EXACT UNLEARNING FOR REINFORCEMENT LEARNING IN LINEAR MDPS

Anonymous authors
Paper under double-blind review

ABSTRACT

We develop a provably efficient learning–unlearning framework for episodic reinforcement learning in linear Markov decision processes. Our framework consists of a pair of a ρ-TV-stable RL learner and a *coupling*-based exact RL unlearner. Our ρ-TV-stable learner releases weighted Gram matrices through a binary tree and releases three value-dependent Bellman vectors only at policy switching. Upon a replacement request, the unlearner constructs an approximately maximal coupling between the randomized output of the learner and its counterfactual output had the replaced data not been present to the learner to begin with. We show that we achieve an efficient exact unlearner, whose retraining probability scales only with $\mathcal{O}(\rho)$. At the same time, the learner is ρ-TV-stable and has regret $\widetilde{\mathcal{O}}(dH^{3/2}\sqrt{K} + d^2H^3\sqrt{K/\rho} + d^9H^8/\rho)$, retaining the the optimal-order in the ρ-independent term; here d is the feature dimension, H horizon and K number of episodes.

1 INTRODUCTION

Modern learning systems are trained on data collected from many users and often retain the influence of this data long after training is completed. This creates a basic data-governance problem when a user's data must later be removed, for example because consent is withdrawn, a deletion request is issued, or the data is found to be incorrect. Deleting the original record from storage is not sufficient, since its effect has already been incorporated into the learned model. Retraining from scratch on the modified dataset removes this effect, but repeating the entire training process after every request is often computationally expensive. Machine unlearning studies how to remove the influence of selected training data while avoiding this full retraining cost (Ginart et al., 2019; Guo et al., 2020).

Two main notions of unlearning have been studied. Approximate unlearning requires the updated learner to be close to retraining under a prescribed notion of discrepancy (Guo et al., 2020; Neel et al., 2021; Sekhari et al., 2021). Exact unlearning instead requires the output after an edit to have exactly the same distribution as a fresh run on the modified data (Ullah et al., 2021; Ullah & Arora, 2023). This gives a direct reference for whether the effect of the edited data has been removed: the unlearned learner is statistically indistinguishable from one that was trained without that data in the first place. The difficulty is computational, since retraining already satisfies this requirement. The goal is therefore to recover the same fresh-run distribution while reusing as much of the existing learner state as possible.

This problem becomes more challenging in reinforcement learning, where the training data are generated through interaction with the learner itself. The policy used in one episode determines which states and actions are visited, and the resulting trajectory is then used to update future policies. Thus the influence of one user episode is not confined to a fixed training record: it can propagate through the subsequent learning process. Nguyen-Tang & Arora (2026) recently showed that efficient exact unlearning is possible in tabular episodic MDPs, establishing that exact unlearning and online reinforcement learning can be combined without always retraining from scratch.

The tabular setting, however, is restrictive for large or continuous state spaces, where function approximation is needed to generalize across states. We study this question in linear MDPs, a standard model of linear function approximation (Jin et al., 2020). Moving beyond the tabular setting introduces a new difficulty: linear RL algorithms rely on regression statistics whose targets depend on learned value functions, so the effect of one episode is no longer captured only by fixed counts or empirical sums. It is therefore unclear whether exact unlearning can remain substantially cheaper than retraining while retaining strong learning guarantees.

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.

We give an affirmative answer. We develop a learner–unlearner framework for episodic linear MDPs such that, after replacing one user episode, the updated policies and retained learner state have exactly the same distribution as a fresh run on the modified user sequence. The same learner achieves sublinear regret. With the noise choice designed to control retraining, retraining occurs with probability at most ρ; when retraining is avoided, the unlearning computational cost is only $\mathcal{O}(\log K)$ for fixed problem dimensions, compared with linear dependence on K for rerunning the learner. We summarize our main contributions below:

- **Exact unlearning for linear MDPs.** We design a provable learning–unlearning framework for episodic linear MDPs, combining a ρ-TV-stable learner with a coupling-based unlearner that is provably *exact* (Theorem 1).
- **Efficient unlearning.** Under linear function approximation, exact unlearning costs $\mathcal{O}(\log K + \rho K)$ versus $\mathcal{O}(K)$ for retraining from scratch, for any fixed $d, H, |\mathcal{A}|$ (Theorem 3, Proposition 3).
- **Regret guarantees.** The ρ-TV-stable learner attains regret $\widetilde{\mathcal{O}}\big(dH^{3/2}\sqrt{K} + d^2H^3\sqrt{K/\rho} + d^9H^8/\rho\big)$, whose ρ-independent term $\widetilde{\mathcal{O}}(dH^{3/2}\sqrt{K})$ matches the minimax-optimal order in (d, H, K) attained by LSVI-UCB^{++}, up to logarithmic factors (Theorem 5).

Overview of Techniques. The main technical difficulty is that the sufficient statistics of a linear RL learner depend on the learning history in fundamentally different ways, compared to exact unlearning in tabular MDPs of Nguyen-Tang & Arora (2026). Our learner is based on variance-weighted least-squares estimation: the weighted Gram matrix is a prefix sum of fixed episode-wise feature contributions, but the Bellman regression vectors use targets defined by the current value functions. Consequently, changing the value functions can change how earlier transitions enter later regressions, so the binary-tree construction used for fixed prefix sums cannot be applied directly to the entire learner state. We address this by maintaining the Gram statistics with a binary tree while recomputing the Bellman vectors only at switch episodes. The challenge is to make these two update schedules compatible with the adaptive switch decisions, the underlying learner's regret guarantees, and the approximately maximal coupling required during unlearning.

Exact unlearning introduces a second difficulty: it is not enough to recompute the corrected statistics or obtain a policy with similar value; the updated retained state must have exactly the law of a fresh run. We therefore add Gaussian noise to the released statistics and couple each affected noisy release with its law under the modified user sequence. Successful couplings reuse the existing noisy state, while the first rejection regenerates the remaining run. We study two choices of noise scales: one controls the joint TV distance with less noise, while the other directly controls the probability of retraining. This yields exact unlearning with sublinear accepted-update cost while preserving the ρ-independent term of order $\widetilde{\mathcal{O}}(dH^{3/2}\sqrt{K})$ in regret. See further technical differences with the literature in our discussion in Section 6.

2 Related Work

Machine unlearning. Machine unlearning studies how to remove the effect of selected training data without retraining a model from scratch. Several works develop deletion or certified-removal guarantees for standard learning problems (Ginart et al., 2019; Guo et al., 2020; Neel et al., 2021; Sekhari et al., 2021), while other work considers adaptive deletion requests and stronger data-deletion guarantees (Gupta et al., 2021; Chourasia & Shah, 2023). Our approach is most closely related to the line based on algorithmic stability. Ullah et al. (2021) introduce TV stability as a useful tool for exact unlearning and construct efficient unlearners through coupling. Ullah & Arora (2023) study a more general adaptive query-release view and give unlearning procedures for structured linear and prefix-sum queries. We use the same general TV-stability and coupling approach, but the RL setting introduces additional dependencies: the statistics are generated from trajectories produced by the learner's policies, and the Bellman statistics themselves depend on value functions computed from earlier noisy releases. In our construction, TV stability provides a convenient way to balance the amount of noise used during learning with the probability of reusing the retained state after a replacement; we do not assume that it is necessary for efficient exact unlearning.

Unlearning and stability in sequential decision making. Unlearning has recently been studied in several sequential settings. Hu et al. (2025) consider online learning and unlearning for online gradient methods, requiring future outputs after an unlearning request to be statistically indistinguishable from those obtained without the deleted point. Ye et al. (2026) study unlearning in offline stochastic multi-armed bandits and characterize trade-offs between the unlearning requirement and

the quality of the resulting decision. These settings do not contain the adaptive state transitions and value estimation that arise in online reinforcement learning.

The closest work to ours is Nguyen-Tang & Arora (2026), who establish efficient exact unlearning for tabular episodic MDPs using TV-stable learning and coupling. Their framework shows that exact unlearning can be substantially cheaper than retraining while retaining strong learning guarantees, and leaves function approximation as an open direction. We address this direction for linear MDPs, where the learner state has a richer dependence on past data and learned value functions. Our results show that efficient exact unlearning remains possible in this setting while preserving near-optimal regret dependence.

A related but different notion is replicability. Eaton et al. (2026) develop provably efficient replicable RL algorithms with linear function approximation, including linear MDPs. Replicability asks different runs on independent samples to produce the same output with high probability, whereas our goal is to update a realized learner state after a replacement so that it has exactly the distribution of a fresh run. We therefore view these results as complementary approaches to controlling the sensitivity of reinforcement-learning algorithms.

Linear MDPs and private reinforcement learning. We work in the episodic linear-MDP setting of Jin et al. (2020). Our learner builds on the variance-weighted LSVI-UCB^{++} algorithm of He et al. (2023), which combines variance-weighted regression with rare policy switching to obtain a nearly minimax regret bound. There is also a substantial literature on differential privacy for reinforcement learning with linear function approximation (Luyo et al., 2021; Zhou, 2022; Ngo et al., 2022). In particular, Ngo et al. (2022) use a low-switching policy-update schedule to reduce the privacy cost, while Sahu (2025) perturb the sufficient statistics of LSVI-UCB^{++} and preserve its leading regret dependence.

3 Problem Setup

Notation. For an integer $n \geq 1$, write $[n] = \{1, \dots, n\}$. For a positive semidefinite matrix B, let $\|x\|_B = \sqrt{x^\top B x}$. We use $\|B\|_{\mathrm{op}}$ and $\|B\|_F$ for the operator and Frobenius norms, and $B \preceq C$ when $C - B$ is positive semidefinite. Write $[x]_+ = \max\{x, 0\}$ and $[x]_{[u,v]} = \min\{v, \max\{u, x\}\}$. A superscript $*$ denotes finite sequences, and $\widetilde{\mathcal{O}}$ suppresses logarithmic factors in the problem parameters and inverse failure probabilities.

3.1 Markov Decision Process

Let $(\mathcal{S}, \mathcal{A}, H, \{P_h\}_{h=1}^H, \{r_h\}_{h=1}^H)$ be a finite-horizon MDP. The state space $\mathcal{S}$ is measurable and may be infinite, the action set $\mathcal{A}$ is finite, and $[H] := \{1, \dots, H\}$ is the set of stages. At stage h, $P_h(\cdot \mid s, a)$ is the next-state distribution and $r_h : \mathcal{S} \times \mathcal{A} \to [0, 1]$ is the known deterministic reward function (without loss of generality).

The learner interacts with one fixed MDP for K episodes, each with H stages, for a total of HK interaction steps. In episode k, the learner receives an initial state s_1^k and selects a policy $\pi^k = (\pi_1^k, \dots, \pi_H^k)$ using the preceding data. At stage h, it takes $a_h^k = \pi_h^k(s_h^k)$, receives reward $r_h(s_h^k, a_h^k)$, and moves to $s_{h+1}^k \sim P_h(\cdot \mid s_h^k, a_h^k)$.

For a policy $\pi = (\pi_1, \dots, \pi_H)$ with $\pi_h : \mathcal{S} \to \mathcal{A}$, define

$$V_h^\pi(s) := \mathbb{E}_\pi[\sum_{j=h}^{H} r_j(s_j, a_j) \mid s_h = s], \quad Q_h^\pi(s, a) := \mathbb{E}_\pi[\sum_{j=h}^{H} r_j(s_j, a_j) \mid s_h = s, a_h = a].$$

For any bounded $V : \mathcal{S} \to \mathbb{R}$, write $[P_h V](s, a) := \mathbb{E}_{s' \sim P_h(\cdot \mid s, a)}[V(s')]$. With $V_{H+1}^\pi = 0$, the Bellman equations are

$$Q_h^\pi(s, a) = r_h(s, a) + [P_h V_{h+1}^\pi](s, a), \qquad V_h^\pi(s) = Q_h^\pi(s, \pi_h(s)).$$

An optimal policy has $V_h^\star(s) = \max_{a \in \mathcal{A}} Q_h^\star(s, a)$. The regret after K episodes is

$$\mathrm{Regret}(K) := \sum_{k=1}^{K} \left[V_1^\star(s_1^k) - V_1^{\pi^k}(s_1^k) \right]. \tag{1}$$

During learning, the agent has no generative model, and we impose no coverage assumption.

3.2 Linear MDP

Assumption 1 (Linear MDP; Assumption A of Jin et al. (2020)). *A finite-horizon MDP is a linear MDP with known measurable feature map $\phi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$ if, for every $h \in [H]$, there exist a vector of d unknown finite signed measures $\mu_h = (\mu_h^{(1)}, \dots, \mu_h^{(d)})$ on $\mathcal{S}$ and an unknown vector $\theta_h \in \mathbb{R}^d$ such that, for every $(s, a) \in \mathcal{S} \times \mathcal{A}$,*

$$P_h(\cdot \mid s, a) = \langle \phi(s, a), \mu_h(\cdot) \rangle, \qquad r_h(s, a) = \langle \phi(s, a), \theta_h \rangle.$$

As a standard normalization, we assume $\|\phi(s,a)\|_2 \le 1$ *and* $\max \|\theta_h\|_2 \le \sqrt{d}$*, and that: for every stage* h *and measurable* $f : \mathcal{S} \to [0,1]$, $\left\|\int f(s')\mu_h(\mathrm{d}s')\right\|_2 \le \sqrt{d}$.

Remark 1. *Prior linear-MDP analyses (Jin et al., 2020; He et al., 2023; Sahu, 2025) use the total-mass condition* $\|\mu_h(\mathcal{S})\|_2 \le \sqrt{d}$ *instead of* $\left\|\int f(s')\mu_h(\mathrm{d}s')\right\|_2 \le \sqrt{d}$*. Their analyses, however, rely on the latter, for example in their Lemma G.1, but infer it from the total-mass condition. This inference fails for signed measures, as shown by Example 1 in Appendix A.1.*

Proposition 1 (Linear action-value functions)**.** *Under Assumption 1, for every policy* π *and stage* $h \in [H]$*, there exists a vector* $w_h^\pi \in \mathbb{R}^d$ *such that small*

$$Q_h^\pi(s,a) = \langle \phi(s,a), w_h^\pi \rangle \qquad \textit{for every } (s,a) \in \mathcal{S} \times \mathcal{A}.$$

A proof is provided in Appendix A. A linear MDP also recovers tabular MDPs with $d = |\mathcal{S}||\mathcal{A}|$.

3.3 Exact unlearning

Users and Environment Oracle. Let $\mathcal{Z}$ be the user space, and let **user** $z_k \in \mathcal{Z}$ arrive in episode k. We represent user z_k by a depth-H response tree that fixes the reward and next state for every possible sequence of actions in that episode. The learner observes only the branch selected by its deployed policy. This branch is the trajectory $x_k := \left(s_h^k, a_h^k, r_h(s_h^k, a_h^k), s_{h+1}^k\right)_{h=1}^H$. An **environment oracle** makes this interaction explicit: $x_k = \mathrm{EO}_k(z_k; \pi^{1:k})$, where $\pi^{1:k} := (\pi^1, \dots, \pi^k)$ and EO_k follows the branch selected by these policies.

Learner. Once the environment oracle is fixed, a learner is a *randomized* map from a finite user sequence to the deployed policies and retained metadata:

$$\mathsf{A} : \mathcal{Z}^* \longrightarrow \mathcal{W}^* \times \mathcal{M}, \qquad \mathsf{A}(z_{1:K}) = (\pi^{1:K}, M). \tag{2}$$

Here $\mathcal{W}$ is the policy space, $\mathcal{M}$ is the metadata space, and M is the metadata that the learner maintains in its internal memory.

Unlearner. An unlearner $\mathcal{U}$, with respect to the learner A, is a randomized map from the learner's output space, the space of user identities, and the space of users, to the space of the policy and the metadata.

$$\mathcal{U} : (\mathcal{W}^* \times \mathcal{M}) \times \mathbb{N} \times \mathcal{Z} \longrightarrow \mathcal{W}^* \times \mathcal{M}.$$

That is, the unlearner takes as input the user position k to be unlearned, the learner's current policies and metadata, and a replacement user z', and outputs an updated policy sequence and metadata.

Definition 1 (Exact unlearning)**.** *An unlearning algorithm* $\mathcal{U}$ *is said to achieve* exact unlearning *with respect to a learning algorithm* A *if, for any* $K \in \mathbb{N}$*, any* $k \in [K]$*, and any sequence* $z_1, \dots, z_K, z'$ *in* $\mathcal{Z}$*, the following holds:*

$$\mathcal{U}\big(\mathsf{A}(z_{1:K}), k, z'\big) \overset{D}{=} \mathsf{A}(z_1, \dots, z_{k-1}, z', z_{k+1}, \dots, z_K),$$

where $X \overset{D}{=} Y$ *means that for any event* $\mathcal{E}$, $\Pr(X \in \mathcal{E}) = \Pr(Y \in \mathcal{E})$.

Achieving learning–unlearning guarantees is a **multi-criteria** design; that is, our goal is to design the learner–unlearner pair $(\mathsf{A}, \mathcal{U})$ such that (i) it satisfies Definition 1, (ii) A retains good utility, *and* (iii) the computational cost of unlearning is much cheaper than that of retraining from scratch upon a replacement request, which trivially satisfies exact unlearning.

TV-stability as a mechanism for efficient unlearning. Efficient unlearning becomes possible when the learner is randomized and outputs an outcome that could be mostly reused as the counterfactual outcome had the learner not seen the deleted user data to begin with. That is, the learner should randomize yet stabilize its output, in the sense that the absence of any certain user data does not change the learner's output distribution much. This is the key intuition behind using TV-stable learners for designing efficient unlearners in Nguyen-Tang & Arora (2026). Here, the total variation (TV) distance between two probability distributions P and Q on the same output space is $\mathrm{TV}(P,Q) := \sup_{\text{measurable event } E} |P(E) - Q(E)|$, and the $\rho-$TV-stable algorithms are defined as:

Definition 2 (ρ-TV-stable algorithms)**.** *Fix any* $\rho \ge 0$*. For a specified output, a randomized algorithm* A *is* ρ*-TV-stable if, for any two user sequences* Z *and* Z' *of the same length that differ in exactly one position, we have*

$$\mathrm{TV}\big(\mathsf{A}(Z), \mathsf{A}(Z')\big) \le \rho.$$

Unless noted otherwise, all probabilities are over the algorithm's internal randomness; events are measurable sets in the specified output space.

Let P and Q be the output distributions of the learner on the original user sequence and the sequence after a replacement request, respectively. Suppose that we already have a sample $X \sim P$. The goal is to transform it into a sample $Y \sim Q$ at low computational cost. A *coupling* of P and Q is a joint distribution of (X, Y) whose marginals are P and Q. A *maximal coupling* minimizes the probability that the two samples differ and satisfies $\Pr(X \neq Y) = \mathrm{TV}(P, Q)$. Thus, under a maximal coupling, the original output can be reused with probability $1 - \mathrm{TV}(P, Q)$. Hence ρ-TV stability implies that a global maximal coupling of the two specified outputs fails with probability at most ρ. The key challenge is to construct a learner–unlearner framework that has (approximately) maximal coupling.

4 TV-STABLE LEARNER

In this section, we describe our construction of a TV-stable RL learner (Algorithm 1) in linear MDPs that attains a near-optimal minimax regret rate while supporting the exact unlearning framework (Section 5) whose computational cost is only a fraction of the full retraining cost.

