \frac{1}{\sigma_t \delta} \sum_{i=1}^{A} \hat{x}_{t,i} = \frac{1}{\sigma_t \delta}$$

For $\alpha = 1$, the quadratic bound gives $A/(2\sigma_t) \le 1/(\delta\sigma_t)$.

E Lower Bound

E.1 Proof of Lemma E.1 (Greedy bucket properties)

Intuition. The greedy buckets identify intervals during which the learner cannot receive any feedback generated inside the same interval. Thus each bucket behaves like one batch. Since the delays are non-increasing, later buckets cannot be longer than earlier ones. Moreover, the square of a bucket length controls the total delay hidden in the next bucket: each point in the next bucket has delay at most the previous bucket length, and the next bucket itself is no longer than the previous one.

Lemma E.1: Greedy bucket properties. Let $d_1, \dots, d_T \in \mathbb{Z}_{\geq 1}$ be non-increasing and satisfy $d_t \leq T+1-t \quad \forall t \in [T]$. Let $B_m = \{b_m, \dots, b_{m+1}-1\}, m = 1, \dots, M$, be the greedy bucket decomposition, i.e., $\forall t \in B_m : t + d_t > b_{m+1} - 1$, and $\exists t_m \in B_m : t_m + d_{t_m} = b_{m+1}$. Write $L_m := |B_m| = b_{m+1} - b_m$. Then:

1. $L_1 \geq L_2 \geq \cdots \geq L_M$.
2. For every $m \in [M-1]$,
$$L_m^2 \geq \sum_{t \in B_{m+1}} d_t.$$

Proof. We borrow the proof idea from Masoudian et al. [46]. For every m, since $b_m \in B_m$, we have

$$b_m + d_{b_m} > b_{m+1} - 1.$$

Since b_m, d_{b_m}, and b_{m+1} are integers, it follows that

$$b_m + d_{b_m} \geq b_{m+1},$$

and therefore

$$L_m = b_{m+1} - b_m \leq d_{b_m}.$$

On the other hand, by construction of the greedy buckets, there exists $t_m \in B_m$ such that

$$t_m + d_{t_m} = b_{m+1}.$$

Hence

$$L_m = b_{m+1} - b_m = d_{t_m} + (t_m - b_m) \geq d_{t_m}.$$

Since $t_m \leq b_{m+1} - 1$ and the delays are non-increasing,

$$d_{t_m} \geq d_{b_{m+1}-1}.$$

Thus

$$L_m \geq d_{b_{m+1}-1}.$$

Now

$$L_{m+1} = b_{m+2} - b_{m+1} \leq d_{b_{m+1}} \leq d_{b_{m+1}-1} \leq L_m,$$

which proves

$$L_1 \geq L_2 \geq \cdots \geq L_M.$$

For the second claim, let $t \in B_{m+1}$. Since $t \geq b_{m+1}$ and the delays are non-increasing,

$$d_t \leq d_{b_{m+1}} \leq d_{b_{m+1}-1} \leq L_m.$$

Therefore

$$\sum_{t \in B_{m+1}} d_t \leq |B_{m+1}|\, L_m = L_{m+1} L_m \leq L_m^2,$$

where the last inequality uses $L_{m+1} \leq L_m$. □

E.2 Proof of Lemma E.2 (Safe lower bound for the batched bandit game)

Intuition. This theorem isolates the statistical obstruction created by batching. We build two environments that differ only in whether arm 2 is slightly worse or slightly better than the baseline arms. If the learner plays arm 2 often, then it violates safety in the environment where arm 2 is worse. If it plays arm 2 rarely, then it suffers large regret in the environment where arm 2 is better. Because feedback inside each block is hidden until the block ends, the learner cannot reliably distinguish these two environments quickly enough. The quantity $V = \sum_m L_m^2$ is the effective variance scale of this batched testing problem.

Lemma E.2: Safe lower bound for the batched bandit game. Let $A \geq 2$, let $n \geq 1$, and let $L_1 \geq L_2 \geq \cdots \geq L_n \geq 1$ be positive integers. Consider the batched bandit game with block lengths $L_1, \ldots, L_n$: in block m, the learner chooses actions $J_{m,1}, \ldots, J_{m,L_m} \in [A]$ sequentially, but without observing any loss from the current block; after the block ends, the learner observes the chosen losses $g_{m,1,J_{m,1}}, \ldots, g_{m,L_m,J_{m,L_m}}$. Define $V := \sum_{m=1}^n L_m^2$. Fix $\delta \in (0, 1/A]$ and $\Lambda \geq 0$, and assume $\delta \geq \frac{L_1}{64V}$. Then there exists a comparator $x^c \in \Delta_A$ with $x^c(i) \geq \delta \qquad \forall i \in [A]$ such that for every batched bandit algorithm there exists an oblivious deterministic block-loss sequence

$$g_{m,s} \in [0,1]^A, \qquad m = 1, \ldots, n,\ s = 1, \ldots, L_m,$$

for which

$$R^{\mathrm{bat}}(x^c; g) \leq \Lambda \quad \Longrightarrow \quad \max_{x \in \Delta_A} R^{\mathrm{bat}}(x; g) \geq \frac{3}{256}\, \delta^{-1/2} \sqrt{\frac{V}{L_1}} - \frac{\sqrt{\Lambda}}{32\sqrt{2}}\, \delta^{-3/4} \left(\frac{V}{L_1}\right)^{1/4} - \Lambda.$$

Here

$$R^{\mathrm{bat}}(x; g) := \mathbb{E}\left[\sum_{m=1}^n \sum_{s=1}^{L_m} g_{m,s,J_{m,s}} - \sum_{m=1}^n \sum_{s=1}^{L_m} \langle x, g_{m,s} \rangle\right],$$

where the expectation is over the learner's randomization.

Proof. Fix any batched bandit algorithm. Define

$$x^c := (1 - (A-1)\delta, \delta, \ldots, \delta) \in \Delta_A.$$

Since $\delta \leq 1/A$, every coordinate of x^c is at least δ.

We first construct two oblivious randomized environments, and then extract a deterministic loss sequence from one of them.

Step 1: randomized hard instance. Pick a parameter $\gamma > 0$, to be fixed later, and define

$$\varepsilon_m := \gamma \frac{L_m}{\sqrt{V}}, \qquad \Delta_m := \gamma \frac{L_m}{\sqrt{V}}, \qquad w_m := \frac{L_m^2}{V}, \qquad m = 1, \ldots, n.$$

Then $w_m \geq 0$ and $\sum_{m=1}^n w_m = 1$.

For each block m and each slot $s = 1, \ldots, L_m$, let

$$Z_{m,s}^+ \sim \mathrm{Bernoulli}\left(\frac{1}{2} + \varepsilon_m\right), \qquad Z_{m,s}^- \sim \mathrm{Bernoulli}\left(\frac{1}{2} - \varepsilon_m\right),$$

independently across all (m, s).

Under $\mathcal{E}_+$, define

$$g_{m,s,1}^{(+)} = \frac{1}{2}, \qquad g_{m,s,2}^{(+)} = Z_{m,s}^+, \qquad g_{m,s,i}^{(+)} = \frac{1}{2} \quad (i = 3, \ldots, A),$$

and under $\mathcal{E}_-$, define

$$g_{m,s,1}^{(-)} = \frac{1}{2}, \qquad g_{m,s,2}^{(-)} = Z_{m,s}^-, \qquad g_{m,s,i}^{(-)} = \frac{1}{2} \quad (i = 3, \ldots, A).$$

Thus every loss lies in $[0,1]^A$.

