Boosting for Reinforcement Learning in Structured MDPs

Anonymous Author(s)
Affiliation
Address
email

Abstract

Boosting is a powerful machine learning technique that constructs a strong learner by sequentially combining weak learners, each of which performs only slightly better than random. Recent work has adapted boosting to reinforcement learning and established global convergence guarantees under the assumption of access to a multiplicative weak learner (Brukhim et al., 2022). These guarantees critically depend on occupancy mismatch terms relative to the optimal policy, however the mismatch ratio between the boosted policy class and the optimal policy can become unbounded unless the policy class ensures sufficient state-space coverage. In this work, we remove this assumption and show that, whenever weak learning is feasible, boosting can achieve convergence guarantees that depend only on the intrinsic complexity of the underlying Markov Decision Process.

1 Introduction

In reinforcement learning, an agent interacts with an environment, specified by a Markov Decision Process (MDP), with the objective of learning a policy for interaction that maximizes its expected cumulative reward. The computational and statistical complexity of this task depends heavily on the structure of the underlying MDP, and while computationally efficient algorithms that provably learn an optimal policy are known for MDPs with simple structure, such as tabular or low-rank MDPs (Brafman and Tennenholtz, 2002; Kearns and Singh, 2002; Jin et al., 2020), such algorithms are known not to exist for some more expressive environments like block MDPs (under reasonable cryptographic assumptions (Kane et al., 2022)).

Nonetheless, heuristic methods like policy gradient methods and deep-Q learning have been used in practice to obtain impressive performance in complex environments such as MDPs with continuous state or action spaces. This observation motivates the study of ensembling approaches for reinforcement learning – in cases where efficient, provable optimization methods are not known to exist, but policies that improve over random actions can be efficiently learned, how can we effectively generate and ensemble a collection of "rule of thumb" policies to obtain something close to optimal?

Prior work of Brukhim et al. (2022) proposed a boosting-style approach to ensembling for RL, inspired by boosting techniques from supervised learning (Freund, 1995). Assuming the existence of a weak supervised learner for a parameterized policy class – a learner that outputs a policy somewhat better than taking random actions, but potentially far from optimal – they give an algorithm to iteratively refine a policy via aggregation with new policies generated by the weak learner. To ensure each update improves the current policy, the reward function of the underlying MDP for the RL instance is modified at each iteration to encourage the weak learner to output policies that capture residual reward missed by the current policy.

The convergence of their algorithm to an approximately optimal policy is *independent of the size of the state space* of the underlying MDP, and so is applicable to environments far more complex than

Submitted to 40th Conference on Neural Information Processing Systems (NeurIPS 2026). Do not distribute.

simple tabular MDPs. However, their convergence guarantees require two assumptions. The first is a reasonable assumption on the expressibility of an optimal policy as a function of policies from the class parameterizing the weak learner. The second is a common, but strong assumption arising from off-policy learning – that the worst case distribution mismatch between the state visitation distribution of the optimal policy and any policy in the parameterized class is bounded.

In this work, we aim to remove this second assumption by providing an alternative approach to boosting in RL that explicitly encourages exploration. By iteratively altering the reward function of the underlying MDP to promote visitation of new states, our algorithm converges independent of distribution mismatch measures, guaranteeing convergence even if the starting state distribution and that of the optimal policy are disjoint.

1.1 Our Results

We present a boosting approach for reinforcement learning that removes the dependence on occupancy mismatch to the optimal policy present in Brukhim et al. (2022). In order to achieve this independence, our algorithms modify the reward function of the underlying MDP to add a bonus term that rewards visitation of under-explored states, using an approach that builds on the L_∞ and pushforward coverability objectives of Amortila et al. (2024). By shifting the dependency from occupancy mismatch to the less restrictive notion of coverability, our framework yields boosting-based RL guarantees with sample complexity that scales polynomially with the MDP's intrinsic parameters. This weakens the coverage assumptions on the policy class required by the prior RL boosting approach in Brukhim et al. (2022).

Our algorithms construct a collection of policies by iteratively invoking the boosting framework of Brukhim et al. (2022). To avoid introducing a dependence on distribution mismatch for the target MDP and policy class, we only invoke the boosting subroutine with a modified policy class and MDP which guarantees efficient Franke-Wolfe style best-iterate convergence (for the modified MDP and policy class). The modified MDP mimics the target MDP for states with good coverage under the current collection of policies, so to ensure convergence for the target MDP, it suffices to show that the modified MDP converges to something that well-approximates the target. We guarantee this by modifying the reward function within each call to the boosting subroutine with an exploration bonus using coverability notions from Amortila et al. (2024), ensuring that reachable states of the MDP are visited with sufficiently high probability.

1.2 Paper organization

We include preliminaries on RL in Section 2, including psuedocode for the boosting framework of Brukhim et al. (2022) which our algorithms use as a subroutine, as well as relevant concepts. In Section 3 we present our two main boosting algorithms, and recall the L_∞ (Section 3.1) and pushforward (Section 3.2) coverability notions from Amortila et al. (2024) that we use to define our exploration bonuses. We summarize our contributions and future directions in Section 4. A glossary of notation can be found in Appendix A. We discuss related work not covered in the introduction in Appendix B. We provide the setup for the extended MDP and truncated policy class in Appendix C. The complete proofs of the main text are in Appendix D and Appendix E. We provide a discussion and results of using boosting for the reward-free setting in Appendix F. An adaptation of the RL boosting framework of Brukhim et al. (2022) to episodic MDPs is given in Appendix G. Finally, the results of empirical validation of our theoretical results can be found in Appendix K.

2 Preliminaries

Markov Decision Process. We consider an infinite-horizon discounted Markov Decision Process (MDP) defined by the tuple $\mathcal{M} = (\mathcal{X}, \mathcal{A}, P, r, \gamma, d_0)$, where $\mathcal{S}$ and $\mathcal{A}$ are the state and action spaces, respectively. The transition kernel $P(x' \mid x, a)$ denotes the probability of transitioning to state x' after taking action a in state x. The reward function is given by $r : \mathcal{X} \times \mathcal{A} \to [0, 1]$, $\gamma \in [0, 1)$ is the discount factor, and $d_0 \in \Delta(\mathcal{X})$ is the initial state distribution.

An agent's behavior is characterized by a stochastic policy $\pi : \mathcal{X} \to \Delta(\mathcal{A})$. The interaction between π and $\mathcal{M}$ induces a distribution over trajectories $\tau = (x_0, a_0, x_1, a_1, \dots)$, defined by $x_0 \sim d_0$, $a_t \sim \pi(\cdot \mid x_t)$, and $x_{t+1} \sim P(\cdot \mid x_t, a_t)$. We define the state-action value function $Q^\pi(x, a)$ and the

state value function $V^\pi(x)$ as:

$$Q^\pi(x,a) = \mathbb{E}_\pi\left[\sum_{t=0}^{\infty}\gamma^t r(x_t,a_t) \mid x_0 = x, a_0 = a\right], \quad V^\pi(x) = \mathbb{E}_{a\sim\pi(\cdot|x)}[Q^\pi(x,a)].$$

For any distribution d_0 over $\mathcal{X}$, we denote $V^\pi_{d_0} = \mathbb{E}_{x\sim d_0}[V^\pi(x)]$. When the distribution is omitted, V^π refers to $V^\pi_{d_0}$, and we denote the optimal value as $V^* = \max_\pi V^\pi$.

The policy π also induces a normalized discounted state occupancy measure $d^\pi_{d_0} \in \Delta(\mathcal{X})$, defined as:

$$d^\pi_{d_0}(x) = (1-\gamma)\sum_{t=0}^{\infty}\gamma^t \mathbb{P}(x_t = x \mid \pi, x_0 \sim d_0). \tag{1}$$

Equivalently, the value function can be expressed via this occupancy measure as $V^\pi_{d_0} = \frac{1}{1-\gamma}\mathbb{E}_{x\sim d^\pi_{d_0}, a\sim\pi(\cdot|x)}[r(x,a)]$.

Modes of Accessing the MDP. We consider the *episodic rollout setting* in which the algorithm can execute a policy, terminate the rollout at any time, and restart from the initial states, and can repeat this process multiple times.

In this work, we consider large-scale MDPs where the state space $\mathcal{X}$ can be very large, or even infinite. Such settings are challenging and statistically intractable in general (Krishnamurthy et al., 2016). However, in many applications, the environment admits some underlying structure that enables both computationally and statistically efficient algorithms; examples of such environments include block MDPs (Misra et al., 2020; Du et al., 2019) and linear/low-rank MDPs (Jiang et al., 2017; Agarwal et al., 2020). We provide a brief description of these structured MDPs in the following paragraphs.

Block MDPs. A block MDP models an environment where a large state space $\mathcal{X}$ can be summarized into a finite latent state $\mathcal{S}$. Specifically, block MDP assumes there exists an emission function $q : \mathcal{S} \to \Delta(\mathcal{X})$ that maps the latent states $\mathcal{S}$ to a distribution over $\mathcal{X}$. The agent interacts with the environment by repeatedly generating trajectories $(s_0, x_0, a_0, r_0, s_1, x_1, a_1, r_1, \ldots)$, where $s_0 \sim d_0$, $s_{t+1} \sim P(\cdot \mid s_t, a_t), x_t \sim q(s_t)$. The agent can observe x_t, but the latent states s_t are hidden from the agent. Tabular MDP is a special case of block MDP where the observation space coincides with the state space, i.e., $\mathcal{X} = \mathcal{S}$.

Low-rank MDPs. A low-rank MDP with dimension d is an MDP where the transition kernel admits a low-rank factorization $\phi : \mathcal{X}\times\mathcal{A} \to \mathbb{R}^{\mathrm{d}}$ and $\mu : \mathcal{X} \to \mathbb{R}^{\mathrm{d}}$ such that $\forall x, x' \in \mathcal{X}, a \in \mathcal{A} : P\left(x' \mid x, a\right) = \langle \phi\left(x,a\right), \mu(x')\rangle$. This structure enables efficient representation and learning in environments with large or continuous state spaces.

2.1 Boosting Framework (Brukhim et al., 2022)

In this section, we restate the key definitions and the boosting procedure in Brukhim et al. (2022).

Definition 2.1 (Distribution Mismatch)**.** Let $\pi^* = \operatorname{argmax}_\pi V^\pi$ denote the optimal policy. We define the following distribution mismatch coefficients: $C_\infty = \max_{\pi\in\Pi}\left\|d^{\pi^*}/d^\pi\right\|_\infty$.

A natural approach to defining a weak learner is to require performance that is marginally better than that of a random policy. In the definition below, let π_{Rand} be a uniformly random policy, i.e. $\forall(x,a) \in \mathcal{X}\times\mathcal{A}, \pi_{\mathsf{Rand}}(a \mid x) = 1/|\mathcal{A}|$.

Definition 2.2 (Weak Supervised Learner)**.** Let $\alpha \in (0,1]$. Consider a class $\mathcal{L}$ of linear loss functions $\ell : \mathbb{R}^{\mathcal{A}} \to \mathbb{R}$, a family $\mathbb{D}$ of distributions that are supported over $S\times\mathcal{L}$, and policy class Π. A weak supervised learning algorithm, for every $\varepsilon_{\mathsf{w}}, \delta > 0$, given $m(\varepsilon_{\mathsf{w}}, \delta) = \frac{\log|\mathcal{W}|}{\varepsilon^2_{\mathsf{w}}}\log\frac{1}{\delta}$ samples D_m from any distribution $\mathcal{D} \in \mathbb{D}$ outputs a policy $\mathcal{W}(D_m) \in \Pi$ such that with probability $1-\delta$,

$$\mathbb{E}_{(s,\ell)\sim\mathcal{D}}\left[\ell(\mathcal{W}(D_m))\right] \le \alpha \min_{\pi^*\in\Pi}\mathbb{E}_{(s,\ell)\sim\mathcal{D}}\left[\ell(\pi^*(s))\right] + (1-\alpha)\,\mathbb{E}_{(s,\ell)\sim\mathcal{D}}\left[\ell(\pi_{\mathsf{Rand}}(s))\right] + \varepsilon_{\mathsf{w}}.$$

Definition 2.3 (Policy Projection)**.** Given $\tilde\pi : \mathcal{X} \to \mathbb{R}^{|\mathcal{A}|}$, define a projected policy $\pi = \Gamma[\tilde\pi]$ to be a policy such that simultaneously for all $x \in \mathcal{X}$, it holds that $\pi(\cdot \mid x) = \Gamma[\tilde\pi(x)]$.

Due to the multiplicative factor α, the policy obtained from the second boosting procedure (Internal Boost; Brukhim et al. (2022, Algorithm 2)) might not be a convex combination of weak policies. The projection operator Γ ensures that the policy obtained from the boosting procedure is valid. The projection can be computed efficiently using a water-filling algorithm (Duchi et al., 2008), which requires sorting $|\mathcal{A}|$ numbers (Brukhim et al., 2022, Claim 12).

The policy obtained from the RL boosting algorithm of Brukhim et al. (2022) can be presented as a two-level tree, which we restate below.

Definition 2.4 (Policy Tree)**.** A Policy Tree $\mathbb{\Pi} \subseteq \mathcal{X} \to \Delta_{\mathcal{A}}$ with respect to $\Pi \subseteq \mathcal{X} \to \Delta_{\mathcal{A}}$ some base policy class, and $N, T \in \mathbb{N}$, is a linear combination of T projected policies $\Gamma[\tilde{\pi}]$, where each $\tilde{\pi}$ is a linear combination of N base policies $\pi \in \Pi$.

Definition 2.5 (Policy Completeness)**.** For any initial state distribution μ, and policy classes $\Pi, \mathbb{\Pi}$, define

$$\mathcal{E}_\mu(\mathbb{\Pi}, \Pi) = \max_{\pi \in \mathbb{\Pi}} \min_{\pi^* \in \Pi} \mathbb{E}_{x \sim d_\mu^\pi} \left[\max_{a \in \mathcal{A}} Q^\pi(x, a) - Q^\pi(x, \cdot)^\top \pi^*(\cdot|x) \right].$$

Policy completeness quantifies how well the restricted policy class Π can approximate the greedy policy induced by the action-value function of any policy in the larger class $\mathbb{\Pi}$, as measured under the state distribution d_μ^π.

We highlight the core boosting framework of Brukhim et al. (2022) in Algorithm 1, and refer to the Appendix L.1 for their remaining subroutines and discussion. At a high level, the framework employs a two-level boosting scheme. The outer level follows a CPI procedure (Kakade and Langford, 2002), which aggregates policies produced by an inner boosting routine (Internal Boost). The Internal Boost is designed to return a policy π'_t (Line 4 of Algorithm 1) that approximately closes the best policy under the state distribution $d^{\pi_{t-1}}$. Informally,

$$\mathbb{E}_{x \sim d^{\pi_{t-1}}}[\max_{\pi \in \Pi} Q^{\pi_{t-1}}(x, \cdot)^\top \pi(x) - Q^{\pi_{t-1}}(x, \cdot)^\top \pi'_t(x)] \leq \frac{2|\mathcal{A}|}{(1-\gamma)\alpha}(\varepsilon_0 + \frac{2}{\sqrt{N}}),$$

where α is the weak-learner multiplicative factor in Definition 2.2, ε_0 is the boosting oracle error, and N is the number of weak learners used in the Internal Boost.

Algorithm 1 RL Boosting (Brukhim et al., 2022)

1: **Input**: reward function r, policy class Π, number of iterations T, initial state distribution μ, and P, N, M parameters for Internal Boost.
2: Initialize a policy $\pi_0 \in \Pi$ arbitrarily.
3: **for** $t = 1$ **to** T **do**
4: Run Internal Boost (Algorithm 2 of Brukhim et al. (2022)) with distribution μ and policy π_{t-1} to obtain π'_t.
5: Update $\pi_t = (1 - \eta_{1,t})\pi_{t-1} + \eta_{1,t}\pi'_t$.
6: **end for**
7: Run each policy π_t for P rollouts to compute an empirical estimate $\widehat{V^{\pi_t}}$ of the expected return.
8: **return** $\bar{\pi} := \pi_{t'}$ where $t' = \operatorname{argmax}_t \widehat{V^{\pi_t}}$.

3 Main results

In this section, we introduce our main algorithms (Algorithm 2, Algorithm 3). The convergence of both algorithms depends on coverability parameters (defined in Section 3.1 and Section 3.2 respectively) that are known to be bounded for many interesting classes of structured MDPs. Notably, neither algorithm assumes bounded distribution mismatch C_∞ for the underlying MDP and policy class, and so we give the first boosting algorithms with provable convergence guarantees without requiring bounded C_∞.

The assumption of bounded C_∞ is highly restrictive. For C_∞ to be bounded, every policy $\pi \in \Pi$ must have an occupancy d^π that covers the optimal policy's visitation. If even a single policy in the class fails this, C_∞ becomes unbounded. This effectively eliminates the need for exploration, a requirement that rarely holds in practice. We provide the following example showing that bounded C_∞ is indeed a strong assumption even under a simple RL setting (Figure 1).