Algorithm 1 ρ-TV-STABLE-LEARN(k_0, M)

Require: Starting episode k_0, retained metadata M; initialize the learner state from M

for $k = k_0, \ldots, K$ **do**
  $\widetilde{G}_{k,h} \leftarrow \mathrm{GetNoisySum}(k-1, \mathcal{T})_h$ and $\widetilde{\Lambda}_{k,h} \leftarrow \widetilde{G}_{k,h} + c_\Lambda I$ for all $h \in [H]$
  switch $\leftarrow \big[k = 1$ or $\big(m < M_{\max}$ and $\exists h' \in [H] : \det(\widetilde{\Lambda}_{k,h'}) \geq 2\det(\widetilde{\Lambda}_{k_{\mathrm{last}},h'})\big)\big]$
  if switch **then**
    $m \leftarrow m+1, \mathcal{K}_{\mathrm{sw}} \leftarrow \mathcal{K}_{\mathrm{sw}} \cup \{k\}, k_{\mathrm{last}} \leftarrow k$
    for $h = H, \ldots, 1$ **do**
      Compute $b_{j,k,h}$ from $\mathcal{D}_{k-1}$ by (5), release $\widetilde{b}_{j,k,h}$ by (6), $j \in [3]$; set $(\widetilde{\widehat{w}}_{k,h}, \widetilde{\check{w}}_{k,h}, \widetilde{\overline{w}}_{k,h}) \leftarrow \widetilde{\Lambda}_{k,h}^{-1}(\widetilde{b}_{1,k,h}, \widetilde{b}_{2,k,h}, \widetilde{b}_{3,k,h})$
      $\widetilde{\widehat{Q}}_{k,h}(s,a) \leftarrow \big[\min\{r_h(s,a) + \langle \phi(s,a), \widetilde{\widehat{w}}_{k,h}\rangle + \widehat{\beta}\|\phi(s,a)\|_{\widetilde{\Lambda}_{k,h}^{-1}}, \widetilde{\widehat{Q}}_{k-1,h}(s,a), H\}\big]_{[0,H]}$
      $\widetilde{\check{Q}}_{k,h}(s,a) \leftarrow \big[\max\{r_h(s,a) + \langle \phi(s,a), \widetilde{\check{w}}_{k,h}\rangle - \check{\beta}\|\phi(s,a)\|_{\widetilde{\Lambda}_{k,h}^{-1}}, \widetilde{\check{Q}}_{k-1,h}(s,a), 0\}\big]_{[0,H]}$
      $\widetilde{\widehat{V}}_{k,h}(s) \leftarrow \max_a \widetilde{\widehat{Q}}_{k,h}(s,a), \widetilde{\check{V}}_{k,h}(s) \leftarrow \max_a \widetilde{\check{Q}}_{k,h}(s,a), \widetilde{\pi}_h^k(s) \in \arg\max_{a \in \mathcal{A}} \widetilde{\widehat{Q}}_{k,h}(s,a)$
    end for
  else
    Point leaf k's vector statistics to leaf k_{last}; reuse regression vectors, Q, V, and policy
  end if
  Receive s_1^k
  for $h = 1, \ldots, H$ **do**
    Take $a_h^k = \widetilde{\pi}_h^k(s_h^k)$; $\overline{\mathbb{V}}_{k,h} \leftarrow \big[[\langle \phi_h^k, \widetilde{\overline{w}}_{k,h}\rangle]_{[0,H^2]} - [\langle \phi_h^k, \widetilde{\widehat{w}}_{k,h}\rangle]_{[0,H]}^2\big]_{[0,H^2]}$
    Compute $E_{k,h}, D_{k,h}$ by (7); $\widetilde{\sigma}_{k,h} \leftarrow \sqrt{\overline{\mathbb{V}}_{k,h} + E_{k,h} + D_{k,h}} + H$
    $\bar{\sigma}_{k,h} \leftarrow \max\{\widetilde{\sigma}_{k,h}, \sqrt{H}, 2d^3H^2u_{k,h}^{1/2}\}$, $X_{k,h} \leftarrow \bar{\sigma}_{k,h}^{-2}\phi_h^k(\phi_h^k)^\top$; receive s_{h+1}^k via $\mathrm{EO}_k(z_k; \widetilde{\pi}^{1:k})$
  end for
  $\mathcal{D}_k \leftarrow \mathcal{D}_{k-1} \cup \{(x_k, (\bar{\sigma}_{k,h})_{h=1}^H)\}$; $\mathrm{TreeUpdate}(X_k, k; \mathcal{T})$ with $X_k = (X_{k,h})_{h=1}^H$
  $M \leftarrow (\mathcal{D}_k, \mathcal{T}, \mathcal{K}_{\mathrm{sw}})$; at a switch, store $\{\widetilde{\Lambda}_{k,h}\}_h$ and $\{(b_{j,k,h}, \widetilde{b}_{j,k,h})\}_{j\in[3],h\in[H]}$ at leaf k
end for

Ensure: Retained metadata M

4.1 VARIANCE-WEIGHTED SUFFICIENT STATISTICS

We first briefly revisit the rare-switching, variance-weighted LSVI-UCB^{++} algorithm of (He et al., 2023), which our learner will build on. For any bounded value function V, the linear MDP assumption implies that $P_hV(s,a)$ is linear in $\phi(s,a)$. LSVI-UCB^{++} therefore estimates these transition expectations using variance-weighted least squares. Writing $\phi_h^i := \phi(s_h^i, a_h^i)$, the weighted Gram matrix is

$$\Lambda_{k,h} = \lambda I + \sum_{i<k} \bar{\sigma}_{i,h}^{-2}\phi_h^i(\phi_h^i)^\top, \tag{3}$$

where $\lambda > 0$ is a regularization parameter and $\bar{\sigma}_{i,h}$ is a variance-dependent scale that assigns smaller weight to transitions with larger uncertainty. The regressions used by the algorithm can be written in terms of three vector statistics:

$$b_{1,k,h} = \sum_{i<k} \bar{\sigma}_{i,h}^{-2}\phi_h^i\widehat{V}_{k,h+1}(s_{h+1}^i), b_{2,k,h} = \sum_{i<k} \bar{\sigma}_{i,h}^{-2}\phi_h^i\check{V}_{k,h+1}(s_{h+1}^i), b_{3,k,h} = \sum_{i<k} \bar{\sigma}_{i,h}^{-2}\phi_h^i\widehat{V}_{k,h+1}(s_{h+1}^i)^2.$$

The corresponding regression vectors are $\widehat{w}_{k,h} = \Lambda_{k,h}^{-1}b_{1,k,h}$, $\check{w}_{k,h} = \Lambda_{k,h}^{-1}b_{2,k,h}$, $\overline{w}_{k,h} = \Lambda_{k,h}^{-1}b_{3,k,h}$. The first two regressions estimate the Bellman expectations associated with the optimistic and pessimistic value functions, while the first and third provide the first and second moments used for variance estimation. These four sufficient statistics, $\Lambda_{k,h}$ and $b_{1,k,h}, b_{2,k,h}, b_{3,k,h}$, are the quantities that we perturb in our TV-stable learner.

4.2 PERTURBING SUFFICIENT STATISTICS

We perturb two main types of sufficient statistics: the Gram matrix and three vector statistics. The Gram matrix has a prefix-sum structure, while the vector statistics do not. Due to the prefix sum in the Gram matrix, we use binary-tree mechanisms to add noise to the Gram matrix, accumulating the noise only over $\mathcal{O}(\log K)$ terms. For the vector statistics, we instead recompute and perturb them only at switch episodes, where the value functions are updated.

Binary mechanism for the Gram matrix. Intuitively, the binary-tree mechanism perturbs prefix sums without accumulating noise from all previous episodes. Suppose that we want to release noisy versions of $\sum_{k=1}^{k'} u_k$ for $k' \le K$. We store each u_k at a leaf of a binary tree, while every non-leaf node stores the sum of all leaf values in its subtree. Hence, each node represents a dyadic interval of consecutive episodes. We add independent noise to each such interval and reconstruct a noisy prefix sum by decomposing $[1, k']$ into $\mathcal{O}(\log K)$ disjoint dyadic intervals. Thus, each query accumulates noise from only $\mathcal{O}(\log K)$ terms.

We apply this mechanism to the Gram statistics. For each episode k and stage h, let $\phi_h^k := \phi(s_h^k, a_h^k) \in \mathbb{R}^d$ and $X_{k,h} := (\bar{\sigma}_{k,h})^{-2}\phi_h^k(\phi_h^k)^\top$ where $\bar{\sigma}_{k,h} \ge \sqrt{H}$ is the variance scale specified later in Algorithm 1. We collect $X_k := (X_{k,1}, \dots, X_{k,H})$ at leaf k and use a single binary tree $\mathcal{T}$ for all stages. A tree interval $\mathcal{I}$ stores $G_\mathcal{I} := (G_{\mathcal{I},1}, \dots, G_{\mathcal{I},H})$, where $G_{\mathcal{I},h} := \sum_{i\in\mathcal{I}} X_{i,h}$. Since the Gram matrix is symmetric and is subsequently used through its inverse, determinant, and induced norm, we perturb it with symmetric noise so that these matrix operations remain well defined in the same symmetric space. For each dyadic interval $\mathcal{I}$ and stage h, draw $W_{\mathcal{I},h} \in \mathbb{R}^{d\times d}$ with independent $\mathcal{N}(0, \sigma_\Lambda^2/2)$ entries, and define

$$K_{1,\mathcal{I},h} := (W_{\mathcal{I},h} + W_{\mathcal{I},h}^\top)/\sqrt{2}, \qquad \widetilde{G}_{\mathcal{I},h} := G_{\mathcal{I},h} + K_{1,\mathcal{I},h}.$$

The matrix $K_{1,\mathcal{I},h}$ is Gaussian Orthogonal Ensemble (GOE) noise: its diagonal entries have variance σ_Λ^2, while its independent off-diagonal entries have variance $\sigma_\Lambda^2/2$. Define the tree-depth parameter by $L_K := \lceil \log_2 K \rceil + 1$. Every prefix of $[1, K]$ can be decomposed into at most L_K dyadic intervals.

Tree operations.

- **Tree update.** After episode k, $\mathrm{TreeUpdate}(X_k, k; \mathcal{T})$ inserts X_k at leaf k and updates the clean Gram sums along the leaf-to-root path.
- **Noisy query.** At the start of each later episode, the noisy prefix sum at stage h is $\mathrm{GetNoisySum}(n, \mathcal{T})_h := \sum_{\mathcal{I}\in\mathcal{C}(n)} \widetilde{G}_{\mathcal{I},h}$, where $\mathcal{C}(n)$ is the decomposition of $[1, n]$ into at most L_K disjoint dyadic intervals, with $\mathcal{C}(0) = \emptyset$; e.g., for $K = 8$, $\mathcal{C}(7) = \{[1, 4], [5, 6], [7, 7]\}$.

Noisy Gram matrix. At the beginning of episode k, the learner computes the noisy Gram matrix from the binary tree as

$$\widetilde{\Lambda}_{k,h} := \mathrm{GetNoisySum}(k-1, \mathcal{T})_h + c_\Lambda I, \tag{4}$$

where $c_\Lambda > 0$ is a deterministic shift, which ensures positive-definiteness with high probability. The choice of c_Λ are given in Lemma 1 in Appendix A.

Noisy vector statistics and switching. Unlike the Gram matrix, the Bellman regression vectors are not prefix sums of episode-fixed terms because their targets depend on the current value functions. We therefore recompute and perturb them only at switch episodes. Choose $C_{\mathrm{sw}} = 9 > \frac{1}{\log(7/6)} + 2$ and $M_{\max} := \lceil C_{\mathrm{sw}} dH \log(KH + 2) \rceil$. The algorithm allows at most $M_{\max}$ switches. Lemma 3 shows that the determinant-doubling rule produces fewer than $M_{\max}$ switches whenever (10) holds. Let m be the number of previous switches and k_{last} the most recent switch episode. Episode k is a *switch episode* if $k = 1$, or if $m < M_{\max}$ and

$$\det(\widetilde{\Lambda}_{k,h}) \ge 2\det(\widetilde{\Lambda}_{k_{\mathrm{last}},h}) \qquad \text{for some } h \in [H].$$

At a switch episode k, let $\widetilde{\widehat{V}}_{k,h}$ and $\widetilde{\check{V}}_{k,h}$ be the optimistic and pessimistic value functions computed from the noisy statistics. We define

$$b_{1,k,h} := \sum_{i<k} (\bar{\sigma}_{i,h})^{-2}\phi_h^i \widetilde{\widehat{V}}_{k,h+1}(s_{h+1}^i), \quad b_{2,k,h} := \sum_{i<k} (\bar{\sigma}_{i,h})^{-2}\phi_h^i \widetilde{\check{V}}_{k,h+1}(s_{h+1}^i), \tag{5}$$

$$b_{3,k,h} := \sum_{i<k} (\bar{\sigma}_{i,h})^{-2}\phi_h^i \big[\widetilde{\widehat{V}}_{k,h+1}(s_{h+1}^i)\big]^2.$$

The first two vectors are used for the optimistic and pessimistic Bellman regressions, while the first and third provide the first and second moments for variance estimation. At each switch, we release

$$\widetilde{b}_{j,k,h} = b_{j,k,h} + \omega_{j,k,h}, \qquad \omega_{j,k,h} \sim \mathcal{N}(0, \sigma_j^2 I_d), \qquad j \in [3], \tag{6}$$

Sensitivity. Let $\Delta_\Lambda, \Delta_1, \Delta_2, \Delta_3$ be the sensitivity of the Gram matrix $X_{k,h}$, and of the three statistics in Eq. (5), respectively. Here, for a clean sufficient statistic c, its sensitivity is defined by $\Delta(c) := \sup_{Z,Z',\widetilde{\mathcal{H}}:d_H(Z,Z')\le 1} \left\| c(Z;\widetilde{\mathcal{H}}) - c(Z';\widetilde{\mathcal{H}}) \right\|$ where $d_H(Z,Z')$ is the Hamming distance between the two user sequences, and $\widetilde{\mathcal{H}}$ is the preceding noisy history. We use the Frobenius norm for matrix statistics and the Euclidean norm for vector statistics. Appendix A.4 shows that $\Delta_\Lambda \le 2/H$, $\Delta_1, \Delta_2 \le 2$, and $\Delta_3 \le 2H$.

Metadata. To support efficient unlearning, we retain additional metadata that allows the clean sufficient statistics to be updated when a user is replaced or an unlearning request is received. However, exact unlearning applies to the entire retained learner state, not only to the policy or noisy statistics. We accordingly keep only the auxiliary information that is necessary for correcting the clean statistics, while ensuring that it can itself be updated consistently by the unlearning procedure.

For simplicity, we also store the Bellman vector statistics at the leaves of the binary tree $\mathcal{T}$. At a switch episode k, leaf k stores $\{\widetilde{\Lambda}_{k,h} : h \in [H]\}$ and $\{(b_{j,k,h}, \widetilde{b}_{j,k,h}) : j \in [3], h \in [H]\}$. At a non-switch episode, the Bellman vector statistics are the same as those from the most recent switch k_{last}. Hence, in practice, we only need to store a reference to the vector statistics at k_{last} rather than duplicate them at every leaf. For simplicity of notation, we treat each leaf as storing the corresponding vector statistics.

Let $\mathcal{K}_{\text{sw}} \subseteq [K]$ denote the set of switch episodes, and let $\mathcal{D}_K := ((x_i, (\bar{\sigma}_{i,h})_{h=1}^H))_{i=1}^K$ collect the trajectories and variance scales. The retained metadata is $M := (\mathcal{D}_K, \mathcal{T}, \mathcal{K}_{\text{sw}})$. The noisy metadata $\widetilde{M}$ contains the noisy Gram and vector statistics, switch indices, and quantities determined by them, while M additionally contains the trajectories and clean statistics required by the unlearner.

4.3 Rare-switching learner

Algorithm 1 maintains the switch count m and the most recent switch episode k_{last}, and applies the determinant rule defined above. For a fresh run, initialize $\widetilde{\widetilde{Q}}_{0,h}(s,a) = H$ and $\widetilde{\widetilde{Q}}_{0,h}(s,a) = 0$ for every (s,a,h). At a switch episode k, set $\widetilde{\widetilde{V}}_{k,H+1} = \widetilde{\widetilde{V}}_{k,H+1} = 0$. The released vectors give

$$\widehat{\widetilde{w}}_{k,h} := \widetilde{\Lambda}_{k,h}^{-1}\widetilde{b}_{1,k,h}, \qquad \widetilde{\widetilde{w}}_{k,h} := \widetilde{\Lambda}_{k,h}^{-1}\widetilde{b}_{2,k,h}, \qquad \widetilde{\widetilde{w}}_{k,h} := \widetilde{\Lambda}_{k,h}^{-1}\widetilde{b}_{3,k,h}.$$

These vectors estimate the upper target, lower target, and upper second moment, respectively, and remain fixed until the next switch. Define the confidence width at the most recent switch by $u_{k,h} := \|\phi_h^k\|_{\widetilde{\Lambda}_{k_{\text{last}},h}^{-1}}$. Treat the positive constants $\widehat{\beta}$, $\check{\beta}$, and $\widetilde{\beta}$ as input confidence radii for the upper value, lower value, and moment regressions, respectively. Their values for the regret guarantee are given in (35). Define

$$E_{k,h} := \min\{H\widetilde{\beta}u_{k,h}, H^2\} + \min\{2H\widetilde{\beta}u_{k,h}, H^2\}, \tag{7}$$
$$D_{k,h} := \min\{4d^3H^2\left[\langle\phi_h^k, \widehat{\widetilde{w}}_{k,h} - \widetilde{\widetilde{w}}_{k,h}\rangle + (\widetilde{\beta}+\check{\beta})u_{k,h}\right]_+, d^3H^3\}.$$

Remark 2 (Relation to prior learning algorithms)**.** *While our learner builds on the variance-weighted updates and determinant-based rare switching of LSVI-UCB^{++} (He et al., 2023), as well as the Gaussian perturbation of its sufficient statistics in Sahu (2025), two structural changes are important for exact unlearning. First, instead of perturbing a separate Gram statistic at every episode and stage, we exploit its prefix-sum structure and release it through a binary tree. Hence, replacing one episode affects only $\mathcal{O}(\log K)$ noisy Gram nodes. Second, we move the computation and release of the three value-dependent Bellman vectors inside the switching step. Thus, no new vector statistic is released between two switches; the learner simply reuses the regression vectors and value functions from the most recent switch. Consequently, a replacement affects the Gram statistics only along one tree path and the Bellman statistics only at switch episodes, which is the structure that enables efficient exact unlearning.*

5 Exact unlearning by Gaussian coupling

The unlearner receives the retained metadata M, the episode k to be unlearned, and a replacement user z'. It first interacts with z' using the policy deployed in episode k to obtain the replacement trajectory. This changes the clean Gram statistics only at tree nodes containing leaf k, and the clean Bellman vectors only at switch episodes after k. Algorithm 2 couples the corresponding noisy releases and updates M in place. If all tests accept, the noisy releases and switch episodes remain unchanged, and the unlearner only updates the affected clean statistics. In particular, it reconstructs the value functions once at each of the at most $M_{\max} = \mathcal{O}(dH\log(KH+2))$ switches and reuses them across the three updates in (8). Section D gives the resulting cost.

The coupling tests follow the learner's release order. Bellman vectors at a switch are released before the episode interaction, whereas Gram nodes are released afterward in post-order. Starting from leaf k, the unlearner follows its path to the root and processes the switch leaves encountered between successive ancestors. At each switch, vector types $1, 2, 3$ are processed in order, with stages $H, \dots, 1$ within each type; non-switch leaves introduce no vector release. This order preserves the dependence of type 1 on the optimistic values, type 2 on the pessimistic values, and type 3 on the optimistic values.

For each affected release, acceptance keeps the stored noisy value and replaces only its clean mean. On the first rejection, reflection produces the new noisy value, after which all strictly later releases are regenerated by Algorithm 1; the reflected value itself is retained and is not redrawn. For a rejected Gram release at node l, let $k^+(l)$ be the episode of the first leaf following l in post-order, with $k^+(l) := K + 1$ if no such leaf exists. The learner resumes from $k^+(l)$ using the updated metadata; when $k^+(l) = K + 1$, it only completes the current tree update and returns.