Let N_m be the number of times the learner plays arm 2 in block m:

$$N_m := \sum_{s=1}^{L_m} \mathbf{1}\{J_{m,s} = 2\}.$$

Define the weighted number of arm 2 being played in block m as:

$$W := \sum_{m=1}^{n} \frac{L_m N_m}{V}.$$

Since $0 \le N_m \le L_m$ and $\sum_{m=1}^{n} L_m^2 = V$, we have

$$0 \le W \le 1.$$

Step 2: safety regret under $\mathcal{E}_+$. Under $\mathcal{E}_+$, arm 2 has mean loss

$$\mathbb{E}[g_{m,s,2}^{(+)}] = \frac{1}{2} + \Delta_m,$$

while all other arms have mean loss $1/2$. Hence the learner's expected loss in block m equals

$$\frac{L_m}{2} + \Delta_m \, \mathbb{E}_+[N_m],$$

and the comparator x^c incurs expected loss

$$L_m \left((1-\delta)\frac{1}{2} + \delta \left(\frac{1}{2} + \Delta_m \right) \right) = \frac{L_m}{2} + \delta L_m \Delta_m.$$

Therefore

$$R_+^{\mathrm{bat}}(x^c) = \sum_{m=1}^{n} \Delta_m \big(\mathbb{E}_+[N_m] - \delta L_m \big).$$

Since $\Delta_m = \gamma L_m / \sqrt{V}$,

$$R_+^{\mathrm{bat}}(x^c) = \gamma \sqrt{V} \left(\mathbb{E}_+[W] - \delta \right).$$

Thus, if

$$R_+^{\mathrm{bat}}(x^c) \le \Lambda,$$

then

$$\mathbb{E}_+[W] \le \delta + \frac{\Lambda}{\gamma \sqrt{V}}. \tag{13}$$

Step 3: KL bound between $\mathcal{E}_+$ and $\mathcal{E}_-$. Let P_+ and P_- denote the laws of the full batched transcript under $\mathcal{E}_+$ and $\mathcal{E}_-$, respectively. Write H_m for the transcript up to the end of block $m-1$.

By the chain rule for KL divergence,

$$\mathrm{KL}(P_+ \| P_-) = \sum_{m=1}^{n} \mathbb{E}_+ \left[\mathrm{KL}\big(P_+(J_{m,1:L_m}, Y_{m,1:L_m} \mid H_m) \| P_-(J_{m,1:L_m}, Y_{m,1:L_m} \mid H_m) \big) \right],$$

where

$$J_{m,1:L_m} := (J_{m,1}, \ldots, J_{m,L_m}), \qquad Y_{m,1:L_m} := (g_{m,s,J_{m,s}})_{s=1}^{L_m}.$$

Because the learner observes no loss from the current block before the block ends, the conditional law of $J_{m,1:L_m}$ given H_m is the same under P_+ and P_-. Hence the block-m contribution reduces to the KL of the observed losses conditioned on the chosen action sequence.

Fix H_m and a realized action sequence $a_{1:L_m}$. For every slot s with $a_s \neq 2$, the observed loss is deterministically $1/2$ under both environments, so the KL contribution is 0. For every slot s with

$a_s = 2$, the observed loss is Bernoulli with parameter $1/2 + \varepsilon_m$ under $\mathcal{E}_+$ and with parameter $1/2 - \varepsilon_m$ under $\mathcal{E}_-$. Since losses are independent across slots, the conditional KL for block m is

$$N_m(a_{1:L_m}) \operatorname{kl}\left(\frac{1}{2} + \varepsilon_m, \frac{1}{2} - \varepsilon_m\right).$$

Taking expectation gives

$$\mathrm{KL}(P_+ \| P_-) = \sum_{m=1}^{n} \mathbb{E}_+[N_m] \operatorname{kl}\left(\frac{1}{2} + \varepsilon_m, \frac{1}{2} - \varepsilon_m\right).$$

For $\varepsilon \in [0, 1/4]$,

$$\operatorname{kl}\left(\frac{1}{2} + \varepsilon, \frac{1}{2} - \varepsilon\right) \le 16\varepsilon^2.$$

We will verify below that $\varepsilon_m \le 1/4$ for all m. Using $\varepsilon_m = \gamma L_m / \sqrt{V}$, we obtain

$$\mathrm{KL}(P_+ \| P_-) \le 16\gamma^2 \sum_{m=1}^{n} \mathbb{E}_+[N_m] \frac{L_m^2}{V}.$$

Since $L_m \le L_1$,

$$\frac{L_m^2}{V} \mathbb{E}_+[N_m] \le L_1 \frac{L_m \mathbb{E}_+[N_m]}{V}.$$

Therefore

$$\mathrm{KL}(P_+ \| P_-) \le 16\gamma^2 L_1 \mathbb{E}_+[W].$$

By Pinsker's inequality,

$$\mathrm{TV}(P_+, P_-) \le \sqrt{\frac{1}{2}\mathrm{KL}(P_+ \| P_-)} \le 2\sqrt{2}\,\gamma \sqrt{L_1 \mathbb{E}_+[W]} \le 4\gamma \sqrt{L_1 \mathbb{E}_+[W]}.$$

Since $W \in [0, 1]$,

$$\mathbb{E}_-[W] - \mathbb{E}_+[W] \le \mathrm{TV}(P_+, P_-) \le 4\gamma \sqrt{L_1 \mathbb{E}_+[W]}.$$

Hence

$$\mathbb{E}_-[W] \le \mathbb{E}_+[W] + 4\gamma \sqrt{L_1 \mathbb{E}_+[W]}. \tag{14}$$

Step 4: regret against the best arm under $\mathcal{E}_-$. Under $\mathcal{E}_-$, arm 2 is the unique best arm in expectation, because

$$\mathbb{E}[g_{m,s,2}^{(-)}] = \frac{1}{2} - \Delta_m,$$

while every other arm has mean loss $1/2$. The learner's expected regret against the pure strategy e_2 is therefore

$$R_-^{\mathrm{bat}}(e_2) = \sum_{m=1}^{n} \Delta_m \big(L_m - \mathbb{E}_-[N_m]\big) = \gamma\sqrt{V}\big(1 - \mathbb{E}_-[W]\big).$$

Combining this with (14) and then with (13), we get: if $R_+^{\mathrm{bat}}(x^c) \le \Lambda$, then

$$R_-^{\mathrm{bat}}(e_2) \ge \gamma\sqrt{V}\left(1 - \delta - \frac{\Lambda}{\gamma\sqrt{V}} - 4\gamma\sqrt{L_1\left(\delta + \frac{\Lambda}{\gamma\sqrt{V}}\right)}\right).$$

Using $\sqrt{x+y} \le \sqrt{x} + \sqrt{y}$,

$$R_-^{\mathrm{bat}}(e_2) \ge \gamma\sqrt{V}\big(1 - \delta - 4\gamma\sqrt{L_1\delta}\big) - \Lambda - 4\sqrt{\Lambda}\,\gamma^{3/2} V^{1/4} \sqrt{L_1}.$$

Choose

$$\gamma := \frac{1}{32\sqrt{L_1\delta}}.$$

Then

$$\varepsilon_m = \frac{L_m}{32\sqrt{L_1\delta}\sqrt{V}} \le \frac{\sqrt{L_1}}{32\sqrt{\delta V}} \le \frac{1}{4},$$

where the last inequality follows from

$$\delta \geq \frac{L_1}{64V}.$$

Thus the Bernoulli KL bound used above is valid.

Moreover,

$$4\gamma\sqrt{L_1\delta} = \frac{1}{8}, \qquad \delta \leq \frac{1}{A} \leq \frac{1}{2},$$

and hence

$$1 - \delta - 4\gamma\sqrt{L_1\delta} \geq \frac{3}{8}.$$

Therefore

$$\gamma\sqrt{V}\big(1 - \delta - 4\gamma\sqrt{L_1\delta}\big) \geq \frac{3}{256}\,\delta^{-1/2}\sqrt{\frac{V}{L_1}}.$$

Also,

$$4\sqrt{\Lambda}\,\gamma^{3/2}V^{1/4}\sqrt{L_1} = \frac{\sqrt{\Lambda}}{32\sqrt{2}}\,\delta^{-3/4}\left(\frac{V}{L_1}\right)^{1/4}.$$

Hence, if $R_+^{\mathrm{bat}}(x^c) \leq \Lambda$, then

$$R_-^{\mathrm{bat}}(e_2) \geq \frac{3}{256}\,\delta^{-1/2}\sqrt{\frac{V}{L_1}} - \frac{\sqrt{\Lambda}}{32\sqrt{2}}\,\delta^{-3/4}\left(\frac{V}{L_1}\right)^{1/4} - \Lambda.$$

Since

$$\max_{x\in\Delta_A} R_-^{\mathrm{bat}}(x) \geq R_-^{\mathrm{bat}}(e_2),$$

we have shown that

$$\max_{x\in\Delta_A} R_-^{\mathrm{bat}}(x) \geq \frac{3}{256}\,\delta^{-1/2}\sqrt{\frac{V}{L_1}} - \frac{\sqrt{\Lambda}}{32\sqrt{2}}\,\delta^{-3/4}\left(\frac{V}{L_1}\right)^{1/4} - \Lambda \tag{15}$$

whenever $R_+^{\mathrm{bat}}(x^c) \leq \Lambda$.