Motivating example. Consider the tabular MDP, where we only have one single starting state, i.e., $d_0(x_1) = 1, d_0(x_2) = 0, d_0(x_3) = 0$. The states x_2 and x_3 are self-loop, and the agent receives zero reward when it is in either of these states. The action space consists of n actions, and transitions are deterministic. Let $\Pi = \{\pi_1, \ldots, \pi_n\}$ denote the set of deterministic policies, where each policy selects a single action. The optimal policy is the deterministic policy π_n that selects a_n, which transitions to the left and yields a reward of 1. Each of the remaining deterministic policies selects one of the other actions and yields a reward of $1 - \varepsilon$. Let $\mathrm{CH}(\Pi)$ denote the convex hull of Π, and let $\pi \in \mathrm{CH}(\Pi)$ be any boosted (stochastic) policy. It is immediate that every policy in $\mathrm{CH}(\Pi)$ achieves near-optimal performance: $V^* - V^\pi \leq \varepsilon, \forall \pi \in \mathrm{CH}(\Pi)$. However, there exists a policy π in $\mathrm{CH}(\Pi)$ that assigns zero probability to the optimal action. For such a policy, the induced state distribution satisfies $d^\pi(x_2) = 0$, while $d^{\pi^*}(x_2) > 0$. As a result, $C_\infty := \sup_{\pi \in \mathrm{CH}(\Pi)} \left\| d^{\pi^*}/d^\pi \right\|_\infty \to \infty$ due to the nature definition of C_∞, a supremum over all policy in $\mathrm{CH}(\Pi)$.

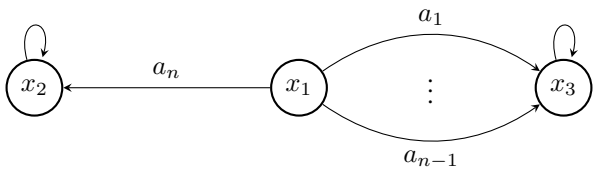

Figure 1: An MDP with 3 states and n actions, play optimal action a_n will get reward of 1, the other $n-1$ actions will get reward of $1-\varepsilon$, and receive no reward when in the self-loop state x_2, x_3.

Our solution to eliminate the dependency on the distribution mismatch coefficient C_∞ is to incorporate an explicit exploration strategy into the boosting framework of Brukhim et al. (2022) (see Appendix L.1 for a comprehensive description of their boosting algorithm, which we use as a subroutine). Specifically, we augment the true reward with an exploration bonus constructed via a policy cover. We leverage recent developments in L_∞ policy coverage and pushforward coverability within online reinforcement learning, which allows the algorithm to adapt automatically to the inherent complexity of the underlying MDP (Amortila et al., 2024). While L_∞ coverability generally requires stronger prior knowledge of the MDP to construct the bonus, the pushforward coverability framework relaxes this requirement by enabling the estimation of an augmented bonus.

Our algorithmic approach follows similarly to Agarwal et al. (2020), which constructs a policy cover in an iterative manner, though their results are restricted to *linear MDPs*. Using this policy cover, they design an exploration bonus that is added to the reward to encourage exploration, and optimize policies by solving $\max_\pi \mathbb{E}_{s_0,a_0 \sim \rho^n_{\mathrm{cov}}} \left[Q^\pi \left(s_0, a_0; r + b \right) \right]$, where b is the bonus function and ρ_{cov} is a distribution constructed from the policy cover. In our framework, we use boosting to solve this policy optimization problem. Our theoretical guarantees are established specifically for the boosting-based solution, whereas in practice, the optimization step can be instantiated with any suitable policy optimization algorithm.

Next, we present our first main result on L_∞ coverability.

3.1 L_∞-coverability

In this section, we analyze the performance of our boosting algorithm under the assumption of access to a reference distribution μ that provides global coverage of the state-action space. This ensures that any admissible occupancy distribution induced by a policy is not "too far" from μ in terms of their density ratio (Amortila et al., 2024; Chen and Jiang, 2019). This requirement is formalized by the following assumption:

Assumption 3.1 (L_∞-concentrability coefficient (Chen and Jiang, 2019))**.** We assume access to a distribution $\mu \in \Delta(\mathcal{X} \times \mathcal{A})$ for which the L_∞-concentrability coefficient $\widetilde{C}_\infty := \sup_\pi \sup_{(x,a) \in \mathcal{X} \times \mathcal{A}} \left\{ \frac{d^\pi(x,a)}{\mu(x,a)} \right\}$ is bounded.

Next, we provide the pseudo-code of our first algorithm in Algorithm 2. The algorithm runs for K iterations to generate policy cover. To ensure sufficient exploration under the L_∞-concentrability assumption, we define a weight function $w^k(x,a)$ that decays as state-action pairs are visited by

Algorithm 2 Boosting with L_∞ coverability

input: number of epoch K, number of iterations T, number of weak learners N, number of sample episodes M, policy class Π.
Initialize policy π^1 arbitrarily, $p^1 = \{\pi^1\}$.
for $k = 1, \cdots, K$ **do**
    Construct a new bonus $b^k(x, a)$ with policy cover d^{p^k} (see Eq. (2)).
    Update policy class $\widehat{\Pi} \leftarrow \mathcal{T}\left(\Pi, b^k\right)$ (see Eq. (3)).
    Run RL boosting (Algorithm 1) with reward function $r + b^k$, policy class $\widehat{\Pi}$, initial distribution d^{p^k} to obtain π^k.
    Update the policy cover $p^{k+1} = \mathrm{Unif}(\pi_1, \ldots, \pi_k)$.
end for
return $\pi := \mathrm{argmax}_{\pi \in \{\pi^1, \ldots, \pi^K\}} V^\pi$.

previous policies. For each iteration k, the weight function and the corresponding exploration bonus $b^k(x, a)$ are defined as follows:

$$w^k(x,a) = \frac{\widetilde{C}_\infty \mu(x,a)}{\sum_{i<k} d^{\pi_i}(x,a) + \widetilde{C}_\infty \mu(x,a)}, \qquad b^k(x,a) = \frac{2}{1-\gamma} w^k(x,a). \tag{2}$$

The weight $w^k(x, a)$ acts as an inverse frequency measure. When the total density of previous policies $\sum d^{\pi_i}$ at a certain (x, a) is low relative to the reference distribution μ, the weight w^k remains high, providing a larger bonus to encourage exploration. As the algorithm collects more samples and the cumulative density grows, the bonus naturally vanishes. The factor $\frac{1}{1-\gamma}$ in the bonus is to cap the maximum value of any value function $V^\pi(x)$.

To facilitate exploration, the algorithm adopts a uniform strategy over actions that remain insufficiently explored. This mechanism is formalized through a policy class transformation, $\mathcal{T}\left(\Pi, b^k\right)$, which maps each original policy $\pi \in \Pi$ to a corresponding transformed policy $\widehat{\pi}$, which we formalize as follows.

Policy transformation $\mathcal{T}(\Pi, b^k)$. Given the well-explored set $\mathcal{B}^k := \left\{x : w^k(x, a) \le \frac{1}{2}, \forall a \in \mathcal{A}\right\}$, the transformed policy $\widehat{\pi}$ is defined as follows:

$$\widehat{\pi}(x) = \begin{cases} \pi(x), & \text{if } x \in \mathcal{B}^k \\ \mathrm{Unif}\left(\left\{a \in \mathcal{A} : w^k(x,a) > \frac{1}{2}\right\}\right), & \text{otherwise} \end{cases} \tag{3}$$

Intuitively, the policy simply takes a random action on the state where the bonus is already large. We note that the policy transformation is introduced purely for abstraction; in practice, we do not need to iterate over the entire policy class Π. Its implementation is straightforward by checking the condition for all the actions (e.g., Algorithm 3 in Agarwal et al. (2020)).

With the policy transformation in place, we present the main result for the RL boosting under L_∞ coverability as follows:

Theorem 3.2. *For any $\varepsilon, \delta \in (0, 1)$, suppose Assumption 3.1 is satisfied, Algorithm 2 is instantiated with the following parameters: $K = O(\frac{\widetilde{C}_\infty \iota}{(1-\gamma)\varepsilon})$, $T = O(\frac{\widetilde{C}_\infty^2 \iota^2}{(1-\gamma)^8 \varepsilon^3})$, $N = O(\frac{\widetilde{C}_\infty^2 |\mathcal{A}|^2 \iota^2}{(1-\gamma)^{10} \alpha^2 \varepsilon^4})$, $M = m\left(\frac{(1-\gamma)^5 \alpha \varepsilon^2}{\widetilde{C}_\infty |\mathcal{A}|}, \frac{\delta}{KTN}\right)$, where $\iota = \log(\frac{\widetilde{C}_\infty}{(1-\gamma)\varepsilon})$. Then Algorithm 2 produces policy π, such that with probability at least $1 - \delta$,*

$$V^* - V^\pi \le \frac{\widetilde{C}_\infty \iota}{(1-\gamma)^5 \varepsilon} \mathcal{E}(\mathbb{\Pi}, \Pi) + \varepsilon.$$

Sample Complexity: *If $m(\varepsilon, \delta) = \frac{\log |\mathcal{W}|}{\varepsilon^2} \log \frac{1}{\delta}$ for some measure of weak learning complexity $|\mathcal{W}|$, then the algorithm uses at most $KTNM \le \widetilde{O}\left(\frac{\widetilde{C}_\infty^7 |\mathcal{A}|^4 \log |\mathcal{W}|}{(1-\gamma)^{29} \alpha^4 \varepsilon^{12}}\right)$ episodes.*

In tabular MDP, we can choose the distribution $\mu(x,a) = \frac{1}{|\mathcal{X}||\mathcal{A}|}$ which admits a closed form and $\widetilde{C}_\infty(\mu) \leq |\mathcal{X}||\mathcal{A}|$. For a Block MDP with latent state space $\mathcal{S}$, emission distribution $q : \mathcal{S} \to \Delta(\mathcal{X})$, and decoder $\phi : \mathcal{X} \to \mathcal{S}$, the distribution $\mu(x,a) := q\left(x \mid \phi(x)\right) \cdot \frac{1}{|\mathcal{S}||\mathcal{A}|}$ achieves $\widetilde{C}_\infty(\mu) \leq |\mathcal{S}||\mathcal{A}|$. However, the choice of distribution μ relies on prior knowledge of the underlying MDP. Pushforward coverability relaxes this requirement by constructing the exploration objective directly from the transition dynamics, which we present the corresponding result in the next section.

3.2 Pushforward coverability

The pushforward coverability is a more general notion that can be applied to various structured MDPs, and does not require the reference distribution like L_∞ coverability. The pseudo-code of the boosting with policy cover algorithm is provided in Algorithm 3. At each episode k, we use the following weight function and bonus as follows:

$$w^k(x \mid x', a') := \frac{P(x \mid x', a')}{\sum_{i<k} d^{\pi^i}(x) + P(x \mid x', a')}, \quad b^k(x \mid x', a') := \frac{2}{1-\gamma} w^k(x \mid x', a'), \tag{4}$$

Intuitively, if a state x is poorly covered by previous policies, the sum $\sum_{i<k} d^{\pi^i}(x)$ will be small, and the resulting weight will be higher, thereby encouraging exploration of that state.

By using the exploration objective in Eq. (4), the sample complexity of the algorithm adapts to the intrinsic complexity of the MDP, which is captured by the pushforward coverability C_{push}, which we recall below.

Definition 3.3 (Pushforward coverability (Xie and Jiang, 2021))**.**

$$C_{\mathsf{push}} = \inf_{\mu \in \Delta(\mathcal{X})} \sup_{(x,a,x') \in \mathcal{X}\times\mathcal{A}\times\mathcal{X}} \left\{ \frac{P(x' \mid x, a)}{\mu(x')} \right\},$$

The pushforward coverability C_{push} is bounded by the intrinsic complexity of the MDP. This is a general notion that can be applied to various structured MDPs. For example, tabular MDPs with state space $\mathcal{X}$ have $C_{\mathsf{push}} \leq |\mathcal{X}|$, block MDPs with latent space $\mathcal{S}$ have $C_{\mathsf{push}} \leq |\mathcal{S}|$, and low-rank MDPs of dimension d have $C_{\mathsf{push}} \leq d$.

Since the reward construction requires the knowledge of the transition, which is unknown in general, we can adopt the estimate the weight function via contrastive learning (Amortila et al., 2024), which we present in Algorithm 4. The weight estimation can be solved efficiently via a convex program (Nguyen et al., 2010); we refer to Katdare et al. (2023) for a practical implementation for estimating the weight. Conceptually, we estimate $\widehat{w}^k(x \mid x', a') \approx w^k(x \mid x', a')$, and use $\widehat{b}^k(x \mid x', a') := \frac{2}{1-\gamma}\widehat{w}^k(x \mid x', a')$ as the exploration bonus. To perform weight function estimation, we assume access to a weight function class $\overline{\mathcal{W}}$, with $\overline{\mathcal{W}} \subseteq (\mathcal{X} \times \mathcal{A} \times \mathcal{X} \to \mathbb{R}_+)$ that is capable of representing the weight function w^k in Eq. (4). We provide this assumption as follows:

Assumption 3.4 (Weight function realizability (Amortila et al., 2024))**.** For all $\pi \in \Pi$:

$$w\left(x' \mid x, a\right) := \frac{P\left(x' \mid x, a\right)}{d^\pi\left(x'\right)} \in \overline{\mathcal{W}}.$$

At a high level, the estimated weight $\widehat{w}$ is closed to the true weight w under the Hellinger distance: $D^2_{\mathbf{H},\nu}\left(w, w'\right) := \mathbb{E}[(\sqrt{\widehat{w}} - \sqrt{w})^2] \lesssim \frac{\|w\|_\infty \log\left(|\overline{\mathcal{W}}|\delta^{-1}\right)}{n}$, where n is number of sample in the estimation procedure. Next, we present our main result for the pushforward coverability.

Theorem 3.5. *For any $\varepsilon, \delta \in (0,1)$, suppose Assumption 3.4 is satisfied. Algorithm 3 is instantiated with the following parameters: $K = O(\frac{C_{\mathsf{push}}\iota}{(1-\gamma)^2\varepsilon})$, $T = O(\frac{C^2_{\mathsf{push}}\iota^2}{(1-\gamma)^{15}\varepsilon^4})$, $N = O(\frac{C^2_{\mathsf{push}}|\mathcal{A}|^2\iota^2}{(1-\gamma)^{18}\alpha^2\epsilon^6})$, $M = m\left(\frac{(1-\gamma)^9\alpha\epsilon^3}{C_{\mathsf{push}}|\mathcal{A}|}, \frac{\delta}{KTN_{\mathsf{weight}}}\right)$, $N_{\mathsf{weight}} = O(\frac{C_{\mathsf{push}}|\mathcal{A}|\log(|\overline{\mathcal{W}}|\delta^{-1})\iota}{(1-\gamma)^{11}\varepsilon^3})$, where $\iota = \log(\frac{C_{\mathsf{push}}}{(1-\gamma)^2\varepsilon})$. Then Algorithm 3 produces policy π, such that with probability at least $1-\delta$,*

$$V^* - V^\pi \leq \frac{C_{\mathsf{push}}\iota}{(1-\gamma)^6\varepsilon}\mathcal{E}(\Pi, \Pi) + \varepsilon.$$

Sample Complexity: *If* $m(\varepsilon, \delta) = \frac{\log |\mathcal{W}|}{\varepsilon^2} \log \frac{1}{\delta}$ *for some measure of weak learning complexity* $|\mathcal{W}|$*, then the algorithm uses at most* $K(TNM + N_{\mathsf{weight}}) \leq \widetilde{O}\left(\frac{C_{\mathsf{push}}^7 |\mathcal{A}|^4 \log(|\mathcal{W}|)}{(1-\gamma)^{53}\alpha^4\varepsilon^{17}} + \frac{C_{\mathsf{push}}^2 |\mathcal{A}| \log(|\overline{\mathcal{W}}|)}{(1-\gamma)^{13}\varepsilon^4}\right)$ *episodes.*

The large polynomial dependence on $(1-\gamma)^{-1}$ arises from the fact that our algorithm operates within the policy learning regime; we do not impose any structural assumption on the Q or V function. Recent work on policy learning in the episodic setting also experienced similar scaling challenges, e.g., $\widetilde{O}(\frac{C_{\mathsf{push}}^4 |S|^{24} |\mathcal{A}|^{30} H^{39}}{\varepsilon^{18}})$ (Krishnamurthy et al., 2025, Theorem 4), where H is the horizon.

Our results offer an advantage over existing boosting framework. Specifically, Brukhim et al. (2022) established an optimality gap of $V^* - V^\pi \leq \frac{C_\infty \mathcal{E}(\mathbb{\Pi}, \Pi)}{1-\gamma} + \varepsilon$, this bound can become vacuous in the regime of unbounded C_∞ , while our bound still gives a meaningful result. In addition, the number of iterations required for their algorithm to converge depends on C_∞, as does the number of weak learners aggregated in the final policy.