At a later switch $r > k$, the clean Bellman vectors change only in the summand contributed by the replaced episode. Reconstruct the switch-r value functions from the noisy statistics, and write $\phi_h'^k := \phi(s_h'^k, a_h'^k)$ for the replacement feature. Subtracting the old summand and adding the new one gives

$$\begin{aligned} b'_{1,r,h} &= b_{1,r,h} - (\bar{\sigma}_{k,h})^{-2} \phi_h^k \widetilde{\widetilde{V}}_{r,h+1}(s_{h+1}^k) + (\bar{\sigma}'_{k,h})^{-2} \phi_h'^k \widetilde{\widetilde{V}}_{r,h+1}(s_{h+1}'^k), \\ b'_{2,r,h} &= b_{2,r,h} - (\bar{\sigma}_{k,h})^{-2} \phi_h^k \widetilde{\widetilde{V}}_{r,h+1}(s_{h+1}^k) + (\bar{\sigma}'_{k,h})^{-2} \phi_h'^k \widetilde{\widetilde{V}}_{r,h+1}(s_{h+1}'^k), \\ b'_{3,r,h} &= b_{3,r,h} - (\bar{\sigma}_{k,h})^{-2} \phi_h^k \left[\widetilde{\widetilde{V}}_{r,h+1}(s_{h+1}^k)\right]^2 + (\bar{\sigma}'_{k,h})^{-2} \phi_h'^k \left[\widetilde{\widetilde{V}}_{r,h+1}(s_{h+1}'^k)\right]^2 \end{aligned} \tag{8}$$

No clean Bellman vector changes at a switch $r \le k$, because its defining sum uses only episodes strictly before r.

Algorithm 2 UNLEARN(M, k, z')

Require: Retained metadata M; position k to unlearn; replacement user z'

Retrieve $(x_k, (\bar{\sigma}_{k,h})_{h=1}^H)$ from $\mathcal{D}_K$, $X_k = (X_{k,h})_{h=1}^H$ from leaf k, and $\mathcal{K}_{\mathrm{sw}}$ from M; process $\mathcal{K}_{\mathrm{sw}}$ in increasing order, reconstructing regression vectors, Q, V, and policies once per switch and reusing Q, V across all three vector updates

Query the oracle at episode k under z' with the deployed policy given by the preceding noisy values; obtain x'_k, recompute $(\bar{\sigma}'_{k,h})_{h=1}^H$, and form $X'_k = (X'_{k,h})_{h=1}^H$

Replace the episode-k entry of $\mathcal{D}_K$ in M by $(x'_k, (\bar{\sigma}'_{k,h})_{h=1}^H)$ (keep the old one locally); $l \leftarrow \mathrm{leaf}(k)$

while $l \neq \emptyset$ **do**

  Retrieve $(G_l, \widetilde{G}_l)$ from M; $G'_l \leftarrow G_l - X_k + X'_k$; draw $U_l^G \sim \mathrm{Unif}(0,1)$; let p_l^G, q_l^G be the old/new joint densities of the H noisy Gram matrices at node l

  if $U_l^G \le \min\{1, q_l^G(\widetilde{G}_l)/p_l^G(\widetilde{G}_l)\}$ **then**

    Store $(G'_l, \widetilde{G}_l)$ in M

  else

    $\widetilde{G}'_{l,h} \leftarrow G'_{l,h} + G_{l,h} - \widetilde{G}_{l,h}$ $\forall h \in [H]$; store $(G'_l, \widetilde{G}'_l)$; **return** ρ-TV-STABLE-LEARN$(k^+(l), M)$

  end if

  $l^+ \leftarrow \mathrm{parent}(l)$; **if** $l^+ = \emptyset$ **then return** M

  for each switch leaf $r \in \mathcal{K}_{\mathrm{sw}}$ after l and before l^+ in post-order **do**

    for $h = H, \dots, 1$ and $j = 1, 2, 3$ **do**

      Compute $b'_{j,r,h}$ by swapping episode k's old contribution for its new one (8); retrieve $(b_{j,r,h}, \widetilde{b}_{j,r,h})$; let $p^b_{j,r,h}, q^b_{j,r,h}$ be the old/new Gaussian densities; draw $U^b_{j,r,h} \sim \mathrm{Unif}(0,1)$

      if $U^b_{j,r,h} \le \min\{1, q^b_{j,r,h}(\widetilde{b}_{j,r,h})/p^b_{j,r,h}(\widetilde{b}_{j,r,h})\}$ **then**

        Store $(b'_{j,r,h}, \widetilde{b}_{j,r,h})$ in M

      else

        $\widetilde{b}'_{j,r,h} \leftarrow b'_{j,r,h} + b_{j,r,h} - \widetilde{b}_{j,r,h}$; store $(b'_{j,r,h}, \widetilde{b}'_{j,r,h})$;

        Resume the vector releases (Appendix A.7) **return** ρ-TV-STABLE-LEARN(r, M)

      end if

    end for

    Keep the noisy statistics; the reconstructed Q, V, and deployed policy at episode r remain valid

  end for

  $l \leftarrow l^+$

end while

return M

Ensure: Updated metadata M

6 MAIN RESULTS

Combining the stability of the noisy releases with the coupling construction above gives the following TV-stability and exact-unlearning guarantees. The first theorem establishes these properties, while the second shows the corresponding regret, unlearning-cost, and memory guarantees. The proofs are given in Appendices B and C, respectively.

Theorem 1 (TV stability and exact unlearning). *Fix any $0 < \rho \le 1$ and use either noise choice in (13)–(14). Then Algorithm 1 is ρ-TV-stable, and Algorithm 2 achieves exact unlearning with respect to Algorithm 1.*

Theorem 2 (Regret and unlearning cost). *Let Assumption 1 hold. Fix any $0 < \rho \le 1$ and instantiate Algorithms 1 and 2 by setting the noise scales equal to the right sides of either Choice (I) (13) or Choice (II) (14). For any $\delta \in (0,1)$, using the shift and confidence radii in Appendix E, the following statements hold.*

1. *Under Assumption 1, with probability at least $1-\delta$, the regret of Algorithm 1 satisfies,*
$$\text{(I)} \quad \mathrm{Regret}(K) \le \min\left\{KH, \widetilde{\mathcal{O}}\left(dH^{3/2}\sqrt{K} + d^2H^{11/4}\sqrt{K/\rho} + d^9H^8/\rho\right)\right\},$$
$$\text{(II)} \quad \mathrm{Regret}(K) \le \min\left\{KH, \widetilde{\mathcal{O}}\left(dH^{3/2}\sqrt{K} + d^2H^3\sqrt{K/\rho} + d^9H^8/\rho\right)\right\}.$$
2. *For a replacement at episode k, let p_q denote the retraining probability under noise choice $q \in \{\mathrm{I}, \mathrm{II}\}$. Then $p_{\mathrm{I}} \le \min\{1, 6\rho H\sqrt{d\log(KH+2)}\}$ and $p_{\mathrm{II}} \le \rho$, and*
$$\mathbb{E}[C_{\mathrm{unlearn}}(k,K)] \le \widetilde{\mathcal{O}}\left(d^4H^2|\mathcal{A}|\log(KH+2) + p_q d^4H^3|\mathcal{A}|K\right),$$
where the two terms correspond to the all-accept cost and full retraining, respectively.
3. *Algorithms 1 and 2 use $\mathcal{O}(HKd^2)$ memory.*

Remark 3. *The regret bounds retain the ρ-independent term $\widetilde{\mathcal{O}}(dH^{3/2}\sqrt{K})$ of LSVI-UCB^{++} (He et al., 2023), which is near-minimax up to logarithmic factors. The second term captures the main cost of the added noise and is smaller under choice (I), reflecting the sharper aggregation obtained by controlling the joint TV distance through the accumulated KL divergence. The final d^9H^8/ρ term is independent of K and represents an additional finite-sample cost from the noisy regressions.*

Comparison with prior work. Compared to the bounds of order $d\sqrt{H^3K} + H^{15/4}d^{7/6}K^{1/2}/\epsilon$ obtained by Sahu (2025), our perturbation also preserves the optimal order of the noise-independent term while differing in the second term and having an additional ρ-dependent K-independent term. For constant $\rho = \epsilon = \mathcal{O}(1)$, our dependence ($H^{2.75}d^2$) on d, H in the second term is better than that in Sahu (2025) ($H^{3.75}d^{1.17}$) for long-horizon problems (i.e., $H > d^{5/6}$).

Such a difference arises from calibrating Gaussian noise to different guarantees. Their privacy analysis first uses zCDP. If ρ_{DP} denotes the total zCDP budget, composition over the $\mathcal{O}(HK)$ releases gives a Gaussian noise scale of order $\widetilde{\mathcal{O}}(\sqrt{HK/\rho_{\mathrm{DP}}})$. Using the standard conversion $\epsilon = \rho_{\mathrm{DP}} + 2\sqrt{\rho_{\mathrm{DP}}\log(1/\delta)}$ leads to $\rho_{\mathrm{DP}} = \widetilde{\Theta}(\epsilon^2)$, and thus the corresponding $\widetilde{\mathcal{O}}(\sqrt{K}/\epsilon)$ dependence in regret. In contrast, we target TV stability directly through KL composition and Pinsker's inequality, which requires perturbations of order $\widetilde{\mathcal{O}}(1/\rho)$. The same Gram perturbation also induces a shift $b_\Lambda = \widetilde{\Theta}(1/\rho)$, and measuring the regression error in the inverse-Gram norm reduces its effective contribution to $(1/\rho)/\sqrt{b_\Lambda} = \widetilde{\mathcal{O}}(1/\sqrt{\rho})$, which leads to the $\widetilde{\mathcal{O}}(\sqrt{K/\rho})$ dependence when accumulating the resulting confidence bonuses over K episodes. The difference is also reflected in the release design, where designing an efficient learner–unlearner pair is more delicate than designing an efficient learner alone; revisit Remark 2 for the different designs.

For one-hot state–action features, $d = SA$, linear MDPs reduce into tabular MDPs and our regret becomes $\widetilde{\mathcal{O}}(SAH^{3/2}\sqrt{K} + S^2A^2H^{11/4}\sqrt{K/\rho} + S^9A^9H^8/\rho)$. We compare to the regret guarantee $\widetilde{\mathcal{O}}(H^2\sqrt{SAK} + H^3S^2A + H^{5/2}S^2A/\rho)$ in the unlearning framework in tabular MDPs of Nguyen-Tang & Arora (2026). Their bound has sharper dependence on S and A, while our ρ-independent term has a better dependence on the horizon, $H^{3/2}$ rather than H^2. This difference reflects the underlying algorithmic structures: the tabular analysis exploits state–action-wise statistics directly, whereas our learner operates through d-dimensional regression statistics for general shared linear features. A similar gap is also present when translating the state-of-the-art regret bound from DP-linear MDPs to DP-tabular MDPs, highlighting a yet-to-be-resolved technical challenge in analyzing stable algorithms when lifting from the tabular setting to the function approximation setting.

7 Conclusion and Discussion

We design the first provably *efficient* learning-unlearning framework for reinforcement learning in linear MDPs, extending the literature beyond tabular MDPs. In particular, our framework guarantees provably exact and efficient unlearning while retaining a regret bound with the ρ-independent term of minimax-optimal order. Future directions include improving/understanding the low-order ρ-dependent terms in regret-stability analysis for unlearning, the memory-utility-computation trade-offs, extensions to richer function classes, and alternative unlearning frameworks to TV stability.

AI USE STATEMENT

Generative AI tools were used to support the development and verification of technical arguments, and to assist with improving the clarity, organization, and presentation of the manuscript. All AI-assisted content was reviewed and verified by the authors. The authors take full responsibility for the final manuscript.

ETHICS STATEMENT

This work is theoretical and does not involve human subjects or empirical datasets. We do not identify any specific ethical concerns arising from the theoretical results presented in this work.

REPRODUCIBILITY STATEMENT

The problem setting, assumptions, and learning and unlearning algorithms are specified in the main text. The appendices provide the complete proofs of the theoretical guarantees, together with the auxiliary results and parameter choices used in the analysis. This work is theoretical and reports no empirical results.

REFERENCES

Rishav Chourasia and Neil Shah. Forget unlearning: Towards true data-deletion in machine learning. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, volume 202 of *PMLR*, 2023. URL https://arxiv.org/abs/2210.08911.

Eric Eaton, Marcel Hussing, Michael Kearns, Aaron Roth, Sikata Bela Sengupta, and Jessica Sorrell. Replicable reinforcement learning with linear function approximation. In *Proceedings of the 14th International Conference on Learning Representations (ICLR)*, 2026. URL https://arxiv.org/abs/2509.08660.

Antonio Ginart, Melody Y. Guan, Gregory Valiant, and James Zou. Making AI forget you: Data deletion in machine learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, 2019. URL https://arxiv.org/abs/1907.05012.

Chuan Guo, Tom Goldstein, Awni Hannun, and Laurens van der Maaten. Certified data removal from machine learning models. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, volume 119 of *PMLR*, 2020. URL https://arxiv.org/abs/1911.03030.

Varun Gupta, Christopher Jung, Seth Neel, Aaron Roth, Saeed Sharifi-Malvajerdi, and Chris Waites. Adaptive machine unlearning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, 2021. URL https://arxiv.org/abs/2106.04378.

Jiafan He, Heyang Zhao, Dongruo Zhou, and Quanquan Gu. Nearly minimax optimal reinforcement learning for linear Markov decision processes. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, volume 202 of *PMLR*, pp. 12790–12822, 2023. URL https://proceedings.mlr.press/v202/he23d.html.

Yaxi Hu, Bernhard Schölkopf, and Amartya Sanyal. Online learning and unlearning. arXiv preprint arXiv:2505.08557, 2025. URL https://arxiv.org/abs/2505.08557.

Chi Jin, Zhuoran Yang, Zhaoran Wang, and Michael I. Jordan. Provably efficient reinforcement learning with linear function approximation. In *Proceedings of the 33rd Conference on Learning Theory (COLT)*, volume 125 of *PMLR*, 2020. URL https://arxiv.org/abs/1907.05388.

Paul Luyo, Evrard Garcelon, Alessandro Lazaric, and Matteo Pirotta. Differentially private exploration in reinforcement learning with linear representation. arXiv preprint arXiv:2112.01585, 2021. URL https://arxiv.org/abs/2112.01585.

Seth Neel, Aaron Roth, and Saeed Sharifi-Malvajerdi. Descent-to-delete: Gradient-based methods for machine unlearning. In *Proceedings of the 32nd International Conference on Algorithmic Learning Theory (ALT)*, volume 132 of *PMLR*, 2021. URL https://arxiv.org/abs/2007.02923.

Dung Daniel Ngo, Giuseppe Vietri, and Zhiwei Steven Wu. Improved regret for differentially private exploration in linear MDP. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*, volume 162 of *PMLR*, 2022. URL https://arxiv.org/abs/2202.01292.

Thanh Nguyen-Tang and Raman Arora. Exact unlearning in reinforcement learning. In *Proceedings of the 43rd International Conference on Machine Learning (ICML)*, volume 306 of *PMLR*, Seoul, South Korea, 2026. URL https://arxiv.org/abs/2606.04182.

Sharan Sahu. Towards optimal differentially private regret bounds in linear MDPs. arXiv preprint arXiv:2504.09339, 2025. URL https://arxiv.org/abs/2504.09339.

Ayush Sekhari, Jayadev Acharya, Gautam Kamath, and Ananda Theertha Suresh. Remember what you want to forget: Algorithms for machine unlearning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, 2021. URL https://arxiv.org/abs/2103.03279.

Enayat Ullah and Raman Arora. From adaptive query release to machine unlearning. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, volume 202 of *PMLR*, pp. 34642–34667, 2023. URL https://proceedings.mlr.press/v202/ullah23a.html.

Enayat Ullah, Tung Mai, Anup Rao, Ryan A. Rossi, and Raman Arora. Machine unlearning via algorithmic stability. In *Proceedings of the 34th Conference on Learning Theory (COLT)*, volume 134 of *PMLR*, 2021. URL https://arxiv.org/abs/2102.13179.

Zichun Ye, Runqi Wang, Xuchuang Wang, Xutong Liu, Shuai Li, and Mohammad Hajiesmaili. Unlearning offline stochastic multi-armed bandits. arXiv preprint arXiv:2605.00638, 2026. URL https://arxiv.org/abs/2605.00638.

Dongruo Zhou and Quanquan Gu. Computationally efficient horizon-free reinforcement learning for linear mixture MDPs. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, 2022. URL https://arxiv.org/abs/2205.11507.

Xingyu Zhou. Differentially private reinforcement learning with linear function approximation. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 6(1):8:1–8:27, 2022. doi: 10.1145/3508028. URL https://doi.org/10.1145/3508028.

CONTENTS

1 Introduction **1**

2 Related Work **2**

3 Problem setup **3**

3.1 Markov decision process 3

3.2 Linear MDP 3

3.3 Exact unlearning 4

4 TV-stable learner **5**

4.1 Variance-Weighted Sufficient Statistics 5

4.2 Perturbing sufficient statistics 6

4.3 Rare-switching learner 7

5 Exact unlearning by Gaussian coupling **7**

6 Main results **8**

7 Conclusion and discussion **9**

A Auxiliary results **14**

A.1 The total-mass condition is insufficient 14

A.2 Proof of Proposition 1 14

A.3 Positive-definiteness of the noisy Gram matrices 14

A.4 Sensitivity of the released statistics 15

A.5 Choosing the noise scales 15

A.6 Reflection coupling 15

A.7 Continuation after a rejected Bellman vector 16

A.8 TV tools 16

B Proof of Theorem 1 **17**

C Proof of Theorem 2 **19**

D Unlearning cost **19**

D.1 Retraining probability 19

D.2 Cost conditional on acceptance 20

D.3 Memory cost 21

E Regret analysis **22**

E.1 Proof overview 22

E.2 Matrix comparison and regression confidence 23
E.3 Bonus and variance sums 30
E.4 Regret bound 33

A AUXILIARY RESULTS

A.1 THE TOTAL-MASS CONDITION IS INSUFFICIENT

Example 1 (A signed-measure counterexample). *Let $\mathcal{S} = \{x, y\}$, let there be one action a, and set $d = 2$. Fix $\varepsilon > 0$. At every stage h, take $\theta_h = 0$, $\phi(s, a) = (1/\sqrt{2}, 0)$ for both states, and define the vector signed measure by*

$$\mu_h(\{x\}) = \left(\frac{1}{\sqrt{2}}, \frac{1}{2\varepsilon}\right), \qquad \mu_h(\{y\}) = \left(\frac{1}{\sqrt{2}}, -\frac{1}{2\varepsilon}\right).$$

The induced transition kernel is valid: for either state, $P_h(\{x\} \mid s, a) = P_h(\{y\} \mid s, a) = 1/2$. Also $\|\phi(s, a)\|_2 = 1/\sqrt{2} \le 1$, while

$$\mu_h(\mathcal{S}) = (\sqrt{2}, 0), \qquad \|\mu_h(\mathcal{S})\|_2 = \sqrt{2} = \sqrt{d}.$$

Now fix $B > 0$ and take the test function $V(x) = B$, $V(y) = 0$, which satisfies $0 \le V \le B$. Then

$$\int_{\mathcal{S}} V(s')\, \mu_h(\mathrm{d}s') = \left(\frac{B}{\sqrt{2}}, \frac{B}{2\varepsilon}\right), \qquad \left\| \int_{\mathcal{S}} V(s')\, \mu_h(\mathrm{d}s') \right\|_2 = B\sqrt{\frac{1}{2} + \frac{1}{4\varepsilon^2}} \longrightarrow \infty \quad \text{as } \varepsilon \downarrow 0.$$

Thus the total-mass condition does not imply the bound $B\sqrt{d}$ for bounded test functions; taking $B = 1$ gives the normalized case in Assumption 1.

