VANILLA POLICY OPTIMIZATION IS BOTH OPTIMAL AND DIFFERENTIALLY PRIVATE FOR STOCHASTIC CONTEXTUAL BANDITS

Anonymous authors
Paper under double-blind review

ABSTRACT

Can vanilla policy optimization explore enough to achieve near-optimal regret in stochastic contextual bandits? We show that standard exponential policy updates driven by offline regression do so under realizability, without exploration bonuses or importance weighting. For A actions, T rounds, and a finite prediction class $\mathcal{F}$, vanilla PO achieves $\widetilde{O}(\sqrt{AT\log(|\mathcal{F}|)})$ regret with high probability. Our analysis reveals an implicit exploration mechanism of independent interest: gradual policy updates prevent actions from losing probability too quickly, allowing the regression oracle to learn their expected losses. We further develop a batched version using only $O(\log T)$ regression calls and policy switches, and show how private regression oracles yield differentially private contextual bandit algorithms without composition across batches. For a finite class, this gives pure $\varepsilon_{\mathrm{priv}}$-DP and regret $\widetilde{O}\left(\sqrt{AT\log(|\mathcal{F}|/\delta)}(1+\varepsilon_{\mathrm{priv}}^{-1/2})\right)$. Finally, experiments across oracle-based contextual bandit algorithms, with and without privacy, demonstrate the practical effectiveness of policy optimization and the value of explicit exploration under stronger privacy constraints.

1 INTRODUCTION

Stochastic contextual multi-armed bandits (CMAB) are a fundamental and extensively studied model of sequential decision making under partial feedback (Agarwal et al., 2012; 2014; Foster et al., 2018; Simchi-Levi & Xu, 2022; Levy & Mansour, 2026). At each round, a learner observes a context, selects an action, and receives feedback only for that action. The goal is to achieve low cumulative loss while learning from this limited feedback. In the stochastic setting, contexts and their associated action losses are drawn independently across rounds from a fixed joint distribution.

Unlike in ordinary multi-armed bandits, the best action can vary with the context, so the learner must learn a policy mapping contexts to action distributions. Viewed as a reinforcement-learning problem with one decision per episode, CMAB retains the challenge of a potentially enormous or infinite state space. Since contexts may rarely repeat, learning separately at each context is ineffective and the learner must generalize across them. This makes CMAB substantially richer than learning which of a fixed set of arms has the lowest mean loss.

This contextual information is also central to the model's practical appeal, particularly for personalization. A context can describe a user's preferences, the available items, or the current task, allowing the learner to adapt its decision to the situation. Personalized news recommendation is a prominent example: user and article features guide which article to display, while feedback is observed only for the displayed article (Li et al., 2010). More broadly, contextual bandits provide a practical framework for adaptive decision systems (Agarwal et al., 2017; Bietti et al., 2021).

A central goal is to develop computationally efficient algorithms with strong statistical guarantees. Oracle-based reductions address this goal through offline supervised learning (Agarwal et al., 2014), particularly regression (Foster et al., 2018; Simchi-Levi & Xu, 2022; Xu & Zeevi, 2020). Related approaches reduce contextual bandit learning to online regression (Foster & Rakhlin, 2020).

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Policy optimization (PO) is a widely used approach to reinforcement learning, with practical successes exemplified by proximal policy optimization (Schulman et al., 2017). Its basic idea is to improve a policy directly by increasing the probability of actions predicted to perform well. Exponential policy updates also have a well-established theoretical foundation (Abbasi-Yadkori et al., 2019; Agarwal et al., 2021). Recently, Levy & Mansour (2026) showed that PO equipped with an exploration bonus achieves near-optimal regret for stochastic CMAB under realizability and offline regression assumptions. Here, function approximation provides the structure needed to generalize across contexts: the expected loss belongs to a known function class, and an offline regression procedure fits a predictor from the observed feedback (Foster et al., 2018; Simchi-Levi & Xu, 2022).

The exploration bonus encourages actions for which the learner has insufficient information, but computing it can be costly. In the method of Levy & Mansour (2026), evaluating the bonus at a new context requires reconstructing historical action probabilities from past predictors, which can become prohibitively expensive over long horizons. This raises the question of whether explicit bonuses are necessary at all: perhaps vanilla exponential updates, which simply favor actions with lower predicted losses, already provide sufficient exploration.

Returning to the personalization motivation, an effective learning algorithm must also protect the people whose data it uses. Contexts and feedback can contain sensitive information whose influence may persist in later predictors and policies. Differential privacy provides a formal way to limit this influence (Dwork et al., 2006; Dwork & Roth, 2014), and has been studied in contextual linear bandits (Shariff & Sheffet, 2018; Chen et al., 2025) and episodic reinforcement learning (Vietri et al., 2020). Closely related work studies the sample complexity of private PO, including methods based on private regression (He & Zhou, 2025). Recent work also establishes regret guarantees for private reinforcement learning with general function approximation (He & Zhou, 2026). We discuss how our approach differs from these works in its feedback model, regression requirements, and theoretical guarantees in Section A.

Our goal is to design efficient contextual bandit algorithms that simultaneously achieve strong regret guarantees, support general function approximation, and protect the data on which they learn. We seek algorithms that access the function class through a suitable private regression oracle. Vanilla policy optimization is a particularly appealing candidate for this approach: it is simple, computationally practical, and broadly compatible with modern regression methods.

Together, these considerations lead to our central questions: *(1) Can vanilla policy optimization achieve near-optimal regret in stochastic contextual bandits without explicit exploration bonuses? (2) Can it be made differentially private through private offline regression while retaining near-optimal regret?*

Our contributions.

1. **Near-optimal regret for vanilla PO (Section 3).** For A actions, T rounds, and a finite realizable function class $\mathcal{F}$, vanilla PO with least-squares regression achieves $\widetilde{O}(\sqrt{AT\log(|\mathcal{F}|/\delta)})$ regret with probability at least $1-\delta$. Our analysis reveals an implicit exploration mechanism: gradual updates preserve enough historical action probability that persistent suboptimal actions require substantial squared prediction error, which is controlled by the regression guarantee.
2. **Batched and private PO (Sections 4 and 5).** We develop batched versions of vanilla PO and the bonus-based method of Levy & Mansour (2026), retaining near-optimal regret with only $O(\log T)$ regression calls. Replacing regression by a private oracle protects the complete sequence of released predictors and policy maps, without a composition factor in the number of epochs. For finite realizable $\mathcal{F}$, the exponential mechanism yields pure $\varepsilon_{\text{priv}}$-DP and regret $\widetilde{O}\left(\sqrt{AT\log(|\mathcal{F}|/\delta)}(1+\varepsilon_{\text{priv}}^{-1/2})\right)$ with probability at least $1-\delta$. More generally, our guarantees accommodate realizable function classes through a suitable private regression oracle, beyond the linear setting studied under joint differential privacy by Shariff & Sheffet (2018).
3. **Private instantiations of other CMAB algorithms (Section 5).** The same privacy argument applies to other oracle-based CMAB algorithms that collect each batch under a fixed policy and use it in only one regression fit. When subsequent policies depend only on fitted predictors and public parameters, replacing the regression oracle with a private one yields

private counterparts while preserving the policy construction. We instantiate and evaluate these additional private algorithms in our experiments.

4. **Empirical evaluation (Section 6).** We compare vanilla PO, exploration-enhanced PO, and several private and non-private oracle-based CMAB algorithms under linear and logistic regression. The experiments show that the privatization scheme is broadly effective, that vanilla PO is a simple and competitive practical method, and that additional exploration can improve the privacy–utility tradeoff.

2 Contextual Multi-Armed Bandit Preliminaries

We consider a stochastic contextual bandit problem (Agarwal et al., 2012; Simchi-Levi & Xu, 2022) with context space $\mathcal{C}$ and a finite action set $\mathcal{A}$ of size $A \geq 2$. A policy $\pi : \mathcal{C} \to \Delta(\mathcal{A})$ maps each context to a distribution over actions. Let $\pi(c, a)$ denote the probability that $\pi(c)$ plays action a. The stochastic loss of action a for context c in round t is denoted by $\ell_t(c, a) \in [0, 1]$. We assume that the context–loss pairs $(c_t, \ell_t(c_t, \cdot))$ are drawn i.i.d. from a fixed unknown joint distribution, independently of the learner's randomness. We write $\mathcal{D}$ for its marginal distribution over contexts.

Let $\mathcal{H}_{t-1}$ denote the history of previous contexts, actions, and observed losses, together with any random choices already used by the learner. At each round $t \leq T$, the learner: (i) chooses a policy π_t based on $\mathcal{H}_{t-1}$; (ii) observes a fresh context $c_t \sim \mathcal{D}$; (iii) samples $a_t \sim \pi_t(c_t, \cdot)$; and (iv) observes only the loss $\ell_t(c_t, a_t)$ of the selected action.

Define the expected loss function by $f^\star(c, a) = \mathbb{E}[\ell_t(c_t, a) \mid c_t = c]$, for $c \in \mathcal{C}$ and $a \in \mathcal{A}$. This function does not depend on t, since the joint distribution is the same in every round. Let $\mathcal{F} \subseteq [0, 1]^{\mathcal{C} \times \mathcal{A}}$ be a known class of predictors.

Assumption 1 (Realizability)**.** The expected loss function belongs to the function class: $f^\star \in \mathcal{F}$.

Let $a^\star(c) \in \arg\min_{a \in \mathcal{A}} f^\star(c, a)$, with ties broken by a fixed ordering, and define the action gap $\Delta(c, a) = f^\star(c, a) - f^\star(c, a^\star(c))$. The expected one-round regret of a policy π is $\mathcal{R}(\pi) = \mathbb{E}_{c \sim \mathcal{D}}\big[\sum_{a \in \mathcal{A}} \pi(c, a)\Delta(c, a)\big]$. We measure performance by the cumulative regret

$$\overline{R}_T = \sum_{t=1}^{T} \sum_{a \in \mathcal{A}} \pi_t(c_t, a)\Delta(c_t, a) = \sum_{t=1}^{T} \mathop{\mathbb{E}}_{a_t \sim \pi_t(c_t, \cdot)} [\Delta(c_t, a_t) \mid \mathcal{H}_{t-1}, c_t]\,. \tag{1}$$

The expectation averages over the current action, with the history and current context held fixed. The observed contexts and learned policies remain random. Our analyses first bound $\sum_{t=1}^{T} \mathcal{R}(\pi_t)$, which averages each learned policy's regret over the context distribution $\mathcal{D}$. The concentration argument in Section B then yields high-probability bounds on $\overline{R}_T$, evaluated at the observed contexts.

Offline regression oracle. The learner estimates $f^\star$ using an offline regression oracle. For a policy π, define the mean squared prediction error $\|f - f^\star\|_\pi^2 = \mathbb{E}_{c \sim \mathcal{D}}\Big[\sum_{a \in \mathcal{A}} \pi(c, a)\big(f(c, a) - f^\star(c, a)\big)^2\Big]$. When the predictor is refitted after each round using all observations collected so far, we require the following accuracy guarantee under the policies used to collect those observations.

Assumption 2 (Regression-oracle accuracy)**.** Let $\widehat{f}_t : \mathcal{C} \times \mathcal{A} \to [0, 1]$ be the predictor returned by the oracle after round t. For every $\delta \in (0, 1)$, with probability at least $1 - \delta$, $\sum_{s=1}^{t} \|\widehat{f}_t - f^\star\|_{\pi_s}^2 \leq \mathrm{Err}_{\mathrm{orc}}(T, \delta)$ simultaneously for all $t \leq T$, where $\mathrm{Err}_{\mathrm{orc}}(T, \delta)$ is a known error bound fixed before the interaction.

Equivalently, $\widehat{f}_t$ has mean squared error at most $\mathrm{Err}_{\mathrm{orc}}(T, \delta)/t$ under the average of the policies used to collect its training data. For a finite class $\mathcal{F}$, least-squares empirical risk minimization (ERM), $\widehat{f}_t \in \arg\min_{f \in \mathcal{F}} \sum_{s=1}^{t} \big(f(c_s, a_s) - \ell_s(c_s, a_s)\big)^2$, satisfies Assumption 2 with $\mathrm{Err}_{\mathrm{orc}}(T, \delta) = O(\log(|\mathcal{F}|/\delta))$. The proof is in Section C.

Differential privacy definitions appear in Section 5. We use $\widetilde{O}(\cdot)$ to suppress logarithmic factors.

3 NEAR-OPTIMAL REGRET FOR VANILLA POLICY OPTIMIZATION

We consider the standard policy-optimization update based on exponential weights (Lattimore & Szepesvári, 2020; Slivkins, 2019; Agarwal et al., 2019). Starting from the uniform policy, at each round t the learner observes c_t, samples an action from $\pi_t(c_t, \cdot)$, and observes its loss. It then fits a predictor $\widehat{f}_t$ using all observations collected so far and defines π_{t+1} by exponentially downweighting actions with larger predicted losses. The algorithm has a simple structure and requires no additional computations for explicit exploration. Its batched implementation in Section 4 is also computationally lightweight. To our knowledge, vanilla PO is the first policy-optimization method shown to achieve near-optimal regret in stochastic contextual bandits without an explicit exploration bonus.

Algorithm 1 Vanilla Policy Optimization (VANILLAPO)

1: **Input:** T, $\mathcal{A}$, $\eta \in (0, 1]$, regression oracle.
2: Initialize $\pi_1(c, a) = 1/A$ for all $(c, a) \in \mathcal{C} \times \mathcal{A}$.
3: **for** $t = 1, \dots, T$ **do**
4: Observe c_t, sample $a_t \sim \pi_t(c_t, \cdot)$, and observe $\ell_t(c_t, a_t)$.
5: Fit $\widehat{f}_t \leftarrow \mathrm{Oracle}(\{(c_s, a_s, \ell_s(c_s, a_s))\}_{s=1}^{t})$.
6: For all $(c, a) \in \mathcal{C} \times \mathcal{A}$: $\pi_{t+1}(c, a) \propto \pi_t(c, a) \exp\{-\eta \widehat{f}_t(c, a)\}$.
7: **end for**

The algorithm enjoys near-optimal regret as stated next.

Theorem 3 (Regret of vanilla PO)**.** *Suppose the offline regression oracle satisfies Assumption 2. Fix $\delta \in (0, 1/2]$. Then vanilla PO with learning rate $\eta = \sqrt{(1 + \log(AT))/\big(AT(1 + \mathrm{Err}_{\mathrm{orc}}(T, \delta/2))\big)}$ satisfies, with probability at least $1 - \delta$, $\overline{R}_T = O\Big(\sqrt{AT\big(1 + \mathrm{Err}_{\mathrm{orc}}(T, \delta/2)\big)}\big(1 + \log(AT)\big)^{3/2} + \sqrt{T \log(2/\delta)}\Big)$. In particular, for a finite realizable class $\mathcal{F}$, using least-squares ERM gives $\overline{R}_T = \widetilde{O}\Big(\sqrt{AT \log(|\mathcal{F}|/\delta)}\Big)$.*

3.1 PROOF OVERVIEW

The proof of Theorem 3 hinges on understanding when a suboptimal action can continue to receive nonnegligible probability. The key insight is that, beyond a gap-dependent initial period, this can happen only if the cumulative squared prediction error is large relative to the action's gap. We use this connection to bound regret deterministically at each context, then average over contexts and apply the regression-oracle guarantee. The complete proof is given in Section D.

Fix a context c and a realized sequence of fitted predictors. We first analyze the resulting policies deterministically. Let $\bar{\pi}_t(c, a) = \frac{1}{t}\sum_{r=1}^{t} \pi_r(c, a)$ be the average of the first t policies, and define the cumulative squared prediction error by $S_t(c) = \sum_{r=1}^{t} \sum_{a \in \mathcal{A}} \bar{\pi}_r(c, a)\big(\widehat{f}_r(c, a) - f^\star(c, a)\big)^2$. After averaging over contexts, these weighted squared errors are controlled by Assumption 2.

Key lemma. The central ingredient is the *late-round probability–error dichotomy* in Lemma 13. The lemma states that, for every suboptimal action a and round t,

$$t - 1 \gtrsim \frac{\log(AT)}{\eta \Delta(c, a)} \quad \Longrightarrow \quad \left[\pi_t(c, a) < \frac{1}{AT} \quad \text{or} \quad \Delta(c, a) \lesssim A\eta S_{t-1}(c)\right]. \tag{2}$$

Thus, after an initial period that is shorter for larger gaps, an action either has negligible probability or its gap is controlled by the cumulative prediction error.

Why the dichotomy in (2) holds. For a suboptimal action a to retain probability at least $1/(AT)$ in a late round, the exponential update requires the predictors to substantially underestimate its loss gap in total. This can result from underestimating the loss of a or overestimating the loss of the optimal action $a^\star(c)$, so we need historical coverage of both actions. Suppose that $S_{t-1}(c)$ is smaller than a sufficiently small constant times $\Delta(c, a)/(A\eta)$. An induction shows that both actions then retain probability $\Omega(1/A)$ during the first $1/(\eta\Delta(c, a))$ updates. Earlier probability bounds control

accumulated errors in the predicted loss gaps, which in turn control the next policy. The argument bounds all action weights together, since they share the same normalizing denominator.

This initial exploration provides lasting historical coverage: the early probabilities remain in the average policies even after the current probabilities decrease. Prediction errors on both actions therefore receive enough weight in $S_{t-1}(c)$ for weighted Cauchy–Schwarz to control the accumulated error in their predicted loss gap. Under the assumed small-error condition, this accumulated error is too small to offset the true cumulative loss gap. This contradicts the underestimation required for action a to persist, establishing the large-error alternative in (2).

Regret decomposition. The dichotomy suggests separating action–round pairs into three regimes: small gaps, large gaps in early rounds, and large gaps in late rounds. Let $\tau(c)$ be a sufficiently large universal constant times $A\eta S_T(c)$, and call a gap small when $\Delta(c,a) \le \tau(c)$. For a suboptimal action, call a round late when the time condition in (2) holds, and early otherwise. Small gaps contribute at most $\tau(c)$ regret per round. For large gaps in early rounds, rearranging the definition of an early round gives $\Delta(c,a) \lesssim \frac{\log(AT)}{\eta(t-1)}$ when $t \ge 2$. Both contributions are bounded by summing action probabilities at each round, rather than paying a separate exploration cost for each action. For large gaps in late rounds, the choice of $\tau(c)$ and $S_{t-1}(c) \le S_T(c)$ rule out the large-error alternative in the dichotomy. Each such action therefore has probability below $1/(AT)$, and their combined regret is at most $1/T$. Consequently, for $t \ge 2$,

$$\sum_{a\in\mathcal{A}} \pi_t(c,a)\Delta(c,a) \lesssim \underbrace{A\eta S_T(c)}_{\text{small gap}} + \underbrace{\frac{\log(AT)}{\eta(t-1)}}_{\substack{\text{large gap}\\ \text{early round}}} + \underbrace{\frac{1}{T}}_{\substack{\text{large gap}\\ \text{late round}}}.$$

Let $R_T(c)$ denote the cumulative regret obtained by evaluating every policy in the learned sequence at the same fixed context c. Summing over rounds and bounding the first-round regret by one yields

$$R_T(c) := \sum_{t=1}^{T}\sum_{a\in\mathcal{A}} \pi_t(c,a)\Delta(c,a) \lesssim A\eta T S_T(c) + \frac{\log^2(AT)}{\eta}. \tag{3}$$

The oracle guarantee, applied with failure probability $\delta/2$, gives $\mathbb{E}_{c\sim\mathcal{D}}[S_T(c)] \le \mathrm{Err}_{\mathrm{orc}}(T,\delta/2)(1+\log T)$. Averaging (3) therefore bounds $\sum_{t=1}^{T}\mathcal{R}(\pi_t)$. A concentration argument transfers this bound to $\overline{R}_T$, adding $\sqrt{(T/2)\log(2/\delta)}$. Substituting the learning rate chosen using $\mathrm{Err}_{\mathrm{orc}}(T,\delta/2)$ and taking a union bound gives the regret guarantee with probability at least $1-\delta$.

4 Batched Implementation of Vanilla Policy Optimization

The vanilla PO implementation in the previous section updates the policy and calls the regression oracle after every round. While conceptually simple, this can be undesirable in practice: fitting the regression model may itself be computationally expensive, and changing a deployed policy may incur an additional switching or deployment cost. Differential privacy provides an important example, since repeatedly releasing updated predictors or policies may incur additional privacy loss. These considerations motivate a batched implementation of PO, in which the predictor and policy are updated only at a small number of epochs. To obtain that, we use geometrically growing batches. Within each epoch, the policy is fixed, and the data collected in that epoch are used once to fit the predictor for the next epoch. Consequently, the algorithm makes only $O(\log T)$ regression-oracle calls and $O(\log T)$ policy switches, while preserving the optimal regret guarantee.

Batched vanilla PO. Let $B_m = 2^{m-1}$ denote the length of epoch m. At the beginning of epoch $m \ge 2$, we fit a predictor $\widetilde{f}_m$ using only the data collected in epoch $m-1$. We then construct a single policy that remains fixed throughout epoch m. To define this policy, consider starting from the uniform distribution and repeatedly applying the exponential update using the same predictor $\widetilde{f}_m$. After k updates, the resulting distribution is proportional to $\exp\{-\eta_m k\widetilde{f}_m(c,a)\}$. We take the average of the distributions corresponding to $k = 0,\dots,B_m-1$:

$$\pi_m(c,a) = \frac{1}{B_m}\sum_{k=0}^{B_m-1} \frac{\exp\{-\eta_m k\widetilde{f}_m(c,a)\}}{\sum_{b\in\mathcal{A}}\exp\{-\eta_m k\widetilde{f}_m(c,b)\}}. \tag{4}$$

The early terms in this average spread probability relatively evenly across actions, whereas the later terms place increasingly more mass on actions with smaller predicted loss. Their average therefore preserves the exploration induced by the early exponential iterates while exploiting the predictions in the later ones. Only the final epoch may be truncated by the horizon. Its policy is still defined using all B_M mixture components in (4); the data collected in this final epoch are simply never used for another regression fit.