Step 5: choose one randomized environment. There are two cases.

Case 1: $R_+^{\mathrm{bat}}(x^c) > \Lambda$. Choose $\mathcal{E} = \mathcal{E}_+$. Then

$$R_{\mathcal{E}}^{\mathrm{bat}}(x^c) \leq \Lambda \quad \Longrightarrow \quad \max_{x\in\Delta_A} R_{\mathcal{E}}^{\mathrm{bat}}(x) \geq B_\Lambda$$

holds vacuously, where

$$B_\Lambda := \frac{3}{256}\,\delta^{-1/2}\sqrt{\frac{V}{L_1}} - \frac{\sqrt{\Lambda}}{32\sqrt{2}}\,\delta^{-3/4}\left(\frac{V}{L_1}\right)^{1/4} - \Lambda.$$

Case 2: $R_+^{\mathrm{bat}}(x^c) \leq \Lambda$. Choose $\mathcal{E} = \mathcal{E}_-$. Then (15) gives

$$\max_{x\in\Delta_A} R_{\mathcal{E}}^{\mathrm{bat}}(x) \geq B_\Lambda,$$

so the implication

$$R_{\mathcal{E}}^{\mathrm{bat}}(x^c) \leq \Lambda \quad \Longrightarrow \quad \max_{x\in\Delta_A} R_{\mathcal{E}}^{\mathrm{bat}}(x) \geq B_\Lambda$$

holds as well.

Thus, for every batched bandit algorithm there exists an oblivious randomized environment $\mathcal{E}$ for which the desired implication holds.

Step 6: extract a deterministic sequence. If we are in Case 1, then

$$R_{\mathcal{E}_+}^{\mathrm{bat}}(x^c) > \Lambda.$$

Since this is the expectation of the regret over the random block-loss sequence drawn from $\mathcal{E}_+$, there exists a deterministic realization for which

$$R^{\mathrm{bat}}(x^c; g) > \Lambda.$$

For this realization, the implication

$$R^{\mathrm{bat}}(x^c; g) \le \Lambda \quad \Longrightarrow \quad \max_{x \in \Delta_A} R^{\mathrm{bat}}(x; g) \ge B_\Lambda$$

holds vacuously.

If we are in Case 2, then

$$\max_{x \in \Delta_A} R^{\mathrm{bat}}_{\mathcal{E}_-}(x) \ge B_\Lambda.$$

Writing $R^{\mathrm{bat}}(x; g)$ for the regret on a deterministic realization g, we have

$$R^{\mathrm{bat}}_{\mathcal{E}_-}(x) = \mathbb{E}_{g \sim \mathcal{E}_-}\left[R^{\mathrm{bat}}(x; g)\right].$$

Therefore

$$\mathbb{E}_{g \sim \mathcal{E}_-}\left[\max_{x \in \Delta_A} R^{\mathrm{bat}}(x; g)\right] \ge \max_{x \in \Delta_A} \mathbb{E}_{g \sim \mathcal{E}_-}\left[R^{\mathrm{bat}}(x; g)\right] \ge B_\Lambda.$$

Hence some deterministic realization g satisfies

$$\max_{x \in \Delta_A} R^{\mathrm{bat}}(x; g) \ge B_\Lambda.$$

For this realization, the implication

$$R^{\mathrm{bat}}(x^c; g) \le \Lambda \quad \Longrightarrow \quad \max_{x \in \Delta_A} R^{\mathrm{bat}}(x; g) \ge B_\Lambda$$

holds.

Thus, in both cases, there exists an oblivious deterministic block-loss sequence with the claimed property. □

E.3 Proof of Lemma E.3 (Uniform simulation of a delayed bandit algorithm by a batched bandit algorithm)

Intuition. This lemma formalizes the reduction from delayed feedback to batching. Once we restrict attention to a suffix of greedy buckets, no feedback generated inside a bucket can be used before that bucket ends. Therefore a batched algorithm can faithfully simulate any delayed algorithm: it pre-simulates the zero prefix, then processes each suffix bucket as one batch, revealing the chosen losses only after the bucket ends. Since the simulated delayed transcript is identical to the original delayed transcript, the regret against every comparator is exactly preserved.

Lemma E.3: Uniform simulation of a delayed bandit algorithm by a batched bandit algorithm. Assume the setting of Lemma E.1, and fix $j \in [M]$. For every delayed bandit algorithm $\mathcal{A}^{\mathrm{del}}$, there exists a batched bandit algorithm $\mathcal{A}^{\mathrm{bat}}$ with block lengths $L_j, \dots, L_M$, depending only on $\mathcal{A}^{\mathrm{del}}$, the delay sequence $d_{1:T}$, and j, such that the following holds.
For every family of vectors

$$g_{m,s} \in [0,1]^A, \qquad m = j, \dots, M, \quad s = 1, \dots, L_m,$$

define a deterministic delayed-loss sequence $\ell(g) = (\ell_1(g), \dots, \ell_T(g))$ by

$$\ell_t(g) := \begin{cases} \mathbf{0}, & t < b_j, \\ g_{m,s}, & t = b_m + s - 1 \text{ for some } m \in \{j, \dots, M\},\ s \in [L_m]. \end{cases}$$

Then for every comparator $x \in \Delta_A$,

$$R_T^{\mathrm{del}}(x; \ell(g)) = R_{j:M}^{\mathrm{bat}}(x; g).$$

where

$$R_T^{\mathrm{del}}(x; \ell(g)) = \mathbb{E}\left[\sum_{t=1}^{T} \left([\ell_t(g)]_{A_t} - \langle x, \ell_t(g)\rangle\right)\right].$$

Similarly,

$$R_{j:M}^{\mathrm{bat}}(x; g) := \mathbb{E}\left[\sum_{m=j}^{M} \sum_{s=1}^{L_m} \left([g_{m,s}]_{J_{m,s}} - \langle x, g_{m,s}\rangle\right)\right],$$

where A_t is the action played by $\mathcal{A}^{\mathrm{del}}$ at delayed round t and $J_{m,s}$ is the action played by $\mathcal{A}^{\mathrm{bat}}$ in slot s of block m. Here the left-hand side is the regret of $\mathcal{A}^{\mathrm{del}}$ on the delayed instance $\ell(g)$, and the right-hand side is the regret of $\mathcal{A}^{\mathrm{bat}}$ on the batched instance g.

Proof. Fix a delayed bandit algorithm $\mathcal{A}^{\mathrm{del}}$. We construct a single batched bandit algorithm $\mathcal{A}^{\mathrm{bat}}$; the construction depends only on $\mathcal{A}^{\mathrm{del}}$, $d_{1:T}$, and j, and not on the eventual choice of g.

Internally, $\mathcal{A}^{\mathrm{bat}}$ maintains a faithful copy of $\mathcal{A}^{\mathrm{del}}$, including its random seed. We use the convention that feedback whose reveal time is t is delivered immediately before the action at round t.

Step 1: pre-simulate the zero prefix. For every round $t < b_j$, the delayed loss vector is $\ell_t(g) = \mathbf{0}$, so the entire prefix is known in advance. The batched simulator therefore runs rounds $1, \dots, b_j - 1$ of $\mathcal{A}^{\mathrm{del}}$ internally before the batched game begins.

More concretely, for each simulated round $t = 1, \dots, b_j - 1$, it first delivers to the internal copy the entire collection of previously generated prefix feedback items whose reveal time equals t. Each such feedback value is 0. The internal copy then chooses an action $I_t \in [A]$. Since the loss vector at round t is $\mathbf{0}$, the chosen loss is also 0. If its reveal time $t + d_t$ lies in $\{t+1, \dots, b_j - 1\}$, then this zero loss will be delivered later during the same pre-simulation. If $t + d_t \in \{b_j, \dots, T\}$, the simulator stores this feedback item, together with its reveal time $t + d_t$, for delivery during the batched simulation. (If $t + d_t = T + 1$, it is irrelevant for decisions up to horizon T and need not be stored.)

After this pre-simulation, the internal copy is in exactly the same state that $\mathcal{A}^{\mathrm{del}}$ would be in immediately before round b_j on the delayed instance $\ell(g)$.