A.2 PROOF OF PROPOSITION 1

Proof. Define

$$w_h^\pi := \theta_h + \int_{\mathcal{S}} V_{h+1}^\pi(s') \mu_h(\mathrm{d}s').$$

Putting the linear reward and transition model into the Bellman equation gives

$$\begin{aligned} Q_h^\pi(s, a) &= \langle \phi(s, a), \theta_h \rangle + \int_{\mathcal{S}} V_{h+1}^\pi(s') \langle \phi(s, a), \mu_h(\mathrm{d}s') \rangle \\ &= \left\langle \phi(s, a), \theta_h + \int_{\mathcal{S}} V_{h+1}^\pi(s') \mu_h(\mathrm{d}s') \right\rangle. \end{aligned}$$

□

A.3 POSITIVE-DEFINITENESS OF THE NOISY GRAM MATRICES

Lemma 1 (Positive-definiteness of the noisy Gram matrices). *For $\delta_\Lambda \in (0, 1)$, define*

$$\nu_\Lambda := \sigma_\Lambda \sqrt{L_K} \left(\sqrt{\frac{d(d+1)}{2}} + \sqrt{2 \log \frac{2HK}{\delta_\Lambda}} \right), \qquad b_\Lambda := c_\Lambda - \nu_\Lambda.$$

Suppose that

$$c_\Lambda \ge \max\left\{ H^{-2} + \nu_\Lambda,\, 4dH\nu_\Lambda \right\}, \tag{9}$$

then, with probability at least $1 - \delta_\Lambda$, simultaneously for all (k, h),

$$\sum_{i<k} X_{i,h} + b_\Lambda I \preceq \widetilde{\Lambda}_{k,h} \preceq \sum_{i<k} X_{i,h} + (c_\Lambda + \nu_\Lambda) I. \tag{10}$$

In particular, every $\widetilde{\Lambda}_{k,h}$ is positive definite.

Proof. By the definition of GetNoisySum,

$$\widetilde{\Lambda}_{k,h} = \sum_{i<k} X_{i,h} + c_\Lambda I + \sum_{\mathcal{I} \in \mathcal{C}(k-1)} K_{1,\mathcal{I},h}.$$

Since $\mathcal{C}(k-1)$ contains at most L_K intervals, the noise sum is a normalized GOE matrix with scale at most $\sigma_\Lambda \sqrt{L_K}$. A Gaussian operator-norm bound and a union bound over (k, h) imply that its operator norm is at most ν_Λ simultaneously for all (k, h) with probability at least $1 - \delta_\Lambda$. Equation (10) follows directly. Finally, (9) gives $b_\Lambda \ge H^{-2} > 0$, and hence $\widetilde{\Lambda}_{k,h} \succ 0$. □

A.4 SENSITIVITY OF THE RELEASED STATISTICS

Replacing one episode changes each affected dyadic Gram sum by the difference of one contribution. Since $\|\phi\phi^\top\|_{\mathrm{F}} = \|\phi\|_2^2$, $\bar{\sigma}_{i,h} \geq \sqrt{H}$, and $\|\phi_h^i\|_2 \leq 1$, we have $\|X_{i,h}\|_{\mathrm{F}} \leq 1/H$. Therefore,

$$\Delta_\Lambda := \sup \|X_{i,h} - X'_{i,h}\|_{\mathrm{F}} \leq \frac{2}{H}. \tag{11}$$

For the vector statistics, conditioning on the same preceding noisy values fixes the switch decision and the current value functions, so only the contribution of the replaced episode can change. Since the clipped value functions lie in $[0, H]$ and $(\bar{\sigma}_{i,h})^{-2} \leq 1/H$, one contribution has norm at most 1, 1, and H for types 1, 2, and 3, respectively. Therefore,

$$\Delta_1 = \Delta_2 := 2, \qquad \Delta_3 := 2H. \tag{12}$$

Since vector releases occur only at switch episodes, each vector type is released at most $HM_{\max}$ times.

A.5 CHOOSING THE NOISE SCALES

Choosing the noise scales. Fix $0 < \rho \leq 1$. We consider two choices of Gaussian noise scales:

$$\text{(I)} \quad \sigma_\Lambda \geq \frac{\Delta_\Lambda \sqrt{HL_K}}{\rho}, \qquad \sigma_j \geq \frac{\Delta_j \sqrt{HM_{\max}}}{\rho}, \qquad j \in [3], \tag{13}$$

$$\text{(II)} \quad \sigma_\Lambda \geq \frac{2\sqrt{H}\, L_K \Delta_\Lambda}{\rho}, \qquad \sigma_j \geq \frac{2HM_{\max}\Delta_j}{\rho}, \qquad j \in [3]. \tag{14}$$

Both choices make the policy sequence and noisy metadata ρ-TV-stable. Choice (I) uses square-root composition and therefore requires less noise, while Choice (II) uses larger noise so that the rejection probabilities of the local coupling tests sum to at most ρ.

Table 1: Comparison of the two noise choices in (K, H, d, ρ); universal constants suppressed.

	Choice (I)	**Choice (II)**
Controls	TV distance	Retraining probability
σ_Λ	$\Theta\big(\sqrt{\log(K+1)}/(\sqrt{H}\rho)\big)$	$\Theta\big(\log(K+1)/(\sqrt{H}\rho)\big)$
σ_1, σ_2	$\Theta\big(H\sqrt{d\log(KH+2)}/\rho\big)$	$\Theta\big(dH^2\log(KH+2)/\rho\big)$
σ_3	$\Theta\big(H^2\sqrt{d\log(KH+2)}/\rho\big)$	$\Theta\big(dH^3\log(KH+2)/\rho\big)$
TV distance	$\leq \rho$	$\mathcal{O}\big(\rho/\sqrt{\log(K+1)}\big)$
$\Pr(\text{retrain})$	$\leq \min\{1,\, 6\rho H\sqrt{d\log(KH+2)}\}$	$\leq \rho$
Regret	$\widetilde{\mathcal{O}}\big(dH^{3/2}\sqrt{K} + d^2H^{11/4}\sqrt{K/\rho} + d^9H^8/\rho\big)$	$\widetilde{\mathcal{O}}\big(dH^{3/2}\sqrt{K} + d^2H^3\sqrt{K/\rho} + d^9H^8/\rho\big)$

Comparison of the two noise choices. For later concentration arguments, for $\delta_b \in (0, 1)$ define

$$\zeta_j := \sigma_j \left(\sqrt{d} + \sqrt{2\log\frac{6HM_{\max}}{\delta_b}}\right), \qquad j \in [3].$$

With probability at least $1 - \delta_b$, all released vector noises satisfy $\|\omega_{j,k,h}\|_2 \leq \zeta_j$.

A.6 REFLECTION COUPLING

Lemma 2 (Reflection coupling for one release)**.** *Let P and Q be Gaussian laws with densities p and q, respective means c and c', and the same positive definite covariance on their vector space. This includes the vector Gaussian and normalized-GOE releases above. Draw $X \sim P$ and independently $U \sim \mathrm{Unif}(0, 1)$. Keep X when*

$$U \leq \min\{1, q(X)/p(X)\}.$$

On rejection, return the reflected value

$$X' := c' + c - X.$$

Then the returned value has law Q, and reflection occurs with probability $\mathrm{TV}(P, Q)$.

Proof. The main point is that reflection swaps the two Gaussian densities. Put $R(x) = c' + c - x$. The map R is its own inverse and has absolute Jacobian determinant one. Because the two Gaussian laws have the same centered density, $p(R(x)) = q(x)$ and $q(R(x)) = p(x)$. The kept part has density $\min\{p, q\}$. At a point y, the reflected part has density

$$\big(p(R(y)) - q(R(y))\big)_+ = (q(y) - p(y))_+.$$

Their sum is $q(y)$. The probability of reflection is $\int (p - q)_+ = \mathrm{TV}(P, Q)$. □

A.7 Continuation after a rejected Bellman vector

The vector-rejection branch of Algorithm 2 resumes the learner within switch episode r. If the test for $\widetilde{b}_{j,r,h}$ rejects, retain its reflected value and updated clean mean. Recompute and release the remaining vectors of type j at stages $h - 1, \ldots, 1$, using the updated value functions. Then process types $j + 1, \ldots, 3$, each at stages $H, \ldots, 1$. After all three vector types are complete, continue episode r and the later episodes with ρ-TV-Stable-Learn. Every release after the rejected one is generated from the updated state; the reflected release is never redrawn.

A.8 TV tools

For distributions with densities p and q, let $D_{\mathrm{KL}}(P\|Q) := \int p(x) \log(p(x)/q(x))\, \mathrm{d}x$ denote their KL divergence. Consider the noisy Gram matrices and Bellman vectors in the order generated by the learner. After each coupling step, we show that all noisy values generated so far have the same joint distribution as in a fresh run on the modified data. Sensitivity bounds and Gaussian divergence control the TV distance of these noisy values, while reflection coupling matches them one at a time. Since these noisy values, together with the user response trees, determine the policies, trajectories, variance scales, and clean sufficient statistics, this also gives the correct distribution for the entire learner state.

Proposition 2 (Gaussian/GOE divergence and adaptive composition)**.** *The following statements hold.*

1. *Equal-covariance vector Gaussians satisfy*

$$D_{\mathrm{KL}}(\mathcal{N}(c, \sigma^2 I)\|\mathcal{N}(c', \sigma^2 I)) = \frac{\|c - c'\|_2^2}{2\sigma^2}.$$

2. *Two normalized GOE releases with symmetric means A, A' and scale σ satisfy*

$$D_{\mathrm{KL}}(P_A\|P_{A'}) = \frac{\|A - A'\|_F^2}{2\sigma^2}.$$

3. *For adaptive releases, the KL divergence of their joint distribution is the expectation of the sum of the conditional KL divergences. Moreover,* $\mathrm{TV}(P, Q) \le \sqrt{D_{\mathrm{KL}}(P\|Q)/2}$.

Proof. We prove the three statements in order.

For two Gaussian vectors, expand the log density ratio and take its expectation. This gives the first formula. For the GOE formula, the diagonal entries are independent Gaussians of variance σ^2 and the upper off-diagonal entries are independent Gaussians of variance $\sigma^2/2$. Summing their scalar KL divergences gives

$$\frac{1}{2\sigma^2}\left(\sum_r (A_{rr} - A'_{rr})^2 + 2\sum_{r<s} (A_{rs} - A'_{rs})^2\right) = \frac{\|A - A'\|_F^2}{2\sigma^2}.$$

For adaptive releases, write the joint density as a product of conditional densities. The log of the ratio is then a sum. Taking expectation gives the chain rule. For Pinsker's inequality, let E attain the TV distance and put $a = P(E)$ and $b = Q(E)$. Grouping the density into E and E^c gives

$$D_{\mathrm{KL}}(P\|Q) \geq a \log \frac{a}{b} + (1-a) \log \frac{1-a}{1-b}.$$

As a function of a, the right-hand side and its first derivative are zero at $a = b$. Its second derivative is $1/[a(1-a)] \geq 4$. Hence it is at least $2(a-b)^2 = 2\,\mathrm{TV}(P,Q)^2$. The chain rule is needed here because each release mean may depend on earlier noisy releases. □

B Proof of Theorem 1

Proof. The proof has two parts. We first bound the change in the output law. We then show that the coupling used by UNLEARN gives exactly the law of a fresh run.

We first prove TV stability. Fix neighboring user sequences Z and Z'. Let R_Z and $R_{Z'}$ be the joint laws of the ordered noisy releases on these inputs, and let P_Z and $P_{Z'}$ be the corresponding laws of the policy sequence and noisy metadata. It suffices to control the release laws, because the policy sequence and noisy metadata are fixed functions of the releases. Condition on the same earlier noisy values. Replacing one episode changes at most H Gram matrices at each of at most L_K tree nodes. Each vector type has at most $HM_{\max}$ Gaussian releases. At a non-switch episode, reusing the most recent released vector is a deterministic operation and adds no conditional KL divergence. Equations (11) and (12), together with the first two parts of Proposition 2, give

$$D_{\mathrm{KL}}(R_Z\|R_{Z'}) \leq \frac{HL_K\Delta_\Lambda^2}{2\sigma_\Lambda^2} + \frac{HM_{\max}}{2}\sum_{j=1}^{3}\frac{\Delta_j^2}{\sigma_j^2}.$$

Under choice (I), the right side is at most $2\rho^2$. Under choice (II), it is at most

$$\frac{\rho^2}{8}\left(\frac{1}{L_K} + \frac{3}{HM_{\max}}\right) \leq \frac{\rho^2}{2}.$$

The adaptive chain rule adds these conditional KL bounds. Pinsker's inequality then gives

$$\mathrm{TV}(R_Z, R_{Z'}) \leq \rho.$$

The policies and $\widetilde{M}$ are fixed functions of these values. A fixed function cannot increase TV distance, so $\mathrm{TV}(P_Z, P_{Z'}) \leq \mathrm{TV}(R_Z, R_{Z'}) \leq \rho$ and Algorithm 1 is ρ-TV-stable. Under choice (II), the same calculation gives the sharper output-law bound

$$\mathrm{TV}(P_Z, P_{Z'}) \leq \frac{\rho}{4}\sqrt{\frac{1}{L_K} + \frac{3}{HM_{\max}}}. \tag{15}$$

Thus choice (II) controls the retraining probability by ρ and gives a smaller upper bound on the TV distance between neighboring output laws.

We next prove exactness. Put

$$Z' := (z_1, \ldots, z_{k-1}, z', z_{k+1}, \ldots, z_K).$$

It is enough to match the joint law of the noisy releases. The fixed tie-breaking rule makes every policy, switch decision, and variance scale a fixed function of the earlier noisy releases. The response trees in Z' then fix the trajectories. The trajectories and noisy releases fix all clean sufficient statistics. Thus all metadata has the fresh-run law once the ordered list of noisy releases has that law.

The only bookkeeping issue is that tree and vector releases are interleaved. To track their order, compare the output of Algorithm 2 with the output of a fresh run on Z'. For $t \in [K]$, let

$$\mathcal{T}_{\leq t} := \{l : \ l \text{ is an ancestor of a leaf in } \{1, \ldots, t\} \text{ and } \mathcal{I}_l \subseteq [1, t]\}.$$

Here $\mathcal{I}_l$ is the interval at node l, and a leaf is its own ancestor. The condition $\mathcal{I}_l \subseteq [1, t]$ means that all leaves below l have been processed. Hence $\mathcal{T}_{\leq t}$ is the set of ancestor nodes whose noisy values have been created after episodes $1, \ldots, t$. The new nodes created after episode t are $\mathcal{T}_{\leq t} \setminus \mathcal{T}_{\leq t-1}$.

The vector releases and policy at episode k occur before x_k. They depend only on $z_1, \dots, z_{k-1}$ and the preceding noisy values. Therefore the old and fresh computations use the same policy $\widetilde{\pi}^k$. Querying the response tree z' with this policy gives the same trajectory x'_k as a fresh run.

Condition on all earlier noisy values at one changed release. Let p and q be the equal-covariance Gaussian densities of the old and fresh values at the next release. On acceptance, the algorithm reuses the old value; on rejection, it reflects that value as in Lemma 2. The two branches contribute

$$\underbrace{p(y)\min\left\{1, \frac{q(y)}{p(y)}\right\}}_{\text{accept and reuse}} + \underbrace{(q(y)-p(y))_+}_{\text{reject and reflect}} = \min\{p(y), q(y)\} + (q(y)-p(y))_+ = q(y). \tag{16}$$

Thus the returned value has the fresh density q. If the clean value is unchanged, then $p = q$, and reuse already has density q.

We now extend the one-release identity to the full adaptive history by induction on $t = k, \dots, K$. The induction statement is that the trajectories through episode t, the noisy values at the nodes in $\mathcal{T}_{\le t}$, and the vector releases through episode t have the same joint distribution as in a fresh run on Z'.

For $t = k$, we have already shown that the replacement trajectory is x'_k. Apply (16) in post-order to the nodes in $\mathcal{T}_{\le k} \setminus \mathcal{T}_{\le k-1}$. This proves the induction statement at k.

Suppose the claim holds at t. The updated noisy Gram values may differ from the old values, but their joint history has the fresh-run law. The determinant test is a fixed function of this history. Therefore the switch decision at episode $t+1$ also has the fresh-run law.

Fix an earlier history and consider the two possible switch decisions. At a switch, the type 1 means depend on higher-stage optimistic values, the type 2 means on higher-stage pessimistic values, and the type 3 means on optimistic values already determined by type 1. Independent Gaussian noises therefore give the same joint conditional law in type-first order. Apply (16) with $j = 1, 2, 3$ and $h = H, \dots, 1$ within each type before the first rejection. Afterward the continuation draws the remaining releases from the updated state. If there is no switch, both runs keep the earlier estimates and policy. In both cases, the policy and the trajectory from z_{t+1} have the fresh conditional law.

Now consider the new nodes in $\mathcal{T}_{\le t+1} \setminus \mathcal{T}_{\le t}$. On the all-accept branch, the algorithm keeps the noisy values and later trajectories. If a node interval does not contain k, its clean value is unchanged, so its old noisy value already has the fresh conditional density. If the interval contains k, use (16). After a rejection, LEARN draws the rest. These cases prove the induction statement at $t+1$. The stored switch decision is valid on the all-accept branch because the earlier noisy values have not changed. After reflection, LEARN uses the updated noisy values in all later tests. Algorithm 2 performs these operations in post-order: it follows the ancestors of leaf k and, between one node and its parent, examines the intervening switch leaves. Skipping the non-switch leaves does not change the order because they contain no new vector release.

It remains to check that several accepted tests preserve the same joint law. Fix a complete history of noisy releases in post-order, including releases that require no rejection test. The switch decisions are determined by this history. Suppose it contains n_{rel} releases. At release s, let p_s be the old learner's conditional density given the fixed preceding noisy values, and let q_s be the fresh learner's conditional density given that same history. The factors for earlier acceptances are kept separately below. Evaluate both densities at the fixed noisy value and set $a_s := \min\{p_s, q_s\}$. For a release that requires no rejection test, $p_s = q_s$, so $a_s = q_s$ and $(q_s - p_s)_+ = 0$. The test uniforms are independent. Thus the all-accept branch contributes $\prod_s a_s$. If the first rejection occurs at test s, reflection contributes $(q_s - p_s)_+$ and Algorithm 1 draws every later noisy value from the updated metadata. The total density is

$$\prod_{s=1}^{n_{\mathrm{rel}}} a_s + \sum_{s=1}^{n_{\mathrm{rel}}} \left(\prod_{u<s} a_u\right)(q_s - p_s)_+ \left(\prod_{u>s} q_u\right) = \prod_{s=1}^{n_{\mathrm{rel}}} (a_s + (q_s - p_s)_+) = \prod_{s=1}^{n_{\mathrm{rel}}} q_s.$$

The first equality groups the paths by their first rejection. The last expression is exactly the fresh joint density. Given this noisy history and Z', the learner fixes every other entry of the metadata. Therefore the output of Algorithm 2 has the same distribution as a fresh run of Algorithm 1 on the modified user sequence. Thus Algorithm 2 achieves exact unlearning with respect to Algorithm 1.

□

C Proof of Theorem 2

Proof. The regret bounds follow directly from Theorem 5 with the stated shift and confidence radii, taking $\delta_\Lambda = \delta_b = \delta_0 = \delta/9$. The retraining guarantees follow from Theorem 3. Since $L_K \le 4\log(KH+2)$ and, for $C_{\mathrm{sw}} = 9$, $M_{\max} \le 10dH\log(KH+2)$, its bound on η_{I} gives $p_{\mathrm{I}} \le \min\{1, 6\rho H\sqrt{d\log(KH+2)}\}$, while choice (II) gives $p_{\mathrm{II}} \le \rho$.

For the computational cost, (23) gives $C_{\mathrm{accept}}(k,K) = \mathcal{O}(d^4H^2|\mathcal{A}|\log(KH+2))$, whereas a rejection incurs at most one additional full run with cost given by (20). Conditioning on whether a rejection occurs yields the stated expected unlearning-cost bound. The memory guarantee follows from Appendix D.3. □

D Unlearning cost

The unlearning cost is determined by whether a coupling test rejects. If all tests accept, Algorithm 2 keeps the noisy statistics unchanged and only updates the clean statistics affected by the replacement. If a test rejects, it performs the preceding updates and then invokes the learner once to regenerate the remaining computation. Therefore,

$$\mathbb{E}[C_{\mathrm{unlearn}}(k,K)] \le C_{\mathrm{accept}}(k,K) + \Pr(\mathrm{retrain})\, C_{\mathrm{full}}(K).$$

D.1 Retraining probability

Exactness holds for either noise choice; the noise scales determine how often a coupling test rejects. The next theorem adds the conditional rejection probabilities of the tests performed by Algorithm 2.