Algorithm 2 Batched Vanilla Policy Optimization

Input: horizon T, action set $\mathcal{A}$, learning rates η_m, and an offline regression oracle.
Set $M \leftarrow \lceil \log_2(T+1) \rceil$ and $B_m \leftarrow 2^{m-1}$ for $m = 1, \dots, M$.
Initialize $\pi_1(c, a) \leftarrow 1/A$ for every $c \in \mathcal{C}$ and $a \in \mathcal{A}$.
for $m = 1, \dots, M$ **do**
  if $m \geq 2$ **then**
    Fit $\widetilde{f}_m \leftarrow \mathrm{Oracle}(\mathcal{S}_{m-1})$ and discard $\mathcal{S}_{m-1}$.
    Define π_m by (4).
  end if
  Initialize $\mathcal{S}_m \leftarrow ()$.
  for $t = B_m, \dots, \min\{2B_m - 1, T\}$ **do**
    Observe c_t, sample $a_t \sim \pi_m(c_t, \cdot)$, and observe $\ell_t(c_t, a_t)$.
    Append $(c_t, a_t, \ell_t(c_t, a_t))$ to $\mathcal{S}_m$.
  end for
end for

Efficient implementation. Although (4) defines π_m for every context–action pair, it never needs to be constructed explicitly over the context space. At round t in epoch m, the learner evaluates $\widetilde{f}_m(c_t, a)$ only for the observed context c_t and the A actions. Moreover, sampling from the mixture in (4) does not require explicitly computing all B_m component distributions. Instead, independently sample $K_t \sim \mathrm{Unif}\{0, \dots, B_m - 1\}$ and then draw a_t according to

$$\mathbb{P}(a_t = a \mid c_t, K_t, \widetilde{f}_m) = \frac{\exp\{-\eta_m K_t \widetilde{f}_m(c_t, a)\}}{\sum_{b \in \mathcal{A}} \exp\{-\eta_m K_t \widetilde{f}_m(c_t, b)\}}. \tag{5}$$

Averaging over K_t recovers exactly $\pi_m(c_t, \cdot)$. Importantly, drawing K_t only samples a component of the fixed mixture; it does not constitute a policy update. Thus, each round requires only A predictor evaluations and $O(A)$ additional arithmetic, while the regression oracle and the deployed policy are updated only $O(\log T)$ times.

Regret guarantee. Because the policy is fixed within each epoch, conditional on the history at its start, the observations in that epoch are i.i.d. under a fixed policy. We can therefore apply a standard offline regression guarantee to each batch, as in FALCON+ (Simchi-Levi & Xu, 2022). Fix $\delta \in (0, 1/2]$ and suppose that, with probability at least $1 - \delta/2$, $\|\widetilde{f}_m - f^\star\|^2_{\pi_{m-1}} \leq \mathrm{Err}^{\mathrm{batch}}_{\mathrm{orc}}(T, \delta)/B_{m-1}$ simultaneously for all $m = 2, \dots, M$, where $\mathrm{Err}^{\mathrm{batch}}_{\mathrm{orc}}(T, \delta) \geq 1$ is a known error bound fixed before the interaction.

Theorem 4 (Regret of batched vanilla PO)**.** *Under the regression oracle guarantee above, batched vanilla PO with learning rates* $\eta_m = \Theta\left(\min\left\{1/A, 1/\sqrt{AB_m \,\mathrm{Err}^{\mathrm{batch}}_{\mathrm{orc}}(T, \delta)}\right\}\right)$ *uses* $O(\log T)$ *regression calls and* $O(\log T)$ *policy switches, and satisfies* $\overline{R}_T = \widetilde{O}\left(\sqrt{AT \;\mathrm{Err}^{\mathrm{batch}}_{\mathrm{orc}}(T, \delta)}\right)$ *with probability at least* $1 - \delta$*. In particular, for a finite realizable class* $\mathcal{F}$*, least-squares ERM with the corresponding error bound gives* $\overline{R}_T = \widetilde{O}\left(\sqrt{AT \log(|\mathcal{F}|/\delta)}\right)$.

The proof appears in Section E. Using each batch in at most one fit also enables the privacy guarantees developed next.

5 DIFFERENTIAL PRIVACY AND POLICY OPTIMIZATION

We now privatize batched PO by replacing each offline regression call with a differentially private regression oracle. Since the batches are disjoint and each batch is used in only one fit, the privacy cost does not accumulate across epochs. This allows us to protect the released predictors and policy maps while retaining the near-optimal regret guarantee of the batched algorithm.

Privacy model. Each round represents one user with record $z_t = (c_t, \ell_t(c_t, \cdot))$, consisting of a context and its potential-loss vector. Each user participates once. Two streams are neighboring if they differ in one record. Our privacy guarantee holds for arbitrary fixed streams, and the stochastic assumptions in Section 2 are needed only for regret.

Definition 5 (Differential privacy)**.** Let $\mathcal{M}$ denote the learning algorithm, whose public output on input stream z is $\mathcal{M}(z) = ((\widetilde{f}_m, \pi_m))_{m=2}^{M}$. Contexts, selected actions, observed losses, and policy evaluations at protected contexts remain internal. The algorithm $\mathcal{M}$ is $(\varepsilon_{\mathrm{priv}}, \delta_{\mathrm{priv}})$-differentially private if, for every pair of neighboring streams z, z' and every event E in the output space,

$$\mathbb{P}(\mathcal{M}(z) \in E) \le e^{\varepsilon_{\mathrm{priv}}} \mathbb{P}(\mathcal{M}(z') \in E) + \delta_{\mathrm{priv}}. \tag{6}$$

The probabilities are over the learner's randomness.

Assumption 6 (Private regression oracle)**.** Each oracle call uses fresh randomness and is $(\varepsilon_{\mathrm{priv}}, \delta_{\mathrm{priv}})$-DP under replacement of one row, for every pair of datasets of the same size. For every $\delta \in (0, 1/2]$, with probability at least $1 - \delta/2$, $\|\widetilde{f}_m - f^\star\|^2_{\pi_{m-1}} \le \mathrm{Err}^{\mathrm{batch}}_{\mathrm{orc}}(T, \delta)/B_{m-1}$ simultaneously for all $m = 2, \dots, M$, where $\mathrm{Err}^{\mathrm{batch}}_{\mathrm{orc}}(T, \delta) \ge 1$ is a known error bound. The function class, the error bound, and all tuning parameters are public and fixed before the interaction.

5.1 DIFFERENTIALLY PRIVATE BATCHED VANILLA PO

We instantiate Algorithm 2 with the private regression oracle of Assumption 6 and release each predictor and its policy map before the corresponding epoch. Each batch enters only one private fit, and subsequent releases depend on that batch only through the private output. Consequently, the complete transcript inherits the privacy parameters of a single oracle call.

Theorem 7 (Private batched vanilla PO)**.** *Under Assumption 6, batched vanilla PO releases an $(\varepsilon_{\mathrm{priv}}, \delta_{\mathrm{priv}})$-DP transcript using $O(\log T)$ private regression calls. With the learning rates of Theorem 4 chosen using the private oracle's error bound, it satisfies $\overline{R}_T = \widetilde{O}\left(\sqrt{AT\ \mathrm{Err}^{\mathrm{batch}}_{\mathrm{orc}}(T, \delta)}\right)$ with probability at least $1 - \delta$.*

For a finite function class, the exponential mechanism gives the following pure-DP instantiation.

Corollary 8 (Finite-class private regression)**.** *Let $\mathcal{F}$ be finite and realizable, fix $\delta \in (0, 1/2]$, and let $\varepsilon_{\mathrm{priv}} > 0$. Batched vanilla PO, instantiated with the exponential mechanism and the parameter choices specified in Section F.2, releases a pure $\varepsilon_{\mathrm{priv}}$-DP transcript and satisfies $\overline{R}_T = \widetilde{O}\left(\sqrt{AT \log(|\mathcal{F}|/\delta)}(1 + \varepsilon_{\mathrm{priv}}^{-1/2})\right)$ with probability at least $1 - \delta$.*

Relation to joint differential privacy. Joint differential privacy (JDP) allows a user's recommendation to depend on their own context while still "hiding" their data from the other users. This is achievable in our construction via the standard billboard model (Kearns et al., 2014; Hsu et al., 2014), in which users personalize their recommendation as a function of a public output (the "billboard") and their own private data. In particular, our finite-class exponential-mechanism instantiation gives pure JDP. A similar connection underlies Shariff & Sheffet (2018), who obtain approximate JDP for contextual linear bandits by maintaining differentially private regression statistics. Our main guarantee is stated directly for the released predictors and policy maps.

Proof overview. The regret and privacy arguments follow from two complementary properties of the batched construction. For regret, we simply invoke Theorem 4 with the prediction-error guarantee of the private regression oracle in Assumption 6; for a finite class, the exponential mechanism provides a concrete pure-DP instantiation whose excess empirical loss yields the required batch prediction-error bound, as shown in Lemma 12. For privacy, each record belongs to a single batch and enters the training data of at most one private oracle call. After that batch is discarded, all subsequent

predictors, policies, and released outputs depend on the record only through this private oracle output and are therefore postprocessing. Thus, the full transcript inherits the privacy guarantee of a single oracle call, with no composition cost across epochs. The full argument appears in Section F.

Extension to other batched algorithms. The same privacy argument applies to other oracle-based CMAB algorithms that collect each batch under a fixed policy and use it in only one regression fit. If subsequent policies depend only on the fitted predictors and public parameters, replacing the regression oracle with a private one protects the complete sequence of released predictors and policy maps with the privacy parameters of a single oracle call. We use this observation to construct private versions of several standard algorithms for the experiments in Section 6.

Batched bonus PO. We also develop a batched private adaptation of the bonus-based PO method of Levy & Mansour (2026), whose exploration bonus favors actions with little accumulated probability. The batched construction satisfies the conditions above and therefore inherits the same privacy guarantee. With the prescribed learning rates and bonus parameters, it satisfies $\overline{R}_T = \widetilde{O}\left(\sqrt{AT\ \mathrm{Err}_{\mathrm{orc}}^{\mathrm{batch}}(T,\delta)}\right)$ with probability at least $1-\delta$. The construction and regret analysis appear in Section G. For the experiments, we develop a faster approximation of this algorithm.

6 Experiments

Following standard practice in the CMAB literature (Bietti et al., 2021), we evaluate the algorithms on randomly permuted multiclass classification datasets from OpenML[1], treating class labels as actions, feature vectors as contexts, and the 0–1 classification error as the loss. We compare `VanillaPO` and `BonusPO` against the regression-based baselines `SquareCB` (Foster & Rakhlin, 2020), `FastCB` (Foster & Krishnamurthy, 2021), `RegCB` (Foster et al., 2018), and `AdaCB` (Foster et al., 2021), as well as `LinUCB` (Li et al., 2010; Shariff & Sheffet, 2018), together with the fully supervised baseline `Supervised`. We outline the experimental setup here and defer full implementation details and privacy accounting to Section H.

Implementation details. All algorithms follow the same doubling batch schedule, updating their policies at the end of each batch. The oracles fit regularized logistic or linear regression models to produce the action-loss predictors used in these updates. To ensure differential privacy of the predictors and, by post-processing, the resulting policies, we employ objective perturbation (Chaudhuri & Monteleoni, 2008; Chaudhuri et al., 2011) and sufficient-statistic perturbation (Wang, 2018). We note that a direct implementation of `BonusPO` incurs $O(T^2)$ computational cost. To make this tractable in our experiments, we use a historical-replay modification inspired by Levy & Mansour (2026): at each context, we replay bonus-based updates using the privatized predictors from previous batches, weighting each update by the corresponding batch size. Since the doubling schedule produces only $O(\log T)$ predictors, replay requires $O(T \log T)$ computation across all T rounds.

Experimental setup and evaluation. We evaluate the algorithms on 100 datasets with horizons $T \in [10^3, 10^5]$, allowing 9 to 16 policy updates per run. For brevity, throughout this section we write ε and δ for $\varepsilon_{\mathrm{priv}}$ and δ_{priv}, respectively. We consider four private regimes with $\varepsilon \in \{8, 4, 2, 1\}$ and $\delta = 10^{-5}$, together with a non-private regime. Performance is measured by progressive-validation (PV) loss, the average incurred loss up to round t: $L_{\mathrm{PV}}(t) = \frac{1}{t}\sum_{s=1}^{t} \ell_s(a_s)$.

Following previous work, we tune each algorithm independently for each dataset and privacy regime, selecting the hyperparameter configuration that minimizes the mean final PV loss. Table 1 lists the search grids. The reported results average ten runs with independently sampled dataset permutations and privacy perturbations, both shared across algorithms and privacy regimes within each run.

Discussion of results in Figure 1. At the strongest tested private regime, $(\varepsilon, \delta) = (1, 10^{-5})$, datasets 901 and 41027 exhibit a clear separation into three groups: `BonusPO` is the strongest bandit method and approaches the supervised reference, followed by `VanillaPO`, `SquareCB`, and `FastCB`, while `LinUCB`, `RegCB`, and `AdaCB` perform worse. On dataset 1489, the algorithms are closer together, although the same separation begins to emerge near the end. Section H shows that these patterns persist more broadly across 100 OpenML datasets.

[1] https://www.openml.org/

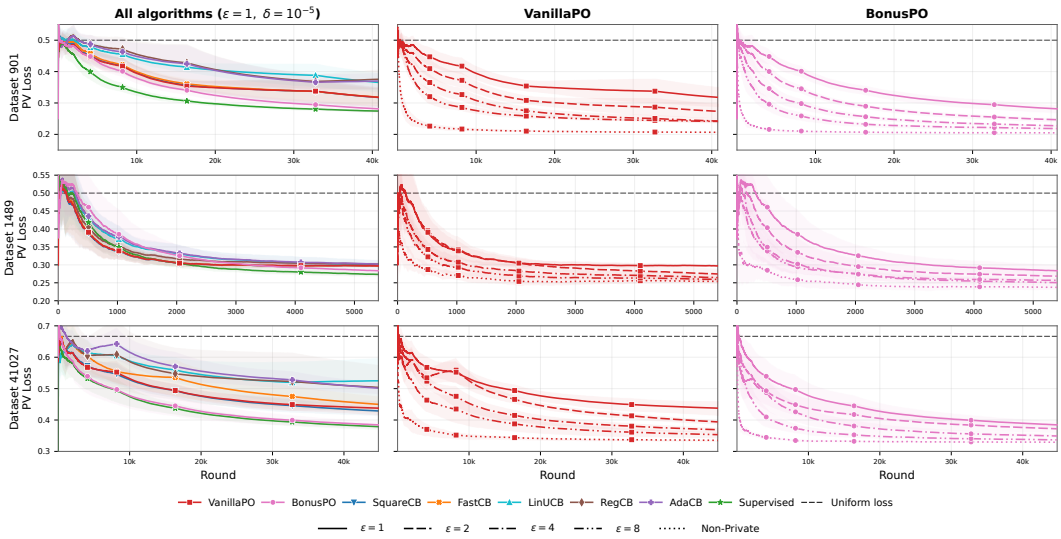


Figure 1: Progressive-validation loss on OpenML datasets 901, 1489, and 41027 (rows). The left column compares all algorithms at $(\varepsilon, \delta) = (1, 10^{-5})$, while the middle and right columns show `VanillaPO` and `BonusPO`, respectively, across privacy levels.

A plausible explanation for this behavior is that stronger privacy introduces more noise into the fitted action-loss predictors. The exponential updates of `VanillaPO` and `BonusPO` change the policy gradually in response to these predictors, whereas the baseline methods translate each updated fit more directly into their action probabilities and can therefore react more strongly to this noise. `BonusPO` additionally benefits from its exploration bonus, which favors actions that have received little past probability and provides additional robustness when the fitted losses are inaccurate.

The remaining panels show that `VanillaPO` and `BonusPO` degrade gracefully as privacy becomes stronger. In Section H, we provide the full privacy-regime and algorithm-wise plots for these three datasets, allowing the same trends to be compared across all methods and privacy regimes.

7 Discussion

Our results establish theoretical guarantees for one of the simplest forms of policy optimization, an approach widely used in practice. Vanilla PO achieves near-optimal regret in stochastic contextual bandits without explicit exploration bonuses under realizability and using an offline regression oracle. Gradual updates preserve exploration during the initial rounds, helping the regression oracle correct predictions that make suboptimal actions appear better than they are. Batching makes this approach attractive for both computation and privacy. The batched algorithms use $O(\log T)$ regression calls and policy switches. Fitting each batch once with a private oracle protects the released predictors and policy maps without additional privacy cost across epochs. Regret and computational efficiency depend on the oracle's accuracy and runtime, respectively.

The experiments suggest that exploration bonuses can remain useful even when they are unnecessary for the regret guarantee. The historical-replay variant of bonus PO performs strongly in our comparisons, including under privacy constraints. These findings motivate understanding when additional exploration improves robustness to private prediction errors.

Our guarantees rely on stochastic contexts, realizability, and exact policy updates. A natural direction is to investigate whether the principles underlying our exploration and privacy analyses can guide algorithm design in more complex interactive settings, such as reinforcement learning. Such extensions may require modified policy updates or additional exploration. Another direction is to improve the privacy–utility tradeoff within our regression-oracle framework.

AI USE STATEMENT

Generative AI assistance was used to check and develop mathematical arguments, verify references, and draft and revise the exposition. The authors are responsible for the final content.

REFERENCES

Yasin Abbasi-Yadkori, Peter Bartlett, Kush Bhatia, Nevena Lazic, Csaba Szepesvari, and Gellért Weisz. POLITEX: Regret bounds for policy iteration using expert prediction. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 3692–3702. PMLR, 2019.

Alekh Agarwal, Miroslav Dudík, Satyen Kale, John Langford, and Robert Schapire. Contextual bandit learning with predictable rewards. In *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics*, volume 22 of *Proceedings of Machine Learning Research*, pp. 19–26. PMLR, 2012.

Alekh Agarwal, Daniel Hsu, Satyen Kale, John Langford, Lihong Li, and Robert Schapire. Taming the monster: A fast and simple algorithm for contextual bandits. In *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pp. 1638–1646. PMLR, 2014.

Alekh Agarwal, Sarah Bird, Markus Cozowicz, Luong Hoang, John Langford, Stephen Lee, Jiaji Li, Dan Melamed, Gal Oshri, Oswaldo Ribas, Siddhartha Sen, and Alex Slivkins. Making contextual decisions with low technical debt. *arXiv preprint arXiv:1606.03966*, 2017.

Alekh Agarwal, Nan Jiang, Sham M Kakade, and Wen Sun. Reinforcement learning: Theory and algorithms. *CS Dept., UW Seattle, Seattle, WA, USA, Tech. Rep*, 32:96, 2019.

Alekh Agarwal, Sham M Kakade, Jason D Lee, and Gaurav Mahajan. On the theory of policy gradient methods: Optimality, approximation, and distribution shift. *Journal of Machine Learning Research*, 22(98):1–76, 2021.

Raef Bassily, Adam Smith, and Abhradeep Thakurta. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *2014 IEEE 55th Annual Symposium on Foundations of Computer Science*, pp. 464–473. IEEE, 2014.

Hamsa Bastani, Mohsen Bayati, and Khashayar Khosravi. Mostly exploration-free algorithms for contextual bandits. *Management Science*, 67(3):1329–1349, 2021.

Alberto Bietti, Alekh Agarwal, and John Langford. A contextual bandit bake-off. *Journal of Machine Learning Research*, 22(133):1–49, 2021.

Mark Bun and Thomas Steinke. Concentrated differential privacy: Simplifications, extensions, and lower bounds. In *Theory of Cryptography*, volume 9985 of *Lecture Notes in Computer Science*, pp. 635–658. Springer, 2016.

Qi Cai, Zhuoran Yang, Chi Jin, and Zhaoran Wang. Provably efficient exploration in policy optimization. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 1283–1294. PMLR, 2020.

Nicolò Cesa-Bianchi, Claudio Gentile, Gabor Lugosi, and Gergely Neu. Boltzmann exploration done right. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Kamalika Chaudhuri and Claire Monteleoni. Privacy-preserving logistic regression. In *Advances in Neural Information Processing Systems*, volume 21, pp. 289–296, 2008.

Kamalika Chaudhuri, Claire Monteleoni, and Anand D. Sarwate. Differentially private empirical risk minimization. *Journal of Machine Learning Research*, 12(29):1069–1109, 2011.

Fan Chen and Alexander Rakhlin. Decision making in changing environments: Robustness, query-based learning, and differential privacy. In *Proceedings of Thirty Eighth Conference on Learning Theory*, volume 291 of *Proceedings of Machine Learning Research*, pp. 983–985. PMLR, 2025.

Fan Chen, Jiachun Li, Alexander Rakhlin, and David Simchi-Levi. Beyond covariance matrix: The statistical complexity of private linear regression. *arXiv preprint arXiv:2502.13115*, 2025.

Sayak Ray Chowdhury and Xingyu Zhou. Differentially private regret minimization in episodic Markov decision processes. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(6): 6375–6383, 2022a.

Sayak Ray Chowdhury and Xingyu Zhou. Shuffle private linear contextual bandits. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 3984–4009. PMLR, 2022b.

Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3–4):211–407, 2014.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography*, volume 3876 of *Lecture Notes in Computer Science*, pp. 265–284. Springer, 2006.

Dylan Foster and Alexander Rakhlin. Beyond UCB: Optimal and efficient contextual bandits with regression oracles. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 3199–3210. PMLR, 2020.

Dylan Foster, Alekh Agarwal, Miroslav Dudik, Haipeng Luo, and Robert Schapire. Practical contextual bandits with regression oracles. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1539–1548. PMLR, 2018.

Dylan Foster, Alexander Rakhlin, David Simchi-Levi, and Yunzong Xu. Instance-dependent complexity of contextual bandits and reinforcement learning: A disagreement-based perspective. In *Proceedings of the 34th Annual Conference on Learning Theory*, volume 134 of *Proceedings of Machine Learning Research*, pp. 2059–2059. PMLR, 2021.

Dylan J. Foster and Akshay Krishnamurthy. Efficient first-order contextual bandits: Prediction, allocation, and triangular discrimination. In *Advances in Neural Information Processing Systems*, volume 34, pp. 18907–18919, 2021.

Evrard Garcelon, Vianney Perchet, Ciara Pike-Burke, and Matteo Pirotta. Local differential privacy for regret minimization in reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 34, pp. 10561–10573, 2021.

Evrard Garcelon, Kamalika Chaudhuri, Vianney Perchet, and Matteo Pirotta. Privacy amplification via shuffling for linear contextual bandits. In *Proceedings of the 33rd International Conference on Algorithmic Learning Theory*, volume 167 of *Proceedings of Machine Learning Research*, pp. 381–407. PMLR, 2022.