Algorithm 3 is oracle efficient: assuming access to an efficient weak learning oracle, and an efficient optimizer for the weight function, our result implies that both computational and statistical complexities scale as $\mathrm{poly}(C_{\mathsf{push}}, (1-\gamma)^{-1}, |\mathcal{A}|, \alpha^{-1}, \varepsilon^{-1}, \log |\mathcal{W}|, \log(|\overline{\mathcal{W}}|), \log \delta^{-1})$:

- *Statistical efficiency.* Theorem 3.5 shows that the sample complexity scales polynomially with the intrinsic complexity of the MDP, as captured by the pushforward coverability C_{push}, where it is bounded in many structured MDPs.

- *Computational efficiency.* Algorithm 3 requires a total of $\widetilde{O}\left(\frac{C_{\mathsf{push}}^5 |\mathcal{A}|^2 \log(|\mathcal{W}|)}{(1-\gamma)^{30}\alpha^2\varepsilon^{10}}\right)$ calls to the weak learning oracle, and K calls to an optimization oracle for the weight function (in Algorithm 4).

Proof sketch. The core of our analysis involves a novel construction of an extended MDP and a truncated policy class, building upon the framework of Amortila et al. (2024). Prior work utilizes this construction for the reward-free setting, optimizing only for the exploration bonuses. In contrast, our algorithm optimizes both the true reward and the exploration bonuses. This creates a non-trivial coupling, and poses a significant analytical challenge, particularly in the discounted MDP setting where the episodic construction of Amortila et al. (2024) does not hold. To resolve this, we propose a redesigned construction of the extended MDP and a new termination condition of the truncated policy class. This construction allows us to partition the state space into well-covered and poorly covered regions, effectively managing the latter via our exploration bonus and the Performance Difference Lemma. Consequently, we can bound the distribution mismatch in the well-covered regions by $\left\| \frac{d^{\pi^*}}{(1-\gamma)d^{p^k}} \right\|_\infty$ with d^{p^k} as our exploration distribution induced by the policy cover. Crucially, this yields a gradient dominance property that is independent of the global distribution mismatch, enabling the direct application of non-convex Frank-Wolfe convergence guarantees of Brukhim et al. (2022). Furthermore, the pushforward analysis requires exploration bonuses to reach a $\frac{1}{1-\gamma}$ threshold on poorly explored states: a condition that typically relies on a known lower bound on MDP transition dynamics. We relax this requirement through an adaptive tuning technique: by treating the transition threshold as a variable within the truncated policy condition and optimizing it in the final bound, we eliminate the need for prior knowledge of the lower bound on the transition dynamics. Importantly, these constructions — the extended MDP, truncated policies and adaptive threshold — serve purely as analytical tools; the underlying algorithms remain agnostic to them, preserving its practical simplicity.

Algorithm 3 Boosting with push forward coverability

input: total iterations T, number of weak learners N, total sample episodes M, policy class Π.
Initialize policy π^1 arbitrarily, $p^1 = \{\pi^1\}$.
for $k = 1, \cdots, K$ **do**
    Run Algorithm 4 to estimate $\widehat{w}^k(x \mid x', a')$ with cover p^k and define bonus $\widehat{b}^k(x \mid x', a')$.
    Run RL boosting (Algorithm 1) with reward $r + \widehat{b}^k$, initial distribution d^{p^k} to obtain π^k.
    Update the policy cover $p^{k+1} = \mathrm{Unif}(\pi_1, \ldots, \pi_k)$.
end for
return $\pi := \mathrm{argmax}_{\pi \in \{\pi^1, \ldots, \pi^K\}} V^\pi$.

Algorithm 4 Weight estimation

function SAMPLECOVER
    Sample $x_0 \sim \mu$, continue to execute π, at any step h with (x_{h-1}, a_{h-1}, x_h) terminate with probability $1 - \gamma$.
    Play action $a \sim \mathrm{Unif}(\mathcal{A})$, and observe (x_h, a, x_{h+1}).
    return (x_h, a, x_{h+1}).
end function
function SAMPLESTATEACTION
    Sample $x_0 \sim \mu$, continue to execute π, at any step h with (x_{h-1}, a_{h-1}, x_h) terminate with probablity $1 - \gamma$.
    return (x_{h-1}, a_{h-1}, x_h).
end function
input: policy cover p.
Initialize $\mathcal{D}_1 = \mathcal{D}_2 = \varnothing$.
For each $j \in [n]$, draw $\pi \sim p$ and sample $(x'_j, a'_j, x_j) \sim \pi$ with `SampleCover`. Add $(x'_j, a'_j, x_j) \sim \pi$ to both $\mathcal{D}_1$ and $\mathcal{D}_2$.
for $i < k$ **do**
    Draw $\{(x'_j, a'_j, x_j)\}_{j \in [n]}$ by drawing $\pi \sim p$ and $(x'_j, a'_j, x_j) \sim \pi$ with `SampleCover`.
    Draw $\{(\tilde{x}'_j, \tilde{a}'_j, \tilde{x}_j)\}_{j \in [n]}$ by sampling $\left(\tilde{x}'^j, \tilde{a}'^j, \tilde{x}^j\right) \sim \pi^i$ with `SampleStateAction`.
    Add $\{(x'_j, a'_j, x_j)\}_{j \in [n]}$ to $\mathcal{D}_1$ and add $\{(x'_j, a'_j, \tilde{x}_j)\}_{j \in [n]}$ to $\mathcal{D}_2$.
end for
Set $\widehat{w} := \arg\max_{w \in \overline{\mathcal{W}}} \widehat{\mathbb{E}}_{\mathcal{D}_1}\left[\log\left(w\left(x_h \mid x_{h-1}, a_{h-1}\right)\right)\right] - k \cdot \widehat{\mathbb{E}}_{\mathcal{D}_2}\left[w\left(x_h \mid x_{h-1}, a_{h-1}\right)\right]$.

4 Conclusions

In this work, we propose a boosting-based approach for reinforcement learning in structured MDPs. Rather than relying on distribution mismatch assumptions commonly used in prior work, we use an intrinsic notion of coverability, which is bounded in a wide class of structured MDPs. Under this condition, we show that the proposed boosting method is both statistically efficient and oracle efficient, with complexity scaling polynomial in the relevant problem parameters.

Several promising directions remain for future research. Our approach leverages policy covers to guide exploration in a principled manner, which plays a crucial role in relaxing dependence on distribution mismatch. It would be interesting to investigate alternative mechanisms for strategic exploration that could further broaden the applicability of the framework while maintaining computational and statistical efficiency. The guarantee provided in this paper assumes access to a *multiplicative* weak-learner, further development for other types of weak learners is another natural direction. While our algorithms have complexities that depend only polynomially on all relevant parameters, the large polynomial dependence on $1 - \gamma$ is undesirable, and we leave open whether this and other dependencies can be improved with alternative approaches. Furthermore, our results show that, given access to a weak learner, efficient boosting is achievable under the notion of pushforward coverability. Whether such a coverability condition is also necessary for efficient boosting remains an open question for future work.

References

Alekh Agarwal, Mikael Henaff, Sham Kakade, and Wen Sun. Pc-pg: Policy cover directed exploration for provable policy gradient learning. *Advances in neural information processing systems*, 33: 13399–13412, 2020.

Naman Agarwal, Brian Bullins, and Karan Singh. Variance-reduced conservative policy iteration. In Shipra Agrawal and Francesco Orabona, editors, *Proceedings of The 34th International Conference on Algorithmic Learning Theory*, volume 201 of *Proceedings of Machine Learning Research*, pages 3–33. PMLR, 20 Feb–23 Feb 2023. URL https://proceedings.mlr.press/v201/agarwal23a.html.

Philip Amortila, Dylan J Foster, and Akshay Krishnamurthy. Scalable online exploration via coverability. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 1392–1455. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/amortila24a.html.

Ronen I Brafman and Moshe Tennenholtz. R-max-a general polynomial time algorithm for near-optimal reinforcement learning. *Journal of Machine Learning Research*, 3(Oct):213–231, 2002.

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym, 2016.

Nataly Brukhim, Elad Hazan, and Karan Singh. A boosting approach to reinforcement learning. *Advances in Neural Information Processing Systems*, 35:33806–33817, 2022.

Kai-Wei Chang, Akshay Krishnamurthy, Alekh Agarwal, Hal Daumé III, and John Langford. Learning to search better than your teacher. In *International Conference on Machine Learning*, pages 2058–2066. PMLR, 2015.

Jinglin Chen and Nan Jiang. Information-theoretic considerations in batch reinforcement learning. In *International conference on machine learning*, pages 1042–1051. PMLR, 2019.

Ching-An Cheng, Andrey Kolobov, and Alekh Agarwal. Policy improvement via imitation of multiple oracles. *Advances in Neural Information Processing Systems*, 33:5587–5598, 2020.

Simon Du, Akshay Krishnamurthy, Nan Jiang, Alekh Agarwal, Miroslav Dudik, and John Langford. Provably efficient rl with rich observations via latent state decoding. In *International Conference on Machine Learning*, pages 1665–1674. PMLR, 2019.

John Duchi, Shai Shalev-Shwartz, Yoram Singer, and Tushar Chandra. Efficient projections onto the l 1-ball for learning in high dimensions. In *Proceedings of the 25th international conference on Machine learning*, pages 272–279, 2008.

Y. Freund. Boosting a weak learning algorithm by majority. *Information and Computation*, 121 (2):256–285, 1995. ISSN 0890-5401. doi: https://doi.org/10.1006/inco.1995.1136. URL https://www.sciencedirect.com/science/article/pii/S0890540185711364.

Udaya Ghai and Karan Singh. Sample-optimal agnostic boosting with unlabeled data. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=hcLeFe7idT.

Elad Hazan and Karan Singh. Boosting for online convex optimization. In *International Conference on Machine Learning*, pages 4140–4149. PMLR, 2021.

Marcel Hussing, Michael Kearns, Aaron Roth, Sikata B Sengupta, and Jessica Sorrell. Oracle-efficient reinforcement learning for max value ensembles. *Advances in Neural Information Processing Systems*, 37:117657–117681, 2024.

Nan Jiang, Akshay Krishnamurthy, Alekh Agarwal, John Langford, and Robert E Schapire. Contextual decision processes with low bellman rank are pac-learnable. In *International Conference on Machine Learning*, pages 1704–1713. PMLR, 2017.

Chi Jin, Zeyuan Allen-Zhu, Sebastien Bubeck, and Michael I Jordan. Is q-learning provably efficient? *Advances in neural information processing systems*, 31, 2018.

Chi Jin, Zhuoran Yang, Zhaoran Wang, and Michael I Jordan. Provably efficient reinforcement learning with linear function approximation. In *Conference on learning theory*, pages 2137–2143. PMLR, 2020.

Sham Kakade and John Langford. Approximately optimal approximate reinforcement learning. In *Proceedings of the nineteenth international conference on machine learning*, pages 267–274, 2002.

Daniel Kane, Sihan Liu, Shachar Lovett, and Gaurav Mahajan. Computational-statistical gap in reinforcement learning. In *Conference on Learning Theory*, pages 1282–1302. PMLR, 2022.

Pulkit Katdare, Nan Jiang, and Katherine Rose Driggs-Campbell. Marginalized importance sampling for off-environment policy evaluation. In *Conference on Robot Learning*, pages 3778–3788. PMLR, 2023.

Michael Kearns and Satinder Singh. Near-optimal reinforcement learning in polynomial time. *Machine learning*, 49:209–232, 2002.

Akshay Krishnamurthy, Alekh Agarwal, and John Langford. Pac reinforcement learning with rich observations. *Advances in Neural Information Processing Systems*, 29, 2016.

Akshay Krishnamurthy, Gene Li, and Ayush Sekhari. The role of environment access in agnostic reinforcement learning. *arXiv preprint arXiv:2504.05405*, 2025.

Andrey Kurenkov, Ajay Mandlekar, Roberto Martin-Martin, Silvio Savarese, and Animesh Garg. Ac-teach: A bayesian actor-critic method for policy learning with an ensemble of suboptimal teachers. *arXiv preprint arXiv:1909.04121*, 2019.

Guohao Li, Matthias Mueller, Vincent Casser, Neil Smith, Dominik L Michels, and Bernard Ghanem. Oil: Observational imitation learning. *arXiv preprint arXiv:1803.01129*, 2018.

Xuefeng Liu, Takuma Yoneda, Chaoqi Wang, Matthew Walter, and Yuxin Chen. Active policy improvement from multiple black-box oracles. In *International Conference on Machine Learning*, pages 22320–22337. PMLR, 2023.

Xuefeng Liu, Takuma Yoneda, Rick Stevens, Matthew Walter, and Yuxin Chen. Blending imitation and reinforcement learning for robust policy improvement. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=eJ0dzPJq1F.

Zak Mhammedi, Adam Block, Dylan J Foster, and Alexander Rakhlin. Efficient model-free exploration in low-rank mdps. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 66782–66817. Curran Associates, Inc., 2023a. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/d2dc4d6c7b102d05f111c02a32e7c6bc-Paper-Conference.pdf.

Zakaria Mhammedi, Dylan J Foster, and Alexander Rakhlin. Representation learning with multi-step inverse kinematics: An efficient and optimal approach to rich-observation rl. In *International Conference on Machine Learning*, pages 24659–24700. PMLR, 2023b.

Dipendra Misra, Mikael Henaff, Akshay Krishnamurthy, and John Langford. Kinematic state abstraction and provably efficient rich-observation reinforcement learning. In *International conference on machine learning*, pages 6961–6971. PMLR, 2020.

XuanLong Nguyen, Martin J Wainwright, and Michael I Jordan. Estimating divergence functionals and the likelihood ratio by convex risk minimization. *IEEE Transactions on Information Theory*, 56(11):5847–5861, 2010.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Robert E Schapire. The strength of weak learnability. *Machine learning*, 5(2):197–227, 1990.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015.

Yanjie Song, Ponnuthurai Nagaratnam Suganthan, Witold Pedrycz, Junwei Ou, Yongming He, Yingwu Chen, and Yutong Wu. Ensemble reinforcement learning: A survey. *Applied Soft Computing*, page 110975, 2023.

Wen Sun, Arun Venkatraman, Geoffrey J Gordon, Byron Boots, and J Andrew Bagnell. Deeply aggrevated: Differentiable imitation learning for sequential prediction. In *International conference on machine learning*, pages 3309–3318. PMLR, 2017.

Tengyang Xie and Nan Jiang. Batch value-function approximation with only realizability. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 11404–11413. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/xie21d.html.

Tengyang Xie, Dylan J Foster, Yu Bai, Nan Jiang, and Sham M Kakade. The role of coverage in online reinforcement learning. In *ICLR*, 2023.

A Notation: List of Symbols

Markov Decision Process

$\mathcal{X}$	Observation space
$\mathcal{S}$	Latent state space
$\mathcal{A}$	Action space
$\Delta_{\mathcal{A}}$	Probability simplex over actions
$Q^\pi(s,a)$	Q function
$V^\pi(s)$	Value function
$d^\pi(x)$	state distribution
γ	Discount factor
$\mathcal{E}_\mu(\mathbb{\Pi}, \Pi)$	Policy completeness
C_∞	Distribution mismatch
$\widetilde{C}_\infty$	L_∞ coefficient
C_{push}	pushforward coverability

Weak Learning and Boosting

α	Weak learning parameter
$\Gamma[\cdot]$	Policy projection
Π	Policy class
$\mathbb{\Pi}$	Policy-Tree class (w.r.t Π)

B Related work

The boosting approach is a prominent strategy for provably reducing the task of strong learning to that of weak learning of classifiers that are simpler to compute but only slightly better than random guessing. While boosting is a well-established technique in supervised learning with strong theoretical guarantees (Schapire, 1990; Freund, 1995), its adaptation to reinforcement learning (RL) has emerged only recently (Brukhim et al., 2022; Agarwal et al., 2023; Ghai and Singh, 2025). These boosting frameworks can achieve global convergence guarantees under the assumption of access to a multiplicative weak learner. However, their analysis crucially depends on occupancy mismatch terms relative to the optimal policy. These terms can grow unbounded unless the policy class provides sufficient state-space coverage, highlighting a key limitation of their approach.

The issue of occupancy mismatch suggests that strategic exploration is necessary to ensure that the policy class adequately covers the state space. A large body of work in value-based RL has studied the optimism principle, which introduces an exploration bonus to encourage visiting underexplored states. Such bonuses promote exploration by increasing the estimated value of rarely visited states, as in Upper Confidence Bound (UCB) methods (Jin et al., 2018, 2020). An alternative line of research focuses on reward-free exploration, where the goal is to construct a policy cover that ensures sufficient reachability across states, independent of specific reward functions. Recent works have explored various notions of coverage and their implications for efficient exploration in RL (Misra et al., 2020; Du et al., 2019; Mhammedi et al., 2023b,a). More recently, Amortila et al. (2024) proposed a generalized exploration objective that unifies prior schemes while enabling intrinsic complexity control, efficient planning, and scalable exploration, and we rely heavily on this objective in our work.