Theorem 3 (Retraining probability)**.** *Consider Algorithms 1 and 2 with positive Gaussian noise scales, and define*

$$\eta := \min\left\{1, \frac{\sqrt{H}\,L_K\Delta_\Lambda}{2\sigma_\Lambda} + \frac{HM_{\max}}{2}\sum_{j=1}^{3}\frac{\Delta_j}{\sigma_j}\right\}. \tag{17}$$

For the two noise choices, define

$$\eta_{\mathrm{I}} := \min\left\{1, \frac{\rho}{2}\left(\sqrt{L_K} + 3\sqrt{HM_{\max}}\right)\right\}, \qquad \eta_{\mathrm{II}} := \rho.$$

For a replacement at episode k, Algorithm 2 calls Algorithm 1 again with probability at most η. Under choice (I), $\eta \le \eta_{\mathrm{I}}$; under choice (II), $\eta \le \eta_{\mathrm{II}}$. Under choice (I), the tree-node tests contribute at most $\rho\sqrt{L_K}/2$, and the tests for the three vector types together contribute at most $3\rho\sqrt{HM_{\max}}/2$. Under choice (II), the Gram tests and the tests for each vector type contribute at most $\rho/4$, giving $\eta_{\mathrm{II}} = \rho$. In either case the learner is called at most once.

Proof. The main idea is to split retraining by the first rejected test. We bound the rejection probability of each test, and then add these bounds.

Enumerate the coupling tests in their execution order. For test s, let $\mathcal{E}_{<s}$ denote acceptance of every earlier test and let $\mathcal{R}_s$ denote rejection at test s. Conditional on $\mathcal{E}_{<s}$, the two runs have the same preceding noisy values and switch decisions. Maximal coupling and Proposition 2 therefore bound the conditional rejection probability by the TV distance between the two current Gaussian laws.

At one Gram node, Algorithm 2 couples all H matrices together. Their squared mean displacement is at most $H\Delta_\Lambda^2$. Pinsker's inequality therefore bounds the conditional rejection probability at this node by $\sqrt{H}\Delta_\Lambda/(2\sigma_\Lambda)$. There are at most L_K ancestors of leaf k, so all Gram tests together contribute at most $L_K\sqrt{H}\Delta_\Lambda/(2\sigma_\Lambda)$.

A type-j vector release has conditional rejection probability at most $\Delta_j/(2\sigma_j)$. The algorithm tests at most $HM_{\max}$ releases of each of the three types. All vector tests therefore contribute at most $(HM_{\max}/2)\sum_{j=1}^{3}\Delta_j/\sigma_j$.

The events $\mathcal{E}_{<s} \cap \mathcal{R}_s$ are disjoint because each one gives a different first rejected test. Thus

$$\Pr(\text{retrain}) = \sum_s \Pr(\mathcal{E}_{<s}) \Pr(\mathcal{R}_s \mid \mathcal{E}_{<s}) \le \sum_s \Pr(\mathcal{R}_s \mid \mathcal{E}_{<s})$$
$$\le \frac{\sqrt{H}\, L_K \Delta_\Lambda}{2\sigma_\Lambda} + \frac{H M_{\max}}{2} \sum_{j=1}^{3} \frac{\Delta_j}{\sigma_j}. \tag{18}$$

The probability is also at most one. Taking the smaller of the two bounds proves (17). Under choice (I), (18) gives

$$\Pr(\text{retrain}) \le \min\left\{1, \underbrace{\frac{\rho}{2}\sqrt{L_K}}_{\text{Gram tests}} + \underbrace{\frac{3\rho}{2}\sqrt{H M_{\max}}}_{\text{vector tests}}\right\} = \eta_{\mathrm{I}}.$$

The factor $\sqrt{L_K}$ appears because one test couples all H Gram matrices at a node, while the algorithm tests the L_K nodes one by one. This calculation uses the local couplings actually implemented by Algorithm 2; the TV stability bound in Theorem 1 does not replace it. Under choice (II), the same bound gives

$$\Pr(\text{retrain}) \le \underbrace{\frac{\rho}{4}}_{\text{Gram tests}} + \underbrace{3 \cdot \frac{\rho}{4}}_{\text{vector tests}} = \rho = \eta_{\mathrm{II}}.$$

□

Retraining and TV stability. Table 1 highlights the distinction between TV stability and retraining probability. Choice (I) uses smaller noise and adaptive composition, so the effect of multiple releases accumulates through square-root terms rather than linearly, leading to the better regret bound. Yet ρ-TV stability only guarantees the existence of a coupling with disagreement probability at most ρ. Consequently, under choice (I), the retraining probability is at most $\min\{1, 6\rho H\sqrt{d \log(KH+2)}\}$ rather than ρ. Choice (II) uses larger noise to control these rejection probabilities directly, giving $\Pr(\text{retrain}) \le \rho$ at the cost of a larger regret term. Thus, the two choices make explicit the trade-off between exact unlearning, retraining probability, regret, and computational cost that must be balanced when designing the learner and unlearner jointly.

The pathwise rejection calculation parallels the tree coupling in Theorem 3.3 of Nguyen-Tang & Arora (2026); the switch-only Bellman vectors are the additional cost of function approximation.

D.2 Cost conditional on acceptance

Put $M_{\mathrm{sw}} := |\mathcal{K}_{\mathrm{sw}}|$. By the deterministic switch cap

$$M_{\mathrm{sw}} \le M_{\max} = \lceil C_{\mathrm{sw}} dH \log(KH+2) \rceil = \mathcal{O}(dH \log(KH+2)). \tag{19}$$

When d and H are fixed, $M_{\max} = \mathcal{O}(\log K)$ and the all-accept cost is $\mathcal{O}(\log K)$. We count dense matrix solves and finite-action value evaluations as arithmetic operations. A complete run then costs $\widetilde{\mathcal{O}}(d^4 H^3 |\mathcal{A}| K)$; the tree updates and the constant number of noisy sufficient statistics do not change this order. Accordingly, define

$$C_{\mathrm{full}}(K) := d^4 H^3 |\mathcal{A}| K. \tag{20}$$

To bound $C_{\mathrm{accept}}(k, K)$, suppose that every coupling test accepts. Then all noisy statistics remain unchanged, and the unlearner only updates the clean statistics. The clean Gram matrices change at the L_K tree nodes containing episode k, at cost $\mathcal{O}(H L_K d^2)$.

Updating the three clean vectors also requires their value targets in (5). These value functions are not retained. The unlearner therefore reconstructs the Q- and value functions at at most $M_{\max}$ switch episodes once from the stored $\widetilde{\Lambda}_{r,h}$ and $\widetilde{b}_{j,r,h}$. The matrix solves and value evaluations over the finite action set cost $\mathcal{O}(H M_{\max}(d^3 + |\mathcal{A}| d^2))$. The same reconstructed functions are used for all three vector types and for the replacement trajectory.

Only switches $r > k$ require a change to the clean vectors. At each such switch, the unlearner subtracts the old episode-k contribution and adds the new one. These corrections cost $\mathcal{O}(HM_{\max}d)$ in total. Thus, when every test accepts, the cost is bounded by

$$C_{\text{accept}}(k, K) := HL_K d^2 + HM_{\text{sw}}(d^3 + |\mathcal{A}|d^2 + d) \tag{21}$$

$$\leq HL_K d^2 + HM_{\max}(d^3 + |\mathcal{A}|d^2 + d). \tag{22}$$

This bound reconstructs the functions at every switch even though only switches after k require clean-vector corrections. Substituting (19) and $L_K = \mathcal{O}(\log(K+1))$ gives

$$C_{\text{accept}}(k, K) = \mathcal{O}\big(Hd^2 \log(K+1) + dH^2 \log(KH+2)(d^3 + |\mathcal{A}|d^2 + d)\big). \tag{23}$$

Consequently,

$$C_{\text{accept}}(k, K) = \widetilde{\mathcal{O}}(d^4H^2 + d^3H^2|\mathcal{A}|). \tag{24}$$

For fixed d, H, and $|\mathcal{A}|$, the two costs are $C_{\text{accept}}(k, K) = \mathcal{O}(\log(K+1))$ and $C_{\text{full}}(K) = \Theta(K)$.

Proposition 3 (Arithmetic cost comparison)**.** *Let $C_{\text{unlearn}}(k, K)$ denote the computational cost of processing one replacement. Under the implementation above,*

$$\mathbb{E}[C_{\text{unlearn}}(k, K)] = \widetilde{\mathcal{O}}(C_{\text{accept}}(k, K) + \eta C_{\text{full}}(K)). \tag{25}$$

Retraining from scratch costs $\widetilde{\mathcal{O}}(C_{\text{full}}(K))$. Under choice (I), η in (25) may be replaced by η_{I}; under choice (II), it may be replaced by η_{II}.

Proof. On the all-accept branch, the operations above cost $\mathcal{O}(C_{\text{accept}}(k, K))$. At the first rejection, the unlearner calls the full learner and therefore incurs at most the full-retraining cost. This happens with probability at most η and occurs at most once. Taking expectation proves (25). □

Equations (20) and (23) give

$$\frac{C_{\text{accept}}(k, K)}{C_{\text{full}}(K)} = \mathcal{O}\left(\frac{\log(KH+2)}{K}\right) = \widetilde{\mathcal{O}}\left(\frac{1}{K}\right). \tag{26}$$

For fixed d and H, the middle expression is $\mathcal{O}(\log(K+1)/K)$. Combining this ratio with Proposition 3 gives

$$\frac{\mathbb{E}[C_{\text{unlearn}}(k, K)]}{C_{\text{full}}(K)} = \begin{cases} \widetilde{\mathcal{O}}(K^{-1} + \eta_{\text{I}}), & \text{under choice (I),} \\ \widetilde{\mathcal{O}}(K^{-1} + \eta_{\text{II}}) = \widetilde{\mathcal{O}}(K^{-1} + \rho), & \text{under choice (II).} \end{cases}$$

The saving does not assume that the clean Bellman vectors can be updated without reconstructing their value targets: the cost of reconstructing the Q- and value functions at every switch is already included in $C_{\text{accept}}(k, K)$.

D.3 Memory cost

The binary tree is the largest stored object. It has $\mathcal{O}(K)$ nodes, and each node stores a constant number of $d \times d$ matrices at every stage. It therefore requires $\mathcal{O}(HKd^2)$ numerical values. The stored noisy matrices at switch episodes require $\mathcal{O}(H \min\{K, M_{\max}\}d^2)$ values, and the vector statistics stored at switches require $\mathcal{O}(H \min\{K, M_{\max}\}d)$ values. The variance scales require $\mathcal{O}(HK)$ values, and the switch indices require $\mathcal{O}(K)$ values. None of these terms exceeds the tree storage. Hence $\text{size}(M) = \mathcal{O}(HKd^2)$ numbers and indices, in addition to the $\mathcal{O}(HK)$ transition records. Policies, value functions, and regression vectors are recomputed and need not be retained.

Algorithm 2 updates these objects in place. It factorizes each stored switch matrix once and retains the factorization and regression vectors during the update. This requires $\mathcal{O}(HM_{\max}d^2)$ temporary values, which is at most $\mathcal{O}(HKd^2)$. Therefore its peak memory remains $\mathcal{O}(HKd^2)$ numbers and indices, together with $\mathcal{O}(HK)$ transition records.

E REGRET ANALYSIS

E.1 PROOF OVERVIEW

This section explains the main ideas behind the regret part of Theorem 2. The proof first turns regret into a cumulative confidence bonus and a martingale term. It then bounds the bonus using the weighted feature norms and controls the total estimated variance through four explicit terms. These steps show the source of each term in the final rate. The following subsections give the concentration arguments and complete calculations.

We begin by decomposing the regret of one episode. Define

$$\Delta_k := V_1^\star(s_1^k) - V_1^{\widetilde{\pi}^k}(s_1^k), \qquad e_{k,h}(s) := \big[\widetilde{\widehat{V}}_{k,h}(s) - V_h^{\widetilde{\pi}^k}(s)\big]_{[0,H]}, \qquad e_{k,H+1} := 0.$$

On the high-probability confidence event, the optimistic estimate satisfies $\widetilde{\widehat{V}}_{k,1}(s_1^k) \geq V_1^\star(s_1^k)$, so $\Delta_k \leq e_{k,1}(s_1^k)$. Because the policy is greedy with respect to $\widetilde{\widehat{Q}}_{k,h}$, the upper confidence bound at the most recent switch gives

$$\widetilde{\widehat{Q}}_{k,h}(s_h^k, a_h^k) \leq r_h(s_h^k, a_h^k) + P_h \widetilde{\widehat{V}}_{k,h+1}(s_h^k, a_h^k) + 2\widehat{\beta} u_{k,h}.$$

Subtracting the Bellman equation of the executed policy from this inequality gives

$$e_{k,h}(s_h^k) \leq 2\min\{\widehat{\beta} u_{k,h}, H\} + e_{k,h+1}(s_{h+1}^k) + \xi_{k,h}, \tag{27}$$

where

$$\xi_{k,h} := P_h e_{k,h+1}(s_h^k, a_h^k) - e_{k,h+1}(s_{h+1}^k)$$

has conditional mean zero. Summing (27) over stages makes the value gaps telescope. Summing over episodes then gives

$$\mathrm{Regret}(K) \leq 2\sum_{k=1}^{K}\sum_{h=1}^{H} \min\{\widehat{\beta} u_{k,h}, H\} + \sum_{k=1}^{K}\sum_{h=1}^{H} \xi_{k,h}. \tag{28}$$

The second term is a bounded martingale sum and is $\widetilde{\mathcal{O}}(H^{3/2}\sqrt{K})$. It remains to control the cumulative confidence bonus.

The confidence bounds and optimism in Lemma 5 are proved together. Suppose at stage $h+1$ that the optimistic and pessimistic values bracket $V_{h+1}^\star$. This ordering makes the estimated variance large enough to apply the Bernstein confidence bound at stage h. The resulting upper and lower Bellman candidates bracket $Q_h^\star$. The running minimum of upper candidates and the running maximum of lower candidates preserve this bracket. A non-switch episode simply reuses the most recent bracket. Backward induction from $V_{H+1}^\star = 0$ therefore proves the confidence bounds and optimism simultaneously.

Because the regression targets are data-dependent, concentration for one fixed target is not enough. We therefore use an ε-net: discretize the vector and inverse-matrix parameters that define the switch-dependent value functions, prove concentration on this deterministic finite set, and transfer the bounds to the actual targets. Rare switching limits each value function to at most $M_{\max} = \mathcal{O}(dH\log(KH+2))$ Bellman candidates, which keeps the covering number manageable.

Let $S := \sum_{k,h} \widetilde{\sigma}_{k,h}^2$ be the total estimated variance. To sum the bonuses, we normalize each feature by its variance scale $\bar{\sigma}_{k,h}$. The matrix determinant lemma bounds the sum of the resulting normalized widths, while the determinant-doubling rule compares the current matrix with the matrix stored at the most recent switch. Cauchy–Schwarz then converts the weighted width sum into $\sqrt{S}$. The result is

$$\sum_{k,h} \min\{\widehat{\beta} u_{k,h}, H\} \leq \widetilde{\mathcal{O}}\Big(d^7H^7 + \widehat{\beta} d^7 H^5 + \widehat{\beta}\sqrt{dH(S + H^2K)}\Big). \tag{29}$$

The first two terms are lower-order costs. The last term contains the leading dependence on K.

It remains to bound S. Define the four terms

$$I_2 := \sum_{k,h} E_{k,h}, \qquad I_3 := \sum_{k,h} D_{k,h},$$

$$I_4 := \sum_{k,h} \left| \mathbb{V}_h \widetilde{\widehat{V}}_{k,h+1} - \mathbb{V}_h V^{\widetilde{\pi}^k}_{h+1} \right|, \qquad I_5 := \sum_{k,h} \mathbb{V}_h V^{\widetilde{\pi}^k}_{h+1},$$

where $\mathbb{V}_h f(s,a)$ denotes the conditional variance of $f(s_{h+1})$ given $(s_h, a_h) = (s,a)$. The variance estimator then gives

$$S \le H^2 K + 2I_2 + I_3 + I_4 + I_5. \tag{30}$$

Thus I_2 is the moment-estimation error, I_3 is the optimistic–pessimistic width, I_4 compares the optimistic variance with the variance under the executed policy, and I_5 is the latter variance itself. The moment confidence bounds reduce I_2 to bonus sums. Applying the same Bellman recursion to $\widetilde{\widehat{V}}_{k,h} - \widetilde{\check{V}}_{k,h}$ reduces I_3 to bonuses and a martingale sum. Optimism and, for $X, Y \in [0,H]$, $|\operatorname{Var}(X) - \operatorname{Var}(Y)| \le 2H\mathbb{E}|X-Y|$ reduce I_4 to the regret recursion, while the law of total variance controls I_5. Hence every term is reduced to the same bonus sum, plus lower-order martingale terms. Substituting the bonus estimate into (30) gives an inequality of the form

$$S \le \widetilde{\mathcal{O}}\big(H^2K + d^{10}H^{10} + B_\beta d^{10}H^8\big) + \widetilde{\mathcal{O}}\Big(d^{7/2}H^{7/2}B_\beta\Big)\sqrt{S}, \tag{31}$$

where $B_\beta := \widehat{\beta} + \check{\beta} + H\widetilde{\beta}$. Using $x \le A + B\sqrt{x} \Rightarrow x \le 2A + 2B^2$ gives

$$S \le \widetilde{\mathcal{O}}\big(H^2K + d^{10}H^{10} + B_\beta d^{10}H^8 + d^7H^7B_\beta^2\big)\,. \tag{32}$$

Equations (28), (29), and (32) reduce the proof to substituting the two noise choices into the confidence radius:

$$\widehat{\beta} = \widetilde{\mathcal{O}}\left(\sqrt{d} + \frac{d^{3/2}H^{5/4}}{\sqrt{\rho}}\right) \quad \text{under choice (I)},$$

$$\widehat{\beta} = \widetilde{\mathcal{O}}\left(\sqrt{d} + \frac{d^{3/2}H^{3/2}}{\sqrt{\rho}}\right) \quad \text{under choice (II)}.$$

The leading bonus term then gives, respectively,

$$\widetilde{\mathcal{O}}\Big(dH^{3/2}\sqrt{K} + d^2H^{11/4}\sqrt{K/\rho}\Big) \quad \text{and} \quad \widetilde{\mathcal{O}}\Big(dH^{3/2}\sqrt{K} + d^2H^3\sqrt{K/\rho}\Big)\,.$$

The quadratic term B_β^2 contributes the common lower-order stability cost $\widetilde{\mathcal{O}}(d^9H^8/\rho)$, which completes the stated bounds.

Remark 4 (Comparison with the regret proof of Sahu (2025))**.** *Our regret analysis follows the same sequence of steps as this prior analysis. Both analyses establish confidence bounds for variance-weighted regressions, use an ε-net for the data-dependent value functions, prove optimism and pessimism, and then bound the cumulative bonuses and variance estimates.* ***The main difference is how the noisy statistics are released****. Their method adds noise to the sufficient statistics at every episode and stage and analyzes their privacy through zCDP. We release Gram prefix sums through a binary tree and release the value-dependent Bellman vectors only when a policy switch occurs, with the noise calibrated directly for TV stability. The tree ensures that one episode changes only $L_K = \mathcal{O}(\log K)$ Gram releases rather than all later Gram prefixes. This removes polynomial dependence on K from the Gram noise and is also essential for the logarithmic cost of an accepted unlearning request in our unlearning design. In the regret proof, the tree and the reuse of estimates between switches require an additional comparison between the current Gram matrix and the matrix stored at the most recent switch.*

E.2 Matrix comparison and regression confidence

For a bounded function f, write $\mathbb{V}_h f := P_h(f^2) - (P_h f)^2$. Let $\mathcal{F}_{k,h}$ contain all observations and learner randomness revealed before the transition from (s_h^k, a_h^k), including that state and action. Thus the variance scales and features are $\mathcal{F}_{k,h}$-measurable, and the conditional next-state law is $P_h(\cdot \mid s_h^k, a_h^k)$. The regret probabilities include both transition randomness and learner randomness in the fixed MDP.