Yuxuan Han, Zhipeng Liang, Yang Wang, and Jiheng Zhang. Generalized linear bandits with local differential privacy. In *Advances in Neural Information Processing Systems*, volume 34, pp. 26511–26522, 2021.

Osama Hanna, Antonious M. Girgis, Christina Fragouli, and Suhas Diggavi. Differentially private stochastic linear bandits: (almost) for free. *IEEE Journal on Selected Areas in Information Theory*, 5:135–147, 2024.

Yi He and Xingyu Zhou. On the sample complexity of differentially private policy optimization. In *Advances in Neural Information Processing Systems*, volume 38, pp. 11181–11217. Curran Associates, Inc., 2025.

Yi He and Xingyu Zhou. Towards differentially private reinforcement learning with general function approximation. *arXiv preprint arXiv:2605.07049*, 2026.

Steven R. Howard, Aaditya Ramdas, Jon McAuliffe, and Jasjeet Sekhon. Time-uniform Chernoff bounds via nonnegative supermartingales. *Probability Surveys*, 17:257–317, 2020.

Justin Hsu, Zhiyi Huang, Aaron Roth, Tim Roughgarden, and Zhiwei Steven Wu. Private matchings and allocations. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, pp. 21–30, 2014.

Sampath Kannan, Jamie H. Morgenstern, Aaron Roth, Bo Waggoner, and Zhiwei Steven Wu. A smoothed analysis of the greedy algorithm for the linear contextual bandit problem. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Michael Kearns, Mallesh M. Pai, Aaron Roth, and Jonathan Ullman. Mechanism design in large games: Incentives and privacy. In *Proceedings of the 5th Conference on Innovations in Theoretical Computer Science*, pp. 403–410, 2014.

Sanath Kumar Krishnamurthy, Vitor Hadad, and Susan Athey. Adapting to misspecification in contextual bandits with offline regression oracles. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 5805–5814. PMLR, 2021.

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

Orin Levy and Yishay Mansour. Optimal regret for policy optimization in contextual bandits. *arXiv preprint arXiv:2602.13700*, 2026.

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*, pp. 661–670. ACM, 2010.

Dong C. Liu and Jorge Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, 45:503–528, 1989.

Haipeng Luo, Chen-Yu Wei, and Chung-Wei Lee. Policy optimization in adversarial MDPs: Improved exploration via dilated bonuses. In *Advances in Neural Information Processing Systems*, volume 34, pp. 22931–22942, 2021.

Nikola Pavlovic, Sudeep Salgia, and Qing Zhao. Differentially private kernelized contextual bandits. In *Proceedings of the 28th International Conference on Artificial Intelligence and Statistics*, volume 258 of *Proceedings of Machine Learning Research*, pp. 4618–4626. PMLR, 2025a.

Nikola Pavlovic, Sudeep Salgia, and Qing Zhao. Differential privacy in kernelized contextual bandits via random projections. *arXiv preprint arXiv:2507.13639*, 2025b.

Dan Qiao and Yu-Xiang Wang. Near-optimal differentially private reinforcement learning. In *Proceedings of the 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pp. 9914–9940. PMLR, 2023.

Hao Qin and Chicheng Zhang. Taming the monster every context: Complexity measure and unified framework for offline-oracle efficient contextual bandits. In *Proceedings of Thirty Ninth Conference on Learning Theory*, volume 336 of *Proceedings of Machine Learning Research*, pp. 5399–5464. PMLR, 2026.

Touqir Sajed and Or Sheffet. An optimal private stochastic-MAB algorithm based on optimal private stopping rule. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 5579–5588. PMLR, 2019.

Sahasrajit Sarmasarkar. Shuffle and joint differential privacy for generalized linear contextual bandits. *arXiv preprint arXiv:2602.00417*, 2026.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Lior Shani, Yonathan Efroni, Aviv Rosenberg, and Shie Mannor. Optimistic policy optimization with bandit feedback. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 8604–8613. PMLR, 2020.

Roshan Shariff and Or Sheffet. Differentially private contextual linear bandits. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

David Simchi-Levi and Yunzong Xu. Bypassing the monster: A faster and simpler optimal algorithm for contextual bandits under realizability. *Mathematics of Operations Research*, 47(3):1904–1931, 2022.

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

Jay Tenenbaum, Haim Kaplan, Yishay Mansour, and Uri Stemmer. Concurrent shuffle differential privacy under continual observation. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 33961–33982. PMLR, 2023.

Giuseppe Vietri, Borja Balle, Akshay Krishnamurthy, and Steven Wu. Private reinforcement learning with PAC and regret guarantees. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 9754–9764. PMLR, 2020.

Yu-Xiang Wang. Revisiting differentially private linear regression: Optimal and adaptive prediction & estimation in unbounded domain. In *Proceedings of the 34th Conference on Uncertainty in Artificial Intelligence*, pp. 93–103. AUAI Press, 2018.

Yunbei Xu and Assaf Zeevi. Upper counterfactual confidence bounds: A new optimism principle for contextual bandits. *arXiv preprint arXiv:2007.07876*, 2020.

Kai Zheng, Tianle Cai, Weiran Huang, Zhenguo Li, and Liwei Wang. Locally differentially private (contextual) bandits learning. In *Advances in Neural Information Processing Systems*, volume 33, pp. 12300–12310. Curran Associates, Inc., 2020.

Xingyu Zhou. Differentially private reinforcement learning with linear function approximation. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 6(1):1–27, 2022.

A Additional related work

Contextual bandits with optimization oracles. Oracle reductions separate exploration from the computational task of learning a predictor or policy. Agarwal et al. (2014) obtain near-optimal regret using a cost-sensitive classification oracle and optimization over policy distributions. Under realizability, Foster et al. (2018) develop regression-based exploration methods, while Simchi-Levi & Xu (2022) obtain optimal rates through inverse-gap sampling. Their FALCON algorithm uses finite-class least-squares regression; FALCON+ accommodates a general offline regression oracle trained on the preceding epoch. Our batched methods share this fixed-policy data-collection structure, but construct the next policy through averaged exponential updates. Xu & Zeevi (2020) develop counterfactual confidence bounds. Further work studies robustness to misspecification (Krishnamurthy et al., 2021) and offline-oracle reductions for large action spaces (Qin & Zhang, 2026). The empirical study of Bietti et al. (2021) motivates comparisons across oracle-based methods beyond their worst-case regret guarantees.

Online regression provides a complementary interface. SquareCB (Foster & Rakhlin, 2020) reduces contextual bandits to online square-loss regression and handles adversarial contexts under realizability, with a guarantee expressed through the online oracle's regret. Our vanilla result instead uses an offline regression guarantee averaged over past data-collection policies. The common stochastic context distribution connects this guarantee to the prediction errors affecting future policy updates. The distinction therefore concerns both the oracle interface and the context model.

Exploration in policy optimization. The empirical success of PO, exemplified by PPO (Schulman et al., 2017), is complemented by theoretical analyses of exponential policy updates and natural policy-gradient methods (Abbasi-Yadkori et al., 2019; Agarwal et al., 2021). Optimistic PO algorithms obtain regret guarantees in tabular and structured Markov decision processes (Shani et al., 2020; Cai et al., 2020), including adversarial settings through exploration bonuses (Luo et al., 2021). For stochastic contextual bandits with general offline function approximation, Levy & Mansour (2026) prove near-optimal regret for PO with a bonus based on accumulated action probabilities. Our first result establishes that the vanilla exponential update already suffices under stochastic contexts, realizability, and the stated regression-oracle guarantee. Our bonus-based result is a batched private adaptation of their method. These results concern the contextual-bandit setting; bonuses can still improve empirical performance, as our experiments illustrate.

Exploration supplied by the context distribution. Prior work shows that greedy contextual-bandit algorithms can succeed under distributional conditions such as covariate diversity or smoothed contexts (Bastani et al., 2021; Kannan et al., 2018). Our algorithm uses gradual exponential updates rather than greedy action selection, and its guarantee does not require these additional conditions on the context distribution. It also differs from Boltzmann exploration based on current empirical arm means, whose limitations are analyzed by Cesa-Bianchi et al. (2017): unrolling our updates gives weights based on the sum of successive fitted predictors.

Private bandits and privacy models. Sajed & Sheffet (2019) study private stochastic multi-armed bandits using private stopping rules and successive elimination. Hanna et al. (2024) study stochastic linear bandits under central, local, and shuffle privacy. For contextual bandits, Shariff & Sheffet (2018) establish regret guarantees for linear models under joint differential privacy, and Chen et al. (2025) connect private linear regression to private contextual-bandit learning. Local privacy for contextual bandits, including generalized linear models, is studied by Zheng et al. (2020); Han et al. (2021). Shuffle-private linear contextual bandits are studied by Garcelon et al. (2022); Chowdhury & Zhou (2022b). Tenenbaum et al. (2023) introduce the concurrent shuffle model, obtain improved guarantees for continual summation, and apply them to contextual linear bandits. Sarmasarkar (2026) studies shuffle and joint privacy for generalized linear contextual bandits. Kernel methods provide another extension beyond linear models: Pavlovic et al. (2025a) study private policy learning in kernelized contextual bandits, and Pavlovic et al. (2025b) establish cumulative-regret guarantees using private kernel regression.

Private reinforcement learning. Vietri et al. (2020) establish PAC and regret guarantees for private tabular reinforcement learning, in the fundamental model of tabular Markov Decision Process

(MDP). Later, Garcelon et al. (2021) study regret minimization under local privacy in MDPs, protecting trajectories on the user's side. Chowdhury & Zhou (2022a) develop private policy-optimization and value-iteration methods for episodic tabular MDPs under joint and local privacy, while Qiao & Wang (2023) obtain improved regret guarantees in these models. Zhou (2022) studies private exploration with linear function approximation in linear-mixture MDPs. The policy-optimization for MDPs methods of Chowdhury & Zhou (2022a); Zhou (2022) use optimistic value estimates, including exploration bonuses. Beyond particular parametric models, Chen & Rakhlin (2025) develop a general framework for decision making that includes local differential privacy. He & Zhou (2026) study online reinforcement learning in finite-horizon episodic MDPs with general value-function approximation under joint differential privacy. Their approach combines batched policy updates with the exponential mechanism and achieves $\widetilde{O}(T^{3/5})$ dependence on the number of episodes T, with additional dependence on coverability, function-class complexity, the horizon, and privacy parameters. In contrast, we study realizable stochastic contextual bandits and analyze vanilla policy optimization, whose policy updates rely solely on exponential weighting of predicted losses. We obtain near-optimal $\widetilde{O}(\sqrt{T})$ dependence on the number of rounds T, without coverability assumptions or additional structural complexity measures: the dependence on the loss-prediction class is captured entirely by its regression-learning complexity. We also consider a different privacy model, requiring differential privacy of the complete released transcript of predictors and policy maps rather than joint differential privacy.

Private PO through regression. Especially close to our approach, He & Zhou (2025) analyze private PG, NPG, and REBEL, including reductions to private regression using fresh batches. We share their principle of using each batch in one private update. Their guarantees include sample-complexity and average policy-suboptimality bounds. For NPG, regression estimates advantages using policy-gradient features, and the guarantee depends on coverage by a base policy; their sampling procedure obtains feedback for two actions at each context. Our algorithms instead fit expected losses from one observed action loss per arriving context. We analyze how the resulting batched policies provide sufficient exploration to control cumulative regret over all interactions, under realizability and the stated regression-oracle guarantee.

Scope of our privacy and computational guarantees. Our guarantee is central differential privacy for the complete sequence of released predictors and policy maps under replacement of one user's context and potential-loss vector. Actions and evaluations at protected contexts remain internal. This release convention differs from joint privacy for user-specific actions (Shariff & Sheffet, 2018; Vietri et al., 2020), local privacy for individual messages, and privacy for shuffled messages. Comparisons must therefore account for the protected records, observable outputs, and trust assumptions. Our dependence on the prediction class enters through the private regression oracle's accuracy. Private empirical risk minimization provides relevant computational tools under suitable conditions on the loss and parameter space (Chaudhuri et al., 2011; Bassily et al., 2014). Computational efficiency additionally requires an efficient implementation of the oracle; in particular, our finite-class exponential-mechanism instantiation does not assume efficient sampling over an arbitrary function class.

B Concentration of pseudo-regret

Our regret analyses control $\sum_{t=1}^{T} \mathcal{R}(\pi_t)$, which evaluates each learned policy under the context distribution $\mathcal{D}$. The regret $\overline{R}_T$ instead evaluates each policy at the context observed in that round, while still averaging over the current action. The policies remain random in both quantities. The following proposition bounds their difference.

Proposition 9. *Suppose that each policy π_t is chosen using $\mathcal{H}_{t-1}$, as in Section 2. For every $\delta \in (0,1)$, with probability at least $1-\delta$,*

$$\overline{R}_T \le \sum_{t=1}^{T} \mathcal{R}(\pi_t) + \sqrt{\frac{T}{2} \log \frac{1}{\delta}}.$$

Proof. Conditional on $\mathcal{H}_{t-1}$, the policy π_t is fixed and the fresh context c_t has distribution $\mathcal{D}$. Therefore,

$$\mathbb{E}\left[\sum_{a\in\mathcal{A}} \pi_t(c_t,a)\Delta(c_t,a) \,\middle|\, \mathcal{H}_{t-1}\right] = \mathcal{R}(\pi_t).$$

Since every gap lies in $[0,1]$ and the action probabilities sum to one,

$$0 \le \sum_{a\in\mathcal{A}} \pi_t(c_t,a)\Delta(c_t,a) \le 1.$$

Conditional Hoeffding's lemma thus gives, for every $\lambda > 0$,

$$\mathbb{E}\left[\exp\left\{\lambda\left(\sum_{a\in\mathcal{A}} \pi_t(c_t,a)\Delta(c_t,a) - \mathcal{R}(\pi_t)\right)\right\} \,\middle|\, \mathcal{H}_{t-1}\right] \le e^{\lambda^2/8}.$$

Iterating this inequality over rounds yields

$$\mathbb{E}\left[\exp\left\{\lambda\left(\overline{R}_T - \sum_{t=1}^{T} \mathcal{R}(\pi_t)\right)\right\}\right] \le e^{T\lambda^2/8}.$$

By exponential Markov's inequality, for every $u > 0$,

$$\mathbb{P}\left(\overline{R}_T - \sum_{t=1}^{T} \mathcal{R}(\pi_t) \ge u\right) \le \inf_{\lambda>0} \exp\left(-\lambda u + \frac{T\lambda^2}{8}\right) = e^{-2u^2/T},$$

where the minimum is attained at $\lambda = 4u/T$. Taking $u = \sqrt{(T/2)\log(1/\delta)}$ proves the claim. □

To combine this proposition with a high-probability bound on $\sum_{t=1}^{T} \mathcal{R}(\pi_t)$, apply each result with failure probability $\delta/2$ and take a union bound. The resulting bound on $\overline{R}_T$ holds with probability at least $1-\delta$ and adds $\sqrt{(T/2)\log(2/\delta)}$ to the bound obtained at confidence level $1-\delta/2$. This additional term does not change the stated finite-class regret rates in $\widetilde{O}$ notation.

These regret quantities are internal performance measures and are not part of the private algorithms' released transcript.

C Finite-class regression guarantees

We establish the regression guarantees used by vanilla PO and its batched variants. The first result handles adaptive action selection and holds simultaneously over all rounds. The remaining results concern a single batch collected under a fixed policy.

C.1 Least-squares ERM under adaptive data collection

Lemma 10 (Least-squares ERM under adaptive data collection)**.** *In the model of Section 2, suppose that $\mathcal{F}$ is finite and $f^\star \in \mathcal{F}$. For every round t, let $\widehat{f}_t$ minimize*

$$\sum_{r=1}^{t} \big(f(c_r,a_r) - \ell_r(c_r,a_r)\big)^2$$

over $f \in \mathcal{F}$, with ties broken by a fixed ordering. Then, for every $\delta \in (0,1)$, Assumption 2 holds with

$$\mathrm{Err}_{\mathrm{orc}}(T,\delta) = 24\log\frac{|\mathcal{F}|}{\delta}.$$

Proof. Fix $f \in \mathcal{F}$ and define its excess squared loss at round r by

$$Z_r(f) = \big(f(c_r,a_r) - \ell_r(c_r,a_r)\big)^2 - \big(f^\star(c_r,a_r) - \ell_r(c_r,a_r)\big)^2.$$

Conditional on $\mathcal{H}_{r-1}$, the policy π_r is fixed. The current context–loss pair is independent of this history and follows the distribution specified in Section 2. Moreover, the action is sampled from $\pi_r(c_r, \cdot)$ using fresh randomness, without observing the current losses. For each fixed action, realizability implies that the expected excess squared loss at context c is $\big(f(c,a) - f^\star(c,a)\big)^2$: expanding the squares cancels the term involving the centered loss. Averaging over the action and context therefore gives

$$\begin{aligned}\nu_r(f) &:= \mathbb{E}[Z_r(f) \mid \mathcal{H}_{r-1}] \\ &= \mathbb{E}_{c\sim\mathcal{D}}\left[\sum_{a\in\mathcal{A}} \pi_r(c,a)\big(f(c,a) - f^\star(c,a)\big)^2\right] = \|f - f^\star\|^2_{\pi_r}.\end{aligned}$$

Let $V_t(f) = \sum_{r=1}^t \nu_r(f)$.

Since losses and predictions lie in $[0,1]$, we have $|Z_r(f)| \le 1$. Factoring the difference of squares also gives

$$\begin{aligned}Z_r(f)^2 &= \big(f(c_r,a_r) - f^\star(c_r,a_r)\big)^2\big(f(c_r,a_r) + f^\star(c_r,a_r) - 2\ell_r(c_r,a_r)\big)^2 \\ &\le 4\big(f(c_r,a_r) - f^\star(c_r,a_r)\big)^2.\end{aligned}$$

Thus $X_r(f) = \nu_r(f) - Z_r(f)$ satisfies

$$\mathbb{E}[X_r(f) \mid \mathcal{H}_{r-1}] = 0, \qquad |X_r(f)| \le 2, \qquad \mathbb{E}[X_r(f)^2 \mid \mathcal{H}_{r-1}] \le 4\nu_r(f).$$

Take $\lambda = 1/12$. The conditional Bernstein moment bound yields

$$\log \mathbb{E}[e^{\lambda X_r(f)} \mid \mathcal{H}_{r-1}] \le \frac{\lambda^2 \mathbb{E}[X_r(f)^2 \mid \mathcal{H}_{r-1}]}{2(1 - 2\lambda/3)} \le \frac{\lambda}{2}\nu_r(f).$$

Consequently,

$$M_t(f) = \exp\left\{\lambda\left(\frac{1}{2}V_t(f) - \sum_{r=1}^t Z_r(f)\right)\right\}$$

is a nonnegative supermartingale with $M_0(f) = 1$. By Ville's inequality, for every $x > 0$, with probability at least $1 - e^{-x}$, simultaneously for all $t \le T$,

$$\sum_{r=1}^t Z_r(f) \ge \frac{1}{2}V_t(f) - 12x. \tag{7}$$

Indeed, a violation would imply that $M_t(f) > e^x$ at some round, whose probability is at most e^{-x}; see Howard et al. (2020).

Set $x = \log(|\mathcal{F}|/\delta)$ and take a union bound over $f \in \mathcal{F}$. With probability at least $1 - \delta$, (7) holds for every $f \in \mathcal{F}$ and every $t \le T$. On this event, it applies to the data-dependent choice $f = \widehat{f}_t$. The ERM property and $f^\star \in \mathcal{F}$ imply

$$0 \ge \sum_{r=1}^t Z_r(\widehat{f}_t) \ge \frac{1}{2}V_t(\widehat{f}_t) - 12x.$$

Therefore,

$$\sum_{r=1}^t \|\widehat{f}_t - f^\star\|^2_{\pi_r} = V_t(\widehat{f}_t) \le 24 \log \frac{|\mathcal{F}|}{\delta}$$

simultaneously for all $t \le T$. This proves Assumption 2 without a union bound over rounds. □

C.2 Regression on a fixed batch and approximate ERM

Fix a policy π. Each observation is generated by drawing a fresh context–loss pair in the model of Section 2, sampling an action from π using independent randomness, and observing its loss. For an ordered batch $\mathcal{S} = ((c_i, a_i, y_i))_{i=1}^n$ of such observations, define

$$L_{\mathcal{S}}(f) = \sum_{i=1}^n \big(f(c_i,a_i) - y_i\big)^2.$$

The following result allows the fitted predictor to approximately minimize the empirical squared loss. This covers both exact ERM and the private oracle analyzed below.

Lemma 11 (Accuracy of approximate ERM on a batch)**.** *Let $\mathcal{S}$ contain $n \geq 1$ i.i.d. observations generated under a fixed policy π, and suppose that $\mathcal{F}$ is finite and $f^\star \in \mathcal{F}$. Fix $\delta \in (0,1)$ and $\alpha \geq 0$. Suppose that a possibly randomized $\widetilde{f} \in \mathcal{F}$ satisfies*

$$L_{\mathcal{S}}(\widetilde{f}) \leq \min_{f \in \mathcal{F}} L_{\mathcal{S}}(f) + \alpha$$

with probability at least $1 - \delta/2$ over its randomness, conditional on each dataset $\mathcal{S}$. Then, jointly over the observations and oracle randomness, with probability at least $1 - \delta$,

$$n\|\widetilde{f} - f^\star\|_\pi^2 \leq 2\alpha + 10 \log \frac{2|\mathcal{F}|}{\delta}.$$

Proof. For a fixed $f \in \mathcal{F}$, define

$$Z_i(f) = \big(f(c_i, a_i) - y_i\big)^2 - \big(f^\star(c_i, a_i) - y_i\big)^2, \qquad \mu_f = \mathbb{E}[Z_i(f)] = \|f - f^\star\|_\pi^2.$$

The equality for μ_f follows by expanding the squares and applying realizability, as in Lemma 10. The same boundedness calculation gives

$$|Z_i(f)| \leq 1, \qquad \mathrm{Var}(Z_i(f)) \leq 4\mu_f, \qquad |Z_i(f) - \mu_f| \leq 2.$$

Bernstein's inequality implies that, for every $u > 0$, with probability at least $1 - e^{-u}$,

$$\sum_{i=1}^n Z_i(f) \geq n\mu_f - \sqrt{8n\mu_f u} - \frac{2u}{3} \geq \frac{n\mu_f}{2} - \frac{14u}{3}.$$

The second inequality uses $\sqrt{8xu} \leq x/2 + 4u$. Rearranging gives

$$n\mu_f \leq 2\sum_{i=1}^n Z_i(f) + \frac{28u}{3} \leq 2\sum_{i=1}^n Z_i(f) + 10u.$$

Set $u = \log(2|\mathcal{F}|/\delta)$ and take a union bound over $f \in \mathcal{F}$. With probability at least $1 - \delta/2$, this inequality holds for every candidate, including the randomly selected $\widetilde{f}$. On the oracle's success event,

$$\sum_{i=1}^n Z_i(\widetilde{f}) = L_{\mathcal{S}}(\widetilde{f}) - L_{\mathcal{S}}(f^\star) \leq \alpha,$$

because $f^\star \in \mathcal{F}$. The oracle's failure probability is at most $\delta/2$ conditional on every dataset, and hence also unconditionally. A union bound over the two failure events proves the claim. □

For ordinary least-squares ERM, the empirical optimality condition holds deterministically with $\alpha = 0$. The lemma therefore gives $n\|\widetilde{f} - f^\star\|_\pi^2 \leq 10 \log(2|\mathcal{F}|/\delta)$ with probability at least $1 - \delta$.