Step 2: simulate the suffix online, one bucket at a time. Now process the suffix buckets $B_j, \dots, B_M$. For a fixed bucket B_m, the batched algorithm goes through the slots corresponding to rounds

$$t = b_m, \dots, b_{m+1} - 1.$$

Before the slot corresponding to round t, the batched simulator delivers to the internal copy the entire collection of stored feedback items whose reveal time equals t. We claim that every feedback item that should be revealed at time t is indeed already known to the batched simulator.

Let u be a round whose feedback is revealed at time t, so $u + d_u = t$. Since $d_u \geq 1$, necessarily $u < t$. There are now three possibilities.

- If $u < b_j$, then u is a prefix round, and its zero feedback was generated during Step 1 and stored there.
- If $u \in B_r$ for some $r < m$, then u lies in a strictly earlier suffix bucket. The chosen loss from round u was observed when block r ended, and was then stored together with its reveal time.
- u cannot lie in the current bucket B_m, because for every $u \in B_m$, the greedy-bucket property gives

  $$u + d_u > b_{m+1} - 1 \geq t.$$

  Likewise, u cannot lie in a future bucket B_r with $r > m$, because then $u \geq b_{m+1} > t$, so $u + d_u \geq u + 1 > t$.

Hence every feedback item due at time t comes either from the prefix or from a strictly earlier suffix bucket, and is already available to the batched simulator.

After receiving all feedback items whose availability time equals t, the internal copy of $\mathcal{A}^{\mathrm{del}}$ outputs an action $A_t \in [A]$. The batched algorithm plays the same action in the corresponding slot

$$s = t - b_m + 1$$

of block m; that is, it sets

$$J_{m,s} := A_t.$$

When block m ends, the batched environment reveals the chosen losses

$$[g_{m,1}]_{J_{m,1}}, \dots, [g_{m,L_m}]_{J_{m,L_m}}.$$

For each $s \in [L_m]$, let

$$u = b_m + s - 1.$$

Since $J_{m,s} = A_u$, the revealed value satisfies

$$[g_{m,s}]_{J_{m,s}} = [g_{m,s}]_{A_u} = [\ell_u(g)]_{A_u}.$$

The simulator stores the delayed feedback item

$$\big(A_u, [g_{m,s}]_{J_{m,s}}\big)$$

with availability time

$$\tau_u := u + d_u + 1,$$

provided $\tau_u \leq T$. If $\tau_u > T$, then this feedback item cannot affect any action up to horizon T, and hence may be ignored.

This storage is valid because the greedy-bucket property ensures that, for every $u \in B_m$,

$$u + d_u > b_{m+1} - 1.$$

Since all quantities are integers, this implies

$$u + d_u \geq b_{m+1},$$

and therefore

$$\tau_u = u + d_u + 1 \geq b_{m+1} + 1.$$

Thus no feedback generated inside the current bucket B_m is available before any action inside B_m. Once block m ends, the batched simulator knows all chosen losses from B_m, and can deliver them to the internal delayed copy at their correct future availability times.

By induction over rounds $t = b_j, \dots, T$, immediately before each suffix round t, the internal copy of $\mathcal{A}^{\mathrm{del}}$ has exactly the same decision-relevant history as $\mathcal{A}^{\mathrm{del}}$ would have on the delayed instance $\ell(g)$. In particular, for every suffix round $t = b_m + s - 1$,

$$J_{m,s} = A_t.$$

Step 3: equality of regrets. For every suffix round $t = b_m + s - 1$, we have

$$\ell_t(g) = g_{m,s} \qquad \text{and} \qquad J_{m,s} = A_t.$$

Therefore the learner incurs the same realized loss in the delayed and batched games:

$$[\ell_t(g)]_{A_t} = [g_{m,s}]_{A_t} = [g_{m,s}]_{J_{m,s}}.$$

The comparator incurs the same loss as well:

$$\langle x, \ell_t(g)\rangle = \langle x, g_{m,s}\rangle.$$

For the prefix rounds $t < b_j$, we have $\ell_t(g) = \mathbf{0}$, so

$$[\ell_t(g)]_{A_t} - \langle x, \ell_t(g)\rangle = 0.$$

Hence, pathwise,

$$\sum_{t=1}^{T} \left([\ell_t(g)]_{A_t} - \langle x, \ell_t(g)\rangle\right) = \sum_{m=j}^{M} \sum_{s=1}^{L_m} \left([g_{m,s}]_{J_{m,s}} - \langle x, g_{m,s}\rangle\right).$$

Taking expectation over the common randomization of the coupled delayed and batched algorithms gives

$$R_T^{\mathrm{del}}(x; \ell(g)) = R_{j:M}^{\mathrm{bat}}(x; g).$$

□

E.4 Proof of Theorem E.1(Delayed safety lower bound)

Intuition. The delayed lower bound combines the previous two ingredients. First, the greedy buckets convert a suffix of the delayed game into a batched game with block lengths $L_j, \dots, L_M$. Second, the batched lower bound applies to this simulated game and gives a regret lower bound governed by $V_j = \sum_{m=j}^{M} L_m^2$. Finally, the greedy bucket geometry shows that this batched complexity dominates the total delay hidden in the suffix: $V_j \geq D_{\overline{S_j}}$. Hence large delayed feedback forces the same safety–optimality tradeoff as large batches.

Theorem E.1: Delayed safety lower bound. Let $d_1, \dots, d_T \in \mathbb{Z}_{\geq 1}$ be non-increasing delays with $d_t \leq T+1-t \quad \forall t \in [T]$. Let $D := \sum_{t=1}^{T} d_t$ be the total delay. Let $B_m = \{b_m, \dots, b_{m+1} - 1\}, \quad m = 1, \dots, M$, be the greedy bucket decomposition, and define $L_m := |B_m|$. Fix $j \in [M]$ and write

$$V_j := \sum_{m=j}^{M} L_m^2, \qquad S_j := \bigcup_{m=1}^{j} B_m, \qquad D_{\overline{S_j}} := \sum_{t \in [T] \setminus S_j} d_t.$$

Fix $\delta \in (0, 1/A]$ and assume $\delta \geq \frac{L_j}{64 V_j}$. Then there exists a comparator $x^c \in \Delta_A$ with $x^c(i) \geq \delta \quad \forall i \in [A]$ such that for every delayed adversarial bandit algorithm there exists an oblivious deterministic delayed-loss sequence $\ell_1, \dots, \ell_T \in [0,1]^A$ for which

$$R_T^{\mathrm{del}}(x^c; \ell) \leq \log D \quad \Rightarrow \quad \max_{x \in \Delta_A} R_T^{\mathrm{del}}(x; \ell) \geq \frac{3}{256} \delta^{-1/2} \sqrt{\frac{V_j}{L_j}} - \frac{\sqrt{\log D}}{32\sqrt{2}} \delta^{-3/4} \left(\frac{V_j}{L_j}\right)^{1/4} - \log D.$$

Moreover, $V_j \geq D_{\overline{S_j}}$, and for the same loss sequence ℓ,

$$R_T^{\mathrm{del}}(x^c; \ell) \leq \log D \quad \Rightarrow \quad \max_{x \in \Delta_A} R_T^{\mathrm{del}}(x; \ell) \geq \frac{3}{256} \delta^{-1/2} \sqrt{\frac{D_{\overline{S_j}}}{L_j}} - \frac{\sqrt{\log D}}{32\sqrt{2}} \delta^{-3/4} \left(\frac{V_j}{L_j}\right)^{1/4} - \log D.$$

Proof. By Lemma E.1, the suffix bucket lengths satisfy

$$L_j \geq L_{j+1} \geq \cdots \geq L_M.$$

Set

$$n := M - j + 1, \qquad \widetilde{L}_r := L_{j+r-1}, \qquad r = 1, \ldots, n.$$

Then

$$\widetilde{L}_1 \geq \widetilde{L}_2 \geq \cdots \geq \widetilde{L}_n \geq 1, \qquad \sum_{r=1}^{n} \widetilde{L}_r^2 = V_j.$$