Fix $\delta \in (0,1)$. In the statements below, we also require (10) and the bounds $\|\omega_{j,k,h}\|_2 \le \zeta_j$ for every released vector. We apply the concentration inequalities before restricting to these realizations. Define the three deterministic perturbation radii

$$\widehat{\gamma} := \frac{(c_\Lambda + \nu_\Lambda)H\sqrt{d} + \zeta_1}{\sqrt{b_\Lambda}}, \qquad \check{\gamma} := \frac{(c_\Lambda + \nu_\Lambda)H\sqrt{d} + \zeta_2}{\sqrt{b_\Lambda}}, \qquad \widetilde{\gamma} := \frac{(c_\Lambda + \nu_\Lambda)H^2\sqrt{d} + \zeta_3}{\sqrt{b_\Lambda}}. \tag{33}$$

The first term in each numerator bounds the regularization and Gram-noise errors. The second is the corresponding vector-noise error. Define

$$\begin{aligned} \varepsilon_{\mathrm{net}} &:= \min\left\{\frac{1}{16d^3H}, \frac{\sqrt{b_\Lambda}}{16KH}\right\}, \\ R_{\mathrm{par}} &:= 1 + b_\Lambda^{-1} + \frac{KH + \zeta_1 + \zeta_2 + \zeta_3}{b_\Lambda} + \widehat{\gamma} + \check{\gamma} + \widetilde{\gamma}, \\ \iota &:= \left(1 + \log\left[\frac{CdHK(M_{\max}+1)}{\delta}\left(1 + \frac{K}{Hb_\Lambda}\right)\left(1 + \frac{R_{\mathrm{par}}}{\varepsilon_{\mathrm{net}}}\right)\right]\right)^2. \end{aligned} \tag{34}$$

Here $C \ge 2$ is a sufficiently large universal constant. Under the noise bounds stated above, R_{par} bounds the regression-vector norms and inverse-matrix norms. Here $\varepsilon_{\mathrm{net}}$ is the uniform approximation error for value functions. Choose

$$\begin{aligned} \widehat{\beta} &= C_0\big(\sqrt{d}\,\iota + \widehat{\gamma}\big), \\ \check{\beta} &= C_0\big(d^{3/2}H\iota + \check{\gamma}\big), \\ \widetilde{\beta} &= C_0\big(d^{3/2}H\iota + \widehat{\gamma} + \check{\gamma} + \widetilde{\gamma}/H\big), \end{aligned} \tag{35}$$

where C_0 is a universal constant. The radius $\widetilde{\beta}$ accounts for a uniform cover of the data-dependent value functions and for both moment estimates. The policy uses the smaller $\widehat{\beta}$ obtained from the conditional-variance bound proved below. Increase C_0 if necessary so that $\widetilde{\beta} \ge \widehat{\beta}$.

The regressions are recomputed only at switch episodes, but regret is charged at every episode. Consequently, the policy bonus uses the matrix at the most recent switch, whereas the determinant potential used later is evaluated at the current episode. The next lemma compares these two matrices. The same determinant calculation bounds the number of switches, which in turn bounds the number of value functions that must enter the uniform concentration argument.

Lemma 3 (Noisy matrices and widths at the most recent switch)**.** *Whenever* (10) *holds, the following statements hold.*

1. *Every matrix $\widetilde{\Lambda}_{k,h}$ is positive definite.*
2. *If $i \le k$, then*
$$\widetilde{\Lambda}_{k,h} \succeq \alpha\widetilde{\Lambda}_{i,h}, \qquad \alpha := \frac{c_\Lambda - \nu_\Lambda}{c_\Lambda + \nu_\Lambda}.$$
3. *If $i = k_{\mathrm{last}}$ is the most recent switch at episode k, then*
$$\|x\|_{\widetilde{\Lambda}_{i,h}^{-1}} \le 3\|x\|_{\widetilde{\Lambda}_{k,h}^{-1}} \qquad \text{for every } x \in \mathbb{R}^d. \tag{36}$$
Moreover, $\|x\|_{\widetilde{\Lambda}_{k,h}^{-1}} \le 2\|x\|_{\widetilde{\Lambda}_{i,h}^{-1}}$.
4. *Recall that*
$$C_{\mathrm{sw}} \ge \frac{1}{\log(7/6)} + 2, \qquad M_{\max} = \lceil C_{\mathrm{sw}}dH\log(KH+2)\rceil,$$
Then the number M_{sw} of policy switches satisfies $M_{\mathrm{sw}} < M_{\max} = \mathcal{O}(dH\log(KH+2))$. Thus the condition $m < M_{\max}$ never prevents a switch.

Proof. The proof has three parts. We compare matrices at two episodes, use their determinants to count switches, and then compare the confidence widths between two switches.

The sandwich (10) and $b_\Lambda \geq H^{-2} > 0$ prove the first claim. For the second, since the clean Gram sums are nondecreasing,

$$\widetilde{\Lambda}_{k,h} \succeq \sum_{j<k} X_{j,h} + (c_\Lambda - \nu_\Lambda) I \succeq \sum_{j<i} X_{j,h} + (c_\Lambda - \nu_\Lambda) I \succeq \alpha \left(\sum_{j<i} X_{j,h} + (c_\Lambda + \nu_\Lambda) I \right) \succeq \alpha \widetilde{\Lambda}_{i,h}.$$

Since $\nu_\Lambda / c_\Lambda \leq 1/(4dH) \leq 1/4$,

$$\log(\alpha^{-1}) = \log \frac{1 + \nu_\Lambda / c_\Lambda}{1 - \nu_\Lambda / c_\Lambda} = 2 \int_0^{\nu_\Lambda / c_\Lambda} \frac{\mathrm{d}t}{1 - t^2} \leq \frac{2(\nu_\Lambda / c_\Lambda)}{1 - (\nu_\Lambda / c_\Lambda)^2} \leq \frac{32 \nu_\Lambda}{15 c_\Lambda}$$

will be used twice.

We first show that $m < M_{\max}$ always holds, so this condition never blocks a switch. As long as $m < M_{\max}$, let $\Pi_k := \prod_{h=1}^{H} \det(\widetilde{\Lambda}_{k,h})$. At a switch, the triggering stage contributes a factor at least two. Part 2 gives a factor at least α^d at every other stage. The preceding integral bound yields

$$\log\bigl(2\alpha^{d(H-1)}\bigr) \geq \log 2 - \frac{8}{15} > \log(7/6).$$

Thus Π_k grows by a factor larger than $7/6$ between two switch episodes. At the start, every eigenvalue is at least b_Λ. At every later episode, $\| \sum_{i<k} X_{i,h} \|_{\mathrm{op}} \leq \sum_{i<k} \bar{\sigma}_{i,h}^{-2} \leq K/H$, so every eigenvalue is at most $K/H + c_\Lambda + \nu_\Lambda$. The shift condition gives $c_\Lambda + \nu_\Lambda \leq b_\Lambda + 2\nu_\Lambda \leq 2 b_\Lambda$. Since $b_\Lambda \geq H^{-2}$,

$$\frac{\Pi_t}{\Pi_1} \leq \left(2 + \frac{K}{H b_\Lambda} \right)^{dH} \leq (KH + 2)^{dH}.$$

Here t is the last switch episode. If there are $m > 1$ switches, the left side is larger than $(7/6)^{m-1}$; for $m = 1$ it equals one. Hence

$$m \leq 1 + \frac{dH \log(KH + 2)}{\log(7/6)}.$$

At the last switch, $m = M_{\mathrm{sw}}$. Set $A := dH \log(KH + 2)$. Since $A \geq \log 3$, we have $2A > 1$. The choice of C_{sw} therefore gives

$$1 + \frac{A}{\log(7/6)} < C_{\mathrm{sw}} A \leq M_{\max}.$$

Thus $m < M_{\max}$ throughout this argument, which proves the fourth claim.

Let i be the most recent switch before episode k. Since there is no switch from $i + 1$ through k, the determinant ratio stays below two. Put $B = \widetilde{\Lambda}_{i,h}^{-1/2} \widetilde{\Lambda}_{k,h} \widetilde{\Lambda}_{i,h}^{-1/2}$. Its eigenvalues are at least α. For $k = i$ the claim is immediate. For $k > i$, the absence of a switch gives $\det(B) < 2$. Hence

$$\lambda_{\max}(B) \leq 2\alpha^{-(d-1)}.$$

The preceding logarithmic bound gives $(d-1)\log(\alpha^{-1}) \leq 8/(15H) < \log 2$. Thus $\alpha^{-(d-1)} < 2$. Therefore $\|x\|^2_{\widetilde{\Lambda}_{i,h}^{-1}} \leq 8 \|x\|^2_{\widetilde{\Lambda}_{k,h}^{-1}}$, which implies (36). The other direction follows from the matrix comparison above and $\alpha^{-1/2} < 2$. □

The matrix comparison resolves the effect of using a fixed policy between switches. It does not control the transition residual in a regression target. That residual is a martingale difference, but its value function depends on the preceding data and noisy releases. The next lemma gives a Hoeffding bound for a finite family and a Bernstein bound when the conditional variance is small. The regression proof will apply it to deterministic covers of the possible value functions.

Lemma 4 (Weighted transition concentration). *For each stage, let $\mathcal{V}$ be a deterministic finite family of functions $f : \mathcal{S} \to [0, H]$ with $\log |\mathcal{V}| \le Cd^3H\iota$, where C is a universal constant. Put*

$$\xi_i(f) := f(s^i_{h+1}) - P_h f(s^i_h, a^i_h), \qquad \psi_{i,h} := \phi^i_h/\bar{\sigma}_{i,h},$$
$$A_{i,h} := b_\Lambda I + \sum_{j<i} \psi_{j,h}\psi_{j,h}^\top, \qquad v^{\mathrm{cl}}_{i,h} := \|\phi^i_h\|_{A^{-1}_{i,h}},$$
$$v_{i,h} := \|\phi^i_h\|_{\widetilde{\Lambda}^{-1}_{i,h}}.$$

For $s > 0$, define the predictable indicator

$$J_i(f; s) := \mathbf{1}\{v^{\mathrm{cl}}_{i,h} \le 3u_{i,h}\}\mathbf{1}\{\mathbb{V}_h f(s^i_h, a^i_h) \le s^2\bar{\sigma}^2_{i,h}\}.$$

The indicator is determined before s^i_{h+1} is observed. Each of the following three bounds holds with probability at least $1 - \delta$, simultaneously over stages, prefixes, and functions in its respective family. For $s = (d^3H)^{-1/2}$,

$$\left\| \sum_{i\le n} \bar{\sigma}^{-2}_{i,h}\phi^i_h J_i(f; s)\xi_i(f) \right\|_{A^{-1}_{n+1,h}} \le C\sqrt{d}\,\iota.$$

For one fixed function per stage the same bound holds with $s = 2$. Without indicators or variance assumptions, the full family satisfies the Hoeffding bound

$$\left\| \sum_{i\le n} \bar{\sigma}^{-2}_{i,h}\phi^i_h \xi_i(f) \right\|_{A^{-1}_{n+1,h}} \le Cd^{3/2}H\iota.$$

Each of these three applications may be assigned its own failure budget δ.

Proof. We first prove a Hoeffding bound that does not use a variance condition. We then prove the two Bernstein bounds, which are smaller when the conditional variance is small.

For a fixed f, the residual has conditional mean zero given $\mathcal{F}_{i,h}$. The normalized residual $\xi_i(f)/\bar{\sigma}_{i,h}$ has absolute value at most $\sqrt{H}$. Conditional Hoeffding therefore gives $\mathbb{E}[\exp(t\xi_i(f)/\bar{\sigma}_{i,h}) \mid \mathcal{F}_{i,h}] \le \exp(t^2H/2)$ for every real t. Thus the sub-Gaussian parameter is $\sqrt{H}$. The self-normalized Hoeffding inequality (Lemma G.5 of He et al. (2023)) gives

$$\left\| \sum_{i\le n} \psi_{i,h}\xi_i(f)/\bar{\sigma}_{i,h} \right\|_{A^{-1}_{n+1,h}} \le \sqrt{2H \log\left(\frac{\det(A_{n+1,h})^{1/2}}{\det(b_\Lambda I)^{1/2}\delta'} \right)}.$$

Here $\|\psi_{i,h}\|_2 \le H^{-1/2}$, and $\det(A_{n+1,h})/\det(b_\Lambda I) \le (1 + K/(dHb_\Lambda))^d$. Taking $\delta' = \delta/(HK|\mathcal{V}|)$ proves the displayed Hoeffding bound by a union bound.

For the Bernstein bounds, put $\bar{\xi}_i = J_i(f; s)\xi_i(f)/\bar{\sigma}_{i,h}$. These residuals remain centered, have conditional variance at most s^2, and satisfy $|\bar{\xi}_i| \le \sqrt{H}$. The definition of $\bar{\sigma}_{i,h}$ gives, on every realization,

$$|\bar{\xi}_i| \min\{1, \|\psi_{i,h}\|_{A^{-1}_{i,h}}\} \le \frac{Hv^{\mathrm{cl}}_{i,h}J_i(f; s)}{\bar{\sigma}^2_{i,h}} \le r_0 := \frac{3}{4d^6H^3}.$$

Theorem 4.3 in the arXiv v1 version of Zhou & Gu (2022), also used in Lemma 4.1 of He et al. (2023), with its auxiliary accuracy set to r_0, yields

$$\left\| \sum_{i\le n} \psi_{i,h}\bar{\xi}_i \right\|_{A^{-1}_{n+1,h}} \le 12s\sqrt{d\ell L} + 30r_0L,$$

$$\ell := \log\left(1 + \frac{K}{dHb_\Lambda}\right), \qquad L := \log\frac{32(1 + \log(\sqrt{H}/r_0))K^2}{\delta'}.$$

Its assumptions hold before we restrict to outcomes that satisfy the noise bounds or the regression confidence inequalities below. For the finite family use $\delta' = \delta/(HK|\mathcal{V}|)$, so $\ell \le \sqrt{\iota}$ and

$L \le Cd^3H\iota$. With $s = (d^3H)^{-1/2}$, the two terms are bounded by $C\sqrt{d}\,\iota^{3/4}$ and $C\iota/(d^3H^2)$, respectively. For one fixed function use $\delta' = \delta/(HK)$ and $s = 2$; now $L \le C\sqrt{\iota}$, giving the claimed $C\sqrt{d}\,\iota$ bound as well. The indicators are only used in this proof. Under (10), the matrix bounds give $\widetilde{\Lambda}_{i,h} \preceq 2A_{i,h}$, and hence $v^{\mathrm{cl}}_{i,h} \le \sqrt{2}v_{i,h} \le 3u_{i,h}$. Whenever the variance inequality also holds, $J_i(f;s) = 1$ and the displayed sum is the actual regression residual. □

The preceding lemma controls the stochastic part of a regression formed with the clean regularized matrix $A_{i,h}$. The learner instead uses a noisy Gram matrix and noisy vector statistics at the most recent switch. The next lemma starts from an exact regression identity, adds the deterministic perturbation radii in (33), and uses the matrix comparison to transfer the resulting confidence bound to every episode between two switches.

Lemma 5 (Hoeffding and Bernstein regression confidence bounds)**.** *Except on a set of probability at most 3δ from the three applications of Lemma 4, the following holds whenever (10) and the bounds $\|\omega_{j,k,h}\|_2 \le \zeta_j$ hold. Simultaneously at every episode k, stage h, and state–action pair (s,a), let k_{last} be the most recent switch episode. Then*

$$
\begin{aligned}
|\langle\phi, \widetilde{\widehat{w}}_{k,h}\rangle - P_h\widetilde{\widehat{V}}_{k,h+1}(s,a)| &\le \widehat{\beta}\|\phi\|_{\widetilde{\Lambda}^{-1}_{k_{\mathrm{last}},h}}, \\
|\langle\phi, \widetilde{\widetilde{w}}_{k,h}\rangle - P_h\widetilde{\widetilde{V}}_{k,h+1}(s,a)| &\le \widetilde{\beta}\|\phi\|_{\widetilde{\Lambda}^{-1}_{k_{\mathrm{last}},h}}, \\
|\langle\phi, \widetilde{\check{w}}_{k,h}\rangle - P_h\widetilde{\check{V}}_{k,h+1}(s,a)| &\le \check{\beta}\|\phi\|_{\widetilde{\Lambda}^{-1}_{k_{\mathrm{last}},h}}, \\
|\langle\phi, \widetilde{\overline{w}}_{k,h}\rangle - P_h(\widetilde{\widehat{V}}_{k,h+1})^2(s,a)| &\le H\widetilde{\beta}\|\phi\|_{\widetilde{\Lambda}^{-1}_{k_{\mathrm{last}},h}}.
\end{aligned}
\tag{37}
$$

Moreover, $\widetilde{\widehat{V}}_{k,h} \ge V^\star_h \ge \widetilde{\check{V}}_{k,h}$ for every k,h.

Proof. The proof has four steps. First, we separate the added Gaussian noise from the transition noise. Second, we obtain uniform bounds for the value functions used by the algorithm. Third, we show that the variance scales are large enough. Last, we use backward induction to prove the first confidence bound, optimism, and pessimism.

First, fix one update, one stage, and one target V. Put

$$\mu_h(V) := \int V(s')\mu_h(\mathrm{d}s').$$

Then $P_hV = \langle\phi, \mu_h(V)\rangle$ and $\|\mu_h(V)\|_2 \le H\sqrt{d}$ by Assumption 1. The noisy regression satisfies the exact identity

$$
\begin{aligned}
\widetilde{\Lambda}_{k,h}(\widetilde{\widehat{w}}_{k,h} - \mu_h(V)) = \sum_{i<k} \bar{\sigma}^{-2}_{i,h}\phi^i_h\left[V(s^i_{h+1}) - P_hV(s^i_h,a^i_h)\right] + \omega_{1,k,h} \\
- \left(\widetilde{\Lambda}_{k,h} - \sum_{i<k}\bar{\sigma}^{-2}_{i,h}\phi^i_h(\phi^i_h)^\top\right)\mu_h(V).
\end{aligned}
\tag{38}
$$

Because $\widetilde{\Lambda}_{k,h} \succeq b_\Lambda I$,

$$
\|\omega_{1,k,h}\|_{\widetilde{\Lambda}^{-1}_{k,h}} \le \zeta_1/\sqrt{b_\Lambda}, \qquad \left\|\left(\widetilde{\Lambda}_{k,h} - \sum_{i<k}\bar{\sigma}^{-2}_{i,h}\phi^i_h(\phi^i_h)^\top\right)\mu_h(V)\right\|_{\widetilde{\Lambda}^{-1}_{k,h}} \le \frac{(c_\Lambda + \nu_\Lambda)H\sqrt{d}}{\sqrt{b_\Lambda}}.
$$

These two terms give $\widehat{\gamma}$. The same calculation for the pessimistic statistic gives $\check{\gamma}$. For the squared target, replace the parameter norm by $H^2\sqrt{d}$ and use ζ_3. This gives $\widetilde{\gamma}$.

We now bound the transition-noise term in (38). A bound for one fixed function is not enough because the algorithm computes each value function from earlier data and noisy releases. We therefore cover all value functions that the algorithm can produce. At one switch, every candidate Q-function has the form

$$\left[r_h(s,a) + \langle\phi(s,a), w\rangle + \beta\sqrt{\phi(s,a)^\top B\phi(s,a)}\right]_{[0,H]},$$

with $w \in \mathbb{R}^d$ and a symmetric $d \times d$ matrix B. The current optimistic value is the maximum over actions of the pointwise minimum of at most $M_{\max}$ such candidates. The pessimistic value uses the corresponding pointwise maximum. Thus a value function is fixed by at most $M_{\max}$ vectors and matrices. Under the stated noise bounds, their norms are at most R_{par} from (34). This follows from $\widetilde{\Lambda} \succeq b_\Lambda I$, targets in $[0, H]$, and the bounds on the vector noise.