If π is chosen using earlier observations, the same result applies conditional on the history before collecting the batch. This fixes π; the new context–loss pairs and fresh action randomness then generate i.i.d. observations under that policy. The application across epochs is given in Section E.

C.3 Exponential-mechanism instantiation

Lemma 12 (Privacy and accuracy of finite-class regression)**.** *Let $\mathcal{F} \subseteq [0,1]^{\mathcal{C} \times \mathcal{A}}$ be a finite prediction class and fix $\varepsilon_{\mathrm{priv}} > 0$. On an ordered batch $\mathcal{S}$ of $n \geq 1$ observations in $\mathcal{C} \times \mathcal{A} \times [0,1]$, consider the exponential mechanism*

$$\mathbb{P}(\widetilde{f} = f \mid \mathcal{S}) = \frac{\exp\{-\varepsilon_{\mathrm{priv}} L_{\mathcal{S}}(f)/2\}}{\sum_{g \in \mathcal{F}} \exp\{-\varepsilon_{\mathrm{priv}} L_{\mathcal{S}}(g)/2\}}, \qquad f \in \mathcal{F}. \tag{8}$$

This oracle is pure $\varepsilon_{\text{priv}}$-DP under replacement of one observation. If $\mathcal{S}$ consists of i.i.d. observations generated under a fixed policy π and $f^\star \in \mathcal{F}$, then, for every $\delta \in (0,1)$, with probability at least $1-\delta$,

$$n\|\widetilde{f} - f^\star\|_\pi^2 \le \left(10 + \frac{4}{\varepsilon_{\text{priv}}}\right) \log \frac{2|\mathcal{F}|}{\delta}.$$

The probability is over the observations and the oracle's randomness.

Proof. We first prove privacy, then show that the sampled predictor has sufficiently small empirical loss to apply Lemma 11.

Privacy. Fix neighboring batches $\mathcal{S}, \mathcal{S}'$ of the same size. Each squared loss lies in $[0,1]$, so $|L_{\mathcal{S}}(f) - L_{\mathcal{S}'}(f)| \le 1$ for every $f \in \mathcal{F}$. Let

$$\mathcal{Z}_{\mathcal{S}} = \sum_{f \in \mathcal{F}} \exp\{-\varepsilon_{\text{priv}} L_{\mathcal{S}}(f)/2\}$$

denote the normalizing constant. For every $f \in \mathcal{F}$,

$$\frac{\mathbb{P}(\widetilde{f} = f \mid \mathcal{S})}{\mathbb{P}(\widetilde{f} = f \mid \mathcal{S}')} = \exp\left\{-\frac{\varepsilon_{\text{priv}}}{2}\big(L_{\mathcal{S}}(f) - L_{\mathcal{S}'}(f)\big)\right\} \frac{\mathcal{Z}_{\mathcal{S}'}}{\mathcal{Z}_{\mathcal{S}}} \le e^{\varepsilon_{\text{priv}}}.$$

The sensitivity bound makes each factor at most $e^{\varepsilon_{\text{priv}}/2}$. For the normalizing constants, this follows by comparing the corresponding summands. Summing over any set of outputs proves pure differential privacy. This argument applies to arbitrary batches and requires no distributional assumption (Dwork & Roth, 2014).

Regression accuracy. The exponential mechanism favors predictors with smaller empirical squared loss. Write $L_{\mathcal{S}}^{\min} = \min_{f \in \mathcal{F}} L_{\mathcal{S}}(f)$. Conditional on any batch $\mathcal{S}$, for every $\alpha \ge 0$,

$$\mathbb{P}\Big(L_{\mathcal{S}}(\widetilde{f}) > L_{\mathcal{S}}^{\min} + \alpha \,\Big|\, \mathcal{S}\Big) \le |\mathcal{F}| e^{-\varepsilon_{\text{priv}}\alpha/2}.$$

Indeed, the total weight of predictors exceeding this threshold is at most $|\mathcal{F}| \exp\{-\varepsilon_{\text{priv}}(L_{\mathcal{S}}^{\min} + \alpha)/2\}$, whereas the normalizing constant is at least $\exp\{-\varepsilon_{\text{priv}} L_{\mathcal{S}}^{\min}/2\}$.

Taking $\alpha = (2/\varepsilon_{\text{priv}}) \log(2|\mathcal{F}|/\delta)$ makes this conditional failure probability at most $\delta/2$. Thus the oracle satisfies the approximate-ERM requirement of Lemma 11. That lemma gives, with probability at least $1-\delta$,

$$n\|\widetilde{f} - f^\star\|_\pi^2 \le 2\alpha + 10 \log \frac{2|\mathcal{F}|}{\delta} = \left(10 + \frac{4}{\varepsilon_{\text{priv}}}\right) \log \frac{2|\mathcal{F}|}{\delta}.$$

□

The application to all fitted predictors, including the choice of $\text{Err}_{\text{orc}}^{\text{batch}}(T, \delta)$, is given in Section F.2.

D Complete proof for vanilla PO

We begin by restating the algorithm and recording the identities used throughout the proof.

Vanilla Policy Optimization (VANILLAPO)

1: **Input:** horizon T, action set $\mathcal{A}$, learning rate $\eta \in (0, 1]$, and an offline regression oracle satisfying Assumption 2.
2: Initialize $\pi_1(c, a) \leftarrow 1/A$ for every $c \in \mathcal{C}$ and $a \in \mathcal{A}$, and let $\mathcal{S}_0 \leftarrow ()$.
3: **for** $t = 1, \dots, T$ **do**
4: Observe the context c_t.
5: Sample $a_t \sim \pi_t(c_t, \cdot)$ and observe $\ell_t(c_t, a_t)$.
6: Append the observation: $\mathcal{S}_t \leftarrow \mathrm{append}(\mathcal{S}_{t-1}, (c_t, a_t, \ell_t(c_t, a_t)))$.
7: Fit the predictor: $\widehat{f}_t \leftarrow \mathrm{Oracle}(\mathcal{S}_t)$.
8: Define the next policy, for every $c \in \mathcal{C}$ and $a \in \mathcal{A}$, by

$$\pi_{t+1}(c, a) = \frac{\pi_t(c, a) \exp\{-\eta \widehat{f}_t(c, a)\}}{\sum_{b \in \mathcal{A}} \pi_t(c, b) \exp\{-\eta \widehat{f}_t(c, b)\}}. \tag{9}$$

9: **end for**

Recall that $\Delta(c, a) = f^\star(c, a) - f^\star(c, a^\star(c))$ is the gap of action a. For each action $a \in \mathcal{A}$, define

$$e_{t,a}(c) = \widehat{f}_t(c, a) - f^\star(c, a),$$

and, for $0 \le t \le T$, let

$$D_{t,a}(c) = \sum_{r=1}^{t} \big(e_{r,a}(c) - e_{r,a^\star(c)}(c)\big), \qquad D_{0,a}(c) = 0.$$

Here, $D_{t,a}(c)$ measures the accumulated prediction error through round t in estimating the loss gap between action a and the optimal action $a^\star(c)$. Starting from the uniform policy and unrolling the exponential update in (9) gives, for $1 \le t \le T + 1$,

$$\frac{\pi_t(c, a)}{\pi_t(c, a^\star(c))} = \exp\{-\eta((t-1)\Delta(c, a) + D_{t-1,a}(c))\}.$$

Define the relative action weights for round t by

$$W_{t,a}(c) = \exp\{-\eta((t-1)\Delta(c, a) + D_{t-1,a}(c))\}.$$

In particular, $W_{t,a^\star(c)}(c) = 1$, and

$$\pi_t(c, a) = \frac{W_{t,a}(c)}{\sum_{b \in \mathcal{A}} W_{t,b}(c)} \le W_{t,a}(c). \tag{10}$$

To measure the errors under past policies, define, for $c \in \mathcal{C}$, $a \in \mathcal{A}$, and $1 \le t \le T$,

$$\bar{\pi}_t(c, a) = \frac{N_{t,a}(c)}{t}, \qquad \text{where} \qquad N_{t,a}(c) = \sum_{r=1}^{t} \pi_r(c, a).$$

We denote the mean squared prediction error of $\widehat{f}_t$ under $\bar{\pi}_t(c, .)$ and the corresponding cumulative mean squared prediction error by

$$s_t(c) = \sum_{a \in \mathcal{A}} \bar{\pi}_t(c, a) e_{t,a}(c)^2, \qquad S_t(c) = \sum_{r=1}^{t} s_r(c).$$

Let $R_T(c)$ denote the cumulative regret obtained by evaluating every policy in the learned sequence at the same fixed context c:

$$R_T(c) = \sum_{t=1}^{T} \sum_{a \in \mathcal{A}} \pi_t(c, a) \Delta(c, a).$$

Holding the learned policies fixed and averaging over a fresh context gives

$$\mathbb{E}_{c \sim \mathcal{D}}[R_T(c)] = \sum_{t=1}^{T} \mathcal{R}(\pi_t).$$

Throughout this appendix, let

$$\Lambda = 1 + \log(AT), \qquad H_T = \sum_{t=1}^{T} \frac{1}{t}, \qquad C = 16.$$

D.1 Proof of Theorem 3

We prove Theorem 3 by first bounding $R_T(c)$ at each context c in terms of $S_T(c)$. Averaging over contexts and applying Assumption 2 then bounds $\sum_{t=1}^{T} \mathcal{R}(\pi_t)$. Finally, the concentration argument in Section B transfers this bound to $\overline{R}_T$. Each summand $\pi_t(c,a)\Delta(c,a)$ in $R_T(c)$ can be bounded through either the loss gap $\Delta(c,a)$ or the action probability $\pi_t(c,a)$. The following dichotomy is the central tool for doing so.

Lemma 13 (Late-round probability–error dichotomy)**.** *Fix a context $c \in \mathcal{C}$, a suboptimal action $a \in \mathcal{A}$ with $\Delta(c,a) > 0$, and a round $t \in [T]$ satisfying $t - 1 \geq \frac{4\Lambda}{\eta\Delta(c,a)}$. Then*

$$\pi_t(c,a) < \frac{1}{AT} \qquad \text{or} \qquad \Delta(c,a) \leq C^2 A\eta\, S_{t-1}(c).$$

The lemma identifies the only two ways a suboptimal action can retain non-negligible probability: its gap is small relative to the accumulated prediction error, or the round is too early in the game. Carefully choosing the learning rate keeps the cumulative cost of both possibilities under control.

We now deduce the theorem from Lemma 13, with the rest of this appendix devoted to proving it.

Proof of Theorem 3. Fix a sequence of predictors $\widehat{f}_1, \ldots, \widehat{f}_T$ and consequently policies $\pi_1, \ldots, \pi_T$. Fix also a context $c \in \mathcal{C}$; we first derive a deterministic bound on $R_T(c)$ in terms of $S_T(c)$.

Define the error-dependent gap threshold $\tau = C^2 A\eta S_T(c)$, and, for each action $a \in \mathcal{A}$, define its action-dependent time scale by $T_a = \frac{4\Lambda}{\eta\Delta(c,a)} + 1$ with $T_a = \infty$ when $\Delta(c,a) = 0$. We partition the action–round pairs $\mathcal{A} \times [T]$ as

$$\underbrace{\{(a,t) : \Delta(c,a) \leq \tau\}}_{\mathcal{S}_1} \cup \underbrace{\{(a,t) : \Delta(c,a) > \tau,\ t \leq T_a\}}_{\mathcal{S}_2} \cup \underbrace{\{(a,t) : \Delta(c,a) > \tau,\ t > T_a\}}_{\mathcal{S}_3}.$$

The three sets are pairwise disjoint. We bound their regret contributions separately.

For every $(a,t) \in \mathcal{S}_1$, $\Delta(c,a) \leq \tau$. Thus

$$\sum_{(a,t)\in\mathcal{S}_1} \pi_t(c,a)\Delta(c,a) \leq T\tau = C^2 A\eta T S_T(c). \tag{11}$$

For every $(a,t) \in \mathcal{S}_2$ with $t \geq 2$, $\Delta(c,a) \leq \frac{4\Lambda}{\eta(t-1)}$. Therefore

$$\sum_{(a,t)\in\mathcal{S}_2} \pi_t(c,a)\Delta(c,a) \leq 1 + \sum_{t=2}^{T} \frac{4\Lambda}{\eta(t-1)} \leq 1 + \frac{4\Lambda H_T}{\eta}. \tag{12}$$

For every $(a,t) \in \mathcal{S}_3$, $S_{t-1}(c) \leq S_T(c) < \frac{\Delta(c,a)}{C^2 A\eta}$, and therefore, by Lemma 13, $\pi_t(c,a) \leq \frac{1}{AT}$. Hence

$$\sum_{(a,t)\in\mathcal{S}_3} \pi_t(c,a)\Delta(c,a) \leq 1. \tag{13}$$

Combining (11)–(13) and using $\Lambda \geq H_T \geq 1$, $\eta \leq 1$, and $C^2 \geq 6$ yields

$$R_T(c) \leq C^2 A\eta T S_T(c) + 2 + \frac{4\Lambda H_T}{\eta} \leq C^2 \left\{ A\eta T S_T(c) + \frac{\Lambda^2}{\eta} \right\}. \tag{14}$$

We now apply Assumption 2 with failure probability $\delta/2$. Fix any realization of the interaction on which this oracle event holds. Holding the fitted predictors and policies fixed, we average over a fresh context $c \sim \mathcal{D}$.

By the definition of $s_t(c)$ and the oracle guarantee,

$$\mathbb{E}_{c\sim\mathcal{D}}[s_t(c)] = \frac{1}{t}\sum_{r=1}^{t} \|\widehat{f}_t - f^\star\|_{\pi_r}^2 \leq \frac{\mathrm{Err}_{\mathrm{orc}}(T, \delta/2)}{t}.$$

Summing over $t = 1, \dots, T$ gives

$$\mathbb{E}_{c\sim\mathcal{D}}[S_T(c)] \le \mathrm{Err}_{\mathrm{orc}}(T, \delta/2) H_T.$$

Integrating (14) over $c \sim \mathcal{D}$ gives $\sum_{t=1}^{T} \mathcal{R}(\pi_t)$ on the left-hand side. Consequently,

$$\sum_{t=1}^{T} \mathcal{R}(\pi_t) \le C^2 \left\{ \frac{\Lambda^2}{\eta} + A\eta T \,\mathrm{Err}_{\mathrm{orc}}(T, \delta/2) H_T \right\}.$$

This inequality holds with probability at least $1 - \delta/2$.

We next transfer this bound to $\overline{R}_T$, which evaluates the policies at the observed contexts. By Proposition 9, with probability at least $1 - \delta/2$,

$$\overline{R}_T \le \sum_{t=1}^{T} \mathcal{R}(\pi_t) + \sqrt{\frac{T}{2} \log \frac{2}{\delta}}.$$

A union bound therefore gives, with probability at least $1 - \delta$,

$$\overline{R}_T \le C^2 \left\{ \frac{\Lambda^2}{\eta} + A\eta T \,\mathrm{Err}_{\mathrm{orc}}(T, \delta/2) H_T \right\} + \sqrt{\frac{T}{2} \log \frac{2}{\delta}}.$$

Substituting $\eta = \sqrt{\Lambda / [AT(1 + \mathrm{Err}_{\mathrm{orc}}(T, \delta/2))]}$ and using $H_T \le \Lambda$ yields

$$\begin{aligned} \overline{R}_T \le\ & 2C^2 \sqrt{AT(1 + \mathrm{Err}_{\mathrm{orc}}(T, \delta/2))}\,(1 + \log(AT))^{3/2} \\ & + \sqrt{\frac{T}{2} \log \frac{2}{\delta}}. \end{aligned} \tag{15}$$

This proves the general-oracle bound with probability at least $1 - \delta$.

For a finite realizable class $\mathcal{F}$, the ERM guarantee in Section C and $\delta \le 1/2$ give $1 + \mathrm{Err}_{\mathrm{orc}}(T, \delta/2) = O(\log(|\mathcal{F}|/\delta))$. Substituting into (15), including the concentration term, yields $\overline{R}_T = \widetilde{O}(\sqrt{AT \log(|\mathcal{F}|/\delta)})$. □

D.2 Proof of Lemma 13

For every action $a \in \mathcal{A}$, define

$$\kappa_{t,a}(c) = \frac{1}{\bar{\pi}_t(c, a)} + \frac{1}{\bar{\pi}_t(c, a^\star(c))}, \qquad B_{t,a}(c) = \sum_{r=1}^{t} \kappa_{r,a}(c). \tag{16}$$

The term $\kappa_{t,a}(c)$ becomes large when the average policy assigns little probability to one of the actions: a or $a^\star(c)$. Prediction errors on that action then receive little weight in $s_t(c)$, so small average squared error alone gives weak control over the estimated loss gap. The sum $B_{t,a}(c)$ accumulates these factors across updates. The following supporting lemmas are used to prove Lemma 13.

Lemma 14 (Bounding the accumulated prediction error)**.** *For every $a \in \mathcal{A}$ and $1 \le r \le t \le T$,*

$$D_{r,a}(c)^2 \le S_r(c) B_{r,a}(c) \le S_t(c) B_{r,a}(c).$$

Lemma 15 (Historical coverage under small prediction error)**.** *Fix a context $c \in \mathcal{C}$, a suboptimal action $a \in \mathcal{A}$ with $\Delta(c, a) > 0$, and a round $t \in \{1, \dots, T\}$. If $S_t(c) < \frac{\Delta(c,a)}{C^2 A\eta}$, then for $1 \le r \le t$,*

$$B_{r,a}(c) \le CA\big(r + \eta \Delta(c, a) r^2\big). \tag{17}$$

Proof of Lemma 13. Suppose, for contradiction, that $\pi_t(c, a) \ge \frac{1}{AT}$ and $S_{t-1}(c) < \frac{\Delta(c,a)}{C^2 A\eta}$.

We show that the bound on $\pi_t(c, a)$ forces the accumulated prediction error to be large and negative. Since $W_{t,a^\star(c)}(c) = 1$, (10) gives $\pi_t(c, a) \le W_{t,a}(c)$. Hence

$$\exp\{-\eta((t-1)\Delta(c, a) + D_{t-1,a}(c))\} = W_{t,a}(c) \ge \pi_t(c, a) \ge \frac{1}{AT}.$$

Taking logarithms and applying $\eta(t-1)\Delta(c,a) \geq 4\Lambda$ and $\log(AT) \leq \Lambda$ yields

$$D_{t-1,a}(c) \leq \frac{\log(AT)}{\eta} - (t-1)\Delta(c,a) \leq -\frac{3}{4}(t-1)\Delta(c,a). \tag{18}$$

We now show that the bound on $S_{t-1}(c)$ makes such a large prediction error impossible.

By Lemma 15, applied to action a and round $t-1 \geq \frac{4\Lambda}{\eta\Delta(c,a)} \geq \frac{4}{\eta\Delta(c,a)}$,

$$B_{t-1,a}(c) \leq CA\big((t-1) + \eta\Delta(c,a)(t-1)^2\big) \leq 2CA\eta\Delta(c,a)(t-1)^2.$$

Therefore, by Lemma 14 and the assumed bound on $S_{t-1}(c)$,

$$|D_{t-1,a}(c)| \leq \sqrt{S_{t-1}(c)B_{t-1,a}(c)} < \sqrt{\frac{2}{C}}\,(t-1)\Delta(c,a) \leq \frac{1}{2}(t-1)\Delta(c,a),$$

where the last inequality uses $C \geq 8$. This contradicts (18) and proves the claim. □

D.3 Proof of Lemma 14

Proof of Lemma 14. For a fixed t and $a \neq a^\star(c)$, using the AM-GM inequality, we can show

$$\begin{aligned}\big(e_{t,a}(c) - e_{t,a^\star(c)}(c)\big)^2 &\leq \big(\bar\pi_t(c,a)e_{t,a}(c)^2 + \bar\pi_t(c,a^\star(c))e_{t,a^\star(c)}(c)^2\big)\left(\frac{1}{\bar\pi_t(c,a)} + \frac{1}{\bar\pi_t(c,a^\star(c))}\right)\\ &\leq s_t(c)\kappa_{t,a}(c).\end{aligned}$$

For $a = a^\star(c)$, it vacuously holds that $\big(e_{t,a}(c) - e_{t,a^\star(c)}(c)\big)^2 = 0 \leq s_t(c)\kappa_{t,a}(c)$.