Apply Lemma E.2 to the block-length sequence $\widetilde{L}_1, \ldots, \widetilde{L}_n$ with

$$\Lambda = \log D.$$

Since $d_t \geq 1$ for all t, we have $D \geq 1$, and hence $\Lambda \geq 0$. Also,

$$\delta \geq \frac{L_j}{64 V_j} = \frac{\widetilde{L}_1}{64 \sum_{r=1}^{n} \widetilde{L}_r^2},$$

so the batched theorem applies. Hence there exists a comparator $x^c \in \Delta_A$ with

$$x^c(i) \geq \delta \qquad \forall i \in [A]$$

such that for every batched bandit algorithm on block lengths $\widetilde{L}_1, \ldots, \widetilde{L}_n$, there exists an oblivious deterministic block-loss sequence

$$\widetilde{g}_{r,s} \in [0,1]^A, \qquad r = 1, \ldots, n, \quad s = 1, \ldots, \widetilde{L}_r,$$

for which

$$R^{\mathrm{bat}}(x^c; \widetilde{g}) \leq \log D \quad \Longrightarrow \quad \max_{x \in \Delta_A} R^{\mathrm{bat}}(x; \widetilde{g}) \geq \frac{3}{256}\, \delta^{-1/2} \sqrt{\frac{V_j}{L_j}} - \frac{\sqrt{\log D}}{32\sqrt{2}}\, \delta^{-3/4} \left(\frac{V_j}{L_j}\right)^{1/4} - \log D.$$

Now fix any delayed adversarial bandit algorithm $\mathcal{A}^{\mathrm{del}}$. By Lemma E.3, there exists a batched bandit algorithm $\mathcal{A}^{\mathrm{bat}}$, depending only on $\mathcal{A}^{\mathrm{del}}$, $d_{1:T}$, and j, such that for every suffix block-loss sequence g and the associated delayed sequence $\ell(g)$,

$$R_T^{\mathrm{del}}(x; \ell(g)) = R_{j:M}^{\mathrm{bat}}(x; g) \qquad \forall x \in \Delta_A.$$

Apply the batched guarantee above to this specific batched algorithm $\mathcal{A}^{\mathrm{bat}}$. Then there exists a deterministic block-loss sequence

$$\widetilde{g}_{r,s} \in [0,1]^A, \qquad r = 1, \ldots, n, \quad s = 1, \ldots, \widetilde{L}_r,$$

such that

$$R^{\mathrm{bat}}(x^c; \widetilde{g}) \leq \log D \quad \Longrightarrow \quad \max_{x \in \Delta_A} R^{\mathrm{bat}}(x; \widetilde{g}) \geq \frac{3}{256}\, \delta^{-1/2} \sqrt{\frac{V_j}{L_j}} - \frac{\sqrt{\log D}}{32\sqrt{2}}\, \delta^{-3/4} \left(\frac{V_j}{L_j}\right)^{1/4} - \log D.$$

Re-index this block-loss sequence by setting

$$g_{m,s} := \widetilde{g}_{m-j+1,s}, \qquad m = j, \ldots, M, \quad s = 1, \ldots, L_m,$$

and define the delayed-loss sequence ℓ by

$$\ell_t := \begin{cases} \mathbf{0}, & t < b_j, \\ g_{m,s}, & t = b_m + s - 1 \text{ for some } m \in \{j, \ldots, M\},\ s \in [L_m]. \end{cases}$$

Since this is only a relabeling of the blocks,

$$R^{\mathrm{bat}}(x; \widetilde{g}) = R_{j:M}^{\mathrm{bat}}(x; g) \qquad \forall x \in \Delta_A.$$

Therefore, by Lemma E.3,

$$R_T^{\mathrm{del}}(x;\ell) = R_{j:M}^{\mathrm{bat}}(x;g) = R^{\mathrm{bat}}(x;\widetilde{g}) \qquad \forall x \in \Delta_A.$$

Thus

$$R_T^{\mathrm{del}}(x^c;\ell) \le \log D \quad \Longrightarrow \quad \max_{x\in\Delta_A} R_T^{\mathrm{del}}(x;\ell) \ge \frac{3}{256}\,\delta^{-1/2}\sqrt{\frac{V_j}{L_j}} - \frac{\sqrt{\log D}}{32\sqrt{2}}\,\delta^{-3/4}\left(\frac{V_j}{L_j}\right)^{1/4} - \log D.$$

It remains to relate V_j to $D_{\overline{S_j}}$. Lemma E.1 gives

$$L_m^2 \ge \sum_{t\in B_{m+1}} d_t \qquad (m = j,\dots,M-1).$$

Summing over $m = j,\dots,M-1$,

$$\sum_{m=j}^{M-1} L_m^2 \ge \sum_{m=j+1}^{M}\sum_{t\in B_m} d_t = D_{\overline{S_j}}.$$

Since

$$V_j = \sum_{m=j}^{M} L_m^2 \ge \sum_{m=j}^{M-1} L_m^2,$$

we obtain

$$V_j \ge D_{\overline{S_j}}.$$

Therefore

$$\sqrt{\frac{V_j}{L_j}} \ge \sqrt{\frac{D_{\overline{S_j}}}{L_j}},$$

and the second displayed lower bound follows for the same loss sequence ℓ. □

Corollary E.1: A delay sequence with linear dependence on total delay Fix integers $q \ge 1$ and $N \ge 1$, set $T := (N+1)q$, and define the non-increasing delay sequence $d_t := \min\{q, T+1-t\}, \quad t = 1,\dots,T$. The greedy bucket decomposition is $B_m = \{(m-1)q+1,\dots,mq\}, \quad m = 1,\dots,N+1$, so that $L_m = q \quad (m = 1,\dots,N+1)$. Taking $j = 1$, if $\delta \in (0, 1/A]$ satisfies $\delta \ge \frac{L_1}{64V_1} = \frac{1}{64\left(\frac{D}{q} + \frac{q-1}{2}\right)}$, then there exists a comparator $x^c \in \Delta_A$ with $x^c(i) \ge \delta \quad \forall i \in [A]$ such that for every delayed adversarial bandit algorithm there exists an oblivious deterministic delayed-loss sequence

$$\ell_1,\dots,\ell_T \in [0,1]^A$$

for which $R_T^{\mathrm{del}}(x^c;\ell) \le \log D$ implies

$$\max_{x\in\Delta_A} R_T^{\mathrm{del}}(x;\ell) \ge \frac{3}{256}\,\delta^{-1/2}\sqrt{D} - \frac{\sqrt{\log D}}{32\sqrt{2}}\,\delta^{-3/4}D^{1/4} - \log D \text{ for } q = 1$$

Proof. The sequence $d_t = \min\{q, T+1-t\}$ is non-increasing and satisfies

$$d_t \le T+1-t \qquad \forall t \in [T].$$

Moreover, $d_t = q$ for $t \le Nq+1$, and on the final $q-1$ rounds the delays decrease as $q-1, q-2,\dots,1$. Hence

$$D = Nq^2 + \sum_{r=1}^{q} r = Nq^2 + \frac{q(q+1)}{2}.$$

For $m = 1,\dots,N+1$, define

$$b_m := (m-1)q + 1.$$

Then $B_m = \{b_m,\dots,b_{m+1}-1\}$ has length q. If $m \le N$, then $d_t = q$ throughout B_m, and therefore for every $t \in B_m$,

$$t + d_t = t + q > b_{m+1} - 1,$$

with equality $b_m + d_{b_m} = b_{m+1}$. For the last bucket B_{N+1}, we have

$$t + d_t = T + 1 = b_{N+2} \qquad \forall t \in B_{N+1}.$$

Thus these are exactly the greedy buckets.

The identities for L_1, V_1, and $D_{\overline{S_1}}$ follow directly:

$$L_1 = q, \qquad V_1 = \sum_{m=1}^{N+1} q^2 = (N+1)q^2,$$

and since $S_1 = B_1$ and $\sum_{t \in B_1} d_t = q^2$,

$$D_{\overline{S_1}} = D - q^2.$$

Dividing by $L_1 = q$ gives

$$\frac{V_1}{L_1} = (N+1)q = \frac{D}{q} + \frac{q-1}{2},$$

and

$$\frac{D_{\overline{S_1}}}{L_1} = \frac{D - q^2}{q} = \frac{D}{q} - q.$$

The regret lower bound is Theorem E.1 applied with $j = 1$. □

F Experimental details

This appendix describes the experimental setup used for the dynamic comparison in Figure 2, which reports fixed one-step, geometric, and Pareto delayed-feedback conditions.. The purpose of the experiment is to test safety-aware bandit algorithms under delayed feedback. In particular, the experiment stresses the following failure mode: if feedback is delayed, an algorithm may incorrectly infer that the safe comparator is suboptimal and may switch to an aggressive exploration mode before enough evidence has actually arrived. The proposed algorithm, PRUDENT-BANKER, is designed to avoid this mistimed transition by using a delay-calibrated restart threshold.