Beyond boosting, more general ensembling methods are an active area of research in RL. Given a pre-trained collection of constituent policies and a squared error regression oracle for the relevant value function class, Hussing et al. (2024) shows how to ensemble the constituent policies to be competitive with a max-following policy (the policy which, in state s, acts consistently with whichever constituent

policy has the highest value in state s). Cheng et al. (2020); Liu et al. (2023, 2024) use policy gradient methods to compete with a similar, but incomparable benchmark class of max-following policies with one-step look-ahead. Notably, these works all consider the question of ensembling non-adaptively; all constituent policies are trained up front, in comparison to boosting, in which the performance of the current policy affects the selection of the next policy to be included in the ensemble. We refer the reader to Song et al. (2023) for a survey of ensembling techniques.

On the more empirical side, (Li et al., 2018; Kurenkov et al., 2019) study the problem of imitation learning from multiple policies and Sun et al. (2017); Schulman et al. (2015); Chang et al. (2015) give algorithms for iterative policy improvement, but these methods are heuristic and lack strong provable guarantees of performance.

C Extended MDP and truncated policy class setup

Extended MDP $\overline{\mathcal{M}}$. We define an extended MDP $\overline{\mathcal{M}}$ by extending the state and action spaces such that $\overline{\mathcal{A}} = \mathcal{A} \cup \{\mathfrak{t}\}$ and $\overline{\mathcal{X}} = \mathcal{X} \cup \{\mathfrak{t}\}$. The extended MDP $\overline{\mathcal{M}}$ matches the original MDP $\mathcal{M}$, with the follow additions:

- **Termination**: Taking the terminal action $\mathfrak{t}$ from any state $x \in \mathcal{X}$ results in a deterministic transition to $\mathfrak{t}$, and $\mathfrak{t}$ is a self-looping terminal state.
- **Absorbing State**: The terminal state $\mathfrak{t}$ is a self-looping absorbing state, where $P(\mathfrak{t} \mid \mathfrak{t}, a) = 1$ for all $a \in \overline{\mathcal{A}}$.
- **Reward Structure**: Entering the terminal state via action $\mathfrak{t}$ yields an immediate reward $r(x, \mathfrak{t}) = \frac{1}{1-\gamma}$. Once in the terminal state, all subsequent rewards are zero.

Truncated policy class $\overline{\Pi}$. Let $\mathcal{B} \subseteq \mathcal{X}$ denote a "well-explored" set (to be formally defined based on specific coverability notions). For every policy π in the original class Π, we define a corresponding truncated policy $\pi' \in \overline{\Pi}$ as:

$$\overline{\pi}(x) = \begin{cases} \pi(x), & x \in \mathcal{B} \\ \mathfrak{t}, & \text{otherwise}, \end{cases}$$

We note that the extended MDP and truncated policy class are defined only for the purpose of analysis. The algorithm itself does not require knowledge of these extensions, nor does it explicitly utilize the terminal state $\mathfrak{t}$ or the set $\mathcal{B}$ during execution.

Notation. We use the bar notation (e.g., $\bar{f}$) to denote quantities related to the extended MDP and the truncated policy class. Specifically, for each episode $k \in [K]$ let $\overline{\Pi}_k$ denote the truncated policy class contructed from Π using $\mathcal{B}^k$. For any $\pi \in \Pi$ we let $\overline{\pi} \in \overline{\Pi}_k$ be the correspond truncated policy. Finally, we denote the state and state action value functions within the extended MDP as $\overline{V}(x)$ and $\overline{Q}(x, a)$, respectively.

We provide the following claim for *extended* MDP with a bonus function.

Claim 1. *Let $\overline{V}_b^{\overline{\pi}}$ denote the value function of the truncated policy $\overline{\pi}$ of π on the extended MDP with a positive bonus function b. Then, for any $x \in \mathcal{X}$, we have*

$$V^{\pi}(x) \leq \overline{V}_b^{\overline{\pi}}(x),$$

Proof. **In-set transitions:** As long as the state x remains within $\mathcal{B}$, the truncated policy $\bar{\pi}$ follows the original policy π. In this regime, the reward in the extended MDP is greater than the original reward because the bonus in positive.

Out-of-set transitions: Upon encountering a state $x \notin \mathcal{B}$, the policy $\bar{\pi}$ takes the termination action $\mathfrak{t}$. This yields an immediate value of $\frac{1}{1-\gamma}$. Since the maximum possible value in the original MDP is bounded by $\frac{1}{1-\gamma}$, the value obtained by terminating is at least as large as the value of continuing with the original policy π.

Combining these cases, the value function in the extended MDP serves as an upper bound for the original value function. □

D Proof of Section 3.1

Theorem 3.2. *For any $\varepsilon, \delta \in (0,1)$, suppose Assumption 3.1 is satisfied, Algorithm 2 is instantiated with the following parameters: $K = O(\frac{\widetilde{C}_\infty \iota}{(1-\gamma)\varepsilon})$,$T = O(\frac{\widetilde{C}_\infty^2 \iota^2}{(1-\gamma)^8 \varepsilon^3})$, $N = O(\frac{\widetilde{C}_\infty^2 |\mathcal{A}|^2 \iota^2}{(1-\gamma)^{10}\alpha^2\varepsilon^4})$, $M = m\left(\frac{(1-\gamma)^5\alpha\varepsilon^2}{\widetilde{C}_\infty|\mathcal{A}|}, \frac{\delta}{KTN}\right)$, where $\iota = \log(\frac{\widetilde{C}_\infty}{(1-\gamma)\varepsilon})$. Then Algorithm 2 produces policy π, such that with probability at least $1-\delta$,*

$$V^* - V^\pi \le \frac{\widetilde{C}_\infty \iota}{(1-\gamma)^5 \varepsilon} \mathcal{E}(\mathbb{\Pi}, \Pi) + \varepsilon.$$

Sample Complexity: *If $m(\varepsilon, \delta) = \frac{\log|\mathcal{W}|}{\varepsilon^2}\log\frac{1}{\delta}$ for some measure of weak learning complexity $|\mathcal{W}|$, then the algorithm uses at most $KTNM \le \widetilde{O}\left(\frac{\widetilde{C}_\infty^7 |\mathcal{A}|^4 \log|\mathcal{W}|}{(1-\gamma)^{29}\alpha^4\varepsilon^{12}}\right)$ episodes.*

Proof of Theorem 3.2. For any episode $k \in [K]$, we define $\mathcal{B}^k := \{x, a : \frac{\widetilde{C}_\infty \mu(x,a)}{d^{p^k}(x,a)} \le k-1\}$ to be the set of a well-explored states relative to the reference distribution μ. We say $x \in \mathcal{B}^k$ if $(x,a) \in \mathcal{B}^k, \forall a \in \mathcal{A}$.

Let $\pi = \text{argmin}_{\pi \in \{\pi^1, \dots, \pi^K\}} V^{\pi^k}$ be the output of the Algorithm 2, we have the following:

$$V_{d_0}^{\pi^*} - V_{d_0}^{\pi} \le \frac{1}{K}\sum_{k=1}^K V_{d_0}^{\pi^*} - V_{d_0}^{\pi^k} \le \frac{1}{K(1-\gamma)}\sum_{k=1}^K V_{d^{p^k}}^{\pi^*} - V_{d^{p^k}}^{\pi^k}. \tag{5}$$

In the last inequality, we use the following properties to switch to using d^{p^k} as our initial distribution in each round, for which we pay a factor of $\frac{1}{1-\gamma}$.

$$\begin{aligned} V_{d_0}^{\pi^*} - V_{d_0}^{\pi} &= \mathbb{E}_{x\sim d_0}\left[V^{\pi^*}(x) - V^\pi(x)\right] \\ &= \mathbb{E}_{x\sim d^{p^k}}\left[\frac{d_0(x)}{d^{p^k}(x)}(V^{\pi^*}(x) - V^\pi(x))\right] \\ &\le \frac{1}{1-\gamma}\left(V_{d^{p^k}}^{\pi^*} - V_{d^{p^k}}^{\pi}\right), \end{aligned}$$

where the last inequality follows from the fact that, $V^*(x) - V^\pi(x) \ge 0$ and $d^{p^k}(x) \ge (1-\gamma)d_0(x), \forall x \in \mathcal{X}$.

To reduce notational clutter throughout the remainder of the proof, we assume d^{p^k} is the initial distribution and omit the subscript d^{p^k} when referring to the value functions, e.g., we simply write $V_{d^{p^k}}^{\pi}$ as V^π.

We proceed to bound the RHS of Eq. (5) by decomposing the summation into the following three terms:

$$\frac{1}{K}\sum_{k=1}^K V^{\pi^*} - V^{\pi^k} = \underbrace{\frac{1}{K}\sum_{k=1}^K V^{\pi^*} - \overline{V}_{b^k}^{\overline{\pi^*}}}_{I_1} + \underbrace{\frac{1}{K}\sum_{k=1}^K \overline{V}_{b^k}^{\overline{\pi^*}} - \overline{V}_{b^k}^{\pi^k}}_{I_2} + \underbrace{\frac{1}{K}\sum_{k=1}^K \overline{V}_{b^k}^{\pi^k} - V^{\pi^k}}_{I_3},$$

where $\overline{\pi^*}$ be the truncated policy counterpart of π^*. First, we have $I_1 \le 0$ follows by Claim 1.

For I_3, we have

$$\begin{aligned}\sum_{k=1}^{K} \overline{V}_{b^k}^{\pi^k} - V^{\pi^k} &= \sum_{k=1}^{K} V_{b^k}^{\pi^k} - V^{\pi^k} \\ &= 2\sum_{k=1}^{K} \mathbb{E}^{\pi^k}\left[\frac{\widetilde{C}_\infty \mu(x,a)}{\sum_{i<k} d^{\pi^i}(x,a) + \widetilde{C}_\infty \mu(x,a)}\right] \\ &= 2\widetilde{C}_\infty \sum_{x\in\mathcal{X}, a\in\mathcal{A}} \sum_{k=1}^{K} \mu(x,a) \frac{d^{\pi^k}(x,a)}{\sum_{i<k} d^{\pi^i}(x,a) + \widetilde{C}_\infty \mu(x,a)}\end{aligned}$$

Since $\sup_{\pi\in\Pi} d^\pi(x,a) \le \widetilde{C}_\infty \mu(x,a)$ for all $(x,a) \in \mathcal{X}\times\mathcal{A}$, Lemma L.3 implies that

$$\sum_{x\in\mathcal{X}, a\in\mathcal{A}} \sum_{k=1}^{K} \mu(x,a) \frac{d^{\pi^k}(x,a)}{\sum_{i<k} d^{\pi^i}(x,a) + \widetilde{C}_\infty \mu(x,a)} \le 2\log(2K)$$

To bound I_2, we proceed to bridge the *extended* bonus MDP to the result of the bonus MDP in order to apply Lemma L.10 for non-convex Frank-Wolfe optimization in Appendix L.1, which requires the guarantee on the gradient domination which we provide as follows:

Lemma D.1. *For any $k \in [K]$, let $\kappa = \frac{k-1}{1-\gamma}, \tau = \frac{2(k-1)}{(1-\gamma)^3}\mathcal{E}(\mathbb{\Pi}, \Pi)$. Let $\overline{\pi} \in \overline{\Pi}_k$ be a policy in the truncated class, let $\widehat{\Pi} \coloneqq \pi \in \mathcal{T}(\Pi, b^k)$ be the transformed policy class, and $\widehat{\mathbb{\Pi}} \coloneqq \mathcal{T}(\mathbb{\Pi}, b^k)$ be the policy tree constructed from $\widehat{\Pi}$. Consider the value function on the extended bonus MDP under the initial distribution d^{p^k}, we have $\overline{V}_{b^k}^{\overline{\pi}}$ is $(\kappa, \tau, \mathbb{\Pi}, \Pi)$-gradient dominated. That is, for any $\pi \in \mathbb{\Pi}$,*

$$\max_{\overline{\pi}\in\overline{\Pi}_k} \overline{V}_{b^k}^{\overline{\pi}} - \overline{V}_{b^k}^{\pi} \le \frac{k-1}{1-\gamma} \max_{\pi'\in\Pi} (\nabla \overline{V}_{b^k}^{\pi})^\top (\pi' - \pi) + \frac{2(k-1)}{(1-\gamma)^3}\mathcal{E}(\mathbb{\Pi}, \Pi).$$

Since $\overline{Q}_{b^k} = Q_{b^k}, \forall x \in \Pi$ since all policy in Π do not pick $\mathfrak{t}$, and by Lemma L.7, the Internal Boost guarantee that

$$\max_{\pi\in\Pi} \mathbb{E}_{(s,\overline{Q})\sim\mathcal{D}_t}\left[\overline{Q}^\top \pi(s)\right] - \mathbb{E}_{(s,\overline{Q})\sim\mathcal{D}_t}\left[\overline{Q}^\top \pi'_t(s)\right] \le \frac{2}{(1-\gamma)^2\alpha}(\varepsilon_{\mathsf{w}} + 2/\sqrt{N}) \tag{6}$$

By Eq. (6), Lemma D.1, and $\overline{V}_{b^k}^{\pi}$ is $\frac{4\gamma}{(1-\gamma)^3}$-smoothness, we can appeal to Lemma L.10, put everything together, we have

$$V_{d_0}^{\pi^*} - V_{d_0}^{\pi^{k'}} \le \frac{2K}{(1-\gamma)^4}\mathcal{E}(\Pi, \mathbb{\Pi}) + \frac{8K^2}{(1-\gamma)^6 T} + \frac{4\varepsilon_{\mathsf{w}} K|\mathcal{A}|}{(1-\gamma)^4\alpha} + \frac{8K|\mathcal{A}|}{(1-\gamma)^4\alpha\sqrt{N}} + \frac{1}{K(1-\gamma)}\widetilde{C}_\infty \log(2K),$$

which also concludes the proof.

□

D.1 Support proof of L_∞ coverability

Claim 2. *For any policy $\overline{\pi} \in \overline{\Pi}_k$, and $\pi \in \mathcal{T}(\Pi, \mathcal{B}^k)$, for any $x \in \mathcal{B}^k$, we have*

$$d^{\overline{\pi}}(x) \le \beta^k d^{p^k}(x)$$

for any $x \notin \mathcal{B}^k$, we have

$$\overline{Q}_{b^k}^{\overline{\pi}}(x, \mathfrak{t}) \le \overline{V}_{b^k}^{\pi}(x).$$

Proof of Claim 2. **Part 1**. For any $x \in \mathcal{B}^k$, $\overline{\pi}$ does not pick the terminal action, so we only need to consider the original action set:

$$d^\pi(x) = \sum_{a\in\mathcal{A}} d^\pi(x,a) \le \sum_{a\in\mathcal{A}} \widetilde{C}_\infty \mu(x,a) \le \sum_{a\in\mathcal{A}} \beta^k d^{p^k}(x,a) \le \beta^k d^{p^k}(x),$$

The first inequality follows by the definition of $\widetilde{C}_\infty$, the second inequality due to $\frac{\widetilde{C}_\infty \mu(x,a)}{d^{p^k}(x,a)} \leq \beta^k, \forall a \in \mathcal{A}$ if $x \in \mathcal{B}^k$.