Put $\beta_{\max} := \max\{\widehat{\beta}, \check{\beta}, \widetilde{\beta}\}$. Use a Euclidean mesh $\varepsilon_{\mathrm{net}}/2$ for the vector parameters. Use Frobenius mesh $(\varepsilon_{\mathrm{net}}/(2\beta_{\max}))^2$ for the positive semidefinite inverse matrices. Choose all centers inside the parameter sets. Since $|\sqrt{x} - \sqrt{y}| \le \sqrt{|x-y|}$, one candidate changes by at most $\varepsilon_{\mathrm{net}}$. Clipping, minima, maxima, and maximization over actions do not increase this error. Use $M_{\max}$ candidates, and add constant candidates if needed. This makes the cover deterministic. The matrix parameters lie in a Frobenius ball of radius $\sqrt{d}R_{\mathrm{par}}$. The volume bound for a ball in dimension $d + d(d+1)/2$ gives

$$\log N_{\varepsilon_{\mathrm{net}}} \le Cd^2 M_{\max} \log\left(2 + \frac{\sqrt{d}R_{\mathrm{par}}(1+\beta_{\max})^2}{\varepsilon_{\mathrm{net}}^2}\right) \le Cd^3 H\iota.$$

The last inequality uses $M_{\max} = \mathcal{O}(dH\log(KH+2))$ and $\beta_{\max} \le Cd^{3/2}HR_{\mathrm{par}}\iota$; in particular $\log(1+\beta_{\max}) \le C\sqrt{\iota}$. If two functions have sup-norm distance a, the difference of their residual vectors has norm at most $2Ka/(H\sqrt{b_\Lambda})$. Our choice of $\varepsilon_{\mathrm{net}}$ makes this at most one, including the normalized squared target V^2/H, whose approximation error is at most $2\varepsilon_{\mathrm{net}}$.

Apply the Hoeffding part of Lemma 4 to the deterministic covers of the optimistic, pessimistic, and normalized squared targets, allocating a total budget δ to these three families. Transfer from the covers by the preceding error bound. This gives radius $Cd^{3/2}H\iota$ for both first moments and $Cd^{3/2}H^2\iota$ for the second moment. Adding the Gaussian-noise terms proves the last three confidence inequalities. This step does not use optimism.

The regret bound needs the first inequality with the smaller radius $\widehat{\beta}$. To obtain it, we use the conditional-variance part of Lemma 4. The algorithm uses an estimated variance to define $\bar{\sigma}_{i,h}$. We must first show that this estimate is large enough. We prove this fact together with optimism by backward induction. Assume at stage $h+1$ that $\widetilde{\widehat{V}}_{k,h+1} \ge V^\star_{h+1} \ge \widetilde{\check{V}}_{k,h+1}$. Let

$$m_1 := P_h\widetilde{\widehat{V}}_{i,h+1}, \qquad m_2 := P_h(\widetilde{\widehat{V}}_{i,h+1})^2,$$

and let $\widehat{m}_1$ and $\widehat{m}_2$ be the clipped regression estimates used in $\overline{\mathbb{V}}_{i,h}$. Clipping cannot increase an error. Thus the two moment confidence bounds give

$$|\widehat{m}_1 - m_1| \le \min\{\widetilde{\beta}u_{i,h}, H\}, \qquad |\widehat{m}_2 - m_2| \le \min\{H\widetilde{\beta}u_{i,h}, H^2\}.$$

Since $\overline{\mathbb{V}}_{i,h} = [\widehat{m}_2 - \widehat{m}_1^2]_{[0,H^2]}$, $m_2 - m_1^2 \in [0, H^2]$, and $|x^2 - y^2| \le 2H|x-y|$ on $[0,H]$,

$$\begin{aligned}|\overline{\mathbb{V}}_{i,h} - (m_2 - m_1^2)| &\le |\widehat{m}_2 - m_2| + |\widehat{m}_1^2 - m_1^2| \\ &\le \min\{H\widetilde{\beta}u_{i,h}, H^2\} + \min\{2H\widetilde{\beta}u_{i,h}, H^2\} = E_{i,h}.\end{aligned}$$

Hence

$$\left|\overline{\mathbb{V}}_{i,h} - \mathbb{V}_h\widetilde{\widehat{V}}_{i,h+1}(s_h^i, a_h^i)\right| \le E_{i,h}. \tag{39}$$

We use two value gaps. For a sample from episode $i < k$, define

$$\Delta_i^\star := \widetilde{\widehat{V}}_{i,h+1} - V^\star_{h+1}, \qquad \Delta_{k|i} := \widetilde{\widehat{V}}_{k,h+1} - V^\star_{h+1}.$$

The first gap lets us compare the variance of $V^\star_{h+1}$ with the estimated variance. The second gap is the part of the regression target at episode k that depends on the data. The running minimum makes $\widetilde{\widehat{V}}_{k,h+1}$ nonincreasing in k, while the running maximum makes $\widetilde{\check{V}}_{k,h+1}$ nondecreasing. Optimism and pessimism therefore imply, for either $W \in \{\Delta_i^\star, \Delta_{k|i}\}$,

$$0 \le W \le \widetilde{\widehat{V}}_{i,h+1} - V^\star_{h+1} \le \widetilde{\widehat{V}}_{i,h+1} - \widetilde{\check{V}}_{i,h+1} \le H.$$

It follows that

$$\mathbb{V}_h(W) \le P_h[W^2] \le HP_h(\widetilde{\widehat{V}}_{i,h+1} - \widetilde{\check{V}}_{i,h+1}).$$

The second and third confidence inequalities at stage h give

$$P_h(\widetilde{\widehat{V}}_{i,h+1} - \widetilde{\check{V}}_{i,h+1}) \le \langle \phi_h^i, \widetilde{\widehat{w}}_{i,h} - \widetilde{\check{w}}_{i,h} \rangle + (\widetilde{\beta} + \check{\beta}) u_{i,h}.$$

The left side lies in $[0, H]$. We now check the two terms in the minimum that defines $D_{i,h}$. The first term is at least $d^3 H\, \mathbb{V}_h(W)$ because

$$\mathbb{V}_h(W) \le H P_h(\widetilde{\widehat{V}}_{i,h+1} - \widetilde{\check{V}}_{i,h+1}).$$

The second term has the same lower bound because $\mathbb{V}_h(W) \le H^2$. Therefore

$$\max\{\mathbb{V}_h(\Delta_i^\star)(s_h^i, a_h^i), \mathbb{V}_h(\Delta_{k|i})(s_h^i, a_h^i)\} \le D_{i,h}/(d^3 H). \tag{40}$$

Equation (39) also gives $\mathbb{V}_h \widetilde{\widehat{V}}_{i,h+1} \le \overline{\mathbb{V}}_{i,h} + E_{i,h}$. Because $\widetilde{\sigma}_{i,h}^2 = \overline{\mathbb{V}}_{i,h} + E_{i,h} + D_{i,h} + H$, we obtain

$$\mathbb{V}_h \widetilde{\widehat{V}}_{i,h+1} \le \widetilde{\sigma}_{i,h}^2, \qquad \mathbb{V}_h(\Delta_i^\star) \le \frac{\widetilde{\sigma}_{i,h}^2}{d^3 H}.$$

Also, $V_{h+1}^\star = \widetilde{\widehat{V}}_{i,h+1} - \Delta_i^\star$ and $\mathbb{V}(X - Y) \le 2\mathbb{V}(X) + 2\mathbb{V}(Y)$ give

$$\mathbb{V}_h V_{h+1}^\star \le 2\widetilde{\sigma}_{i,h}^2 + \frac{2\widetilde{\sigma}_{i,h}^2}{d^3 H} \le 4\bar{\sigma}_{i,h}^2.$$

We can now prove the smaller confidence bound. Split the optimistic transition-noise vector into a fixed part and a data-dependent gap:

$$\sum_i \bar{\sigma}_{i,h}^{-2} \phi_h^i \xi_i(V_{h+1}^\star) + \sum_i \bar{\sigma}_{i,h}^{-2} \phi_h^i \xi_i(\Delta_{k|i}).$$

For the fixed function $V_{h+1}^\star$, use the lemma with $s = 2$. The variance bound above makes its indicator equal to one under the induction hypothesis. Use one failure budget δ for all stages. For the second term, form a deterministic cover of $[V - V_{h+1}^\star]_{[0,H]}$ from the optimistic cover. Subtracting a fixed function and clipping keep the same cover size and error. For the representative f of the actual gap $\Delta_{k|i}$, both functions lie in $[0, H]$, and

$$|\mathbb{V}_h f - \mathbb{V}_h \Delta_{k|i}| \le 4H\varepsilon_{\text{net}} \le \frac{1}{4d^3}.$$

The slack $+H$ in $\widetilde{\sigma}_{i,h}^2$ gives

$$\mathbb{V}_h f \le \frac{D_{i,h}}{d^3 H} + \frac{1}{4d^3} \le \frac{\bar{\sigma}_{i,h}^2}{d^3 H}.$$

Thus every indicator for this representative equals one under the induction hypothesis. Apply the finite-family bound with another budget δ, then transfer it to the actual gap. The transfer error is at most one because

$$\frac{2K\varepsilon_{\text{net}}}{H\sqrt{b_\Lambda}} \le 1.$$

The two residual bounds are $C\sqrt{d}\,\iota$. This proves the first line of (37) with radius $C(\sqrt{d}\,\iota + \widehat{\gamma})$. The backward induction starts from $V_{H+1} = 0$ and goes from $h = H$ to 1. We used the regression argument only at switch episodes. Between two switches, the algorithm reuses the three regression vectors and both value functions from the most recent switch. Thus the same four inequalities hold with the fixed matrix $\widetilde{\Lambda}_{k_{\text{last}},h}$.

It remains to turn the four confidence inequalities into upper and lower bounds on the value. Use the Bellman equations and the same backward induction. Assume the value bounds hold at stage $h+1$. At a switch episode, the first confidence inequality gives

$$r_h + \langle \phi, \widetilde{\widehat{w}}_{k,h} \rangle + \widehat{\beta}\|\phi\|_{\widetilde{\Lambda}^{-1}} \ge r_h + P_h \widetilde{\widehat{V}}_{k,h+1} \ge r_h + P_h V_{h+1}^\star = Q_h^\star.$$

The running minimum and clipping at H keep this inequality true. For the lower estimate,

$$r_h + \langle \phi, \widetilde{\check{w}}_{k,h} \rangle - \check{\beta}\|\phi\|_{\widetilde{\Lambda}^{-1}} \le r_h + P_h \widetilde{\check{V}}_{k,h+1} \le r_h + P_h V_{h+1}^\star = Q_h^\star.$$

The running maximum and clipping at zero keep this inequality true. In a non-switch episode, the algorithm reuses both bounds from the most recent switch. Backward induction proves optimism and pessimism. □

E.3 Bonus and variance sums

Lemma 5 bounds each regression error by a confidence radius times $u_{k,h}$. The next lemma bounds the sum of these terms by a function of $S = \sum_{k,h} \widetilde{\sigma}^2_{k,h}$. Lemma 7 then substitutes the moment confidence bounds into the definition of $\widetilde{\sigma}^2_{k,h}$ and solves the resulting inequality $S \le A + B\sqrt{S}$.

Lemma 6 (Sum of bonuses)**.** *Put*

$$S := \sum_{k,h} \widetilde{\sigma}^2_{k,h}, \qquad B_\sigma := 16d^6H^5, \qquad \Gamma := 6d^3H^2,$$

and, for $\beta, \tau \ge 1$, define

$$\mathcal{B}(\beta, \tau; S) := C_1 \left[\tau H d B_\sigma \iota + \beta \Gamma^2 dH\iota + \beta\sqrt{dH\iota(S + H^2K)}\right]. \tag{41}$$

Here C_1 is a universal constant. Whenever (10) holds, for every $\beta, \tau \ge 1$,

$$\sum_{k=1}^{K}\sum_{h=1}^{H} \min\{\beta u_{k,h}, \tau\} \le \mathcal{B}(\beta, \tau; S).$$

Proof. Put $A_{k,h} = b_\Lambda I + \sum_{i<k} X_{i,h}$, $v^{\mathrm{cl}}_{k,h} = \|\phi^k_h\|_{A^{-1}_{k,h}}$, $z_{k,h} = v^{\mathrm{cl}}_{k,h}/\bar{\sigma}_{k,h}$, and $\sigma^+_{k,h} = \widetilde{\sigma}_{k,h} + \sqrt{H}$. The proof has three steps. The determinant identity bounds the sum of $z^2_{k,h}$. The matrix and variance-scale definitions then bound each $u_{k,h}$. Finally, Cauchy–Schwarz turns the sum of the $\sigma^+_{k,h}$ terms into the total variance S.

The determinant identity and $\|\phi^k_h\|_2 \le 1$ give

$$\sum_{k,h} \min\{z^2_{k,h}, 1\} \le 2\sum_h \log \frac{\det(A_{K+1,h})}{\det(A_{1,h})} \le 2dH\iota. \tag{42}$$

Indeed, $\det(A_{k+1,h}) = \det(A_{k,h})(1 + z^2_{k,h})$, $\min\{x, 1\} \le 2\log(1+x)$ for $x \ge 0$, and $\mathrm{tr}(A_{K+1,h}) \le db_\Lambda + K/H$. The matrix comparison gives $u_{k,h} \le 3v^{\mathrm{cl}}_{k,h}$. The definition of $\bar{\sigma}_{k,h}$ gives $\bar{\sigma}_{k,h} \le \sigma^+_{k,h} + \Gamma\sqrt{v^{\mathrm{cl}}_{k,h}}$. Drop the indices for this calculation. The inequality $2ab \le a^2 + b^2$ gives

$$v^{\mathrm{cl}} = z\bar{\sigma} \le z\sigma^+ + \Gamma z\sqrt{v^{\mathrm{cl}}} \le z\sigma^+ + \tfrac{1}{2}v^{\mathrm{cl}} + \tfrac{1}{2}\Gamma^2 z^2, \qquad v^{\mathrm{cl}} \le 2z\sigma^+ + \Gamma^2 z^2.$$

If $z \le 1$, it follows that $\min\{\beta u, \tau\} \le 6\beta z\sigma^+ + 3\beta\Gamma^2 z^2$. If $z > 1$, use $\min\{\beta u, \tau\} \le \tau$. There are at most $2dH\iota$ such indices by (42). Finally, $\sum_{k,h}(\sigma^+_{k,h})^2 \le 2(S + H^2K)$. Cauchy–Schwarz gives

$$\sum_{k,h} \min\{\beta u_{k,h}, \tau\} \le 2\tau dH\iota + 6\beta\Gamma^2 dH\iota + 12\beta\sqrt{dH\iota(S + H^2K)}. \tag{43}$$

Since $B_\sigma \ge 1$, this proves the stated bound with $C_1 \ge 12$. □

Lemma 6 expresses the bonus sum in terms of $S = \sum_{k,h} \widetilde{\sigma}^2_{k,h}$, but the definition of each $\widetilde{\sigma}^2_{k,h}$ itself contains confidence corrections. The next lemma closes this dependence. It separates S into explicit corrections, the optimistic–pessimistic gap, the variance gap, and the variance of the executed policies. The first three parts are controlled by confidence, bonus sums, and transition martingales. The last part is controlled by the conditional variance of the return.

Lemma 7 (Bonus and variance sums)**.** *For sufficiently large universal constants C_4 and C_5, let $B_\beta := \widehat{\beta} + \check{\beta} + H\widetilde{\beta}$ and define*

$$S_0 := C_4\iota^2\left(H^2K + d^{10}H^{10} + B_\beta d^{10}H^8 + d^7H^7B^2_\beta\right). \tag{44}$$

Whenever the conclusions of Lemma 5 hold, except on three additional sets of total probability at most 3δ,

$$\sum_{k,h} \widetilde{\sigma}^2_{k,h} \le S_0, \tag{45}$$

$$\sum_{k,h} \min\{\widehat{\beta}u_{k,h}, H\} \le C_5\iota\left[d^7H^7 + \widehat{\beta}d^7H^5 + \widehat{\beta}\sqrt{dH(S_0 + H^2K)}\right]. \tag{46}$$

Proof. We split S into four parts and bound each part. After substitution, the bound has the form $S \le A + B\sqrt{S}$, which can be solved directly.

Let $S = \sum_{k,h} \widetilde{\sigma}^2_{k,h}$. The variance-estimation bound (39) gives

$$S \le H^2K + 2I_2 + I_3 + I_4 + I_5,$$
$$I_2 := \sum_{k,h} E_{k,h}, \qquad I_3 := \sum_{k,h} D_{k,h},$$
$$I_4 := \sum_{k,h} |\mathbb{V}_h \widetilde{\widetilde{V}}_{k,h+1} - \mathbb{V}_h V^{\widetilde{\pi}^k}_{h+1}|, \qquad I_5 := \sum_{k,h} \mathbb{V}_h V^{\widetilde{\pi}^k}_{h+1}.$$

The first three parts reduce to confidence bonuses and transition martingales. The last part only depends on the policies that were used and is bounded by the return of each episode. Lemma 6 bounds all confidence bonuses.

We first bound I_2. Definition (7) and Lemma 6 give

$$I_2 \le \mathcal{B}(H\widetilde{\beta}, H^2; S) + \mathcal{B}(2H\widetilde{\beta}, H^2; S).$$

Use (41) twice and include the factor two in the constant. This gives

$$I_2 \le C\left[H^3 d B_\sigma \iota + H\widetilde{\beta}\Gamma^2 dH\iota + H\widetilde{\beta}\sqrt{dH\iota(S + H^2K)}\right]. \tag{47}$$

We next bound I_3. The two confidence inequalities imply

$$D_{k,h} \le 4d^3H^2 P_h(\widetilde{\widetilde{V}}_{k,h+1} - \widetilde{\check{V}}_{k,h+1})(s^k_h, a^k_h) + 16d^3H^2 \min\{(\widetilde{\beta} + \check{\beta})u_{k,h}, H\}.$$

To sum this gap over time, put $g_{k,h} := [\widetilde{\widetilde{V}}_{k,h} - \widetilde{\check{V}}_{k,h}]_{[0,H]}$. When the bounds in Lemma 5 hold, clipping does not change this gap. On every outcome, clipping keeps the gap in $[0, H]$. Thus the martingale bound below does not need to condition on optimism and pessimism. At the action chosen by the optimistic policy, the second and third confidence inequalities in (37) give

$$g_{k,h}(s^k_h) \le P_h g_{k,h+1}(s^k_h, a^k_h) + 2\min\{\widetilde{\beta}u_{k,h}, H\} + 2\min\{\check{\beta}u_{k,h}, H\}.$$

Define

$$b^g_{k,h} := 2\min\{\widetilde{\beta}u_{k,h}, H\} + 2\min\{\check{\beta}u_{k,h}, H\}, \qquad \xi^g_{k,h} := P_h g_{k,h+1}(s^k_h, a^k_h) - g_{k,h+1}(s^k_{h+1}).$$

Add and subtract the value at the observed next state. Then $g_{k,h}(s^k_h) \le g_{k,h+1}(s^k_{h+1}) + b^g_{k,h} + \xi^g_{k,h}$. Since $g_{k,H+1} = 0$, applying this inequality from stage j to H gives

$$g_{k,j}(s^k_j) \le \sum_{t=j}^{H} (b^g_{k,t} + \xi^g_{k,t}).$$

Also $P_h g_{k,h+1} = g_{k,h+1}(s^k_{h+1}) + \xi^g_{k,h}$. Therefore

$$\sum_{h=1}^{H} P_h g_{k,h+1} \le \sum_{h=1}^{H}\sum_{t=h+1}^{H} (b^g_{k,t} + \xi^g_{k,t}) + \sum_{h=1}^{H} \xi^g_{k,h}$$
$$\le H\sum_{h=1}^{H} b^g_{k,h} + \sum_{h=1}^{H} h\xi^g_{k,h}.$$