The displayed run uses horizon

$$T = 50{,}000$$

and number of arms

$$A = 100.$$

At each round t, the learner selects an arm $a_t \in [A]$ and incurs loss ℓ_{t,a_t}, where

$$\ell_t = (\ell_{t,1}, \dots, \ell_{t,A}) \in [0,1]^A.$$

Smaller loss is better. Although the full loss table is generated before the experiment, each algorithm only observes the loss of the arm it selected, and that observation may arrive after a delay.

F.1 Synthetic non-stationary loss process

The displayed experiment uses the flexible non-stationary loss generator denoted `env_any` in the implementation. Time is divided into

$$B = 500$$

blocks. For each arm $i \in [A]$ and each block $b \in [B]$, the environment draws an arm-block-specific mean and standard deviation,

$$x_{i,b} \sim \mathrm{Unif}(0,1), \qquad \sigma_{i,b} \sim \mathrm{Unif}(0.1, 0.2).$$

The implementation assigns round t to block

$$b(t) = 1 + \min\left\{ \left\lfloor \frac{t-1}{\lfloor T/B \rfloor + 1} \right\rfloor, B-1 \right\}.$$

Conditional on these parameters, the loss of arm i at time t is sampled from a normal distribution truncated to the unit interval:

$$\ell_{t,i} \sim \mathrm{TN}_{[0,1]}\left(x_{i,b(t)}, \sigma^2_{i,b(t)}\right).$$

Thus the environment is non-stationary: an arm that is good in one block may be bad in another block. This makes the setting more demanding than a stationary stochastic bandit problem, because algorithms must learn while the quality of arms changes over time.

All algorithms in a given delay column are run on the same realized loss table and the same realized delay sequence. Therefore, differences between curves are due to the algorithms rather than to different random environments.

F.2 Safe comparator

To isolate safety behavior conditional on a strong safe reference, we choose the reference arm in hindsight. Let $i^\star \in \arg\min_{i \in [A]} \sum_{t=1}^T \ell_{t,i}$ be the best fixed arm for the realized loss table. PRUDENT-BANKER receives the full-support comparator x^c defined by $x^c(i^\star) = 1 - (A-1)\delta$ and $x^c(i) = \delta$ for $i \neq i^\star$. In the displayed run, $A = 100$ and $\delta = 0.001$, so $x^c(i^\star) = 0.901$. The conservative baselines require a deterministic safe default arm and a known default reward; we therefore give them the same anchor $\mathrm{arm}\, i^\star$ and $r_0 = \frac{1}{T}\sum_{t=1}^T (1 - \ell_{t,i^\star})$. This oracle choice is favorable to methods that frequently play the default arm and should be interpreted as a diagnostic setting, not as an online procedure for constructing a safe comparator.

Table 1: Delay settings in the dynamic experiment. Each delayed column uses the same realized delay sequence for all algorithms; the geometric setting has the largest realized total delay in the plotted run.

Column	Delay model	Realized D
Fixed one-step	$\Pr(d_t = 1) = 0.03$ and $\Pr(d_t = 0) = 0.97$.	1545
Geometric	With probability 0.03, feedback is delayed and $d_t \sim \mathrm{Geom}(0.4)$ on $\{1, 2, \ldots\}$; otherwise $d_t = 0$.	3927
Pareto	With probability 0.03, feedback is delayed and $d_t = 1 + \lfloor Z_t \rfloor$, where Z_t is generated by NumPy's Pareto sampler, $Z_t \sim \mathrm{Lomax}(\alpha = 2.5$, scale $= 1$), i.e. $Z_t \geq 0$; otherwise $d_t = 0$.	2065

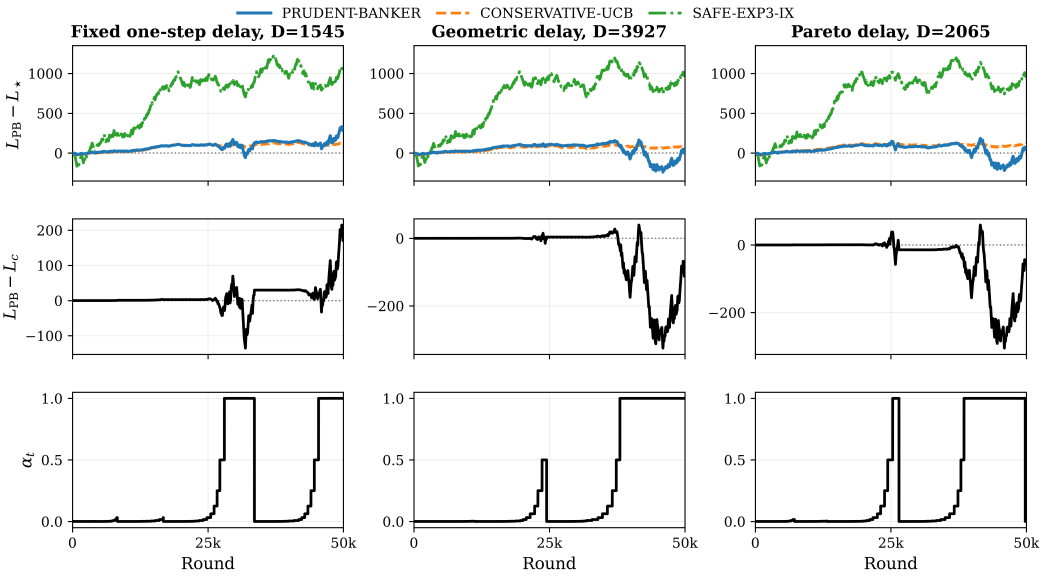


Figure 2: Dynamic performance against the best-fixed-arm comparator. The restart-threshold multiplier is 10^{-3}. The top row plots cumulative loss minus the cumulative loss of the best fixed arm in hindsight, the middle row plots PRUDENT-BANKER loss minus comparator loss, and the bottom row plots the PRUDENT-BANKER mixing coefficient α_t. The key takeaway is that delayed feedback changes the timing of safety decisions: under delay, PRUDENT-BANKER repeatedly restarts conservatively and only becomes aggressive after crossing the empirical restart threshold.

F.3 Delay models

Feedback from round s arrives at the end of round $s + d_s$, where d_s is the realized delay. Under this end-of-round convention, feedback from round s with $d_s = 0$ is observed after the action and loss at round s, and is first usable for the decision at round $s + 1$. More generally, feedback arriving at the end of round $s + d_s$ is first usable from round $s + d_s + 1$. The realized total delay is

$$D = \sum_{t=1}^{T} d_t.$$

The experiment compares three delay settings, summarized in Table 1.

The Fixed one-step setting has only short delays: each round is delayed by one step with probability 0.03, and otherwise has no delay. The geometric setting has the largest realized total delay in the displayed run. The Pareto setting has a smaller realized total delay than the geometric setting, but it is heavy-tailed: rare longer delays can create local bursts of missing feedback. This is important because the effect of delays is not determined only by the scalar quantity D; the temporal placement of delayed observations can also affect when an algorithm becomes aggressive or conservative.

F.4 Evaluation metrics

Let $p_t^B \in \Delta_A$ denote the action distribution used by algorithm B at time t. The plotted curves are pseudo-loss curves:

$$L_B(t) = \sum_{s=1}^{t} \langle p_s^B, \ell_s \rangle.$$

Thus the plotted loss is the expected loss under the algorithm's action distribution, not merely the realized loss of the sampled arm. This reduces sampling noise in the visualization while preserving the bandit-feedback learning dynamics used by the algorithms.

The cumulative loss of the best realized fixed arm is

$$L_\star(t) = \sum_{s=1}^{t} \ell_{s,i^\star}.$$

The top row of Figure 2 plots

$$R_B^\star(t) = L_B(t) - L_\star(t).$$

This is the cumulative loss of algorithm B minus the cumulative loss of the best fixed arm in hindsight. Lower is better. Since $i^\star$ is selected using the entire horizon, a curve can be slightly negative at early times if an algorithm temporarily outperforms the arm that is best over the full horizon.

The middle row plots the PRUDENT-BANKER advantage over comparator:

$$L_{\text{PRUDENT-BANKER}}(t) - L_c(t),$$

where

$$L_c(t) = \sum_{s=1}^{t} \langle x_c, \ell_s \rangle.$$

Positive values mean that PRUDENT-BANKER has more cumulative loss than the safe comparator. Negative values mean that PRUDENT-BANKER has accumulated less loss than the comparator.