Applying Cauchy–Schwarz across rounds gives

$$\begin{aligned}D_{t,a}(c)^2 &= \left(\sum_{r=1}^{t} \frac{e_{r,a}(c) - e_{r,a^\star(c)}(c)}{\sqrt{\kappa_{r,a}(c)}}\sqrt{\kappa_{r,a}(c)}\right)^2\\ &\leq \left(\sum_{r=1}^{t} \frac{(e_{r,a}(c) - e_{r,a^\star(c)}(c))^2}{\kappa_{r,a}(c)}\right)\left(\sum_{r=1}^{t}\kappa_{r,a}(c)\right) \leq S_t(c)B_{t,a}(c).\end{aligned}$$

Applying this bound at an earlier round r and using $S_r(c) \leq S_t(c)$ for $r \leq t$ proves the claim. □

D.4 Proof of Lemma 15

Proof of Lemma 15. With c and a fixed throughout the proof, for every action $b \in \mathcal{A}$, write

$$\Gamma_b = \max\{\Delta(c,a), \Delta(c,b)\}.$$

To control the normalizing denominator, we prove by induction that, for every $1 \leq r \leq t$ and $b \in \mathcal{A}$,

$$B_{r,b}(c) \leq CA\big(r + \eta\Gamma_b r^2\big). \tag{19}$$

Alongside this bound, we establish the early-round action probability bounds

$$r - 1 \leq \frac{1}{\eta\Gamma_b} \quad\Longrightarrow\quad \pi_r(c,b) \geq \frac{e^{-2}}{A}. \tag{20}$$

For the base case $r = 1$, the policy is uniform and $B_{1,b}(c) = 2A$, so both (19) and (20) hold.

The proof of the induction step consists of two parts. First, we show that the bounds on $B_{r-1,b}(c)$ control the accumulated prediction errors and therefore the action probabilities $\pi_r(c,b)$. Second, summing the early-round action probabilities allows us to bound $N_{r,b}(c)$ and consequently $B_{r,b}(c)$.

Part 1: Lower-bound $\pi_r(c,b)$. Fix $2 \le r \le t+1$ and suppose that (19) holds at round $r-1$ for every $b \in \mathcal{A}$. To show (20) at round r, suppose that $r-1 \le 1/(\eta\Delta(c,a))$.

Let $x_b = \eta\Gamma_b(r-1) \ge 0$. By Lemma 14, the induction hypothesis, $\Gamma_b \ge \Delta(c,a)$, and $C \ge 8$,

$$\eta^2 D_{r-1,b}(c)^2 \le \eta^2 S_t(c) B_{r-1,b}(c) \le \frac{\Delta(c,a)\, x_b(1+x_b)}{C\Gamma_b} \le \frac{\max\{1, x_b^2\}}{4}. \tag{21}$$

As $\Gamma_b = \max\{\Delta(c,b), \Delta(c,a)\}$ and $\eta\Delta(c,a)(r-1) \le 1$, it holds that

$$\eta\Delta(c,b)(r-1) \le x_b \qquad \text{and} \qquad \eta\Delta(c,b)(r-1) \ge x_b\, \mathbb{I}\{x_b > 1\}. \tag{22}$$

Combining (21) and (22) with the identity $W_{r,b}(c) = \exp\{-\eta((r-1)\Delta(c,b) + D_{r-1,b}(c))\}$,

$$\exp\left\{-x_b - \tfrac{1}{2}\max\{1, x_b\}\right\} \le W_{r,b}(c) \le \exp\left\{-x_b\, \mathbb{I}\{x_b > 1\} + \tfrac{1}{2}\max\{1, x_b\}\right\}.$$

In particular, when $x_b \le 1$, $W_{r,b}(c) \in [e^{-3/2}, e^{1/2}]$, while when $x_b > 1$, $W_{r,b}(c) \in (0, e^{-1/2})$.

The normalizing denominator for $\pi_r(c,.)$, therefore, satisfies

$$\sum_{k \in \mathcal{A}} W_{r,k}(c) \le A e^{1/2}.$$

For any action b satisfying $r-1 \le 1/(\eta\Gamma_b)$, $x_b \le 1$, and so

$$\pi_r(c,b) \ge \frac{e^{-3/2}}{A e^{1/2}} = \frac{e^{-2}}{A}.$$

This establishes (20) for probabilities $\pi_r(c,b)$ at round r using (19) for $B_{r-1,b}(c)$ at round $r-1$.

Part 2: Lower-bound $N_{r,b}(c)$ and upper-bound $B_{r,b}(c)$. Now let $2 \le r \le t$. By the induction hypothesis and Part 1, the probability bounds in (20) hold on their respective intervals for all policies $\pi_1, \ldots, \pi_r$.

Consider arbitrary $b \in \mathcal{A}$ and $1 \le u \le r$. It holds that

$$N_{u,b}(c) = \sum_{s=1}^{u} \pi_s(c,b) \ge \frac{e^{-2}}{A} \min\left\{u, \frac{1}{\eta\Gamma_b}\right\}.$$

By (16), as $\Gamma_{a^\star(c)} \le \Gamma_b$, it follows that

$$\kappa_{u,b}(c) = \frac{u}{N_{u,b}(c)} + \frac{u}{N_{u,a^\star(c)}(c)} \le 2e^2 A \max\{1, \eta\Gamma_b u\}.$$

Summing over u yields

$$B_{r,b}(c) \le 2e^2 A \sum_{u=1}^{r} \max\{1, \eta\Gamma_b u\} \le 2e^2 A\big(r + \eta\Gamma_b r^2\big) \le CA\big(r + \eta\Gamma_b r^2\big).$$

This proves (19) at round r and completes the induction.

Since $\Gamma_a = \Delta(c,a)$, taking $b = a$ gives (17) for all $1 \le r \le t$, completing the proof. □

E Complete proof for batched vanilla PO

We prove Theorem 4 for the algorithm in Algorithm 2, under the stochastic model of Section 2. Throughout this section, π_m denotes the policy used throughout epoch m, and the predictors take values in $[0,1]$. Fix $\delta \in (0, 1/2]$ and use the known bound $\mathrm{Err}^{\mathrm{batch}}_{\mathrm{orc}}(T,\delta) \ge 1$ from the main body. We reserve failure probability $\delta/2$ for regression accuracy and $\delta/2$ for concentration over the observed contexts. Accordingly, assume that, with probability at least $1 - \delta/2$,

$$\|\widetilde{f}_m - f^\star\|^2_{\pi_{m-1}} \le \frac{\mathrm{Err}^{\mathrm{batch}}_{\mathrm{orc}}(T,\delta)}{B_{m-1}} \quad \text{simultaneously for } m = 2, \ldots, M. \tag{23}$$

We use the following explicit learning rates, which realize the order stated in the theorem:

$$\eta_m = \min\left\{\frac{1}{A}, \frac{1}{\sqrt{6AB_m \operatorname{Err}^{\mathrm{batch}}_{\mathrm{orc}}(T,\delta)}}\right\}, \qquad m = 2, \dots, M. \tag{24}$$

We first prove a deterministic regret bound for every realized predictor sequence satisfying (23). Averaging exponential updates gives both a lower bound on action probabilities and a bound on predicted regret. These probability bounds let us transfer prediction accuracy from the preceding epoch's policy to other policies. An induction then compares predicted and true regret, after which we sum their population regrets over epochs. The concentration result in Section B then converts this bound to $\overline{R}_T$. Finally, Section E.6 shows how an i.i.d. regression guarantee supplies the theorem's accuracy condition and proves its finite-class ERM conclusion.

E.1 Policy representation and the epoch schedule

Recall that $B_m = 2^{m-1}$ and $M = \lceil \log_2(T+1) \rceil$. Since $\sum_{j<m} B_j = B_m - 1$, epoch m starts at round B_m and contains

$$n_m = \min\{B_m, T - B_m + 1\}$$

rounds. Every started epoch has $B_m \le T$. For $m < M$, the batch $\mathcal{S}_m$ has exactly B_m observations; only the final epoch may be shortened. Thus every batch used to fit a predictor is complete, and the algorithm makes exactly $M-1$ oracle calls. When $T = 1$, it uses the uniform policy once and makes no calls.

Datasets are ordered lists, so repeated observations occupy distinct rows. The predictor $\widetilde{f}_m$ is fitted only to $\mathcal{S}_{m-1}$, after which that batch is discarded. No fit uses $\mathcal{S}_M$.

Proposition 16 (Batched vanilla PO as an average of policy updates)**.** *Fix an epoch $m \ge 2$ and a fitted predictor $\widetilde{f}_m$. For every context c, define $\pi_{m,1}(c,a) = 1/A$ and, for $s = 1, \dots, B_m$,*

$$\pi_{m,s+1}(c,a) = \frac{\pi_{m,s}(c,a)\exp\{-\eta_m \widetilde{f}_m(c,a)\}}{\sum_{b\in\mathcal{A}} \pi_{m,s}(c,b)\exp\{-\eta_m \widetilde{f}_m(c,b)\}}.$$

Then the policy in (4) satisfies

$$\pi_m(c,a) = \frac{1}{B_m}\sum_{s=1}^{B_m} \pi_{m,s}(c,a). \tag{25}$$

The sampler in (5) draws an action from this distribution exactly.

Proof. Unrolling the fixed-predictor updates gives

$$\pi_{m,s}(c,a) = \frac{\exp\{-\eta_m (s-1)\widetilde{f}_m(c,a)\}}{\sum_{b\in\mathcal{A}} \exp\{-\eta_m (s-1)\widetilde{f}_m(c,b)\}}.$$

This is the summand indexed by $k = s-1$ in (4). Conditional on $K_t = s-1$, the sampler in (5) has this same action distribution. Averaging over the uniform choice of K_t proves both claims. □

The sequence $\pi_{m,s}$ is an auxiliary sequence used to analyze the policy. The learner stores $(\widetilde{f}_m, \eta_m, B_m)$ and, for each observed context, evaluates the A predicted losses and samples one softmax distribution. This requires no iteration over the context space and no explicit evaluation of all B_m mixture components. In a shortened final epoch, the policy and sampler still use the nominal value B_M.

E.2 Action probabilities and predicted regret

For any predictor $\widetilde{f}_m$, define

$$d_m(c,a) = \widetilde{f}_m(c,a) - \min_{b\in\mathcal{A}} \widetilde{f}_m(c,b) \tag{26}$$

and the corresponding predicted regret

$$\widehat{\mathcal{R}}_m(\pi) = \mathbb{E}_{c\sim\mathcal{D}} \sum_{a\in\mathcal{A}} \pi(c,a) d_m(c,a).$$

Thus $d_m(c,a)$ is the predicted loss gap, and $\widehat{\mathcal{R}}_m(\pi)$ is the regret of π if the predicted losses were correct. These definitions also apply to the bonus variant introduced in Section G.

Lemma 17 (Action probabilities and predicted regret of batched vanilla PO)**.** *For every epoch $m \geq 2$, context c, and action a,*

$$\frac{1}{\pi_m(c,a)} \leq eA\big(1 + B_m \eta_m d_m(c,a)\big). \tag{27}$$

Consequently, for every policy π,

$$\mathbb{E}_{c\sim\mathcal{D}} \sum_{a\in\mathcal{A}} \frac{\pi(c,a)}{\pi_m(c,a)} \leq eA\big(1 + B_m \eta_m \widehat{\mathcal{R}}_m(\pi)\big). \tag{28}$$

The policy deployed in epoch m also satisfies

$$\widehat{\mathcal{R}}_m(\pi_m) \leq \frac{2\log A}{B_m \eta_m}. \tag{29}$$

Proof. Fix m and c. Subtracting the minimum predicted loss from every action leaves each exponential update unchanged. Hence

$$\pi_{m,s}(c,a) = \frac{\exp\{-\eta_m(s-1)d_m(c,a)\}}{\sum_{b\in\mathcal{A}} \exp\{-\eta_m(s-1)d_m(c,b)\}} \geq \frac{1}{A} \exp\{-\eta_m(s-1)d_m(c,a)\},$$

because all predicted gaps are nonnegative.

For every integer $B \geq 1$ and $x \geq 0$,

$$\frac{1}{B} \sum_{k=0}^{B-1} e^{-kx} \geq e^{-1} \min\left\{1, \frac{1}{Bx}\right\}, \tag{30}$$

where $1/(Bx) = +\infty$ when $x = 0$. If $Bx \leq 1$, every term is at least e^{-1}. If $Bx > 1$ and $x \geq 1$, the term $k = 0$ suffices. Otherwise, the terms $k = 0, \ldots, \lfloor 1/x \rfloor$ are all present and are each at least e^{-1}; their number is at least $1/x$. Applying (30) to (25) gives

$$\pi_m(c,a) \geq \frac{e^{-1}}{A} \min\left\{1, \frac{1}{B_m \eta_m d_m(c,a)}\right\}.$$

Taking reciprocals and using $\max\{1,x\} \leq 1 + x$ proves (27). Multiplying by $\pi(c,a)$, summing over actions, and averaging over contexts proves (28). This implication is deterministic and holds for every policy simultaneously.

To bound predicted regret, let

$$W_{m,s}(c) = \sum_{a\in\mathcal{A}} \exp\{-\eta_m(s-1)d_m(c,a)\}.$$

For $x \geq 0$, $e^{-x} \leq 1 - x + x^2/2$. Therefore

$$\log \frac{W_{m,s+1}(c)}{W_{m,s}(c)} \leq -\eta_m \sum_{a\in\mathcal{A}} \pi_{m,s}(c,a) d_m(c,a) + \frac{\eta_m^2}{2} \sum_{a\in\mathcal{A}} \pi_{m,s}(c,a) d_m(c,a)^2.$$

Initially $W_{m,1}(c) = A$, and $W_{m,B_m+1}(c) \geq 1$ because an action with zero predicted gap retains weight one. Summing the last inequality gives

$$\sum_{s=1}^{B_m} \sum_{a\in\mathcal{A}} \pi_{m,s}(c,a) d_m(c,a) \leq \frac{\log A}{\eta_m} + \frac{\eta_m}{2} \sum_{s=1}^{B_m} \sum_{a\in\mathcal{A}} \pi_{m,s}(c,a) d_m(c,a)^2.$$

Since $0 \leq d_m(c,a) \leq 1$ and $\eta_m \leq 1$, the last sum can be absorbed into the left-hand side, leaving an upper bound $2\log A/\eta_m$. Divide by B_m and average over c to obtain (29). □

The first bound quantifies the exploration retained by the average: inverse action probabilities grow at most linearly with the predicted gap. The second shows that this exploration does not prevent the policy from achieving small predicted regret.

E.3 Transferring regression accuracy between policies

The predictor for epoch m is trained under π_{m-1}. To analyze the new policy π_m, we must control its prediction error under a different action distribution. Define the error in the predicted population loss of a policy by

$$E_m(\pi) = \left| \mathbb{E}_{c\sim\mathcal{D}} \sum_{a\in\mathcal{A}} \pi(c,a)\big(\widetilde{f}_m(c,a) - f^\star(c,a)\big) \right|.$$

All statements below are deterministic once the predictors and policies have been fixed. In particular, they apply to policies chosen using those predictors.

Lemma 18 (Prediction error under another policy)**.** *Fix $m \geq 2$. Suppose $\pi_{m-1}(c,a) > 0$ for every context and action, and the accuracy bound (23) holds at epoch m. Then, for every policy π,*

$$E_m(\pi) \leq \sqrt{\frac{\mathrm{Err}^{\mathrm{batch}}_{\mathrm{orc}}(T,\delta)}{B_{m-1}} \mathbb{E}_{c\sim\mathcal{D}} \sum_{a\in\mathcal{A}} \frac{\pi(c,a)}{\pi_{m-1}(c,a)}}. \tag{31}$$

For batched vanilla PO and $m \geq 3$, this gives

$$E_m(\pi) \leq \sqrt{\frac{eA\,\mathrm{Err}^{\mathrm{batch}}_{\mathrm{orc}}(T,\delta)}{B_{m-1}}\big(1 + B_{m-1}\eta_{m-1}\widehat{\mathcal{R}}_{m-1}(\pi)\big)}.$$

For $m = 2$, uniform sampling gives $E_2(\pi) \leq \sqrt{A\,\mathrm{Err}^{\mathrm{batch}}_{\mathrm{orc}}(T,\delta)/B_1}$.

Proof. Write $e_m(c,a) = \widetilde{f}_m(c,a) - f^\star(c,a)$. Cauchy–Schwarz, with weights $\pi_{m-1}(c,a) > 0$, gives

$$\begin{aligned} E_m(\pi) &\leq \|e_m\|_{\pi_{m-1}} \left(\mathbb{E}_{c\sim\mathcal{D}} \sum_{a\in\mathcal{A}} \frac{\pi(c,a)^2}{\pi_{m-1}(c,a)} \right)^{1/2} \\ &\leq \|e_m\|_{\pi_{m-1}} \left(\mathbb{E}_{c\sim\mathcal{D}} \sum_{a\in\mathcal{A}} \frac{\pi(c,a)}{\pi_{m-1}(c,a)} \right)^{1/2}, \end{aligned}$$

where the second inequality uses $\pi(c,a)^2 \leq \pi(c,a)$. Apply (23) to obtain (31). For $m \geq 3$, use (28) for the preceding epoch. For $m = 2$, the expectation in (31) equals A, since $\pi_1(c,a) = 1/A$. □

E.4 Comparing predicted regret with true regret

We now show that policies with small predicted regret also have small true regret. Following the regression-oracle analysis of Simchi-Levi & Xu (2022), the induction uses the preceding epoch's comparison to control the next predictor's error. The argument depends only on action probabilities and regression accuracy, so we state it in a form that also applies to the bonus variant.

Lemma 19 (Comparison of true and predicted regret)**.** *Suppose (23) holds for every $m = 2, \ldots, M$, and π_1 is uniform. Let $C_0 \geq 1$ be a constant and let $\gamma_2, \ldots, \gamma_M$ be positive and nondecreasing. Assume*

$$\pi_m(c,a)^{-1} \leq C_0\big(A + \gamma_m d_m(c,a)\big), \qquad \gamma_m^2 \leq \frac{AB_{m-1}}{C_0\,\mathrm{Err}^{\mathrm{batch}}_{\mathrm{orc}}(T,\delta)} \quad (m \geq 2). \tag{32}$$

Then, simultaneously for every epoch $m \geq 2$ and every policy π,

$$\mathcal{R}(\pi) \leq 2\widehat{\mathcal{R}}_m(\pi) + \frac{30A}{\gamma_m}, \tag{33}$$

$$\widehat{\mathcal{R}}_m(\pi) \leq 2\mathcal{R}(\pi) + \frac{30A}{\gamma_m}. \tag{34}$$

Proof. Let $\pi^\star$ choose a true optimal action at each context, and let $\widehat{\pi}_m$ choose an action minimizing $\widetilde{f}_m(c,\cdot)$, with ties broken by a fixed ordering. Define

$$L(\pi) = \mathbb{E}_{c\sim\mathcal{D}} \sum_{a\in\mathcal{A}} \pi(c,a) f^\star(c,a), \qquad \widetilde{L}_m(\pi) = \mathbb{E}_{c\sim\mathcal{D}} \sum_{a\in\mathcal{A}} \pi(c,a) \widetilde{f}_m(c,a).$$

Then $\mathcal{R}(\pi) = L(\pi) - L(\pi^\star)$ and $\widehat{\mathcal{R}}_m(\pi) = \widetilde{L}_m(\pi) - \widetilde{L}_m(\widehat{\pi}_m)$. The optimality of these two comparator policies implies

$$\mathcal{R}(\pi) \le \widehat{\mathcal{R}}_m(\pi) + E_m(\pi) + E_m(\pi^\star), \tag{35}$$

$$\widehat{\mathcal{R}}_m(\pi) \le \mathcal{R}(\pi) + E_m(\pi) + E_m(\widehat{\pi}_m). \tag{36}$$

For example, the first inequality follows by inserting $\widetilde{L}_m(\pi)$ and $\widetilde{L}_m(\pi^\star)$ into $L(\pi) - L(\pi^\star)$ and using $\widetilde{L}_m(\widehat{\pi}_m) \le \widetilde{L}_m(\pi^\star)$. The second follows by the analogous decomposition and $L(\pi^\star) \le L(\widehat{\pi}_m)$.

For $m = 2$, Lemma 18 and (32) give

$$E_2(\pi) \le \sqrt{\frac{A \operatorname{Err}^{\mathrm{batch}}_{\mathrm{orc}}(T,\delta)}{B_1}} \le \frac{A}{\gamma_2}$$

for every policy. Substitution into (35) and (36) proves both claimed comparisons at epoch two, even with coefficient one on the regret and additive term $2A/\gamma_2$.

Suppose the comparisons hold at epoch $m-1$, where $m \ge 3$. Integrating the coverage inequality in (32) and applying (31) gives

$$\begin{aligned} E_m(\pi) &\le \sqrt{\frac{C_0 \operatorname{Err}^{\mathrm{batch}}_{\mathrm{orc}}(T,\delta)}{B_{m-1}} \big(A + \gamma_{m-1}\widehat{\mathcal{R}}_{m-1}(\pi)\big)} \\ &\le \sqrt{\frac{C_0 \operatorname{Err}^{\mathrm{batch}}_{\mathrm{orc}}(T,\delta)}{B_{m-1}} \big(31A + 2\gamma_{m-1}\mathcal{R}(\pi)\big)}. \end{aligned}$$

The parameter condition gives $C_0 \operatorname{Err}^{\mathrm{batch}}_{\mathrm{orc}}(T,\delta)/B_{m-1} \le A/\gamma_m^2$, and monotonicity gives $\gamma_{m-1} \le \gamma_m$. Consequently,

$$\begin{aligned} E_m(\pi) &\le \sqrt{31}\frac{A}{\gamma_m} + \sqrt{\frac{2A\mathcal{R}(\pi)}{\gamma_m}} \\ &\le \frac{1}{8}\mathcal{R}(\pi) + 10\frac{A}{\gamma_m}. \end{aligned} \tag{37}$$

Here we used $\sqrt{2xy} \le x/8 + 4y$ and $\sqrt{31} + 4 < 10$.