Part 2. By the definition of the truncated policy, for any $x \notin \mathcal{B}^k$, the policy deterministically selects the terminal action, i.e., $\pi(x) = \mathfrak{t}$. Choosing action $\mathfrak{t}$ yields

$$\overline{Q}^{\pi}_{b^k}(x, \mathfrak{t}) = 1 + \gamma \overline{V}^{\pi}_{b^k}(x)$$

since the transition under $\mathfrak{t}$ deterministically returns to state x. Moreover, because π selects actions uniformly from the set $\{a : (x,a) \notin \mathcal{B}^k\}$, implying $\overline{V}^{\pi}_{b^k}(x) \geq \frac{1}{1-\gamma}$. Therefore,

$$\overline{Q}^{\pi}_{b^k}(x, \mathfrak{t}) - \overline{V}^{\pi}_{b^k}(x) = 1 - (1-\gamma)\overline{V}^{\pi}_{b^k}(x) \leq 0$$

This completes the proof. □

Proof of Lemma D.1. Let $\overline{\pi}$ be any truncated policy, by Claim 2, we have $\overline{Q}^{\overline{\pi}}_{b^k}(x, \mathfrak{t}) \leq \overline{V}^{\pi}_{b^k}(x)$ for all $x \notin \mathcal{B}^k$, apply Lemma L.1 yielding

$$\begin{aligned}
\overline{V}^{\overline{\pi}}_{b^k} - \overline{V}^{\pi}_{b^k} &\leq \frac{1}{1-\gamma} \sum_{x,a} d^{\overline{\pi}}(x)\, \overline{\pi}(a \mid x) \left(\overline{Q}^{\pi}_{b^k}(x,a) - \overline{V}^{\pi}_{b^k}(x)\right) \mathbb{1}\{x \in \mathcal{B}^k\} \\
&\leq \frac{k-1}{(1-\gamma)^2} \sum_{x,a} d^{\pi}(x)\, \overline{\pi}(a \mid x) \left(\overline{Q}^{\pi}_{b^k}(x,a) - \overline{V}^{\pi}_{b^k}(x)\right) \mathbb{1}\{x \in \mathcal{B}^k\} \\
&\leq \frac{k-1}{(1-\gamma)^2} \sum_{x} d^{\pi}(x) \left(\max_a \overline{Q}^{\pi}_{b^k}(x,a) - \overline{V}^{\pi}_{b^k}(x)\right) \\
&= \frac{k-1}{(1-\gamma)^2} \max_{\pi_0} \sum_{x,a} d^{\pi}(x)\, \pi_0(a \mid x) \left(\overline{Q}^{\pi}_{b^k}(x,a) - \overline{V}^{\pi}_{b^k}(x)\right) \\
&= \frac{k-1}{1-\gamma} \max_{\pi_0} (\nabla \overline{V}^{\pi}_{b^k})^\top (\pi_0 - \pi),
\end{aligned}$$

where the second inequality follows by $\frac{d^{\overline{\pi}}(x)}{d^{\pi}(x)} \leq \frac{d^{\overline{\pi}}(x)}{d^{p^k}(x)(1-\gamma)} \leq \frac{k-1}{1-\gamma}$.

For any $\pi \in \mathbb{\Pi}$, we decompose

$$\max_{\pi_0} (\nabla \overline{V}^{\pi}_{b^k})^\top (\pi_0 - \pi) \leq \left(\max_{\pi_0} (\nabla \overline{V}^{\pi}_{b^k})^\top \pi_0 - \max_{\pi' \in \Pi} (\nabla \overline{V}^{\pi}_{b^k})^\top \pi'\right) + \max_{\pi' \in \Pi} (\nabla \overline{V}^{\pi}_{b^k})^\top (\pi' - \pi).$$

Finally,

$$\begin{aligned}
\max_{\pi_0} (\nabla \overline{V}^{\pi}_{b^k})^\top \pi_0 - \max_{\pi' \in \Pi} (\nabla \overline{V}^{\pi}_{b^k})^\top \pi' &= \min_{\pi' \in \Pi} \frac{1}{1-\gamma} \mathbb{E}_{x \sim d^{\pi}} \left[\max_a \overline{Q}^{\pi}_{b^k}(x,a) - \overline{Q}^{\pi}_{b^k}(x,\cdot)^\top \pi'\right] \\
&\leq \frac{2}{(1-\gamma)^2}\, \mathcal{E}(\mathbb{\Pi}, \Pi),
\end{aligned}$$

where the additional factor $2/(1-\gamma)$ follows from the bound $r(x,a) + b^k(x,a) \leq 2/(1-\gamma)$ and the definition of $\mathcal{E}(\Pi, \mathbb{\Pi})$. Combining the above inequalities completes the proof. □

E Proof for Section 3.2

For any episode $k \in [K]$, we define a well-explored set $\mathcal{B}^k := \{x : \frac{1}{d^{p^k}(x)} \leq \frac{k-1}{\zeta}\}$, where ζ is a parameter will be tuned later in the subsequent analysis.

Theorem 3.5. *For any $\varepsilon, \delta \in (0,1)$, suppose Assumption 3.4 is satisfied. Algorithm 3 is instantiated with the following parameters: $K = O(\frac{C_{\mathsf{push}}\iota}{(1-\gamma)^2\varepsilon})$, $T = O(\frac{C^2_{\mathsf{push}}\iota^2}{(1-\gamma)^{15}\varepsilon^4})$, $N = O(\frac{C^2_{\mathsf{push}}|\mathcal{A}|^2\iota^2}{(1-\gamma)^{18}\alpha^2\epsilon^6})$,*

$M = m\left(\frac{(1-\gamma)^9\alpha\epsilon^3}{C_{\mathsf{push}}|\mathcal{A}|}, \frac{\delta}{KTNN_{\mathsf{weight}}}\right)$, $N_{\mathsf{weight}} = O(\frac{C_{\mathsf{push}}|\mathcal{A}|\log(|\overline{\mathcal{W}}|\delta^{-1})\iota}{(1-\gamma)^{11}\varepsilon^3})$, *where* $\iota = \log(\frac{C_{\mathsf{push}}}{(1-\gamma)^2\varepsilon})$. *Then Algorithm 3 produces policy* π*, such that with probability at least* $1-\delta$*,*

$$V^* - V^\pi \le \frac{C_{\mathsf{push}}\iota}{(1-\gamma)^6\varepsilon}\mathcal{E}(\mathbb{\Pi}, \Pi) + \varepsilon.$$

Sample Complexity: *If* $m(\varepsilon,\delta) = \frac{\log|\mathcal{W}|}{\varepsilon^2}\log\frac{1}{\delta}$ *for some measure of weak learning complexity* $|\mathcal{W}|$*, then the algorithm uses at most* $K(TNM + N_{\mathsf{weight}}) \le \widetilde{O}\left(\frac{C_{\mathsf{push}}^7|\mathcal{A}|^4\log(|\mathcal{W}|)}{(1-\gamma)^{53}\alpha^4\varepsilon^{17}} + \frac{C_{\mathsf{push}}^2|\mathcal{A}|\log(|\overline{\mathcal{W}}|)}{(1-\gamma)^{13}\varepsilon^4}\right)$ *episodes.*

Proof of Theorem 3.5. We follow a similar argument to the proof of Theorem 3.2, and obtain:

$$\frac{1}{K}\sum_{k=1}^K V^{\pi^*} - V^{\pi^k} = \underbrace{\frac{1}{K}\sum_{k=1}^K V^{\pi^*} - \overline{V}^{\overline{\pi^*}}_{\widehat{b}^k}}_{I_1} + \underbrace{\frac{1}{K}\sum_{k=1}^K \overline{V}^{\overline{\pi^*}}_{\widehat{b}^k} - \overline{V}^{\pi^k}_{\widehat{b}^k}}_{I_2} + \underbrace{\frac{1}{K}\sum_{k=1}^K \overline{V}^{\pi^k}_{\widehat{b}^k} - V^{\pi^k}}_{I_3},$$

where $\overline{\pi^*}$ be the truncated policy counterpart of π^*. We have $I_1 \le 0$ follows by Claim 1.

For I3,

$$\begin{aligned}
\overline{V}^{\pi^k}_{\widehat{b}^k,\beta^k} - V^{\pi^k} &= \mathbb{E}^{\pi^k}\left[\frac{2}{1-\gamma}\hat{w}^k(x \mid x', a)\right] \\
&\le \frac{6}{1-\gamma}\mathbb{E}^{\pi^k}\left[w^k(x \mid x', a)\}\right] + \frac{4}{1-\gamma}\mathbb{E}^{\pi^k}\left[\left(\sqrt{\widehat{w}^k(x \mid x', a')} - \sqrt{w^k(x \mid x', a')}\right)^2\right],
\end{aligned}$$

where the inequality follows from Lemma L.5,

Let $g^i = \mathbb{E}^{\pi^i}\left[\left(\sqrt{\widehat{w}^i(x \mid x', a')} - \sqrt{w^i(x \mid x', a')}\right)^2 \mid a' = \pi^i(x')\right]$, apply Lemma L.6

$$\begin{aligned}
&\sum_{k=1}^K \mathbb{E}^{\pi^k}\left[\left(\sqrt{\widehat{w}^k(x \mid x', a')} - \sqrt{w^k(x \mid x', a')}\right)^2\right] \\
&\le \sqrt{2C_{\mathsf{push}}\log(K)\sum_{k\in[K]}\sum_{i<k}\mathbb{E}^{\pi^i}\left[\left(\mathbb{E}\left[\left(\sqrt{\widehat{w}^i\left(x \mid x', a'\right)} - \sqrt{w^i\left(x \mid x', a'\right)}\right)^2 \mid a' = \pi^t(x')\right]\right)^2\right]} + 4C_{\mathsf{push}} \\
&\le \sqrt{2C_{\mathsf{push}}\log(2K)\sum_{k=1}^K\sum_{i<k}\mathbb{E}^{\pi^i\circ\pi^t}\left[\left(\sqrt{\widehat{w}^i\left(x \mid x', a'\right)} - \sqrt{w^i\left(x \mid x', a'\right)}\right)^4\right]} + 4C_{\mathsf{push}} \\
&\le \sqrt{2C_{\mathsf{push}}|\mathcal{A}|\log(2K)\sum_{k=1}^K\sum_{i<k}\mathbb{E}^{\pi^i\circ\pi_{\mathrm{unif}}}\left[\left(\sqrt{\widehat{w}^i\left(x \mid x', a'\right)} - \sqrt{w^i\left(x \mid x', a'\right)}\right)^4\right]} + 4C_{\mathsf{push}} \\
&\le \sqrt{8C_{\mathsf{push}}|\mathcal{A}|\log(K)\sum_{k=1}^K\sum_{i<k}\mathbb{E}^{\pi^i\circ\pi_{\mathrm{unif}}}\left[\left(\sqrt{\widehat{w}^i\left(x \mid x', a'\right)} - \sqrt{w^i\left(x \mid x', a'\right)}\right)^2\right]} + 4C_{\mathsf{push}} \\
&\le \sqrt{8C_{\mathsf{push}}|\mathcal{A}|K^2\log(K)\varepsilon_{\mathsf{weight}}^2} + 4C_{\mathsf{push}}.
\end{aligned} \tag{7}$$

Next, we apply the following lemma to bound $\sum_{k=1}^K \mathbb{E}^{\pi^k}\left[w^k(x \mid x', a)\right]$,

Lemma E.1. *Let* $\pi^1, \ldots, \pi^K$ *be the set of policy obtained from Algorithm 3, we have that*

$$\sum_{k=1}^K \mathbb{E}^{\pi^k}\left[\frac{P(x' \mid x, a)}{\sum_{i=1}^{k-1} d^{\pi^i}(x') + P(x' \mid x, a)}\right] \le 4C_{\mathsf{push}}\log(2K)$$

Thus, we have the following bound for I_3:

$$\frac{1}{K}\sum_{k=1}^{K}\overline{V}^{\pi^k_{\beta^k}}_{\widehat{b}^k} - V^{\pi^k} \le \frac{24C_{\mathsf{push}}\log(2K)}{(1-\gamma)K} + \frac{4}{1-\gamma}\sqrt{8C_{\mathsf{push}}|\mathcal{A}|\log(K)\varepsilon^2_{\mathsf{weight}}} + \frac{16C_{\mathsf{push}}}{(1-\gamma)K}$$

For I2, similar to Theorem 3.2, we apply Lemma L.10, which requires a guarantee for the gradient domination, we provide the following lemma:

Lemma E.2. *For any $k \in [K]$, let $\kappa = \frac{k-1}{\zeta(1-\gamma)}, \tau = \frac{2(k-1)}{\zeta(1-\gamma)^3}\mathcal{E}(\mathbb{\Pi}, \Pi) + \frac{4}{(1-\gamma)^2}\sqrt{\frac{(k-1)|\mathcal{A}|\varepsilon^2_{\mathsf{weight}}}{\zeta}} + \frac{\zeta}{(1-\gamma)^2}$. We have $\overline{V}^{\pi_{k-1}}_{\widehat{b}^k}$ is $(\kappa, \tau, \mathbb{\Pi}, \Pi)$-gradient dominated. That is, for any $\pi \in \mathbb{\Pi}$,*

$$\max_{\overline{\pi}\in\overline{\Pi}_k} \overline{V}^{\overline{\pi}}_{\widehat{b}^k} - \overline{V}^{\pi}_{\widehat{b}^k} \le \kappa \max_{\pi\in\Pi}(\nabla\overline{V}^{\pi}_{\widehat{b}^k})^\top(\pi' - \pi) + \tau.$$

Put everything together, we have

$$\begin{aligned} V^{\pi^*}_{d_0} - V^{\pi^{k'}}_{d_0} \le{}& \frac{2K}{(1-\gamma)^4\zeta}\mathcal{E}(\mathbb{\Pi}, \Pi) + \frac{8K^2}{\zeta^2(1-\gamma)^6 T} + \frac{4\varepsilon_{\mathsf{w}}K|\mathcal{A}|}{\zeta(1-\gamma)^4\alpha} + \frac{8K|\mathcal{A}|}{\zeta(1-\gamma)^4\alpha\sqrt{N}} \\ &+ \frac{4}{(1-\gamma)^3}\sqrt{\frac{K|\mathcal{A}|\varepsilon^2_{\mathsf{weight}}}{\zeta}} + \frac{\zeta}{(1-\gamma)^3} + \frac{4}{(1-\gamma)^2}\sqrt{8C_{\mathsf{push}}|\mathcal{A}|\log(2K)\varepsilon^2_{\mathsf{weight}}} \\ &+ \frac{48C_{\mathsf{push}}\log(2K)}{(1-\gamma)^2 K}, \end{aligned}$$

we conclude the proof by choosing $\zeta = (1-\gamma)^3\varepsilon$. □

E.1 Support proof for pushforward coverability

Proof of Lemma E.2. Let $\overline{\pi} \in \overline{\Pi}_k$. By the performance difference lemma (Lemma L.1), we have the following:

$$\begin{aligned} \overline{V}^{\overline{\pi}}_{\widehat{b}^k} - \overline{V}^{\pi}_{\widehat{b}^k} = {}& \frac{1}{1-\gamma}\sum_{x,a}\mathbb{E}_{x\sim d^{\overline{\pi}}}\left[\left(\overline{Q}^{\pi}_{\widehat{b}^k}(x,a) - \overline{V}^{\pi}_{\widehat{b}^k}(x)\right)\mathbb{1}\{x \in \mathcal{B}^k\}\right] \\ &+ \frac{1}{1-\gamma}\mathbb{E}_{x\sim d^{\overline{\pi}}}\left[\left(\overline{Q}^{\pi}_{\widehat{b}^k}(x,a) - \overline{V}^{\pi}_{\widehat{b}^k}(x)\right)\mathbb{1}\{x \notin \mathcal{B}^k, x \neq \mathfrak{t}\}\right], \end{aligned} \tag{8}$$

Due to the construction of the extended MDP, $Q^{\pi}_{\widehat{b}^k}(\mathfrak{t}, a) = 0, \forall a \in \bar{\mathcal{A}}$, which remove term with $x = \mathfrak{t}$. For the first term in Eq. (8):

$$\begin{aligned} &\frac{1}{1-\gamma}\sum_{x,a} d^{\overline{\pi}}(x)\,\overline{\pi}(a \mid x)\left(\overline{Q}^{\pi}_{\widehat{b}^k}(x,a) - \overline{V}^{\pi}_{\widehat{b}^k}(x)\right)\mathbb{1}\{x \in \mathcal{B}\} \\ &\le \frac{k-1}{\zeta(1-\gamma)^2}\sum_{x,a} d^{\pi}(x)\,\overline{\pi}(a \mid x)\left(\overline{Q}^{\pi}_{\widehat{b}^k}(x,a) - \overline{V}^{\pi}_{\widehat{b}^k}(x)\right)\mathbb{1}\{x \in \mathcal{B}\} \\ &\le \frac{k-1}{\zeta(1-\gamma)^2}\sum_{x} d^{\pi}(x)\left(\max_a \overline{Q}^{\pi}_{\widehat{b}^k}(x,a) - \overline{V}^{\pi}_{\widehat{b}^k}(x)\right) \\ &= \frac{k-1}{\zeta(1-\gamma)^2}\max_{\pi_0}\sum_{x,a} d^{\pi}(x)\,\pi_0(a \mid x)\left(\overline{Q}^{\pi}_{\widehat{b}^k}(x,a) - \overline{V}^{\pi}_{\widehat{b}^k}(x)\right) \\ &= \frac{k-1}{\zeta(1-\gamma)}\max_{\pi_0}(\nabla\overline{V}^{\pi}_{\widehat{b}^k})^\top(\pi_0 - \pi) \\ &\le \frac{k-1}{\zeta(1-\gamma)}\max_{\pi'\in\Pi}(\nabla\overline{V}^{\pi}_{b^k})^\top(\pi' - \pi) + \frac{2(k-1)}{\zeta(1-\gamma)^3}\mathcal{E}(\mathbb{\Pi}, \Pi) \end{aligned}$$

where the first inequality follows by $\frac{d^{\overline{\pi}}(x)}{d^{\pi}(x)} \le \frac{d^{\overline{\pi}}(x)}{d^{p^k}(x)(1-\gamma)} \le \frac{k-1}{\zeta(1-\gamma)}$, the last inequality follows a similar argument in Lemma D.1.