The bottom row plots the mixing coefficient α_t used by PRUDENT-BANKER. Its played distribution is

$$x_t = (1 - \alpha_t)x^c + \alpha_t \hat{x}_t,$$

where x^c is the safe comparator and $\hat{x}_t$ is the BANKER-OMD distribution. Thus α_t measures aggressiveness:

$$\alpha_t \approx 0 \quad \Longrightarrow \quad \text{PRUDENT-BANKER is close to the safe comparator,}$$

whereas

$$\alpha_t = 1 \quad \Longrightarrow \quad \text{PRUDENT-BANKER fully trusts the BANKER-OMD learner.}$$

Thus, in the middle row, downward movement indicates improvement relative to the comparator.

F.5 Algorithms

F.5.1 PRUDENT-BANKER

The proposed method combines a delayed-feedback BANKER-OMD learner with a safety-aware phased-aggression mechanism. The implementation labels this method as PRUDENT-BANKER in the figure.

The BANKER-OMD component uses the negative-entropy regularizer

$$\psi(x) = \sum_{i=1}^{A} x_i \log x_i.$$

The regret budget uses Algorithm 2's explicit expression. Therefore,

$$C_1 = \log A = \log 100, \qquad C_2 = \frac{1}{\delta} = 1000,$$

and

$$\sigma_t = \sqrt{\frac{C_2}{C_1}} \left(\frac{1}{\sqrt{t - \text{start}_{k(s)} + 1}} + \mathfrak{d}_t^{k(s)} \sqrt{\frac{\ln\left(\mathfrak{D}_t^{k(s)} + 1\right)}{\mathfrak{D}_t^{k(s)}}} \right)^{-1}.$$

The displayed runs multiply $2\widehat{R}_s + \widehat{\xi}_s$ by 10^{-3}, an empirical tuning outside the theorem's guarantee.

F.5.2 CONSERVATIVE-UCB

CONSERVATIVE-UCB is based on Conservative UCB of Wu et al. [62], who introduced the conservative bandit framework and proposed a UCB-style algorithm that maintains a reward constraint relative to a default arm uniformly over time. In this experiment, the safe default is the hindsight best fixed arm $i^\star$. The baseline is also given the default arm's average reward

$$r_0 = \frac{1}{T}\sum_{t=1}^{T}(1-\ell_{t,i^\star}).$$

This makes CONSERVATIVE-UCB particularly strong in the top-row metric because repeatedly playing the default arm is already close to optimal with respect to the best-fixed-arm comparator.

The implementation uses delayed observations. That is, if feedback from round s arrives at time $s+d_s$, then the reward from round s is not used to update the empirical mean or confidence interval before that arrival time. However, CONSERVATIVE-UCB is not delay-calibrated in the same sense as PRUDENT-BANKER: it does not maintain a delay-budget estimate, it does not use a delay-corrected comparator-suboptimality threshold, and it does not restart phases when the realized delay budget exceeds an estimate.

Let $O_{<t} = \{s < t : s + d_s < t\}$ be the set of feedback items available before choosing at round t . Let $N_i^{\mathrm{obs}}(t)$ be the number of rewards from $\mathrm{arm}\, i$ in $O_{<t}$, and let $S_i(t)$ be their sum. For an arm with $N_i^{\mathrm{obs}}(t) > 0$, define

$$\widehat{r}_i(t) = \frac{S_i(t)}{N_i^{\mathrm{obs}}(t)}$$

and a confidence radius

$$c_i(t) = \sqrt{\frac{2\log\left(\max\left\{3, \frac{2A(t+1)^2}{\delta_{\mathrm{ucb}}}\right\}\right)}{N_i^{\mathrm{obs}}(t)}}.$$

where $\delta_{\mathrm{UCB}} = \frac{1}{\max\{T,2\}} = 2\times 10^{-5}$ in our experiment.

The lower and upper confidence bounds are

$$\mathrm{LCB}_i(t) = \max\{0, \widehat{r}_i(t) - c_i(t)\}, \qquad \mathrm{UCB}_i(t) = \min\{1, \widehat{r}_i(t) + c_i(t)\}.$$

For an unobserved arm, the implementation uses

$$\mathrm{LCB}_i(t) = 0, \qquad \mathrm{UCB}_i(t) = 1.$$

For the default arm, the confidence interval is collapsed to the known value:

$$\mathrm{LCB}_{i^\star}(t) = \mathrm{UCB}_{i^\star}(t) = r_0.$$

At time t, the algorithm first selects the optimistic candidate

$$j_t \in \arg\max_{i\in[A]} \mathrm{UCB}_i(t).$$

It then checks whether playing j_t would keep the pessimistic reward budget above the conservative safety requirement. Let $N_i^{\mathrm{play}}(t)$ be the number of times arm i has been played before round t. The implementation plays j_t only if

$$\sum_{i=1}^{A} N_i^{\mathrm{play}}(t)\mathrm{LCB}_i(t) + \mathrm{LCB}_{j_t}(t) \geq (1-\alpha_{\mathrm{safe}})(t+1)r_0.$$

Otherwise, it plays the safe default arm $i^\star$. We pick $\alpha_{\mathrm{safe}} = 0.1$.

F.5.3 SAFE-EXP3-IX

SAFE-EXP3-IX combines the adversarial safe-playing strategy of Wu et al. [62] with EXP3-IX as the base adversarial bandit learner. EXP3-IX is the implicit-exploration variant of EXP3 introduced by Kocák et al. [36]; we use the standard implicit exploration estimator also analyzed in high-probability

form by Neu [49]. In our experiments both baselines are implemented with delayed observations by applying their updates only when feedback arrives. The implementation uses

$$\eta = \min\left\{\frac{1}{2}, \sqrt{\frac{\log A}{AT}}\right\}, \qquad \gamma = \frac{\eta}{2}.$$

Let $w_{t,i}$ be the EXP3 weight of arm i. When the safety wrapper allows the base learner to act, the base distribution is

$$q_t(i) = \frac{w_{t,i}}{\sum_{j=1}^{A} w_{t,j}}.$$

When feedback from a base-learner round s arrives, the algorithm forms the implicit-exploration estimator

$$\widehat{\ell}_{s,a_s} = \frac{\ell_{s,a_s}}{q_s(a_s) + \gamma},$$

and updates the corresponding log-weight by

$$\log w_{s+1,a_s} = \log w_{s,a_s} - \eta \widehat{\ell}_{s,a_s}.$$

All other coordinates are unchanged by that feedback item.

The safety wrapper maintains a reward budget relative to the known default arm. At round t, the required budget is

$$(1 - \alpha_{\text{safe}}) r_0(t+1).$$

If the current budget is large enough, the algorithm allows EXP3-IX to act. If not, the algorithm plays the safe default arm. As with CONSERVATIVE-UCB, rewards from non-default arms are credited only when the delayed feedback arrives.

Thus SAFE-EXP3-IX is a delayed-feedback baseline: it updates the adversarial bandit learner when feedback arrives. However, unlike PRUDENT-BANKER, it does not use a delay-calibrated comparator-suboptimality test. Consequently, it can continue to explore broadly even when the comparator is already very strong.

F.6 Interpretation of Figure 2

F.6.1 Overall patterns

The top row shows cumulative loss relative to the best fixed arm. Since the safe comparator puts most weight on hindsight best arm, a conservative method that often plays the default arm can perform very well in this metric. This explains why CONSERVATIVE-UCB remains close to the zero line in most columns.

SAFE-EXP3-IX accumulates substantially larger loss relative to the best fixed arm. This is expected in this diagnostic setting: EXP3-style algorithms maintain broad exploration for adversarial robustness, and this exploration is costly when the comparator is already highly concentrated on the best fixed arm.

PRUDENT-BANKER moves between these behaviors. Its behavior is best understood by looking at the bottom row. When α_t is small, the algorithm is close to the safe comparator. When α_t approaches 1, the algorithm becomes fully aggressive and behaves like the adaptive BANKER-OMD learner. Therefore, changes in the top and middle rows are closely tied to the trajectory of α_t.

F.6.2 Fixed one-step delays

In the Fixed one-step column, only 3% of feedback events are delayed, and each active delay has length one. The realized total delay is

$$D = 1545.$$

The bottom row shows repeated cycles in α_t. The coefficient rises as the algorithm collects evidence, but it drops when the observable delay count exceeds the current stage estimate and a restart is triggered.