Applying (37) in (35), with $\mathcal{R}(\pi^\star) = 0$, yields

$$\frac{7}{8}\mathcal{R}(\pi) \le \widehat{\mathcal{R}}_m(\pi) + 20\frac{A}{\gamma_m}.$$

Since $8/7 \le 2$ and $160/7 \le 30$, this proves (33). In particular, $\mathcal{R}(\widehat{\pi}_m) \le 30A/\gamma_m$. Using this bound in (36) gives

$$\begin{aligned} \widehat{\mathcal{R}}_m(\pi) &\le \frac{9}{8}\mathcal{R}(\pi) + 20\frac{A}{\gamma_m} + \frac{1}{8}\mathcal{R}(\widehat{\pi}_m) \\ &\le 2\mathcal{R}(\pi) + 30\frac{A}{\gamma_m}. \end{aligned}$$

This proves (34) and closes the induction. □

For vanilla PO, the lemma applies with $C_0 = e$ and the analysis quantity

$$\gamma_m = AB_m\eta_m = \min\left\{B_m, \sqrt{\frac{AB_{m-1}}{3\operatorname{Err}^{\mathrm{batch}}_{\mathrm{orc}}(T,\delta)}}\right\}.$$

This introduces no additional algorithm parameter: it is determined by the learning rate in (24) and $B_m = 2B_{m-1}$. Both entries in the minimum increase with m, so γ_m is nondecreasing. Since $3 > e$, its square is at most $AB_{m-1}/(e\,\mathrm{Err}^{\mathrm{batch}}_{\mathrm{orc}}(T,\delta))$. Together with (27), this verifies the other conditions of the comparison lemma. The constant C_0 is used only in the analysis: here $C_0 = e$, while the bonus variant satisfies the same conditions with $C_0 = 1$. Combining (33) with (29) therefore gives

$$\mathcal{R}(\pi_m) \le \frac{30 + 4\log A}{B_m \eta_m}. \tag{38}$$

E.5 The cumulative regret bound

Regret bound in Theorem 4. If $T = 1$, the regret is at most one and there are no oracle calls, so both conclusions hold. Assume $T \ge 2$ and fix a realization satisfying (23).

We first bound the regret averaged over the context distribution. The first epoch contributes at most one, and (38) bounds every later epoch. Since epoch m uses the same policy for its n_m rounds,

$$\begin{aligned}\sum_{m=1}^{M} n_m \mathcal{R}(\pi_m) &\le 1 + (30 + 4\log A) \sum_{m=2}^{M} \frac{1}{\eta_m} \\ &\le 1 + (30 + 4\log A)\left(A(M-1) + \sqrt{6A\,\mathrm{Err}^{\mathrm{batch}}_{\mathrm{orc}}(T,\delta)} \sum_{m=2}^{M} \sqrt{B_m} \right).\end{aligned}$$

The first inequality uses $n_m \le B_m$; the second uses

$$\frac{1}{\eta_m} = \max\{A, \sqrt{6AB_m\,\mathrm{Err}^{\mathrm{batch}}_{\mathrm{orc}}(T,\delta)}\} \le A + \sqrt{6AB_m\,\mathrm{Err}^{\mathrm{batch}}_{\mathrm{orc}}(T,\delta)}.$$

Since the epoch lengths double and $B_M \le T$,

$$\sum_{m=1}^{M} \sqrt{B_m} \le \frac{\sqrt{B_M}}{1 - 2^{-1/2}} < 3.5\sqrt{T}. \tag{39}$$

Consequently,

$$\sum_{m=1}^{M} n_m \mathcal{R}(\pi_m) \le 1 + (30 + 4\log A)\big(AM + 9\sqrt{AT\ \mathrm{Err}^{\mathrm{batch}}_{\mathrm{orc}}(T,\delta)}\big).$$

This bound holds on the simultaneous regression event, whose probability is at least $1 - \delta/2$.

The target regret $\overline{R}_T$ evaluates the policies at the contexts actually observed. Applying Proposition 9 to the deployed sequence of round policies, with failure probability $\delta/2$, gives

$$\overline{R}_T \le \sum_{m=1}^{M} n_m \mathcal{R}(\pi_m) + \sqrt{\frac{T}{2}\log\frac{2}{\delta}}.$$

This concentration argument is applied unconditionally, rather than after conditioning on the regression event. A union bound therefore yields, with probability at least $1 - \delta$,

$$\begin{aligned}\overline{R}_T \le\ & 1 + (30 + 4\log A)\big(AM + 9\sqrt{AT\ \mathrm{Err}^{\mathrm{batch}}_{\mathrm{orc}}(T,\delta)}\big) \\ &+ \sqrt{\frac{T}{2}\log\frac{2}{\delta}}.\end{aligned} \tag{40}$$

To simplify the bound, first suppose $A \le \mathrm{Err}^{\mathrm{batch}}_{\mathrm{orc}}(T,\delta)T$. Then $A \le \sqrt{AT\ \mathrm{Err}^{\mathrm{batch}}_{\mathrm{orc}}(T,\delta)}$, so the AM term is absorbed up to logarithmic factors. If instead $A > \mathrm{Err}^{\mathrm{batch}}_{\mathrm{orc}}(T,\delta)T$, then $\sqrt{AT\ \mathrm{Err}^{\mathrm{batch}}_{\mathrm{orc}}(T,\delta)} > T$, and the deterministic bound $\overline{R}_T \le T$ suffices. Because $\mathrm{Err}^{\mathrm{batch}}_{\mathrm{orc}}(T,\delta) \ge 1$, the additional concentration term is also absorbed by logarithmic factors in $1/\delta$. Thus $\overline{R}_T = \widetilde{O}(\sqrt{AT\ \mathrm{Err}^{\mathrm{batch}}_{\mathrm{orc}}(T,\delta)})$. The schedule makes exactly $M - 1 = O(\log T)$ regression calls. □

E.6 Verifying the prediction-accuracy condition

The preceding argument uses only the simultaneous accuracy condition (23). The following lemma explains how ordinary regression guarantees for one fixed-policy batch imply that condition. This is the offline-oracle setting used by FALCON+ (Simchi-Levi & Xu, 2022).

Lemma 20 (Simultaneous accuracy of the batch predictors)**.** *Fix $\delta \in (0, 1/2]$ and a bound $\mathrm{Err}_{\mathrm{orc}}^{\mathrm{batch}}(T, \delta) \geq 1$ before the interaction. Suppose that, for every fixed policy π and every completed batch size $n \in \{B_1, \ldots, B_{M-1}\}$, fitting the oracle to n i.i.d. observations under π gives $n\|\widetilde{f} - f^\star\|_\pi^2 \leq \mathrm{Err}_{\mathrm{orc}}^{\mathrm{batch}}(T, \delta)$ with probability at least $1 - \delta/(2M)$. Then (23) holds simultaneously for all $m = 2, \ldots, M$ with probability at least $1 - \delta/2$. This applies to any sequence of policies fixed before their epochs, using fresh randomness for each action and oracle call.*

Proof. The claim is vacuous when $T = 1$. Otherwise, fix $m \geq 2$ and condition on the history before epoch $m - 1$, including the randomness used to construct π_{m-1}. This policy is now fixed. The new context–loss pairs are i.i.d. and independent of that history, and fresh action randomness makes the observed triples conditionally i.i.d. under π_{m-1}. This batch contains exactly B_{m-1} observations because $m - 1 < M$.

The assumed single-batch guarantee, including the oracle's fresh randomness, bounds the conditional failure probability by $\delta/(2M)$. Averaging over the history gives the same unconditional bound. A union bound over the $M - 1$ fitted predictors yields total failure probability at most $(M - 1)\delta/(2M) \leq \delta/2$. No independence between fits is required. □

Finite-class conclusion of Theorem 4. For a finite realizable class $\mathcal{F}$, ordinary least-squares ERM satisfies Lemma 11 with $\alpha = 0$. Applying that result at failure probability $\delta/(2M)$ shows that the choice

$$\mathrm{Err}_{\mathrm{orc}}^{\mathrm{batch}}(T, \delta) = 10 \log \frac{4|\mathcal{F}|M}{\delta} \tag{41}$$

satisfies Lemma 20. This bound is at least one. Thus all batch predictors meet (23) with probability at least $1 - \delta/2$. Using this bound in (24) and applying (40), including its concentration term, gives $\overline{R}_T = \widetilde{O}(\sqrt{AT \log(|\mathcal{F}|/\delta)})$. □

F Complete proof for private batched vanilla PO

The private algorithm uses the same schedule and policy construction as Algorithm 2, with a private regression oracle. Before epoch $m \geq 2$, it releases $(\widetilde{f}_m, \pi_m)$. The policy map is specified by the private predictor and the public parameters; contexts, selected actions, observed losses, and evaluations at protected contexts remain internal, as in Definition 5.

The regret proof above applies with the private oracle's error bound. It remains to prove privacy of all releases together. Disjoint batches alone do not justify treating future datasets as fixed: changing an early private fit can change later policies and observed rows. The proof instead shows that this later dependence is postprocessing of the single private fit that uses the changed record.

F.1 Privacy of the complete transcript

Lemma 21 (Privacy when each batch is used in one fit)**.** *Consider the fixed epoch schedule of Algorithm 2. Suppose the first policy is uniform and each later epoch policy is a deterministic function of the preceding private predictor and public parameters, remaining fixed while its batch is collected. Each batch except the final one is used in one $(\varepsilon_{\mathrm{priv}}, \delta_{\mathrm{priv}})$-DP oracle call under replacement of one row, then discarded. The final batch is never fitted, and all oracle calls and action draws use fresh randomness. Then the public output*

$$\mathcal{M}(z) = ((\widetilde{f}_m, \pi_m))_{m=2}^{M}$$

is $(\varepsilon_{\mathrm{priv}}, \delta_{\mathrm{priv}})$-DP under replacement of one record $z_t = (c_t, \ell_t(c_t, \cdot))$. This statement holds for arbitrary fixed record streams.

Proof. Fix neighboring deterministic streams z, z' that differ only at round i, in epoch j. The schedule is public and does not depend on the records.

Only one row changes in the affected batch. Condition on a common realization of all learner randomness before epoch j. The earlier records agree, so the states and transcripts before that epoch agree as well. In particular, π_j is the same fixed map in both executions.

Also fix common values of the independent action-selection seeds within epoch j. For the efficient vanilla sampler, these determine both K_t and the sample from its softmax distribution. At every round in this epoch other than i, the record, policy, and action seed are the same, so the observed triple is the same. At round i, changing the record may change the context, selected action, and observed loss, but affects only that row. Thus $\mathcal{S}_j$ and $\mathcal{S}'_j$ differ in at most one row. The distribution of the conditioning variables is identical in the two executions and independent of the changed record.

The remaining transcript is postprocessing of one fit. If $j = M$, no oracle is fitted to the affected batch. Its records therefore have no effect on the released transcript.

Otherwise, the oracle call producing $\widetilde{f}_{j+1}$ is $(\varepsilon_{\mathrm{priv}}, \delta_{\mathrm{priv}})$-DP on the two neighboring batches. The map π_{j+1} is a deterministic function of that output and public parameters. After this call, the affected batch is discarded.

For the fixed earlier state and fixed future record stream, there is a single randomized continuation procedure that maps $\widetilde{f}_{j+1}$ to all subsequent releases: it constructs the policies, interacts with the future records, and runs the later oracles with fresh randomness. This procedure is the same in both executions because all records after epoch j agree. Later actions and collected rows need not agree; they are generated by this common procedure from its potentially different private input.

By postprocessing, for any event E involving the complete released transcript and any fixed value v of the common conditioning variables,

$$\mathbb{P}(\mathcal{M}(z) \in E \mid v) \le e^{\varepsilon_{\mathrm{priv}}} \mathbb{P}(\mathcal{M}(z') \in E \mid v) + \delta_{\mathrm{priv}}.$$

The earlier outputs are already fixed as part of v and can be included in E. Integrating over the common distribution of v gives (6), with the same additive δ_{priv}. No oracle-accuracy event or stochastic assumption on the record stream was used. □

Proof of Theorem 7. Each epoch uses a fixed policy, and its batch enters at most one private fit. Thus Lemma 21 gives privacy of the transcript. For regret, Assumption 6 supplies (23) with probability at least $1 - \delta/2$. The proof of Theorem 4, using the private oracle's error bound in (24), first bounds $\sum_{t=1}^{T} \mathcal{R}(\pi_t)$ on this event. The concentration step, with failure probability $\delta/2$, then gives the stated bound on $\overline{R}_T$ with probability at least $1 - \delta$. The schedule makes exactly $M - 1$ private oracle calls, including zero when $T = 1$. □

The absence of a composition factor comes from using each batch in only one private fit and discarding it afterward. The doubling schedule controls the oracle count and regret; privacy itself does not require geometrically increasing batch lengths. In contrast, refitting after every round on all observations would use the same record repeatedly and would require a different privacy analysis.

F.2 Finite-class private regression

Let $\mathcal{F}$ be finite and realizable, and fix $\delta \in (0, 1/2]$ and $\varepsilon_{\mathrm{priv}} > 0$. Use the exponential mechanism in (8) on each completed training batch. By Lemma 12, every fit is pure $\varepsilon_{\mathrm{priv}}$-DP under row replacement. Applying its regression guarantee at failure probability $\delta/(2M)$ gives the choice

$$\mathrm{Err}_{\mathrm{orc}}^{\mathrm{batch}}(T, \delta) = \left(10 + \frac{4}{\varepsilon_{\mathrm{priv}}}\right) \log \frac{4|\mathcal{F}|M}{\delta}. \tag{42}$$

This is a known bound of at least one; its dependence on $\varepsilon_{\mathrm{priv}}$ is implicit in the oracle notation. By Lemma 20, all fitted predictors satisfy (23) with probability at least $1 - \delta/2$.

For batched vanilla PO, use this bound in the explicit learning rates (24). The regret analysis in Section E then gives, with probability at least $1-\delta$,

$$\overline{R}_T = \widetilde{O}\Big(\sqrt{AT\log(|\mathcal{F}|/\delta)}\big(1+\varepsilon_{\mathrm{priv}}^{-1/2}\big)\Big).$$

Independently of this accuracy event, Lemma 21 gives pure $\varepsilon_{\mathrm{priv}}$-DP for the complete transcript. This proves Corollary 8. The same oracle bound also applies to batched bonus PO, with its parameters specified in Section G. The nonprivate ERM choice is given in (41).

G Policy optimization with an exploration bonus

G.1 From the original bonus to a batched algorithm

The optimistic policy-optimization algorithm of Levy & Mansour (2026) subtracts an exploration bonus from the predicted loss before clipping at zero and applying an exponential update. Its bonus decreases with the sum of the probabilities that earlier policies assign to an action at the same context. An action with little past probability receives a larger bonus, which makes it more attractive to the policy. The original algorithm fits predictors throughout the interaction.

Our batched adaptation fits one predictor per epoch from the preceding batch and holds it fixed. It constructs an auxiliary sequence of bonus-adjusted exponential updates, starting from uniform, and deploys their average throughout the epoch. The probabilities entering the bonus are those of this auxiliary sequence. These updates require no additional observations or oracle calls.

We also subtract the minimum predicted loss before applying the bonus and clipping. This ensures that a predicted best action always has adjusted loss zero, which is used in the proof below. Centering before clipping can change the update, so it is an explicit part of the algorithm specification. The theorem below analyzes this centered, batched construction.

G.2 The batched bonus policy

Fix $\delta \in (0, 1/2]$ and use the doubling schedule of Algorithm 2, with a uniform first-epoch policy and one fit to the preceding batch. Let the known bound $\mathrm{Err}_{\mathrm{orc}}^{\mathrm{batch}}(T,\delta) \geq 1$ satisfy (23) for these policies with probability at least $1-\delta/2$. As in Section E, the remaining failure probability $\delta/2$ is reserved for concentration over contexts. For each $m \geq 2$, set

$$\begin{aligned}\gamma_m &= \min\left\{B_m, \sqrt{\frac{AB_{m-1}}{\mathrm{Err}_{\mathrm{orc}}^{\mathrm{batch}}(T,\delta)}}\right\}, \quad \Lambda_m = \frac{B_m}{\gamma_m},\\ \eta_m &= \min\left\{1, \sqrt{\frac{2\log A}{B_m}}\right\}.\end{aligned} \tag{43}$$

Here γ_m controls exploration, Λ_m sets the bonus scale, and η_m is the learning rate for the auxiliary updates. The learning rate here is specific to the bonus algorithm.

For every context c, define $d_m(c,a)$ by (26), initialize $\pi_{m,1}(c,a) = 1/A$, and let

$$N_{m,s}(c,a) = \sum_{r=1}^{s} \pi_{m,r}(c,a), \qquad N_{m,0}(c,a) = 0. \tag{44}$$

The index s runs over auxiliary policies within an epoch's construction: $\pi_{m,s}$ is obtained after $s-1$ updates. The quantity $N_{m,s}(c,a)$ sums the probabilities assigned to action a by the first s auxiliary policies at the same context c; it does not count observed actions in a batch. For $s = 1, \ldots, B_m$, define

$$\begin{aligned}b_{m,s}(c,a) &= \min\left\{1, \frac{2\Lambda_m}{1+N_{m,s-1}(c,a)}\right\},\\ u_{m,s}(c,a) &= [d_m(c,a) - b_{m,s}(c,a)]_+,\\ \pi_{m,s+1}(c,a) &= \frac{\pi_{m,s}(c,a)\exp\{-\eta_m u_{m,s}(c,a)\}}{\sum_{b\in\mathcal{A}} \pi_{m,s}(c,b)\exp\{-\eta_m u_{m,s}(c,b)\}},\end{aligned} \tag{45}$$

where $[x]_+ = \max\{x, 0\}$. The deployed policy is

$$\pi_m(c, a) = \frac{1}{B_m} \sum_{s=1}^{B_m} \pi_{m,s}(c, a) = \frac{N_{m,B_m}(c, a)}{B_m}. \tag{46}$$

When an action has accumulated little probability, its bonus is large and its adjusted loss $u_{m,s}(c, a)$ can be zero. Its weight then remains unchanged while weights of other actions can decrease. As it accumulates probability, the bonus decreases and its predicted gap has more influence on the update.

Algorithm 3 Batched Policy Optimization with an Exploration Bonus

Input: horizon T, action set $\mathcal{A}$, confidence δ, a regression oracle, and a known bound $\mathrm{Err}_{\mathrm{orc}}^{\mathrm{batch}}(T, \delta)$ for (23).
Set $M \leftarrow \lceil \log_2(T + 1) \rceil$ and $B_m \leftarrow 2^{m-1}$ for $m = 1, \ldots, M$.
Initialize $\pi_1(c, a) \leftarrow 1/A$ for every $c \in \mathcal{C}$, $a \in \mathcal{A}$.
for $m = 1, \ldots, M$ **do**
  if $m \geq 2$ **then**
    Fit $\widetilde{f}_m \leftarrow \mathrm{Oracle}(\mathcal{S}_{m-1})$ and discard $\mathcal{S}_{m-1}$.
    Set $\gamma_m, \Lambda_m, \eta_m$ by (43).
    Represent π_m by $\widetilde{f}_m$ and the public construction (44)–(46).
    In the private version, release $(\widetilde{f}_m, \pi_m)$.
  end if
  Initialize $\mathcal{S}_m \leftarrow ()$.
  for $t = B_m, \ldots, \min\{2B_m - 1, T\}$ **do**
    Observe c_t and evaluate $\pi_m(c_t, \cdot)$ by the procedure below.
    Sample $a_t \sim \pi_m(c_t, \cdot)$ with fresh randomness and observe $\ell_t(c_t, a_t)$.
    Append $(c_t, a_t, \ell_t(c_t, a_t))$ to $\mathcal{S}_m$.
  end for
end for
Policy evaluation for epoch m at context c:
  If $m = 1$, return the uniform distribution.
  Otherwise, evaluate $\widetilde{f}_m(c, a)$ for all $a \in \mathcal{A}$ and form $d_m(c, a)$.
  Initialize $\pi_{m,1}(c, a) \leftarrow 1/A$ and $N_{m,0}(c, a) \leftarrow 0$ for every action.
  for $s = 1, \ldots, B_m$ **do**
    Compute $b_{m,s}(c, a)$, $u_{m,s}(c, a)$, and $\pi_{m,s+1}(c, a)$ by (45).
    Set $N_{m,s}(c, a) \leftarrow N_{m,s-1}(c, a) + \pi_{m,s}(c, a)$ for each action.
  end for
  Return $\pi_m(c, a) = N_{m,B_m}(c, a)/B_m$.

Implementation. The predictor and policy map remain fixed throughout an epoch. The evaluation procedure computes their values only at the queried context, using A predictor evaluations and $O(AB_m)$ arithmetic. The sums $N_{m,s}(c, a)$ are initialized anew for each evaluation; they are not carried over between bandit rounds. Repeating an evaluation at the same context therefore gives the same distribution. The update producing π_{m,B_m+1} can be omitted, because that iterate is not included in the average. A shortened final epoch still uses all B_M terms in (46), and its batch is never fitted.

G.3 Regret and privacy guarantees

Theorem 22 (Batched bonus PO)**.** *Fix $\delta \in (0, 1/2]$ and suppose (23) holds with probability at least $1 - \delta/2$, for a known bound $\mathrm{Err}_{\mathrm{orc}}^{\mathrm{batch}}(T, \delta) \geq 1$ fixed before the interaction. Then Algorithm 3, with*

parameters (43), makes exactly $M-1$ oracle calls and, with probability at least $1-\delta$, satisfies

$$\overline{R}_T \le 1 + \big(32 + 8\log(T+1)\big)\big(AM + 5\sqrt{AT\ \mathrm{Err}^{\mathrm{batch}}_{\mathrm{orc}}(T,\delta)}\big) + \sqrt{\frac{T}{2}\log\frac{2}{\delta}}. \tag{47}$$

Together with $\overline{R}_T \le T$, this gives $\overline{R}_T = \widetilde{O}(\sqrt{AT\ \mathrm{Err}^{\mathrm{batch}}_{\mathrm{orc}}(T,\delta)})$. If the oracle also satisfies Assumption 6, then the released transcript is $(\varepsilon_{\mathrm{priv}}, \delta_{\mathrm{priv}})$-DP for arbitrary fixed record streams, with no change to the regret guarantee.

Proof overview. The bonus keeps an action's unnormalized weight at its initial value until that action has accumulated enough probability. Either it receives this much probability or its probability remains at least $1/A$ throughout the auxiliary sequence. This gives an explicit lower bound on the averaged policy. To bound predicted regret, we add the exponential-weights bound for the adjusted losses to the total bonuses. Since each bonus decreases as accumulated probability grows, its probability-weighted sum is bounded by the bonus scale times a logarithmic factor. The probability bound lets us apply Lemma 19; the predicted-regret bound then controls the population regret in each epoch. Summing over epochs and applying concentration gives the bound on $\overline{R}_T$. Privacy follows from the same argument as for batched vanilla PO: each batch enters one private fit, and subsequent releases depend on that batch only through the private predictor.