To bound second term in Eq. (8), we consider the following:

$$
\begin{aligned}
&\mathbb{E}_{x\sim d^{\overline{\pi}}}\left[\left(\overline{Q}^{\pi}_{\widehat{b}^k}(x,\mathfrak{t})-\overline{V}^{\pi}_{\widehat{b}^k}(x)\right)\mathbb{1}\{x\notin\mathcal{B},x\neq\mathfrak{t}\}\right]\\
&\leq\mathbb{E}_{x\sim d^{\overline{\pi}}}\left[\left(\overline{Q}^{\pi}_{\widehat{b}^k}(x,\mathfrak{t})-\frac{2}{1-\gamma}\mathbb{E}^{\pi}_{x',a'}[\widehat{w}^k(\tilde{x}\mid x',a')\mid\tilde{x}=x]\right)\mathbb{1}\{x\notin\mathcal{B},x\neq\mathfrak{t}\}\right]\\
&=\mathbb{E}_{x\sim d^{\overline{\pi}}}\Bigg[\underbrace{\left(\overline{Q}^{\pi}_{\widehat{b}^k}(x,\mathfrak{t})-\frac{2}{1-\gamma}\mathbb{E}^{\pi}_{x',a'}[w^k(\tilde{x}\mid x',a')\mid\tilde{x}=x]\right)\mathbb{1}\{x\notin\mathcal{B},x\neq\mathfrak{t}\}}_{A_1}\Bigg]\\
&\quad+\frac{2}{1-\gamma}\underbrace{\mathbb{E}_{x\sim d^{\overline{\pi}}}\left[\left(\mathbb{E}^{\pi}_{x',a'}[w^k(\tilde{x}\mid x',a')-\widehat{w}^k(\tilde{x}\mid x',a')\mid\tilde{x}=x]\right)\mathbb{1}\{x\notin\mathcal{B},x\neq\mathfrak{t}\}\right]}_{A_2}
\end{aligned}
$$

For any $x\notin\mathcal{B}^k$ (i.e., $\frac{1}{d^{p^k}(x)}\geq\frac{k-1}{\zeta}$), any $P(\tilde{x}\mid x',a')\geq\zeta$, we have

$$
b^k(\tilde{x}\mid x',a')=\frac{2}{1-\gamma}w^k(\tilde{x}\mid x',a')=\frac{2}{1-\gamma}\frac{P(\tilde{x}\mid x',a')}{\sum_{i<k}d^{\pi^i}(\tilde{x})+P(\tilde{x}\mid x',a')}\geq\frac{1}{1-\gamma},
$$

also, due to the dynamics of the extended MDP, we have $\overline{Q}^{\pi}_{\widehat{b}^k}(x,\mathfrak{t})=\frac{1}{1-\gamma}$, therefore, we can bound A_1 as

$$
\begin{aligned}
A_1&\leq\mathbb{E}^{\pi}_{x',a'}\left[\frac{1}{1-\gamma}\mathbb{1}\{x\notin\mathcal{B}^k,P(\tilde{x}\mid x',a')<\zeta\}\right]\\
&\leq\frac{1}{1-\gamma}\sum_{x',a'\in\mathcal{X}\times\mathcal{A}}d^{\pi}(x',a')P(\tilde{x}\mid x',a')\mathbb{1}\{x\notin\mathcal{B}^k,P(\tilde{x}\mid x',a')<\zeta\}\\
&<\frac{\zeta}{1-\gamma}\sum_{x',a'\in\mathcal{X}\times\mathcal{A}}d^{\pi}(x',a')\\
&=\frac{\zeta}{1-\gamma}
\end{aligned}
\tag{9}
$$

By Lemma L.5, we have

$$
\mathbb{E}^{\pi}_{x',a'}[w^k(\tilde{x}\mid x',a')-\widehat{w}^k(\tilde{x}\mid x',a')\mid\tilde{x}\neq\mathfrak{t}]\leq 2\sqrt{\mathbb{E}^{\pi}_{x',a'}\left[\left(\sqrt{w^k(\tilde{x}\mid x',a')}-\sqrt{\widehat{w}^k(\tilde{x}\mid x',a')}\right)^2\mid\tilde{x}\neq\mathfrak{t}\right]}
$$

We complete the proof by bounding A_2 as follows:

$$
\begin{aligned}
A_2 &\le 2\mathbb{E}_{x\sim d^{\overline{\pi}}}\sqrt{\mathbb{E}^{\pi}_{x',a'}\left[\left(\sqrt{w^k\left(x\mid x',a'\right)}-\sqrt{\widehat{w}^k\left(x\mid x',a'\right)}\right)^2 \mid x\neq \mathfrak{t}\right]} && \text{(Lemma L.5)}\\
&\le 2\sqrt{\mathbb{E}_{x\sim d^{\overline{\pi}}}\left[\mathbb{E}^{\pi}_{x',a'}\left[\left(\sqrt{w^k\left(x\mid x',a'\right)}-\sqrt{\widehat{w}^k\left(x\mid x',a'\right)}\right)^2 \mid x\neq \mathfrak{t}\right]\right]} && \text{(Jensen's inequality)}\\
&\le 2\sqrt{\sum_{x\in\mathcal{X}:x\neq\mathfrak{t}} d^{\overline{\pi}}(x)\sum_{\substack{x',a'\in\mathcal{X}\times\mathcal{A}:\\ x'\neq\mathfrak{t},a'\neq\mathfrak{t}}} d^{\pi}(x')\pi(a'\mid x')P(x\mid x',a')\left(\sqrt{w^k\left(x\mid x',a'\right)}-\sqrt{\widehat{w}^k\left(x\mid x',a'\right)}\right)^2}\\
&\le 2\sqrt{\frac{k-1}{\zeta}\sum_{\substack{x\in\mathcal{X}:\\ x\neq\mathfrak{t}}}\sum_{\substack{x',a'\in\mathcal{X}\times\mathcal{A}:\\ x'\neq\mathfrak{t},a'\neq\mathfrak{t}}} d^{p^k}(x')\pi(a'\mid x')P(x\mid x',a')\left(\sqrt{w^k\left(x\mid x',a'\right)}-\sqrt{\widehat{w}^k\left(x\mid x',a'\right)}\right)^2}\\
&\le 2\sqrt{\frac{(k-1)|\mathcal{A}|}{\zeta}\sum_{\substack{x\in\mathcal{X}:\\ x\neq\mathfrak{t}}}\sum_{\substack{x',a'\in\mathcal{X}\times\mathcal{A}:\\ x'\neq\mathfrak{t},a'\neq\mathfrak{t}}} d^{p^k}(x')\pi_{\mathrm{unif}}(a'\mid x')P(x\mid x',a')\left(\sqrt{w^k\left(x\mid x',a'\right)}-\sqrt{\widehat{w}^k\left(x\mid x',a'\right)}\right)^2}\\
&= 2\sqrt{\frac{(k-1)|\mathcal{A}|}{\zeta(k-1)}\sum_{i<k}\mathbb{E}^{\pi^i\circ\pi_{\mathrm{unif}}}\left[\left(\sqrt{w^k\left(x\mid x',a'\right)}-\sqrt{\widehat{w}^k\left(x\mid x',a'\right)}\right)^2\right]}\\
&= 2\sqrt{\frac{(k-1)|\mathcal{A}|\varepsilon^2_{\mathsf{weight}}}{\zeta}},
\end{aligned}
\tag{10}
$$

where in the third inequality, since $x\neq\mathfrak{t}$, we have $x'\neq\mathfrak{t}, a'\neq\mathfrak{t}$ for the state x' and action a' in the previous step before reaching x. □

Proof of Lemma E.1. Let $\widetilde{d^k}(x)=\sum_{i=1}^{k-1} d^{\pi^i}(x)$, let $\mu\in\Delta(\mathcal{X})$ attain the value of C_{push}, with $\epsilon=1$, and $\delta=C_{\mathsf{push}}$, applying Lemma E.3:

$$
\begin{aligned}
\mathbb{E}^{\pi^k}\left[\frac{P(x'\mid x,a)}{\widetilde{d^k}(x')+P(x'\mid x,a)}\right] &\le \mathbb{E}^{\pi^k}\left[\frac{P\left(x'\mid x,a\right)}{\widetilde{d^k}(x')+C_{\mathsf{push}}\mu\left(x'\right)}\right]+C_{\mathsf{push}}\cdot\mathbb{E}^{\pi^k}\left[\frac{\mu\left(x'\right)}{\widetilde{d^k}(x')+C_{\mathsf{push}}\mu\left(x'\right)}\right]\\
&\le 2C_{\mathsf{push}}\cdot\mathbb{E}^{\pi^k}\left[\frac{\mu\left(x'\right)}{\widetilde{d^k}(x')+C_{\mathsf{push}}\mu\left(x'\right)}\right].
\end{aligned}
$$

Since $\sup_{\pi\in\Pi} d^{\pi}(x)\le\sup_{x'\in\mathcal{X},a\in\mathcal{A}} P\left(x\mid x',a\right)\le C_{\mathsf{push}}\mu(x)$ for all $x\in\mathcal{X}$, Lemma L.3 implies that

$$
\sum_{x\in\mathcal{X}}\sum_{k=1}^{K}\mu(x)\frac{d^{\pi^k}(x)}{\widetilde{d^k}(x)+C_{\mathsf{push}}\mu(x)}\le 2\log(2K),
$$

which concludes the proof. □

Lemma E.3. *For any policy π, state weight vectors $d,\mu\in\mathbb{R}^{\mathcal{X}}_{+}$, and constants $\varepsilon,\delta>0$, we have:*

$$
\mathbb{E}^{\pi}\left[\frac{P\left(x'\mid x,a\right)}{d\left(x'\right)+\varepsilon\cdot P\left(x'\mid x,a\right)}\right]\le\mathbb{E}^{\pi}\left[\frac{P\left(x'\mid x,a\right)}{d\left(x'\right)+\delta\cdot\mu\left(x'\right)}\right]+\frac{\delta}{\varepsilon}\cdot\mathbb{E}^{\pi}\left[\frac{\mu\left(x'\right)}{d\left(x'\right)+\delta\cdot\mu\left(x'\right)}\right]
$$

Proof.

$$
\begin{aligned}
&\mathbb{E}^{\pi}\left[\frac{P\left(x' \mid x, a\right)}{d\left(x'\right)+\varepsilon \cdot P\left(x' \mid x, a\right)}\right]-\mathbb{E}^{\pi}\left[\frac{P\left(x' \mid x, a\right)}{d\left(x'\right)+\delta \cdot \mu\left(x'\right)}\right] \\
&=\sum_{x, a, x'} d^{\pi}(x, a) \frac{P\left(x' \mid x, a\right)\left(\delta \cdot \mu\left(x'\right)-\varepsilon \cdot P\left(x' \mid x, a\right)\right)}{\left(d\left(x'\right)+\varepsilon \cdot P\left(x' \mid x, a\right)\right)\left(d\left(x'\right)+\delta \cdot \mu\left(x'\right)\right)} \\
&\leq \sum_{x, a, x'} d^{\pi}(x, a) \frac{P\left(x' \mid x, a\right)\left(\delta \cdot \mu\left(x'\right)\right)}{\left(d\left(x'\right)+\varepsilon \cdot P\left(x' \mid x, a\right)\right)\left(d\left(x'\right)+\delta \cdot \mu\left(x'\right)\right)} \\
&\leq \frac{\delta}{\varepsilon} \sum_{x, a, x'} d^{\pi}(x, a) \frac{P\left(x' \mid x, a\right) \mu\left(x'\right)}{d\left(x'\right)+\delta \cdot \mu\left(x'\right)} \\
&=\frac{\delta}{\varepsilon} \cdot \mathbb{E}^{\pi}\left[\frac{\mu\left(x'\right)}{d\left(x'\right)+\delta \cdot \mu\left(x'\right)}\right].
\end{aligned}
$$

□

Weight estimation. To estimate the weight function, consider the following setting. Let $\mathcal{Z}$ be a set. We receive samples $z_\mu^1, \ldots, z_\mu^n \in \mathcal{Z}$ and $z_\nu^1, \ldots, z_\nu^n \in \mathcal{Z}$, where $z_\mu^t \sim \mu^t \in \Delta(\mathcal{Z})$ and $z_\nu^t \sim \nu^t \in \Delta(\mathcal{Z})$. The distributions μ^t and ν^t can be chosen in an adaptive fashion based on $z_\mu^1, z_\nu^1, \ldots, z_\mu^{t-1}, z_\nu^{t-1}$. We define $\mu = \frac{1}{n}\sum_{t=1}^n \mu^t$ and $\nu = \frac{1}{n}\sum_{t=1}^n \nu^t$, and our goal is to estimate the density ratio

$$
w^\star(z) := \frac{\mu(z)}{\nu(z)}.
$$

We assume that $\|w_\star\|_\infty \leq B$, and assume access to a realizable weight function class $\mathcal{W}$ with $w^\star \in \mathcal{W}$. We consider the estimator

$$
\widehat{w} := \arg\max_{w \in \mathcal{W}} \widehat{\mathbb{E}}_\mu[\log(w)] - \widehat{\mathbb{E}}_\nu[w], \tag{11}
$$

where $\widehat{\mathbb{E}}_\mu[\cdot]$ denotes the empirical expectation with respect to $z_\mu^1, \ldots, z_\mu^n$ and $\widehat{\mathbb{E}}_\nu[\cdot]$ denotes the empirical expectation with respect to $z_\nu^1, \ldots, z_\nu^n$. The following theorem gives a finite-sample bound for this estimator, which may be of independent interest.

Lemma E.4. *Suppose that $w^\star \in \mathcal{W}$ and $\sup_{w \in \mathcal{W}} \|w\|_\infty \leq B$. The estimator in Eq. (11) ensures that with probability at least $1-\delta$,*

$$
D^2_{\mathrm{H},\nu}\left(\widehat{w}, w^\star\right) \leq \frac{20B \log\left(|\mathcal{W}|\delta^{-1}\right)}{n},
$$

where $D^2_{\mathbf{H},\nu}\left(w, w'\right) := \mathbb{E}_\nu\left[\left(\sqrt{w}-\sqrt{w'}\right)^2\right]$.

To state the guarantee for the weight estimation, we use $\pi \circ \pi'$ to denote the policy that plays the policy π' *one step* after π is "terminated" Specifically, under the episodic reset interpretation of a discounted MDP, we execute policy π and terminate with probability $1-\gamma$; upon termination, we then execute π' one additional step. For example, in the following weight estimate lemma, we use $\pi^i \circ \pi_{\mathrm{unif}}$, meaning that we run π^i with probability γ, and with probability $1-\gamma$, we run one time step with π_{unif} and terminate.

Lemma E.5. *For any $k \in [K]$, for any $\varepsilon_{\mathsf{weight}}, \delta \in (0,1)$, distribution $p^k \in \Delta(\Pi)$ and $\pi^1, \ldots, \pi^1 \in \Pi$, Algorithm 4 ensures that with probability at least $1-\delta$, the output $\widehat{w}^k$ satisfies*

$$\frac{1}{k-1}\sum_{i<k}\mathbb{E}^{\pi^i\circ\pi_{\mathrm{unif}}}\left[\left(\sqrt{\widehat{w}^i\left(x\mid x',a'\right)}-\sqrt{w^i\left(x\mid x',a'\right)}\right)^2\right]\leq\varepsilon^2_{\mathsf{weight}},$$

and does so using at most $N_{\mathsf{weight}}=40\frac{\log\left(|\overline{\mathcal{W}}|\delta^{-1}\right)}{\varepsilon^2_{\mathsf{weight}}}$ *episodes.*

Proof. This proof follows similarly to Lemma J.7 of Amortila et al. (2024), Let $\overline{w}^k:=k\cdot w^k$ and let $\check{w}^k:=k\cdot\widehat{w}^k$. Observe that solving the optimization problem in Line 10 of Algorithm 5 is equivalent to solving the optimization problem in over the class $t\cdot\mathcal{W}_h$, which has $\left\|w'\right\|_\infty\leq t$ for all $w'\in\mathcal{W}$. As such, we can appeal to Lemma E.5 with

$$\mu\left(x'\mid x,a\right)=P\left(x'\mid x,a\right),\quad\nu\left(x'\mid x,a\right)=\frac{1}{k}\left(\sum_{i<k}d_{\pi^i}\left(x'\right)+P\left(x'\mid x,a\right)\right),$$

and

$$\omega(x,a)=\frac{1}{2}\left(d^{p^k}(x,a)+\frac{1}{k-1}\sum_{i<k}d^{\pi^i\circ\pi_{\mathrm{unif}}}(x,a)\right).$$

Under the weight assumption, we have

$$\frac{\mu\left(x'\mid x,a\right)}{\nu\left(x'\mid x,a\right)}=\overline{w}\left(x'\mid x,a\right)\in k\cdot\overline{\mathcal{W}},$$

so Lemma E.5 imply that

$$\mathbb{E}_{(x,a)\sim\omega}\left[\left(\sqrt{\check{w}^k_h\left(x'\mid x,a\right)}-\sqrt{\overline{w}^k\left(x'\mid x,a\right)}\right)^2\right]\leq\frac{20k\log\left(|\overline{\mathcal{W}}|\delta^{-1}\right)}{n},$$

or equivalently,

$$\mathbb{E}_{(x,a)\sim\omega}\left[\left(\sqrt{\widehat{w}^k\left(x'\mid x,a\right)}-\sqrt{w^k\left(x'\mid x,a\right)}\right)^2\right]\leq\frac{20\log\left(|\overline{\mathcal{W}}|\delta^{-1}\right)}{n}.$$

□

F Reward-free Exploration

In this section, we provide a result on a reward-free setting. We note that the Algorithm 3 can be directly used in a reward-free setting by simply setting $r(x,a) = 0, \forall x \in \mathcal{X}, a \in \mathcal{A}$. Which results in an algorithm that only optimizes to maximize the total bonus. We can safely ignore the factor $\frac{1}{1-\gamma}$ since this is a constant. In other word, the optimization at each iteration k is

$$\pi = \operatorname*{argmax}_{\pi} \mathbb{E}^{\pi}[w^k(x' \mid x, a)].$$

So basically, the policy operates in the same MDP with the reward $r = w^k$ in iteration k instead.