The top row shows that PRUDENT-BANKER incurs substantially less loss than SAFE-EXP3-IX, but more loss than CONSERVATIVE-UCB at the horizon. The middle row is mostly positive, so

PRUDENT-BANKER pays some extra cumulative loss relative to the smoothed best-arm comparator. However, the key qualitative point is that PRUDENT-BANKER does not remain permanently aggressive after its first transition. Delayed feedback causes the algorithm to restart conservatively and revalidate the evidence.

F.6.3 Geometric delays

The geometric-delay column has the largest realized total delay:

$$D = 3927.$$

Although only 3% of rounds activate a delay, the active delay length is geometric, so the backlog can be substantially larger than in the fixed-delay setting.

This is the clearest panel for the delay-calibrated behavior of PRUDENT-BANKER. The bottom row shows longer conservative periods and fewer aggressive intervals. As the realized delay budget grows, the threshold becomes more conservative, and the algorithm requires stronger evidence before increasing α_t.

The middle row is more favorable in this column than in the fixed-delay and Pareto-delay columns. For a substantial interval, it is negative, meaning that PRUDENT-BANKER temporarily has lower cumulative loss than the smoothed comparator. Later, after a sharp transition, the gap returns closer to zero. This behavior is consistent with the intended role of the delay correction: the algorithm does not overreact to incomplete feedback, but it can still exploit evidence when the evidence is strong enough after accounting for delay.

F.6.4 Pareto delays

The Pareto-delay column has realized total delay

$$D = 2065.$$

This is smaller than the geometric-delay total in the displayed run, but the delay distribution is heavy-tailed. A few longer delays can hide feedback from important periods, which can make the comparator-suboptimality test less stable.

The bottom row again shows several restarts of α_t. These restarts occur because the algorithm discovers that its current delay estimate is too small and therefore returns to a conservative phase. The top row shows that PRUDENT-BANKER improves over SAFE-EXP3-IX and ends below CONSERVATIVE-UCB relative to the hindsight best arm. The middle row is mostly negative, indicating less loss relative to the smoothed comparator.

F.6.5 Main takeaways

The experiment supports three qualitative conclusions.

First, The fixed one-step, geometric, and Pareto panels show that the delay-aware restart threshold affects the trajectory of α_t, not only the final cumulative loss

Second, the delay calibration is visible in the trajectory, not only in final regret. The bottom row shows the operational role of the delay-aware restart threshold: PRUDENT-BANKER does not treat incomplete feedback as definitive evidence. Instead, it restarts when the observable delay count exceeds its estimate and requires new evidence before becoming aggressive again.

Third, the baselines represent different behaviors. CONSERVATIVE-UCB is extremely conservative and performs well in the top-row metric because the default arm is the hindsight best fixed arm in this diagnostic. SAFE-EXP3-IX is adversarially motivated and delay-updated, but it explores much more broadly and therefore incurs larger loss relative to the best fixed arm. PRUDENT-BANKER attempts to balance these objectives: it stays close to the comparator when delay uncertainty is large, but it can increase aggression when the observed evidence is strong enough after delay calibration.

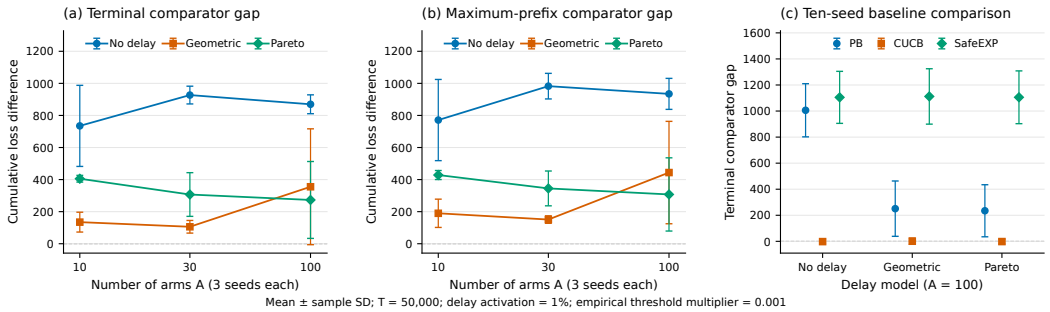


Figure 3: Arm-count and seed sensitivity. Left and center: three seeds per arm count; right: ten seeds at $A = 100$. Error bars show sample standard deviations. All runs use $T = 50000$, $\delta = .001$, and empirical threshold multiplier .001. CUCB can outperform Prudent-Banker under this comparator.

NeurIPS Paper Checklist

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

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

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

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

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

IMPORTANT, please:

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

1. **Claims**

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

   Answer: [Yes].

   Justification: The abstract and introduction state the main contributions: PRUDENT-BANKER for adversarial delayed bandits, near-minimax regret, near-constant regret relative

to a full-support safe comparator, and matching lower bounds up to logarithmic factors. These claims are supported by Theorem 4.1, Corollary 4.1, Theorem 4.2, and the detailed proof appendices.

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 discusses important limitations and assumptions, including the need for a full-support safe comparator, the dependence on the comparator margin δ, the inherited delay-side dependence from the Banker-OMD backbone, and the fact that experiments are synthetic and diagnostic rather than real-world deployments. These limitations are discussed in the problem setup, the arm-dependence discussion, and the experimental interpretation.

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 states the delayed adversarial bandit protocol, the oblivious-adversary assumption, the full-support comparator assumption, and the regularizer assumptions. Proof sketches are provided in the main text, while complete proofs of the main lemmas, theorem, corollary, and lower bound are deferred to Appendices C, D, E.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

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

Answer: [Yes].

Justification: The experimental setup specifies the horizon, number of arms, synthetic non-stationary loss generator, safe comparator construction, delay models, realized delay values corresponding to each plotted figure, baseline algorithms, hyperparameters, and evaluation metrics. Additional implementation-level details are given in Appendices F, F.5, F.6.

Guidelines:

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

some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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

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

   Answer: [No].

   Justification: The current manuscript provides detailed descriptions of the synthetic data generation process and algorithms, but it does not state that code or generated experimental data are publicly released. The experiments are nevertheless reproducible from the described synthetic generator and parameter settings.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes].

   Justification: The paper specifies the synthetic environment, horizon $T = 50{,}000$, number of arms $A = 100$, block structure, delay distributions, comparator margin $\delta = 0.001$, baseline parameters, and the PRUDENT-BANKER regularizer and constants used in the experiments. Since the setting is an online bandit experiment, there are no train/test splits.

   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: [No].

   Justification: The experimental results are presented as a diagnostic single-run comparison using the same realized loss table and delay sequence across algorithms. The current manuscript does not report error bars, confidence intervals, multiple seeds, or statistical significance tests.

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: [No].

   Justification: The current manuscript describes the experimental setup and algorithms but does not report hardware, memory usage, runtime per experiment, or total compute required. The experiments appear lightweight synthetic bandit simulations, but compute details are not explicitly provided.

   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 work is theoretical and uses synthetic experiments. It does not involve human subjects, private data, scraped datasets, sensitive personal information, or high-risk model release. To the best of our knowledge, the research conforms to 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: [No].

    Justification: The paper motivates the method through safety-critical sequential decision-making settings such as clinical decision-making, robotics, online advertising, and other delayed-feedback systems. However, the current manuscript does not include a dedicated broader-impact discussion covering both positive and negative societal impacts.

    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 does not release a high-risk model, pre-trained generative model, scraped dataset, or other artifact requiring controlled access or misuse safeguards.

    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 experiments use synthetically generated data rather than external datasets or models. Prior algorithms and theoretical frameworks used for comparison or inspiration are cited, but the current manuscript does not indicate use of existing code, datasets, or model assets requiring license disclosure.

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 current manuscript does not state that it releases new datasets, code packages, trained models, or benchmarks. The synthetic experimental setup is documented in the paper and appendices, but no standalone asset release is described.

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, user studies, human participants, or human-subject data collection.

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 human subjects, crowdsourcing, user studies, or human-subject data. Therefore, IRB approval or equivalent review is not applicable.

    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, theoretical analysis, lower bounds, and experiments do not involve LLMs as an important, original, or non-standard component. Any ordinary writing, editing, or formatting assistance, if used, would not affect the scientific methodology or originality of the work.

    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.