Lemma 23 (Action probabilities and predicted regret of batched bonus PO)**.** *For every epoch $m \ge 2$, context c, and action a,*

$$\pi_m(c,a) \ge \min\left\{\frac{1}{A}, \frac{1}{\gamma_m d_m(c,a)}\right\}, \qquad \pi_m(c,a)^{-1} \le A + \gamma_m d_m(c,a), \tag{48}$$

where the second term in the minimum is $+\infty$ when $d_m(c,a) = 0$. Moreover,

$$\widehat{\mathcal{R}}_m(\pi_m) \le \big(1 + 4\log(B_m+1)\big)\frac{A}{\gamma_m}. \tag{49}$$

Proof. Fix m and c. Define unnormalized weights by $w_{m,1}(c,a) = 1$ and

$$w_{m,s+1}(c,a) = w_{m,s}(c,a)\exp\{-\eta_m u_{m,s}(c,a)\}.$$

These weights never increase, since $u_{m,s}(c,a) \ge 0$, and $\pi_{m,s}(c,a) = w_{m,s}(c,a)/\sum_b w_{m,s}(c,b)$.

Lower bounds on action probabilities. If $d_m(c,a) = 0$, then $u_{m,s}(c,a) = 0$ at every step. Hence $w_{m,s}(c,a) = 1$ and $\pi_{m,s}(c,a) \ge 1/A$ throughout.

Now suppose $d_m(c,a) > 0$. Since $\Lambda_m \ge 1 \ge d_m(c,a)$, whenever $N_{m,s-1}(c,a) < \Lambda_m/d_m(c,a)$ we have

$$\frac{2\Lambda_m}{1 + N_{m,s-1}(c,a)} > \frac{2\Lambda_m d_m(c,a)}{\Lambda_m + d_m(c,a)} \ge d_m(c,a).$$

Capping this quantity at one still leaves it at least $d_m(c,a)$, so $u_{m,s}(c,a) = 0$. The counts are nondecreasing, so all preceding adjusted losses of this action were also zero. Its weight is therefore still one, and $\pi_{m,s}(c,a) \ge 1/A$.

Either $N_{m,B_m}(c,a) \ge \Lambda_m/d_m(c,a)$, or this threshold is never reached and $\pi_{m,s}(c,a) \ge 1/A$ at every step. In both cases,

$$N_{m,B_m}(c,a) \ge \min\left\{\frac{B_m}{A}, \frac{\Lambda_m}{d_m(c,a)}\right\}.$$

Divide by B_m and use $\Lambda_m/B_m = 1/\gamma_m$ to obtain the first inequality in (48). Taking reciprocals and using $\max\{A, \gamma_m d_m(c,a)\} \le A + \gamma_m d_m(c,a)$ gives the second.

Regret for the adjusted losses. Choose an action a_0 with $d_m(c, a_0) = 0$. It has adjusted loss zero and weight one at every step. Thus the total weight is initially A and is at least one after all B_m updates. Applying the same log-weight calculation as in the proof of Lemma 17, now to the varying adjusted losses, gives

$$\sum_{s=1}^{B_m} \sum_{a \in \mathcal{A}} \pi_{m,s}(c,a) u_{m,s}(c,a) \le \frac{\log A}{\eta_m} + \frac{\eta_m}{2} \sum_{s=1}^{B_m} \sum_{a \in \mathcal{A}} \pi_{m,s}(c,a) u_{m,s}(c,a)^2 \le \frac{\log A}{\eta_m} + \frac{\eta_m B_m}{2}, \tag{50}$$

because every adjusted loss lies in $[0, 1]$.

Adding back the bonuses. The definition of the adjusted loss gives $d_m(c, a) \le u_{m,s}(c, a) + b_{m,s}(c, a)$. For each action, the identity $N_{m,s}(c, a) = N_{m,s-1}(c, a) + \pi_{m,s}(c, a)$ and the inequality $x \le 2 \log(1 + x)$ for $0 \le x \le 1$ yield

$$\begin{aligned} \sum_{s=1}^{B_m} \pi_{m,s}(c,a) b_{m,s}(c,a) &\le 2\Lambda_m \sum_{s=1}^{B_m} \frac{\pi_{m,s}(c,a)}{1 + N_{m,s-1}(c,a)} \\ &\le 4\Lambda_m \sum_{s=1}^{B_m} \log \frac{1 + N_{m,s}(c,a)}{1 + N_{m,s-1}(c,a)} \\ &= 4\Lambda_m \log\bigl(1 + N_{m,B_m}(c,a)\bigr) \\ &\le 4\Lambda_m \log(B_m + 1). \end{aligned}$$

The logarithms telescope because the probability added to the count at each step is the same probability multiplying the bonus. Sum over actions, combine with (50), divide by B_m, and average over contexts to get

$$\widehat{\mathcal{R}}_m(\pi_m) \le \frac{\log A}{\eta_m B_m} + \frac{\eta_m}{2} + \frac{4A \log(B_m + 1)}{\gamma_m}. \tag{51}$$

If $B_m < 2 \log A$, then $\gamma_m \le B_m < 2 \log A \le A$, so the simpler bound $\widehat{\mathcal{R}}_m(\pi_m) \le 1 \le A/\gamma_m$ already proves (49). Otherwise $\eta_m = \sqrt{2 \log A / B_m}$, and the first two terms in (51) sum to $\sqrt{2 \log A / B_m}$. By (43), $B_m = 2B_{m-1}$, and $\mathrm{Err}_{\mathrm{orc}}^{\mathrm{batch}}(T, \delta) \ge 1$,

$$\gamma_m \sqrt{\frac{2 \log A}{B_m}} \le \sqrt{\frac{A \log A}{\mathrm{Err}_{\mathrm{orc}}^{\mathrm{batch}}(T, \delta)}} \le \sqrt{A \log A} \le A.$$

Thus these two terms are at most A/γ_m. Substitution into (51) proves (49). □

Proof of Theorem 22. As in Section E.1, the schedule makes exactly $M - 1$ fits, and every batch used in a fit is complete. If $T = 1$, the only policy is uniform and the regret is at most one.

For $T \ge 2$, work on the simultaneous accuracy event (23), which has probability at least $1 - \delta/2$ by hypothesis. The sequence $(\gamma_m)_{m=2}^{M}$ in (43) is nondecreasing and satisfies

$$\gamma_m^2 \le \frac{A B_{m-1}}{\mathrm{Err}_{\mathrm{orc}}^{\mathrm{batch}}(T, \delta)}.$$

Together with (48), this verifies (32) with $C_0 = 1$. Thus Lemma 19 applies to every comparison policy, in particular the deployed policy π_m. Combining it with (49) gives

$$\mathcal{R}(\pi_m) \le \bigl(32 + 8 \log(B_m + 1)\bigr) \frac{A}{\gamma_m} \le \bigl(32 + 8 \log(T + 1)\bigr) \frac{A}{\gamma_m}.$$

Let $n_m = \min\{B_m, T - B_m + 1\}$ be the number of rounds in epoch m. Using $n_m \le B_m$, the first epoch's population regret bound of one, and

$$\frac{A B_m}{\gamma_m} = \max\{A, \sqrt{2A \, \mathrm{Err}_{\mathrm{orc}}^{\mathrm{batch}}(T, \delta) B_m}\} \le A + \sqrt{2A \, \mathrm{Err}_{\mathrm{orc}}^{\mathrm{batch}}(T, \delta) B_m},$$

we obtain

$$\sum_{m=1}^{M} n_m \mathcal{R}(\pi_m) \le 1 + \big(32 + 8\log(T+1)\big) \sum_{m=2}^{M} \frac{AB_m}{\gamma_m}$$
$$\le 1 + \big(32 + 8\log(T+1)\big)\big(AM + 5\sqrt{AT\ \mathrm{Err}_{\mathrm{orc}}^{\mathrm{batch}}(T,\delta)}\big).$$

The last step uses (39) and $3.5\sqrt{2} < 5$. This bounds the regret averaged over contexts on the accuracy event.

Each deployed policy is determined before its epoch begins, so Proposition 9, applied with failure probability $\delta/2$, gives

$$\overline{R}_T \le \sum_{m=1}^{M} n_m \mathcal{R}(\pi_m) + \sqrt{\frac{T}{2}\log\frac{2}{\delta}}$$

with probability at least $1 - \delta/2$. This concentration statement holds without conditioning on the oracle's accuracy event. A union bound with that event proves (47) with probability at least $1 - \delta$. Combining this with $\overline{R}_T \le T$ gives the stated $\widetilde{O}$ bound, as in the proof of Theorem 4; the additional concentration term is absorbed up to logarithmic factors in $1/\delta$, because $A \ge 2$ and $\mathrm{Err}_{\mathrm{orc}}^{\mathrm{batch}}(T,\delta) \ge 1$.

For privacy, the full policy construction (44)–(46) is a deterministic function of $\widetilde{f}_m$ and public parameters. In particular, its auxiliary counts use no raw batch observations. Each batch is collected under a fixed map and used in only one private fit. The conditions of Lemma 21 therefore hold, proving privacy of the full transcript for arbitrary fixed streams, independently of the accuracy event. □

Corollary 24 (Finite-class batched bonus PO)**.** *Let $\mathcal{F}$ be finite and realizable, and fix $\delta \in (0, 1/2]$. With least-squares ERM and the error bound (41), batched bonus PO satisfies*

$$\overline{R}_T = \widetilde{O}\Big(\sqrt{AT\log(|\mathcal{F}|/\delta)}\Big)$$

with probability at least $1 - \delta$. For any $\varepsilon_{\mathrm{priv}} > 0$, with the exponential mechanism in (8) and $\mathrm{Err}_{\mathrm{orc}}^{\mathrm{batch}}(T,\delta)$ from (42), its transcript is pure $\varepsilon_{\mathrm{priv}}$-DP and, with probability at least $1 - \delta$,

$$\overline{R}_T = \widetilde{O}\Big(\sqrt{AT\log(|\mathcal{F}|/\delta)}\big(1 + \varepsilon_{\mathrm{priv}}^{-1/2}\big)\Big).$$

Proof. The oracle bounds in Section C, applied across epochs by Lemma 20, verify the accuracy condition of Theorem 22 with the stated error bounds. Use each bound in the bonus parameters (43). Substituting the corresponding $\mathrm{Err}_{\mathrm{orc}}^{\mathrm{batch}}(T,\delta)$ proves the regret bounds. For the exponential mechanism, $\delta_{\mathrm{priv}} = 0$, so the transcript guarantee is pure DP. □

H Experiments

Following the contextual-bandit evaluation protocol of Bietti et al. (2021), we evaluate the algorithms on randomly permuted multiclass classification datasets from OpenML. The following sections describe the regression oracles and their private counterparts, the algorithm implementations and hyperparameter tuning, the experimental protocol, and the resulting empirical comparisons.

H.1 Function classes and regression objectives

We first describe the feature representation and regression oracles used throughout the experiments. In the quadratic-loss setting, we use linear predictors fitted by least squares, while in the logistic-loss setting, we use logistic predictors fitted by logistic regression. All methods use the same feature construction and regression objectives; they differ only in the observations supplied to the oracle and, for the private variants, how the fit is privatized.

Action-dependent features. We represent each context–action pair using a block-structured feature map. Given a context $c \in \mathbb{R}^{d_0}$ and an action $a \in \mathcal{A} = [A]$, we first project the context onto the unit Euclidean ball and then place it in the block corresponding to action a:

$$\phi(c, a) = e_a \otimes \bar{c} \in \mathbb{R}^d, \qquad \bar{c} = \frac{c}{\max\{1, \|c\|_2\}}, \qquad d = Ad_0,$$

where $e_a \in \mathbb{R}^A$ is the ath standard basis vector and $\otimes$ denotes the Kronecker product.[2] Thus, $\phi(c, a)$ has $\bar{c}$ in its ath block and zeros elsewhere, and in particular $\|\phi(c, a)\|_2 \le 1$.

Function classes. We consider linear and logistic predictors over these features:

$$\mathcal{F}_{\text{lin}} = \left\{(c, a) \mapsto \phi(c, a)^\top \theta : \theta \in \mathbb{R}^d\right\}, \qquad \mathcal{F}_{\log} = \left\{(c, a) \mapsto \sigma\big(\phi(c, a)^\top \theta\big) : \theta \in \mathbb{R}^d\right\},$$

where $\sigma(s) = (1 + e^{-s})^{-1}$ is the sigmoid function.

Batching and regression objectives. All algorithms follow the same doubling schedule, with batch j having size $B_j = 2^{j-1}$, up to truncation of the final batch. Let $n_j = B_j$ for the bandit algorithms and $n_j = AB_j$ for the supervised reference. For each completed batch, let

$$X_j \in \mathbb{R}^{n_j \times d}, \qquad y_j \in \mathbb{R}^{n_j},$$

denote the corresponding feature matrix and vector of observed 0–1 classification losses, where the loss is 0 for a correct prediction and 1 otherwise. For a bandit algorithm, each round contributes the single observation $(\phi(c_t, a_t), \ell_t(a_t))$, whereas the supervised reference contributes A observations.

For the linear class, define the regularized quadratic objective

$$L_j^{\text{lin}}(\theta) = \frac{1}{2n_j}\|X_j\theta - y_j\|_2^2 + \frac{\lambda}{2}\|\theta\|_2^2. \tag{52}$$

For the logistic class, define the regularized logistic objective

$$L_j^{\log}(\theta) = \frac{1}{n_j}\sum_{i=1}^{n_j}\left[\log\left(1 + e^{x_{ji}^\top\theta}\right) - y_{ji}x_{ji}^\top\theta\right] + \frac{\lambda}{2}\|\theta\|_2^2, \tag{53}$$

where $x_{ji}^\top$ denotes row i of X_j. Both objectives are averaged over the n_j observations available to the oracle, so the same definitions apply directly to the supervised reference.

H.2 Private regression mechanisms

We use two standard mechanisms to privatize the regression oracles. For the quadratic objective, we perturb the sufficient statistics to obtain an (ε, δ)-DP predictor, while for the logistic objective, we use objective perturbation to obtain a pure ε-DP predictor. The corresponding non-private oracles use the same regression procedures with all perturbations set to zero.

Privacy is defined at the level of one original context–label record. A bandit record contributes one fitting row, whereas a supervised record contributes A rows. Since the batches are disjoint, releases across batches compose in parallel, and all subsequent policy computations are post-processing of the private oracle outputs.

Private quadratic oracle. For batch j, let

$$G_j = X_j^\top X_j, \qquad q_j = X_j^\top y_j.$$

Their record-level sensitivities in Frobenius and Euclidean norm are

$$(\Delta_G, \Delta_q) = \begin{cases} (\sqrt{2},\ 2), & \text{bandit feedback,} \\ (\sqrt{2A},\ 2\sqrt{A-1}), & \text{supervised feedback,} \end{cases}$$

where the supervised response bound uses the single-label 0–1 loss structure.

[2] For simplicity, the notation suppresses the constant bias coordinate used in the implementation. The bound $\|\phi(c, a)\|_2 \le 1$ applies to the full feature vector, including this coordinate.

To calibrate the Gaussian noise, define

$$\rho(\varepsilon,\delta)=\left(\sqrt{\log(1/\delta)+\varepsilon}-\sqrt{\log(1/\delta)}\right)^2.$$

We split this zCDP budget equally between the two sufficient statistics and release

$$\widetilde{G}_j=G_j+E_j,\qquad \widetilde{q}_j=q_j+Z_j,\qquad \widehat{\theta}^{\mathrm{lin}}_{j+1}=(\widetilde{G}_j+n_j\lambda I)^{-1}\widetilde{q}_j, \tag{54}$$

where

$$Z_j\sim\mathcal{N}\Big(0,\tfrac{\Delta_q^2}{\rho(\varepsilon,\delta)}I_d\Big),\qquad E_j=\tfrac{W_j+W_j^\top}{2},$$

and the entries of W_j are independent $\mathcal{N}(0,\Delta_G^2/\rho(\varepsilon,\delta))$ variables. In the Frobenius geometry of symmetric matrices, this assigns half of the total zCDP budget to each statistic. Their joint release is therefore (ε,δ)-DP (Bun & Steinke, 2016), and the resulting predictor is private by post-processing.

Private logistic oracle. Let ε_P denote the privacy budget allocated to the predictor and set

$$h=\begin{cases}1, & \text{bandit feedback},\\ A, & \text{supervised feedback},\end{cases}\qquad \varepsilon_{\mathrm{fit}}=\frac{\varepsilon_P}{h}.$$

The logistic loss has gradient norm at most one and scalar second derivative at most $c_0=1/4$. Following Chaudhuri et al. (2011), define

$$u_j=\varepsilon_{\mathrm{fit}}-2\log\Big(1+\tfrac{c_0}{n_j\lambda}\Big),$$

and set

$$(\kappa_j,\varepsilon_j')=\begin{cases}(0,u_j), & u_j>0,\\ \left(\dfrac{c_0}{n_j(e^{\varepsilon_{\mathrm{fit}}/4}-1)}-\lambda,\dfrac{\varepsilon_{\mathrm{fit}}}{2}\right), & u_j\le 0.\end{cases}$$

We draw $b_j\in\mathbb{R}^d$ with density

$$p(b_j)\propto\exp\left(-\frac{\varepsilon_j'}{2}\|b_j\|_2\right)$$

and return the minimizer

$$\widehat{\theta}^{\log}_{j+1}=\arg\min_{\theta\in\mathbb{R}^d}\left\{L_j^{\log}(\theta)+\frac{\kappa_j}{2}\|\theta\|_2^2+\frac{b_j^\top\theta}{n_j}\right\}. \tag{55}$$

The resulting estimator is $\varepsilon_{\mathrm{fit}}$-DP under replacement of one fitting row; group privacy therefore gives pure ε_P-DP at the original-record level. We solve the perturbed objective numerically using L-BFGS, a limited-memory quasi-Newton method (Liu & Nocedal, 1989).[3]

Private confidence radii. Our implementations of several algorithms rely on confidence radii constructed from empirical Gram matrices. Given privatized Gram matrices $(\widetilde{G}_i)_{i=1}^j$, we use the radius

$$r_j(c,a)=\sqrt{\max\left\{\phi(c,a)^\top(\textstyle\sum_{i=1}^j\widetilde{G}_i+I)^{-1}\phi(c,a),0\right\}}.$$

For the quadratic oracle, privatized $\widetilde{G}_j$ are already released as part of (54) and can therefore be reused without any additional privacy cost. For the logistic oracle, algorithms requiring confidence radii allocate $\varepsilon_P=\varepsilon/2$ to the predictor and $(\varepsilon/2,\delta)$ to a separate Gaussian release of G_j. Specifically, we release $\widetilde{G}_j=G_j+E_j$ with symmetric Gaussian noise of Frobenius-coordinate standard deviation $\sigma_G=\Delta_G/\sqrt{2\rho(\varepsilon/2,\delta)}$. Here $\Delta_G=\sqrt{2}$ for the bandit algorithms considered. The predictor and Gram matrix are therefore jointly (ε,δ)-DP. Logistic methods that do not require confidence radii instead allocate the full budget $\varepsilon_P=\varepsilon$ to the predictor, yielding pure ε-DP.

[3]The stated privacy guarantee corresponds to the exact minimizer; L-BFGS approximates this solution.

H.3 Algorithm implementations and hyperparameter tuning

Here we describe the implementation choices and algorithm-specific hyperparameters used in the experiments. All algorithms share the regression regularization parameter λ; the remaining hyperparameters are specific to the policy. At each policy update, the algorithms use the released loss predictions $\widehat{f}_a = \widehat{f}(c, a)$ and, when required, the confidence radius $r_a = r_j(c, a)$ defined in the previous section. All subsequent policy computations are therefore post-processing.

New algorithms. `VanillaPO` (Algorithm 2) uses only the predicted losses $\widehat{f}_a$. We parameterize its learning rate as $\eta_m = \eta_0 \min\{1/A, 1/\sqrt{AB_m}\}$, where η_0 is the tunable hyperparameter.

A direct implementation of `BonusPO` (Algorithm 3) would require $O(T^2)$ computation. We therefore use a historical-replay modification with $O(T \log T)$ computation, defined as follows. For a context c in batch m, let $f_{j,a} = \widehat{f}_{j+1}(c, a)$ denote the prediction released after batch j. Initialize $q_0(a) = 1/A$ and $D_0(a) = 0$. At replay step j, let $T_j = \sum_{i=1}^{j} B_i$ and define

$$b_{j,a} = \min\left\{1,\ \frac{\gamma\sqrt{T_j/A}}{1+D_{j-1}(a)}\right\}, \qquad \widetilde{f}_{j,a} = \max\left\{0,\ f_{j,a} - b_{j,a}\right\}.$$

The policy and cumulative action mass are then updated as

$$q_j(a) \propto q_{j-1}(a)\exp\left(-\eta B_j \widetilde{f}_{j,a}\right), \qquad D_j(a) = D_{j-1}(a) + B_j q_{j-1}(a).$$

The policy used in batch m is q_{m-1}. The hyperparameters are the learning rate η and bonus scale γ.

Prediction-only baselines. `SquareCB` and `FastCB` use only the predicted losses $\widehat{f}_a$. For both algorithms, we use a batch-wise schedule $\gamma_m = \gamma_0(\sum_{j<m} B_j)^\rho$, where γ_0 controls the overall exploration scale and ρ controls how quickly the policy becomes more concentrated as additional data are collected. Larger values of γ_m place less probability on actions with larger predicted loss gaps. We evaluate `SquareCB` with both quadratic and logistic oracles. `FastCB`, whose allocation rule is designed for log-loss regression, is evaluated only with the logistic oracle.

Exploration-radius baselines. Confidence-based contextual bandit algorithms use action-dependent uncertainty estimates to guide exploration. In our implementations of `LinUCB`, `RegCB`, and `AdaCB`, we use the common elliptical radius r_a as a proxy for this uncertainty. `LinUCB` selects actions by minimizing the optimistic loss score $\widehat{f}_a - \beta r_a$, where β scales the exploration bonus; we use this rule with both quadratic and logistic oracles. `RegCB` forms raw scores $g_a = \phi(c, a)^\top \widehat{\theta}$ and retains the survivor set $\mathcal{S} = \{a : g_a - \beta r_a \le \min_b(g_b + \beta r_b)\}$, sampling uniformly over $\mathcal{S}$. Here β scales the confidence widths, so, for fixed predictions and radii, larger values can retain more actions. For the logistic oracle, these comparisons are still performed in raw-score space; monotonicity of the sigmoid preserves the corresponding ordering after applying the link function. `AdaCB` uses the same survivor set, then applies the `SquareCB` allocation to the predicted losses $\widehat{f}_a$ restricted to $\mathcal{S}$, with zero probability outside the set. Its parameters γ_0 and ρ have the same roles as for `SquareCB`, while β controls the elimination step.