Thus, after summing over episodes,

$$\sum_{k,h} P_h g_{k,h+1}(s^k_h, a^k_h) \le 2H \sum_{k,h}\Big[\min\{\widetilde{\beta}u_{k,h}, H\} + \min\{\check{\beta}u_{k,h}, H\}\Big] + \sum_{k,h} \bar{\xi}^g_{k,h},$$

where $\bar{\xi}^g_{k,h} := h\xi^g_{k,h}$ has conditional mean zero and absolute value at most H^2. Conditional Hoeffding gives $\mathbb{E}[\exp(\lambda\bar{\xi}^g_{k,h}) \mid \mathcal{F}_{k,h}] \le \exp(\lambda^2H^4/2)$. Apply this bound over all k, h. Markov's

inequality with $\lambda = \sqrt{2\iota/(H^4KH)}$ then gives $\sum_{k,h} \bar{\xi}^g_{k,h} \le H^2\sqrt{2HK\iota}$. Put this result into the bound on $D_{k,h}$, and apply Lemma 6. We obtain

$$I_3 \le C_6 d^3 H^3 \left[\mathcal{B}(\widetilde{\beta} + \check{\beta}, H; S) + H\sqrt{HK\iota}\right]. \tag{48}$$

For I_4, optimism gives $0 \le V^{\widetilde{\pi}^k}_{h+1} \le \widetilde{\widehat{V}}_{k,h+1} \le H$. Therefore

$$|\mathbb{V}_h \widetilde{\widehat{V}}_{k,h+1} - \mathbb{V}_h V^{\widetilde{\pi}^k}_{h+1}| \le 2H\, P_h(\widetilde{\widehat{V}}_{k,h+1} - V^{\widetilde{\pi}^k}_{h+1}).$$

To see this, let $0 \le Y \le X \le H$. Both $\mathbb{E}(X^2 - Y^2)$ and $(\mathbb{E}X)^2 - (\mathbb{E}Y)^2$ belong to $[0, 2H\mathbb{E}(X-Y)]$. Their difference has absolute value at most $2H\mathbb{E}(X-Y)$. Put $e_{k,h} := [\widetilde{\widehat{V}}_{k,h} - V^{\widetilde{\pi}^k}_h]_{[0,H]}$. Again, clipping changes nothing when the optimism conclusion of Lemma 5 holds. The first confidence inequality in (37) gives

$$e_{k,h}(s^k_h) \le P_h e_{k,h+1}(s^k_h, a^k_h) + 2\min\{\widehat{\beta} u_{k,h}, H\}.$$

Put $b^e_{k,h} := 2\min\{\widehat{\beta} u_{k,h}, H\}$ and $\xi^e_{k,h} := P_h e_{k,h+1}(s^k_h, a^k_h) - e_{k,h+1}(s^k_{h+1})$. The same sum over stages gives

$$\sum_{k,h} P_h e_{k,h+1} \le H \sum_{k,h} b^e_{k,h} + \sum_{k,h} h\xi^e_{k,h}.$$

Here $|h\xi^e_{k,h}| \le H^2$. The same exponential-martingale calculation gives

$$\sum_{k,h} h\xi^e_{k,h} \le H^2\sqrt{2HK\iota}.$$

Multiplying by the factor $2H$ in the variance-gap inequality and applying Lemma 6 gives

$$I_4 \le C_7 H^2 \left[\mathcal{B}(\widehat{\beta}, H; S) + H\sqrt{HK\iota}\right]. \tag{49}$$

To bound I_5, consider one episode and let $R_k = \sum_{h=1}^H r_h(s^k_h, a^k_h)$. Reveal the trajectory one stage at a time. The conditional-variance identity $\mathbb{V}(X) = \mathbb{E}[\mathbb{V}(X \mid Y)] + \mathbb{V}(\mathbb{E}[X \mid Y])$, applied successively to the remaining return, gives

$$\mathbb{E}\left[\sum_{h=1}^H \mathbb{V}_h V^{\widetilde{\pi}^k}_{h+1}(s^k_h, a^k_h) \,\middle|\, \mathcal{F}_{k,1}\right] \le \mathbb{V}(R_k \mid \mathcal{F}_{k,1}) \le H^2.$$

Write the random sum in episode k as Y_k. It satisfies $0 \le Y_k \le H^3$ and $\mathbb{E}[Y_k \mid \text{past}] \le H^2$. For $0 \le x \le H^3$, convexity gives $e^{x/H^3} \le 1 + (e-1)x/H^3$. Therefore

$$\mathbb{E}[e^{Y_k/H^3} \mid \mathcal{F}_{k,1}] \le 1 + \frac{e-1}{H^3}\mathbb{E}[Y_k \mid \mathcal{F}_{k,1}] \le \exp((e-1)/H).$$

Apply this conditional bound over all k, and then use Markov's inequality. Except on an event of probability $e^{-\iota}$, we have $I_5/H^3 \le (e-1)K/H + \iota$. A larger constant gives

$$I_5 \le 3H^2K + 3H^3\iota. \tag{50}$$

We now combine the four bounds and solve for S. Recall $B_\sigma = 16d^6H^5$, $\Gamma = 6d^3H^2$, and $B_\beta = \widehat{\beta} + \check{\beta} + H\widetilde{\beta}$. Equation (47) is at most

$$C\left[\iota(d^7H^8 + B_\beta d^7 H^5) + B_\beta\sqrt{dH\iota(S + H^2K)}\right].$$

Equations (48) and (41) give

$$I_3 \le C\left[\iota(d^{10}H^{10} + B_\beta d^{10}H^8) + B_\beta d^3H^3\sqrt{dH\iota(S+H^2K)} + d^3H^4\sqrt{HK\iota}\right].$$

Likewise, (49) gives

$$I_4 \le C\left[\iota(d^7H^9 + \widehat{\beta} d^7H^7) + \widehat{\beta}H^2\sqrt{dH\iota(S+H^2K)} + H^3\sqrt{HK\iota}\right].$$

Add these three displays, (50), and H^2K. Every $\widetilde{\sigma}_{k,h}^2$ contains the summand H, so $S \geq H^2K$ and $\sqrt{S+H^2K} \leq \sqrt{2S}$. Since $d, H, \iota, B_\beta \geq 1$, every term independent of $\sqrt{S}$ is bounded by $C\iota^2(H^2K + d^{10}H^{10} + B_\beta d^{10}H^8)$, while every coefficient of $\sqrt{S}$ is bounded by $C\iota d^{7/2}H^{7/2}B_\beta$. Thus

$$S \leq C_8\iota^2\left(H^2K + d^{10}H^{10} + B_\beta d^{10}H^8\right) + C_8\iota d^{7/2}H^{7/2}B_\beta\sqrt{S}.$$

For nonnegative x, A, B, $x \leq A + B\sqrt{x}$ implies $x \leq 2A + 2B^2$, because $B\sqrt{x} \leq x/2 + B^2/2$. The square of the coefficient of $\sqrt{S}$ is $C\iota^2 d^7 H^7 B_\beta^2$. This proves (45) term by term.

Finally, apply Lemma 6 with $(\beta, \tau) = (\widehat{\beta}, H)$:

$$\sum_{k,h} \min\{\widehat{\beta}u_{k,h}, H\} \leq C\left[H^2 dB_\sigma\iota + \widehat{\beta}\Gamma^2 dH\iota + \widehat{\beta}\sqrt{dH\iota(S+H^2K)}\right].$$

Use $B_\sigma = 16d^6H^5$, $\Gamma = 6d^3H^2$, and $S \leq S_0$. Include $\sqrt{\iota}$ in the outer factor ι. This proves (46). □

E.4 Regret bound

We now have the two estimates needed for regret: Lemma 5 gives optimism and bounds each Bellman error, while Lemma 7 bounds their sum. The next theorem connects them. Its proof subtracts the Bellman equation of the executed policy from the optimistic estimate and sums the resulting recursion along each trajectory.

Theorem 4 (Regret under general Gaussian noise scales)**.** *Under Assumption 1, choose the shift and confidence radii by (9) and (35). With probability at least $1 - \delta_\Lambda - \delta_b - 7\delta$, Algorithm 1 satisfies*

$$\mathrm{Regret}(K) \leq C\iota\left[d^7H^7 + \widehat{\beta}d^7H^5 + \widehat{\beta}\sqrt{dH(S_0+H^2K)} + H\sqrt{HK\iota}\right], \tag{51}$$

where S_0 is defined in (44). Equivalently, after suppressing logarithmic factors,

$$\mathrm{Regret}(K) \leq \min\left\{HK, \widetilde{\mathcal{O}}\left(\begin{array}{l} d^7H^7 + \widehat{\beta}d^7H^5 + H^{3/2}\sqrt{K} \\ \quad + \widehat{\beta}\sqrt{dH\left(H^2K + d^{10}H^{10} + B_\beta d^{10}H^8 + d^7H^7B_\beta^2\right)}\end{array}\right)\right\}. \tag{52}$$

Proof. Write the regret in one episode as a sum of confidence bonuses and martingale differences, then sum this inequality over all episodes.

Fix an episode k, and let k_{last} be its most recent switch episode. Optimism gives $V_1^\star(s_1^k) \leq \widetilde{\widehat{V}}_{k,1}(s_1^k)$. Define

$$e_{k,h}(s) := [\widetilde{\widehat{V}}_{k,h}(s) - V_h^{\widetilde{\pi}^k}(s)]_{[0,H]}, \qquad e_{k,H+1} := 0.$$

When the optimism conclusion of Lemma 5 holds, clipping does not change this gap. On every outcome, the gap is at most H. At the visited state, the policy chooses $a_h^k \in \arg\max_a \widetilde{\widehat{Q}}_{k,h}(s_h^k, a)$. The first confidence inequality in (37), applied to the estimates frozen at k_{last}, gives

$$\widetilde{\widehat{Q}}_{k,h}(s_h^k, a_h^k) \leq r_h(s_h^k, a_h^k) + P_h\widetilde{\widehat{V}}_{k,h+1}(s_h^k, a_h^k) + 2\widehat{\beta}u_{k,h}.$$

Both values lie in $[0, H]$. If $\widehat{\beta}u_{k,h} > H$, use the bound by H. Otherwise, use the last display. Subtract the Bellman identity

$$V_h^{\widetilde{\pi}^k}(s_h^k) = r_h(s_h^k, a_h^k) + P_hV_{h+1}^{\widetilde{\pi}^k}(s_h^k, a_h^k)$$

to obtain

$$e_{k,h}(s_h^k) \leq 2\min\{\widehat{\beta}u_{k,h}, H\} + P_he_{k,h+1}(s_h^k, a_h^k).$$

Define

$$\xi_{k,h}^e := P_he_{k,h+1}(s_h^k, a_h^k) - e_{k,h+1}(s_{h+1}^k).$$

Add and subtract the value at the observed next state. The recursion becomes

$$e_{k,h}(s_h^k) \le 2\min\{\widehat{\beta}u_{k,h}, H\} + e_{k,h+1}(s_{h+1}^k) + \xi_{k,h}^e.$$

We now sum this recursion along the trajectory. Sum the display over $h = 1, \dots, H$. The middle $e_{k,h}$ terms cancel, and $e_{k,H+1} = 0$. Hence

$$e_{k,1}(s_1^k) \le 2\sum_{h=1}^{H} \min\{\widehat{\beta}u_{k,h}, H\} + \sum_{h=1}^{H} \xi_{k,h}^e.$$

Since the per-episode regret is at most $e_{k,1}(s_1^k)$, summing over episodes gives

$$\mathrm{Regret}(K) \le 2\sum_{k,h} \min\{\widehat{\beta}u_{k,h}, H\} + \sum_{k,h} \xi_{k,h}^e.$$

Lemma 7 bounds the first sum. For the second sum, $\mathbb{E}[\xi_{k,h}^e \mid \mathcal{F}_{k,h}] = 0$ and $|\xi_{k,h}^e| \le H$. Hoeffding's lemma gives

$$\mathbb{E}[e^{\lambda \xi_{k,h}^e} \mid \mathcal{F}_{k,h}] \le e^{\lambda^2 H^2/2}.$$

Apply this inequality over all k, h. Then $\exp(\lambda \sum_{k,h} \xi_{k,h}^e - \lambda^2 H^2 KH/2)$ has expectation at most one. Markov's inequality and $\lambda = \sqrt{2\iota/(H^2 KH)}$ give $\sum_{k,h} \xi_{k,h}^e \le H\sqrt{2HK\iota}$. Substituting (46) proves (51). Substituting S_0 from (44) gives (52); the bound by HK uses rewards in $[0, 1]$. Finally, we add the failure probabilities. The three applications used in Lemma 5 fail with total probability at most 3δ. The three additional concentration bounds used for (45) fail with total probability at most 3δ, and the final martingale bound fails with probability at most δ. Adding the failure probabilities for the Gram and vector noise bounds gives the stated probability. □

The general theorem keeps the noise magnitudes inside the perturbation radii. The remaining task is algebraic: substitute each of the two Gaussian noise choices into those radii and then into (52). No additional concentration argument is needed.

Theorem 5 (Rates under the two noise choices)**.** *Under Assumption 1, let $0 < \rho \le 1$, set the noise scales equal to the right sides of (13) or (14), and split a failure budget δ among the events above. For choice (I), choose $c_\Lambda = \Theta(H^{-2} + dH\nu_\Lambda)$; for choice (II), choose $c_\Lambda = \Theta(H^{-2} + dH\nu_\Lambda + d^2H/\rho)$, with sufficiently large hidden universal constants in both choices. Under choice (I), the quantities in the general bound (52) satisfy*

$$\widehat{\beta} = \widetilde{\mathcal{O}}\left(\sqrt{d} + \frac{d^{3/2}H^{5/4}}{\sqrt{\rho}}\right), \qquad B_\beta = \widetilde{\mathcal{O}}\left(d^{3/2}H^2 + \frac{d^{3/2}H^{9/4}}{\sqrt{\rho}}\right).$$

Under choice (II), they satisfy

$$\widehat{\beta} = \widetilde{\mathcal{O}}\left(\sqrt{d} + \frac{d^{3/2}H^{3/2}}{\sqrt{\rho}}\right), \qquad B_\beta = \widetilde{\mathcal{O}}\left(d^{3/2}H^2 + \frac{d^{3/2}H^{5/2}}{\sqrt{\rho}}\right).$$

Consequently, its term that grows with K is

$$\text{(I)} \quad \widetilde{\mathcal{O}}\left(dH^{3/2}\sqrt{K} + d^2H^{11/4}\sqrt{\frac{K}{\rho}}\right), \tag{53}$$

$$\text{(II)} \quad \widetilde{\mathcal{O}}\left(dH^{3/2}\sqrt{K} + d^2H^3\sqrt{\frac{K}{\rho}}\right). \tag{54}$$

For $d, H \ge 1$, the complete bounds simplify to

$$\text{(I)} \quad \mathrm{Regret}(K) \le \min\left\{HK, \widetilde{\mathcal{O}}\left(dH^{3/2}\sqrt{K} + d^2H^{11/4}\sqrt{\frac{K}{\rho}} + \frac{d^9H^8}{\rho}\right)\right\}, \tag{55}$$

$$\text{(II)} \quad \mathrm{Regret}(K) \le \min\left\{HK, \widetilde{\mathcal{O}}\left(dH^{3/2}\sqrt{K} + d^2H^3\sqrt{\frac{K}{\rho}} + \frac{d^9H^8}{\rho}\right)\right\}. \tag{56}$$

Proof. The proof only substitutes the two noise choices into the general regret bound. We do this in three steps: bound the noise, bound the confidence radii, and simplify the regret terms.

Under choice (I), the Gaussian noise bounds give

$$\nu_\Lambda = \widetilde{\mathcal{O}}\Big(\frac{d}{\sqrt{H}\,\rho}\Big), \qquad \zeta_1, \zeta_2 = \widetilde{\mathcal{O}}\Big(\frac{dH}{\rho}\Big), \qquad \zeta_3 = \widetilde{\mathcal{O}}\Big(\frac{dH^2}{\rho}\Big).$$

Under choice (II), they give

$$\nu_\Lambda = \widetilde{\mathcal{O}}\Big(\frac{d}{\sqrt{H}\,\rho}\Big), \qquad \zeta_1, \zeta_2 = \widetilde{\mathcal{O}}\Big(\frac{d^{3/2}H^2}{\rho}\Big), \qquad \zeta_3 = \widetilde{\mathcal{O}}\Big(\frac{d^{3/2}H^3}{\rho}\Big).$$

The shifts in the theorem satisfy (9). They also have $b_\Lambda = \Theta(c_\Lambda)$ and obey

$$\text{(I)} \quad c_\Lambda = \widetilde{\mathcal{O}}\Big(H^{-2} + \frac{d^2\sqrt{H}}{\rho}\Big), \qquad \text{(II)} \quad c_\Lambda = \widetilde{\mathcal{O}}\Big(H^{-2} + \frac{d^2 H}{\rho}\Big).$$

We next use these bounds in (33). This gives

$$\text{(I)} \quad \frac{(c_\Lambda + \nu_\Lambda)H\sqrt{d}}{\sqrt{b_\Lambda}} \le \widetilde{\mathcal{O}}\Big(\sqrt{d} + \frac{d^{3/2}H^{5/4}}{\sqrt{\rho}}\Big), \quad \frac{\zeta_1+\zeta_2}{\sqrt{b_\Lambda}} + \frac{\zeta_3}{H\sqrt{b_\Lambda}} \le \widetilde{\mathcal{O}}\Big(\frac{d^{3/2}H^{5/4}}{\sqrt{\rho}}\Big),$$
$$\text{(II)} \quad \frac{(c_\Lambda + \nu_\Lambda)H\sqrt{d}}{\sqrt{b_\Lambda}} \le \widetilde{\mathcal{O}}\Big(\sqrt{d} + \frac{d^{3/2}H^{3/2}}{\sqrt{\rho}}\Big), \quad \frac{\zeta_1+\zeta_2}{\sqrt{b_\Lambda}} + \frac{\zeta_3}{H\sqrt{b_\Lambda}} \le \widetilde{\mathcal{O}}\Big(\frac{d^{3/2}H^{3/2}}{\sqrt{\rho}}\Big).$$

Together with (35), these estimates give the two stated bounds on $\widehat{\beta}$ and B_β.

Finally, use these bounds in the general result (52). Its H^2K term gives (53) and (54).

For choice (I), the other three terms inside the square root in (52) are

$$\widetilde{\mathcal{O}}\Big(\frac{d^7H^{27/4}}{\sqrt{\rho}}\Big), \qquad \widetilde{\mathcal{O}}\Big(\frac{d^{31/4}H^{55/8}}{\rho^{3/4}}\Big), \qquad \widetilde{\mathcal{O}}\Big(\frac{d^7H^{15/2}}{\rho}\Big).$$

For choice (II), these terms are

$$\widetilde{\mathcal{O}}\Big(\frac{d^7H^7}{\sqrt{\rho}}\Big), \qquad \widetilde{\mathcal{O}}\Big(\frac{d^{31/4}H^{29/4}}{\rho^{3/4}}\Big), \qquad \widetilde{\mathcal{O}}\Big(\frac{d^7H^8}{\rho}\Big).$$

The terms outside the square root are

$$\widetilde{\mathcal{O}}\Big(d^7H^7 + \frac{d^{17/2}H^{25/4}}{\sqrt{\rho}}\Big), \qquad \widetilde{\mathcal{O}}\Big(d^7H^7 + \frac{d^{17/2}H^{13/2}}{\sqrt{\rho}}\Big).$$

For $d, H \ge 1$ and $0 < \rho \le 1$, each added term is bounded by $\widetilde{\mathcal{O}}(d^9H^8/\rho)$. Finally, regret is at most HK because rewards lie in $[0,1]$. This proves the two compact bounds. □

The leading non-private term in (52) is minimax optimal. Releasing Gram prefix sums through a tree and vector statistics only at switches yields a $\sqrt{K}$ dependence, up to logarithmic factors, in the noise-dependent regret term. We do not claim that the dependence on ρ is optimal. The current proof includes the Gram-matrix shift in the confidence bonus at every episode; removing that repeated cost requires a different estimator or a charging argument over the episodes between two policy changes. The second noise choice reduces expected retraining time by adding more noise and guaranteeing a smaller TV distance between neighboring output laws. Its per-release scaling loses an additional factor $H^{1/4}$ in the displayed stability-dependent regret term.