Supervised reference. `Supervised` serves as a full-feedback reference, observing the losses of all A actions at each context and fitting the same regression oracles as above. It selects uniformly among actions minimizing the predicted loss and has no additional policy hyperparameters.

H.4 Experimental protocol

We describe the datasets, privacy settings, evaluation metric, and model-selection procedure used throughout the experiments.

Datasets. We evaluate the algorithms on 100 unique OpenML classification datasets,[4] spanning a broad range of problem sizes and dimensions: the datasets contain between 1,000 and 89,640

[4] OpenML (`https://www.openml.org/`) contains multiple versions or copies of some underlying datasets. We screened the collection to avoid repeated datasets and ensure diversity across the benchmark.

Algorithm	Policy hyperparameters	λ	Oracle loss
VanillaPO	$\eta_0 \in 2^{\{1,3,5,7,9\}}$	$10^{\{-6,-4,-2,-1,0\}}$	$\ell_{\mathrm{sq}}, \ell_{\log}$
BonusPO	$\eta \in 10^{\{-3,-2,-1,0\}},\ \gamma \in 10^{\{-3,-2,-1\}}$	$10^{\{-6,-4,-2,-1,0\}}$	$\ell_{\mathrm{sq}}, \ell_{\log}$
SquareCB	$\gamma_0 \in 2^{\{0,2,4,6,8\}},\ \rho \in \{0.25, 0.5\}$	$10^{\{-6,-4,-2,-1,0\}}$	$\ell_{\mathrm{sq}}, \ell_{\log}$
FastCB	$\gamma_0 \in 10^{\{-1,0,1,2,3\}},\ \rho \in \{0.25, 0.5\}$	$10^{\{-6,-4,-2,-1,0\}}$	$\ell_{\log}$
LinUCB	$\beta \in 2^{\{-12,-8,-7,-5,-2,0,1,2,3\}}$	$10^{\{-6,-4,-2,-1,0\}}$	$\ell_{\mathrm{sq}}, \ell_{\log}$
RegCB	$\beta \in 10^{\{-5,-3,-2,-1,0,1\}}$	$10^{\{-6,-4,-2,-1,0\}}$	$\ell_{\mathrm{sq}}, \ell_{\log}$
AdaCB	$\gamma_0 \in 10^{\{-1,0,1,2\}},\ \rho \in \{0.25, 0.5\},\ \beta \in 10^{\{-2,-1,0,1\}}$	$10^{\{-6,-4,-2,-1,0\}}$	$\ell_{\mathrm{sq}}, \ell_{\log}$
Supervised	–	$10^{\{-6,-4,-2,-1,0\}}$	$\ell_{\mathrm{sq}}, \ell_{\log}$

Table 1: Hyperparameter grids used in the experiments.

examples and between 2 and 10 classes. We interpret each example as one contextual-bandit round and each class as an action. For each run, we use the full dataset, shuffled without replacement.

Evaluation settings. We evaluate each method in the non-private setting and in the (ε, δ)-DP setting with $\varepsilon \in \{8, 4, 2, 1\}$ and $\delta = 10^{-5}$. All methods follow the same doubling batch schedule and regression-objective normalization described above. We run 10 random seeds for every configuration. All algorithms except FastCB are evaluated with both quadratic and logistic oracles, while FastCB uses only the logistic oracle. Across 100 datasets, five privacy settings, and 10 seeds, the full hyperparameter sweep comprises $100 \times 5 \times 10 \times 800 = 4{,}000{,}000$ runs.

Evaluation metric. We report progressive-validation loss $L_{\mathrm{PV}}(t) = t^{-1} \sum_{s=1}^{t} \ell_s(a_s)$. For algorithms evaluated on the same sequence of examples, differences in final progressive-validation loss equal differences in realized regret relative to a common comparator, divided by T.

Model selection. We tune the common regression parameter λ and the algorithm-specific hyperparameters over the grids in Table 1. Here, ℓ_{sq} and $\ell_{\log}$ denote the quadratic and logistic regression oracle losses, respectively. For algorithms evaluated with both oracles, the oracle type is included in model selection. For each algorithm, dataset, and privacy level, we report the configuration with the lowest mean final progressive-validation loss across seeds.

H.5 Results and discussion

We report results at two levels. We first aggregate the final progressive-validation losses over all 100 OpenML datasets, after selecting the best hyperparameter configuration separately for each algorithm, dataset, and privacy regime. We then examine three representative datasets in more detail to show how the learning dynamics change with privacy.

H.5.1 Aggregate results across datasets

Pairwise win rates. Figure 2 summarizes pairwise win rates across the 100 datasets. For each ordered pair of algorithms, the corresponding entry gives the fraction of datasets on which the first algorithm attains lower final progressive-validation loss than the second. The same pattern appears consistently across private regimes and agrees with the trends seen in the main text. In particular, BonusPO outperforms every competing bandit method on roughly 70%–95% of datasets, making it the clear empirical winner in this comparison. The next tier consists of VanillaPO, FastCB, and SquareCB, whose performance is broadly comparable. The final tier consists of LinUCB, RegCB, and AdaCB, with AdaCB typically performing slightly better than the other two. In the non-private setting, BonusPO remains dominant, while the other bandit algorithms perform comparably, with AdaCB holding a slight edge.

This separation under privacy is consistent with the structure of the algorithms. The stronger performance of BonusPO indicates that the bonus remains useful under private batched regression, as it incorporates uncertainty information that is absent from the raw predicted losses alone. By contrast, LinUCB, RegCB, and AdaCB rely directly on a privatized exploration radius, which appears to make them more sensitive to the noise introduced by the privacy mechanism.

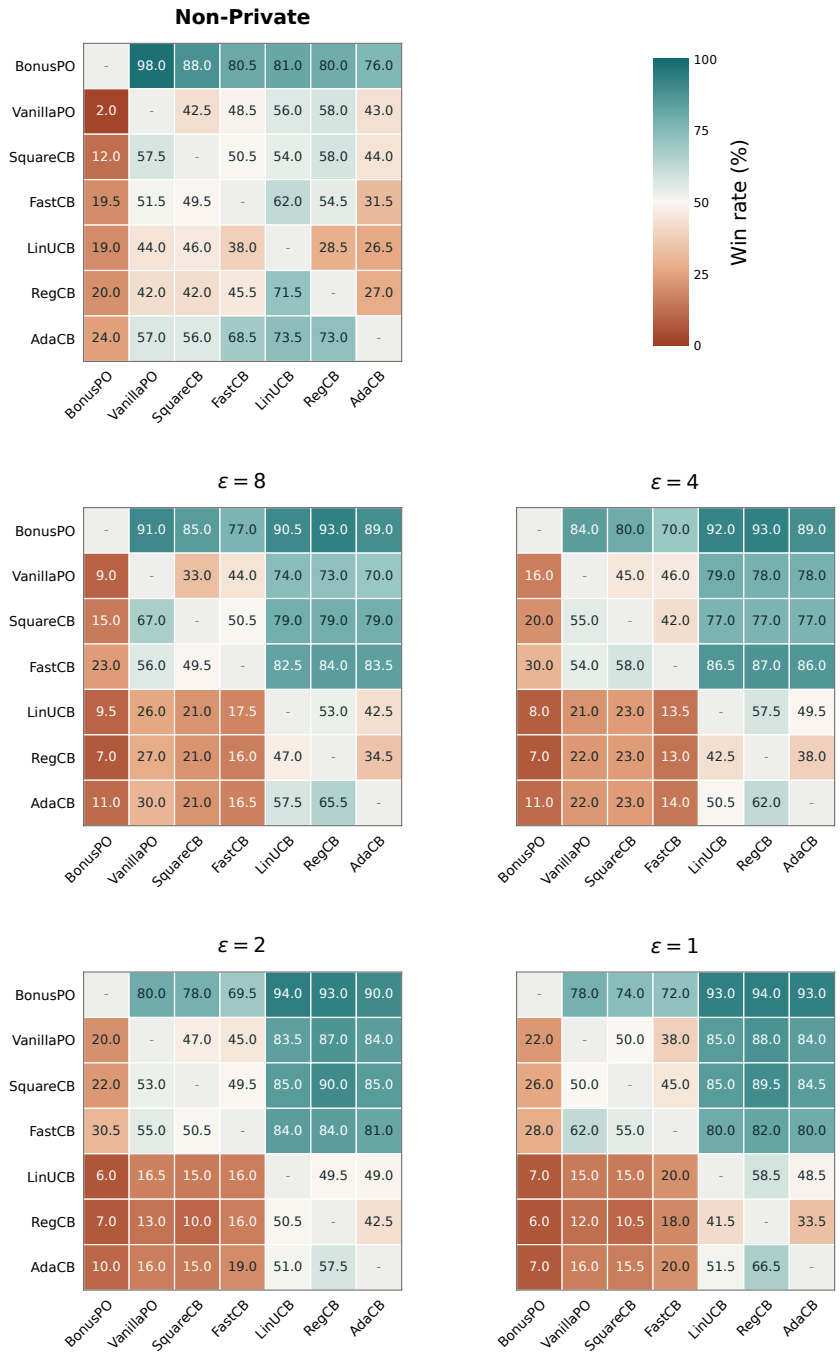


Figure 2: Pairwise win rates based on final progressive-validation loss over 100 OpenML datasets. Each entry reports the percentage of datasets on which the row algorithm achieves lower mean final progressive-validation loss than the column algorithm, with ties receiving half credit. We show the non-private setting and each private regime.

Utility–privacy trade-off. Figure 3 shows the increase in final progressive-validation loss relative to $(8, 10^{-5})$-DP baseline as privacy becomes stronger. This gives a direct empirical view of the privacy–utility trade-off. As expected, stronger privacy leads to larger loss for all methods. The supervised baseline degrades the least. Among the bandit algorithms, the effect of privatization is much more similar.

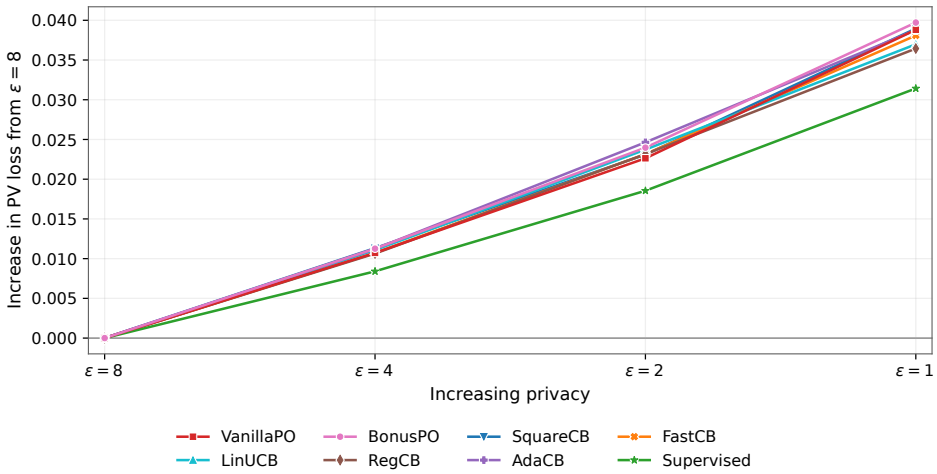


Figure 3: Increase in final progressive-validation loss relative to the $(8, 10^{-5})$-DP baseline, aggregated over 100 OpenML datasets.

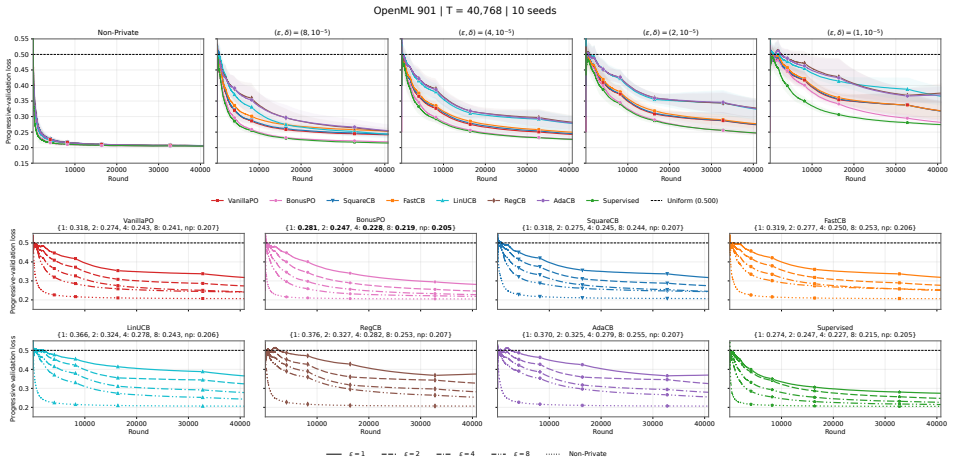
Overall, the aggregate results support the main empirical conclusion of the appendix: `BonusPO` is the strongest bandit method across privacy regimes, followed by a middle group consisting of `VanillaPO`, `FastCB`, and `SquareCB`, while `LinUCB`, `RegCB`, and `AdaCB` are consistently less competitive. Importantly, strengthening the privacy requirement does not substantially alter this relative ordering, consistent with the patterns in Figures 2 and 3.

H.5.2 Results on selected datasets

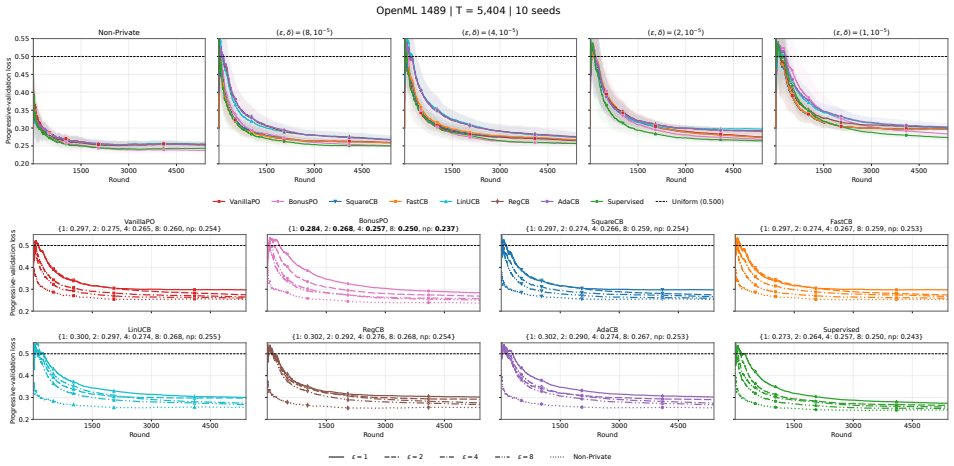
Finally, we examine in greater detail the three OpenML datasets introduced in Section 6, 901, 1489, and 41027, together with three additional datasets, 1462, 1560, and 44743. Figures 4 and 5 show progressive-validation classification loss over rounds, averaged over 10 seeds. For each algorithm and privacy regime, we report the best configuration from the hyperparameter grid.

Several consistent trends emerge. First, performance degrades gracefully as privacy becomes stronger: on all six datasets, increasing privacy raises the loss but does not qualitatively change the learning behavior. Second, the supervised baseline performs best overall, while `BonusPO` is the strongest bandit method and often comes closest to supervised performance.

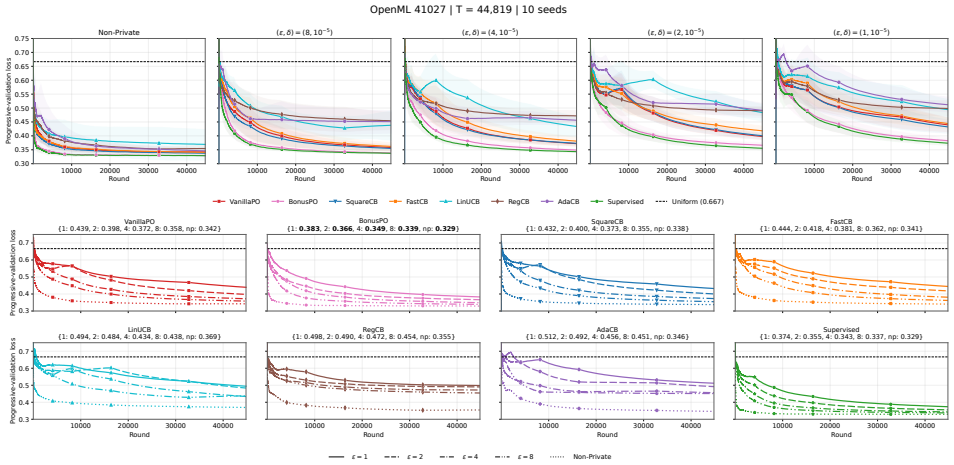
The separation between methods is clearest on datasets 901, 1462, 41027, 44743. In the non-private regime, most algorithms are relatively close, but under stronger privacy they split into three groups: the supervised baseline, with `BonusPO` closest to it; a middle group consisting of `VanillaPO`, `SquareCB`, and `FastCB`; and a lower-performing group consisting of `LinUCB`, `RegCB`, and `AdaCB`. Datasets 1489 and 1560 show the same pattern more weakly: the algorithms remain closer together across privacy levels, but the same ordering begins to emerge as privacy becomes more stringent.



(a) OpenML 901.

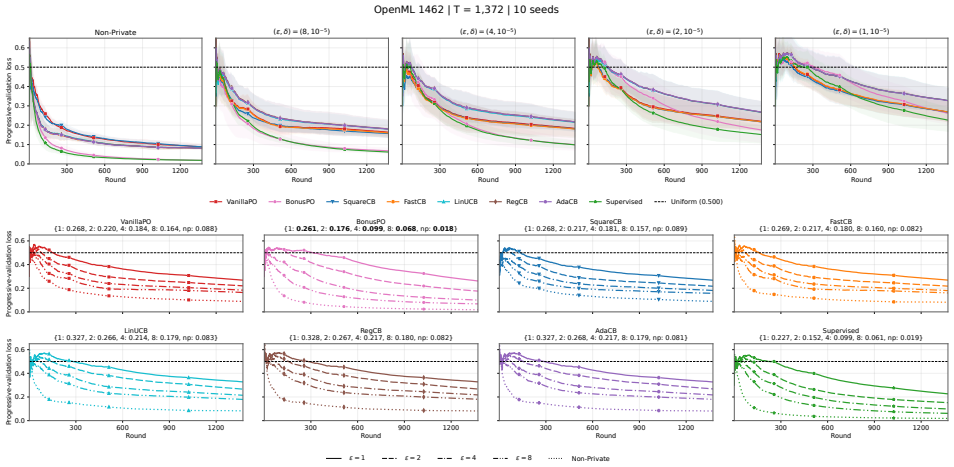


(b) OpenML 1489.

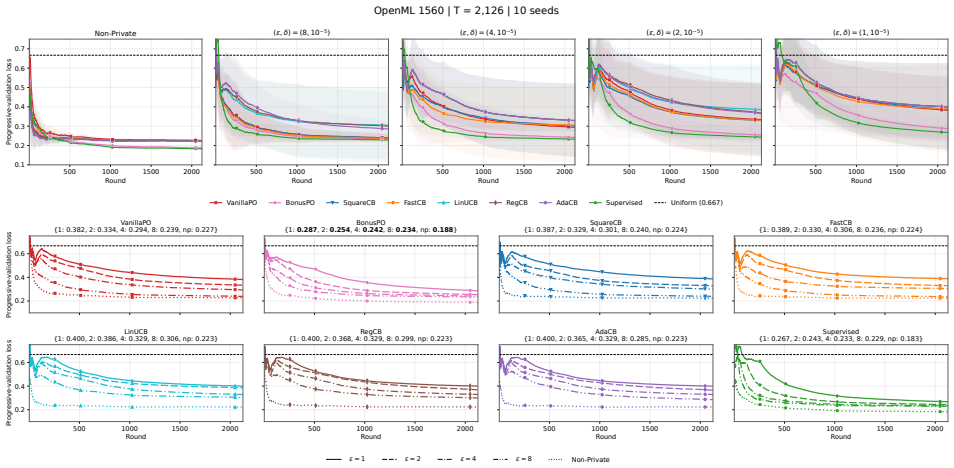


(c) OpenML 41027.

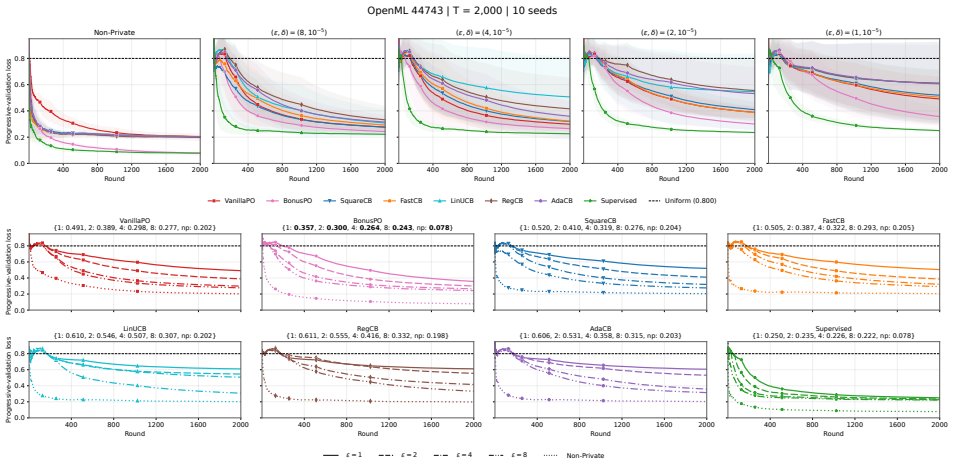
Figure 4: Detailed comparison on OpenML datasets 901, 1489, and 41027. Curves show progressive-validation loss over rounds, averaged over 10 seeds, using the best hyperparameter configuration for each algorithm and privacy regime. The top row compares algorithms at fixed privacy regimes, while the remaining rows show the effect of privacy on each algorithm.



(a) OPENML 1462.



(b) OPENML 1560.



(c) OPENML 44743.

Figure 5: Results on three additional OPENML datasets: 1462, 1560, and 44743. We report progressive-validation loss over rounds, averaged over 10 seeds, with hyperparameters selected separately for each algorithm and privacy regime. The layout mirrors Figure 4.