The policy cover obtained from Algorithm 3 can be later used on a downstream task with a specific-reward function. To see this, we first define the L_1-coverage from Amortila et al. (2024) as follows:

$$\Psi_{\varepsilon}^{M}(p) = \sup_{\pi \in \Pi} \mathbb{E}^{\pi}\left[\frac{d^{\pi}(x,a)}{d^{p}(x,a) + \varepsilon \cdot d^{\pi}(x,a)}\right].$$

It can be shown that L_1 coverage enables downstream task optimization by using a standard offline optimizer, where the sample complexity scales with the $\Psi_{\varepsilon}(p)$ (Amortila et al., 2024). Furthermore, we can relate $\Psi_{\varepsilon}(p)$ with the pushforward coverability by

$$\Psi_{\varepsilon}^{M}(p) \leq |\mathcal{A}| \cdot \Psi_{\mathsf{push};\varepsilon}^{M}(p),$$

where we define $\Psi_{\mathsf{push};\varepsilon}^{M}(p) = \sup_{\pi \in \Pi} \mathbb{E}^{\pi}\left[\frac{P^M(x|x',a')}{d^p(x) + \varepsilon \cdot P(x|x',a')}\right]$.

We show that $\Psi_{\varepsilon}(p)$ is bounded and depends on C_{push}, where p is the policy cover obtained from Algorithm 3.

Theorem F.1. *In the reward-free setting, suppose Algorithm 3 is instantiated with the following parameters:* $T = O(\frac{1}{\varepsilon^3(1-\gamma)^4})$, $N = O(\frac{|\mathcal{A}|^2}{(1-\gamma)^4\alpha^2\epsilon^4})$, $M = m\left(\frac{(1-\gamma)^2\alpha\epsilon^2}{|\mathcal{A}|}, \frac{\delta}{KNT}\right)$, $N_{\mathsf{weight}} = O(\frac{C_{\mathsf{push}}|\mathcal{A}|\log(|\overline{\mathcal{W}}|\delta^{-1})}{(1-\gamma)^2\varepsilon^2})$, *we have with probability* $1-\delta$,

$$\Psi_{\mathsf{push};h,\varepsilon}(p_h) \leq \frac{9\mathcal{E}(\mathbb{\Pi}, \Pi)}{(1-\gamma)^2\varepsilon^2} + \frac{173}{1-\gamma} C_{\mathsf{push}} \log(\varepsilon^{-1}).$$

Proof of Theorem F.1. For the analysis of the reward-free setting, we use the following truncated policy class $\overline{\Pi}_K$ such that for any policy $\pi \in \overline{\Pi}_K$:

$$\overline{\pi}(x) = \begin{cases} \pi(x), & \frac{\bar{d}^{\overline{\pi}}(x)}{\bar{d}^{p^k}(x)} \leq K \\ \mathfrak{t}, & \text{otherwise}, \end{cases}$$

We construct the *extended* MDP such that when a policy pick the terminal action $\mathfrak{t}$ it deterministically transition to the terminal state $\mathfrak{t}$ and immediately receive zero reward. Let r be any reward function and the reward function at each time step is bounded by 1, so whenever the truncated policy pick $\mathfrak{t}$, it transitions to the self-absorb state, therefore, we have the following:

$$\sup_{\pi \in \Pi} \mathbb{E}^{\pi}[r] - \sup_{\pi \in \overline{\Pi}_k} \overline{\mathbb{E}}^{\pi}[r] \leq \frac{1}{1-\gamma} \sup_{\pi \in \overline{\Pi}_k} \overline{P}^{\pi}\left[\frac{\bar{d}^{\pi}(x)}{\bar{d}^{p^k}(x)} > K, x \neq \mathfrak{t}\right]. \tag{12}$$

$$
\begin{aligned}
\sup_{\pi\in\overline{\Pi}_k} \overline{\mathbb{E}}^{\pi}\left[\sum_{\ell=1}^{h-1} r_\ell\right] &= \sup_{\pi\in\overline{\Pi}_k} \overline{\mathbb{E}}^{\pi}\left[\mathbb{E}\left[\frac{P\left(x \mid x', a'\right)}{\sum_{i<k} d^{\pi^i}(x) + P\left(x \mid x', a'\right)}\middle| x', a'\right] \mathbb{1}\{x', a' \neq \mathfrak{t}\}\right] \\
&= \sup_{\pi\in\overline{\Pi}_k} \overline{\mathbb{E}}^{\pi}\left[\overline{\mathbb{E}}\left[\frac{\overline{P}\left(x \mid x', a'\right)}{\sum_{i<k} \bar{d}^{\pi^i}(x) + \overline{P}\left(x \mid x', a'\right)}\middle| x', a'\right] \mathbb{1}\{x', a' \neq \mathfrak{t}\}\right] \\
&= \sup_{\pi\in\overline{\Pi}_k} \overline{\mathbb{E}}^{\pi}\left[\frac{\overline{P}\left(x \mid x', a'\right)}{\sum_{i<k} d^{\pi^i}(x) + \overline{P}\left(x \mid x', a'\right)} \mathbb{1}\{x', a' \neq \mathfrak{t}\}\right] \\
&= \sup_{\pi\in\overline{\Pi}_k} \overline{\mathbb{E}}^{\pi}\left[\frac{\overline{P}\left(x \mid x', a'\right)}{\sum_{i<k} d^{\pi^i}(x) + \overline{P}\left(x \mid x', a'\right)} \mathbb{1}\{x' \neq \mathfrak{t}\}\right]
\end{aligned} \tag{13}
$$

where we use $\varepsilon = \frac{1}{k-1}$ in the last line. Furthermore, we have the following result in the discounted MDP

Lemma F.2. *For all $K > 0$, it holds that* [1]

$$
\sup_{\pi\in\overline{\Pi}_K} \overline{P}^{\pi}\left[\frac{\bar{d}^{\pi}(x)}{\bar{d}^{p^k}(x)} > K, x \neq \mathfrak{t}\right] \leq 2 \sup_{\pi\in\overline{\Pi}_K} \overline{\mathbb{E}}^{\pi}\left[\frac{\overline{P}\left(x \mid x, a\right)}{\sum_{i<K+1} \bar{d}^{\pi^i}(x) + \overline{P}\left(x \mid x, a\right)} \mathbb{1}\{x \neq \mathfrak{t}\}\right]
$$

From Eqs. (12) and (13), Lemma F.2, we have that:

$$
\sup_{\pi\in\Pi} \mathbb{E}^{\pi}\left[w^K\right] \leq \frac{3}{1-\gamma} \sup_{\pi\in\overline{\Pi}_K} \overline{\mathbb{E}}^{\pi}\left[\frac{\overline{P}\left(x \mid x, a\right)}{\sum_{i<K+1} \bar{d}^{\pi^i}(x) + \overline{P}\left(x \mid x, a\right)} \mathbb{1}\{x \neq \mathfrak{t}\}\right] \tag{14}
$$

and hence it suffices to bound the quantity on the right-hand side. Next, note that for all $t \in [T]$, we have that

$$
\sup_{\pi\in\overline{\Pi}_K} \overline{\mathbb{E}}^{\pi}\left[\frac{\overline{P}\left(x \mid x', a'\right)}{\tilde{d}^{k}(x) + \overline{P}\left(x \mid x', a'\right)} \mathbb{1}\{x \neq \mathfrak{t}\}\right] \leq \sup_{\pi\in\overline{\Pi}_K} \overline{\mathbb{E}}^{\pi}\left[\frac{\overline{P}\left(x \mid x', a'\right)}{\tilde{d}^{k-1}(x) + \overline{P}\left(x \mid x', a'\right)} \mathbb{1}\{x \neq \mathfrak{t}\}\right]
$$

Therefore,

$$
K \cdot \sup_{\pi\in\overline{\Pi}_K} \overline{\mathbb{E}}^{\pi}\left[\frac{\overline{P}\left(x \mid x', a'\right)}{\tilde{d}^{K}(x) + \overline{P}\left(x \mid x', a'\right)} \mathbb{1}\{x \neq \mathfrak{t}\}\right] \leq \sum_{k=1}^{K} \sup_{\pi\in\overline{\Pi}_K} \overline{\mathbb{E}}^{\pi}\left[\frac{\overline{P}\left(x \mid x', a'\right)}{\tilde{d}^{k}(x) + \overline{P}\left(x \mid x', a'\right)} \mathbb{1}\{x \neq \mathfrak{t}\}\right] \tag{15}
$$

Now, note that if $x \neq \mathfrak{t}$, the dynamics of $\overline{\mathcal{M}}$ imply that we must have $x', a' \neq \mathfrak{t}$ as well. In this case, we have

$$
\frac{\overline{P}\left(x \mid x', a'\right)}{\tilde{d}_h^t(x) + \overline{P}\left(x \mid x', a'\right)} = w^k\left(x \mid x', a'\right),
$$

since $\overline{P}\left(\cdot \mid x', a'\right) = P\left(\cdot \mid x', a'\right)$ with $x', a' \neq \mathfrak{t}$, and since $d_h^{\pi^i}(x) = \bar{d}_h^{\pi^i}(x)$ when $x \neq \mathfrak{t}$ (as the policies $\pi^1, \ldots, \pi^k$ never take the terminal action). As a result, using Lemma J.2, we have that

[1] This is Lemma J.12 of Amortila et al. (2024), although their results apply for episodic MDP, but this also holds for discounted MDP.

$$
\begin{aligned}
&\sum_{k=1}^{K} \sup_{\pi \in \overline{\Pi}_K} \overline{\mathbb{E}}^{\pi}\left[\frac{\overline{P}\left(x \mid x', a'\right)}{\tilde{d}^k(x)+\overline{P}\left(x \mid x', a'\right)} \mathbb{1}\{x \neq \mathfrak{t}\}\right] \\
&=\sum_{k=1}^{K} \sup_{\pi \in \overline{\Pi}_K} \overline{\mathbb{E}}^{\pi}\left[w^k\left(x \mid x', a'\right) \mathbb{1}\{x \neq \mathfrak{t}\}\right] \\
&\leq 3 \sum_{k=1}^{K} \sup_{\pi \in \overline{\Pi}_K} \overline{\mathbb{E}}^{\pi}\left[\widehat{w}^k\left(x \mid x', a'\right) \mathbb{1}\{x \neq \mathfrak{t}\}\right] \\
&\quad+2 \sum_{k=1}^{K} \sup_{\pi \in \overline{\Pi}_K} \overline{\mathbb{E}}^{\pi}\left[\left(\sqrt{\widehat{w}^k\left(x \mid x', a'\right)}-\sqrt{w^k\left(x \mid x', a'\right)}\right)^2 \mathbb{1}\{x \neq \mathfrak{t}\}\right]
\end{aligned} \tag{16}
$$

In addition, we have

$$
\begin{aligned}
&\sup_{\pi \in \overline{\Pi}_k} \overline{\mathbb{E}}^{\pi}\left[\left(\sqrt{\widehat{w}^k\left(x \mid x', a'\right)}-\sqrt{w^k\left(x \mid x', a'\right)}\right)^2 \mathbb{1}\{x \neq \mathfrak{t}\}\right] \\
&\leq \sup_{\pi \in \overline{\Pi}_k} \sum_{x \in \mathcal{X}} \sum_{x', a' \in \mathcal{X} \times \mathcal{A}} \bar{d}^{\pi}(x') \pi(a' \mid x') P(x \mid x', a')\left[\left(\sqrt{\widehat{w}^k\left(x \mid x', a'\right)}-\sqrt{w^k\left(x \mid x', a'\right)}\right)^2 \mathbb{1}\{x \neq \mathfrak{t}\}\right] \\
&\leq K \sup_{\pi \in \overline{\Pi}_k} \sum_{x \in \mathcal{X}} \sum_{\substack{x', a' \in \mathcal{X} \times \mathcal{A}: \\ x' \neq \mathfrak{t}, a' \neq \mathfrak{t}}} \bar{d}^{p^k}(x') \pi(a' \mid x') P(x \mid x', a')\left[\left(\sqrt{\widehat{w}^k\left(x \mid x', a'\right)}-\sqrt{w^k\left(x \mid x', a'\right)}\right)^2 \mathbb{1}\{x \neq \mathfrak{t}\}\right] \\
&\leq K|\mathcal{A}| \sum_{x \in \mathcal{X}} \sum_{\substack{x', a' \in \mathcal{X} \times \mathcal{A}: \\ x' \neq \mathfrak{t}, a' \neq \mathfrak{t}}} \bar{d}^{p^k}(x') \pi_{\mathrm{unif}}(a' \mid x') P(x \mid x', a')\left[\left(\sqrt{\widehat{w}^k\left(x \mid x', a'\right)}-\sqrt{w^k\left(x \mid x', a'\right)}\right)^2 \mathbb{1}\{x \neq \mathfrak{t}\}\right] \\
&\leq K|\mathcal{A}| \sum_{x \in \mathcal{X}} \sum_{x', a' \in \mathcal{X} \times \mathcal{A}} d^{p^k}(x') \pi_{\mathrm{unif}}(a' \mid x') P(x \mid x', a')\left[\left(\sqrt{\widehat{w}^k\left(x \mid x', a'\right)}-\sqrt{w^k\left(x \mid x', a'\right)}\right)^2 \mathbb{1}\{x \neq \mathfrak{t}\}\right] \\
&\leq \frac{K|\mathcal{A}|}{k-1} \sum_{i<k} \mathbb{E}^{\pi^i \circ \pi_{\mathrm{unif}}}\left[\left(\sqrt{\widehat{w}^k\left(x \mid x', a'\right)}-\sqrt{w^k\left(x \mid x', a'\right)}\right)^2\right] \\
&\leq K|\mathcal{A}| \varepsilon_{\mathsf{weight}}^2
\end{aligned} \tag{17}
$$

From Eqs. (15) to (17), we have

$$
\sup_{\pi \in \overline{\Pi}_K} \overline{\mathbb{E}}^{\pi}\left[w^K\right] \leq \frac{3}{K} \sum_{k=1}^{K} \sup_{\pi \in \overline{\Pi}_K} \overline{\mathbb{E}}^{\pi}\left[\widehat{w}^k\left(x \mid x', a'\right) \mathbb{1}\{x \neq \mathfrak{t}\}\right]+|\mathcal{A}| \varepsilon_{\mathsf{weight}}^2 \tag{18}
$$

Next, we proceed to bound $\sum_{k=1}^{K} \sup_{\pi \in \overline{\Pi}_k} \overline{\mathbb{E}}^{\pi}\left[\widehat{w}^k\left(x \mid x', a'\right) \mathbb{1}\{x \neq \mathfrak{t}\}\right]$, fix any $k \in [K]$, we have:

$$
\begin{aligned}
&\sup_{\pi \in \overline{\Pi}_k} \overline{\mathbb{E}}^{\pi}\left[\widehat{w}^k\left(x \mid x', a'\right) \mathbb{1}\{x \neq \mathfrak{t}\}\right]-\overline{\mathbb{E}}^{\pi^k}\left[\widehat{w}^k\left(x \mid x', a'\right) \mathbb{1}\{x \neq \mathfrak{t}\}\right] \\
&=\sup_{\pi \in \overline{\Pi}_k} \overline{\mathbb{E}}^{\pi}[\widehat{w}^k\left(x \mid x', a'\right) \mathbb{1}\{x', a' \neq \mathfrak{t}\}]-\overline{\mathbb{E}}^{\pi^k}[\widehat{w}^k\left(x \mid x', a'\right) \mathbb{1}\{x', a' \neq \mathfrak{t}\}] \\
&=\overline{V}^{\overline{\pi}}-\overline{V}^{\pi^k}=\varepsilon_{\mathrm{opt}},
\end{aligned}
$$

where $\overline{V}^{\overline{\pi}}=\max_{\pi' \in \overline{\Pi}_K} \overline{V}^{\pi'}$, now we can apply boosting procedure in Brukhim et al. (2022) to bound the policy optimization. We first start by showing the following Gradient domination:

Lemma F.3 (Gradient domination)**.** *For any $\overline{\pi} \in \overline{\Pi}$, we have*

$$\overline{V}^{\overline{\pi}} - \overline{V}^{\pi} \leq K \max_{\pi' \in \Pi} (\nabla \overline{V}^{\pi})^{\top} (\pi' - \pi) + \frac{K}{1-\gamma} \mathcal{E}(\mathbb{\Pi}, \Pi)$$

Proof. Apply the performance difference lemma, we have

$$\begin{aligned}
\overline{V}^{\overline{\pi}} - \overline{V}^{\pi} &= \sum_{x \in \mathcal{X}} \bar{d}^{\overline{\pi}}(x) \left(\overline{Q}^{\pi}(x, \overline{\pi}(x)) - \overline{Q}^{\pi}(x, \pi(x)) \right) \\
&\leq \sum_{x \in \mathcal{X}} \bar{d}^{\overline{\pi}}(x) \left(\overline{Q}^{\pi}(x, \overline{\pi}(x)) - \overline{Q}^{\pi}(x, \pi(x)) \right) \mathbb{1}\{\bar{d}^{\overline{\pi}}(x) / \bar{d}^{p^k}(x) \leq K\} \\
&\leq \sum_{x \in \mathcal{X}} \bar{d}^{\overline{\pi}}(x) \left(\overline{Q}^{\pi}(x, \overline{\pi}(x)) - \overline{Q}^{\pi}(x, \pi(x)) \right) \mathbb{1}\{\bar{d}^{\overline{\pi}}(x) / \bar{d}^{p^k}(x) \leq K\} \\
&\leq \sum_{x \in \mathcal{X}} \frac{K}{1-\gamma} \bar{d}^{\pi}(x) \left(\overline{Q}^{\pi}(x, \overline{\pi}(x)) - \overline{Q}^{\pi}(x, \pi(x)) \right) \mathbb{1}\{\bar{d}^{\overline{\pi}}(x) / \bar{d}^{p^k}(x) \leq K\} \\
&\leq K \max_{\pi' \in \Pi} (\nabla \overline{V}^{\pi})^{\top} (\pi' - \pi) + \frac{K}{1-\gamma} \mathcal{E}(\mathbb{\Pi}, \Pi)
\end{aligned}$$

where we use the fact that $\overline{Q}^{\pi}(x, \mathfrak{t}) - \overline{Q}^{\pi}(x, \pi) \leq 0$ due to the dynamics of the extended MDP. □

For the Internal Boost guarantee, we have that $\varepsilon_0 = \frac{2|\mathcal{A}|}{(1-\gamma)\alpha}(\varepsilon_{\mathsf{w}} + \frac{1}{\sqrt{N}})$. Note that the Oracle guarantee run on the true MDP, but we can still apply the result extended MDP, due the the fact that all the policy π we operate in the boosting procedure never pick $\mathfrak{t}$, therefore, $\overline{Q}^{\pi} \equiv Q^{\pi}$. Now, we can apply the Lemma L.10, we have that

$$\sum_{k=1}^{K} \sup_{\pi \in \overline{\Pi}_k} \overline{\mathbb{E}}^{\pi} \left[\widehat{w}^k \left(x \mid x', a' \right) \mathbb{1}\{x \neq \mathfrak{t}\} \right] \leq \sum_{k=1}^{K} \overline{\mathbb{E}}^{\pi^k} \left[\widehat{w}^k \left(x \mid x', a' \right) \mathbb{1}\{x \neq \mathfrak{t}\} \right] + K \varepsilon_{\mathrm{opt}}, \tag{19}$$

where $\varepsilon_{\mathrm{opt}} = \frac{2K^2}{(1-\gamma)^3 T} + \frac{K}{1-\gamma}\mathcal{E}(\mathbb{\Pi}, \Pi) + \frac{2K|\mathcal{A}|}{(1-\gamma)}(\varepsilon_{\mathsf{w}} + \frac{1}{\sqrt{N}})$. For the summation above, we can apply Lemma L.5:

$$\begin{aligned}
&\sum_{k=1}^{K} \overline{\mathbb{E}}^{\pi^k} \left[\widehat{w}^k \left(x \mid x', a' \right) \mathbb{1}\{x \neq \mathfrak{t}\} \right] \\
&\leq 3 \sum_{k=1}^{K} \overline{\mathbb{E}}^{\pi^k} \left[w^k \left(x \mid x', a' \right) \mathbb{1}\{x \neq \mathfrak{t}\} \right] + 2 \sum_{k=1}^{K} \overline{\mathbb{E}}^{\pi^k} \left[\left(\sqrt{\widehat{w}^k \left(x \mid x', a' \right)} - \sqrt{w^k \left(x \mid x', a' \right)} \right)^2 \mathbb{1}\{x \neq \mathfrak{t}\} \right]
\end{aligned} \tag{20}$$

Finally, note that since $\pi^k \in \Pi$ never select the terminal action (and in particular never reach the terminal state), we have

$$\begin{aligned}
\sum_{k=1}^{K} \overline{\mathbb{E}}^{\pi^k} \left[w^k \left(x \mid x', a' \right) \mathbb{1}\{x \neq \mathfrak{t}\} \right] &= \sum_{k=1}^{K} \overline{\mathbb{E}}^{\pi^k} \left[\frac{P\left(x \mid x', a' \right)}{\widetilde{d}_h^t(x) + P\left(x \mid x', a' \right)} \mathbb{1}\{x \neq \mathfrak{t}\} \right] \\
&= \sum_{k=1}^{K} \mathbb{E}^{\pi^k} \left[\frac{P\left(x \mid x', a' \right)}{\widetilde{d}_h^t(x) + P\left(x \mid x', a' \right)} \right] \\
&\leq 4 C_{\mathrm{push}} \log(2T), \qquad \text{(Lemma E.1)}
\end{aligned} \tag{21}$$

where in the second inequality, we use the fact that $\overline{\mathbb{E}}^{\pi^k} \equiv \mathbb{E}^{\pi}$ since π^k never pick the terminal action. The weight estimaation error follows similar to Eq. (7) in the proof of Theorem 3.5.

$$\sum_{k=1}^{K} \overline{\mathbb{E}}^{\pi^k} \left[\left(\sqrt{\widehat{w}^k \left(x \mid x', a' \right)} - \sqrt{w^k \left(x \mid x', a' \right)} \right)^2 \mathbb{1}\{x \neq \mathfrak{t}\} \right] \leq \sqrt{8 C_{\mathsf{push}} |\mathcal{A}| K^2 \log(K) \varepsilon_{\mathsf{weight}}^2} + 4 C_{\mathsf{push}}. \tag{22}$$

From Eqs. (18) to (22), and for $\varepsilon = \frac{1}{K}$ we have:

$$\begin{aligned}
\Psi_{\mathsf{push};\varepsilon} &\leq K \cdot \sup_{\pi \in \Pi} \mathbb{E}^{\pi} \left[\frac{P\left(x \mid x', a'\right)}{\sum_{i=1}^{K} d^{\pi^i}(x) + P\left(x \mid x', a'\right)} \right] \\
&\leq \frac{18K^3}{(1-\gamma)^4 T} + \frac{9K^2 \mathcal{E}(\mathbb{\Gamma}, \Pi)}{(1-\gamma)^2} + \frac{6K^2|\mathcal{A}|}{(1-\gamma)^2 \alpha}(\varepsilon_{\mathsf{w}} + \frac{1}{\sqrt{N}}) + \frac{98}{(1-\gamma)} C_{\mathsf{push}} \log(2K) \\
&+ \frac{24}{1-\gamma} \sqrt{K^2 C_{\mathsf{push}} |\mathcal{A}| \log(K) \varepsilon^2_{\mathsf{weight}}} + \frac{24 C_{\mathsf{push}}}{1-\gamma} + \frac{2K|\mathcal{A}|}{1-\gamma} \varepsilon^2_{\mathsf{weight}},
\end{aligned}$$

which also complete the proof. □

G RL Boosting in Episodic MDP

Markov Decision Processes (MDPs). We consider the episodic MDPs defined as $M = \{H, \mathcal{X}, \mathcal{A}, \mu, \{P_h\}_{h=1}^H\}$, where H is the episode horizon, $\mathcal{X}$ is the state space, $\mathcal{A}$ is the action space, μ is the initial distribution and $P_h : \mathcal{X} \times \mathcal{A} \to \Delta(\mathcal{X})$ is the transition kernel at step h. A (possibly randomized) non-stationary policy is given by $\pi = (\pi_1, \ldots, \pi_H)$, where each $\pi_h : \mathcal{X} \to \Delta(\mathcal{A})$ specifies a distribution over actions. Let Π^H denote the class of such policies constructed from a base policy class Π, i.e., $\pi_h \in \Pi$ for all $h \in [H]$. At the beginning of each episode, an initial state $x_1 \sim d_0$ is drawn. Then, for each step $h = 1, \ldots, H$, the agent selects an action $a_h \sim \pi_h(x_h)$, transitions to the next state $x_{h+1} \sim P_h(\cdot \mid x_h, a_h)$, and receives a reward $r_h(x_h, a_h)$.

For a policy π, a state x, and $h \in \{1, \ldots H\}$, we define the value function $V_h^\pi : \mathcal{X} \to \mathbb{R}$ as

$$V_h^\pi(x) = \mathbb{E}^\pi \left[\sum_{h'=h}^{H} r_h\left(x_{h'}, a_{h'}\right) \mid \pi, x_{h'} = x \right],$$

and state-action value function $Q_h^\pi : \mathcal{X} \times \mathcal{A} \to \mathbb{R}$ as

$$Q_h^\pi(x, a) = \mathbb{E}^\pi \left[\sum_{h'=h}^{H} r_h\left(x_{h'}, a_{h'}\right) \mid \pi, x_{h'} = x, a_{h'} = a \right],$$

Furthermore, we define the optimal policy as $\pi^* = \operatorname{argmax}_\pi V_1^\pi(d_0)$, where $V_1^\pi(\mu) = \mathbb{E}_{x_1 \sim d_0}[V_1^\pi(x_1)]$. For simplicity, we denote $V^\pi = V_1^\pi(\mu)$ and $V^{\pi^*} = V_1^{\pi^*}(\mu)$ when it is clear from the context.

We define the *occupancy measure* for layer h as

$$d_h^\pi(x, a) = P^\pi(x_h = x, a_h = a)$$

and $d_h^\pi(x) = P^\pi(x_h = x)$. We assume that the cumulative reward is bounded as $\mathbb{E}\left[\sum_{h=1}^H r_h\right] \leq R_{\max}$. In the episodic MDP, it is common to normalize rewards such that $R_{\max} = 1$ for (Amortila et al., 2024; Misra et al., 2020)). Although we consider an episodic MDP, our results remain comparable to those in discounted MDP (Brukhim et al., 2022) by setting the effective horizon $H \approx \frac{1}{1-\gamma}$, and $R_{\max}$ is bounded by $\frac{1}{1-\gamma}$ where γ is the discount factor. Therefore, we purposely keep $R_{\max}$ general in our analysis to facilitate comparison across different MDP formulations.

Similar to the result in discounted MDPs, our algorithm can work on large-scale MDPs where the state space $\mathcal{X}$ can be very large, or even infinite. When the environment admits some underlying structure that allows us to design algorithms that can be computationally and statistically efficient. We provide a brief description of these structures in episodic MDPs in the following paragraphs.

Block MDPs. A block MDP models an environment where a large observation space $\mathcal{X}$ can be summarized into a finite state $\mathcal{S}$. Specifically, block MDP assumes there exists an emission function $q : \mathcal{X} \to \Delta(\mathcal{S})$ that maps the latent states to a distribution over observations. The agent interacts with the environment by repeatedly generating H-step trajectories $(s_1, x_1, a_1, r_1, \ldots, s_H, x_H, a_h, r_H)$, where $s_1 \sim \mu$, $s_{h+1} \sim P(\cdot \mid s_h, a_h), x_h \sim q(s_h)$, for all $h \in [H]$. The agent can observe x_h, but the states $s_1, \ldots, s_H$ are hidden from the agent. Tabular MDP is a special case of block MDP where the observation space coincides with the state space, i.e., $\mathcal{X} = \mathcal{S}$.

Low-rank MDPs. A low-rank MDP with dimension d is an MDP where the transition kernel admits a low-rank factorization: $\forall h \in [H]$, there exist $\phi_h : \mathcal{X} \times \mathcal{A} \to \mathbb{R}^{\mathrm{d}}$ and $\mu_h : \mathcal{X} \to \mathbb{R}^{\mathrm{d}}$ such that $\forall x_h, x_{h+1} \in \mathcal{X}, a_h \in \mathcal{A} : P_h\left(x_{h+1} \mid x_h, a_h\right) = \left\langle \phi_h\left(x_h, a_h\right), \mu_h\left(x_{h+1}\right) \right\rangle$. This structure enables efficient representation and learning in environments with large or continuous state spaces.

Next, we restate several key definitions from the boosting-based reinforcement learning framework of Brukhim et al. (2022) with some adjustment in the context of episodic MDPs.

Definition G.1 (Policy Tree)**.** A Policy Tree $\mathbb{\Pi} \subseteq \mathcal{X} \to \Delta_A$ is linear combination of T policy in $\widetilde{\Pi} := \Pi^H \subseteq \mathcal{X} \to \Delta_A$, for some base policy class Π. Policy at each layer of $\tilde{\pi} \in \widetilde{\Pi}$ is a projected policy $\Gamma(\overline{\pi})$, where each $\overline{\pi}$ is a linear combination of N policy in Π. $\Pi \subseteq \mathcal{X} \to \Delta_{\mathcal{A}}$.

We note that in episodic MDP, policy structure is different where the policy is non-stationary which the policy can be different at different time step. A function policy can be view a concatenation of H policy for each layer, where each policy belong to some base policy class.

Definition G.2 (Policy completeness)**.** For any $h \in [H]$, distribution μ_h, and any two policy class $\mathbb{\Pi}, \Pi$, we define:

$$\mathcal{E}(\mathbb{\Pi}, \Pi) = \max_{\pi \in \mathbb{\Pi}} \min_{\pi'_h \in \Pi} \mathbb{E}_{x_h \sim \mu_h} \left[\max_a Q_h^\pi(x_h, a) - \mathbb{E}_{a \sim \pi'_h} Q_h^\pi(x_h, a) \right]$$

The policy completeness measures the ability of the policy class Π at each layer to approximate the greedy policy with respect to any Q-function induced by a policy in $\mathbb{\Pi}$.

Definition G.3 (Distribution mismatch)**.** Let $\pi^* = \operatorname{argmax}_\pi V^\pi$. We define the following distribution mismatch coefficients: $C_\infty = \max_{h \in [H]} \sup_{\pi \in \mathbb{\Pi}} \left\| \frac{d_h^{\pi^*}}{d_h^\pi} \right\|_\infty$.

Further notation. We denote sequences using the ":" operator, e.g., $\pi_{1:H} = (\pi_1, \ldots, \pi_H)$. We use π_{unif} to denote the uniform policy. We define $\pi \circ_h \pi'$ as the policy that follows π for layers $h' < h$ and follows π' for $h' \geq h$.

H Warming up: Boosting in Episodic MDPs

The algorithm presented in this section serves as a building block for the policy cover in Appendix I. These results are a straightforward extension of Brukhim et al. (2022) where we adapt their boosting approach to episodic MDPs. We note that the algorithm for the policy only needs the first level boosting algorithm (PSDP Boosting Algorithm 6) as the policy optimization procedure. Nevertheless, we still provide a similar two-level boosting as in Brukhim et al. (2022), which we provide the pseudo-code in Algorithm 5. Similar to the boosting procedure in Brukhim et al. (2022), the algorithm runs for T iterations; at each iteration, it calls a subroutine in Algorithm 6 to obtain a new policy π'_t, which is then aggregated with the previous policy π_{t-1} to form the new policy π_t. The aggregation is done via a convex combination with step size $\eta_{1,t}$. Here, we use π_{t-1} to collect samples in Algorithm 6, and we denote $d_h^{\pi_{t-1}}$ as the state distribution at layer h induced by π_{t-1}. The algorithm outputs the final policy π^T after T iterations.

Algorithm 5 RL Boosting

1: **input**: number of iterations T, number of weak learner N, number of sample episodes M.
2: Initialize policy π^0 arbitrarily.
3: **for** $t = 1, \cdots, T$ **do**
4: Run Algorithm 6 with $\{d_h^{\pi_{t-1}}\}_{h=1}^H$, reward r, number of weak learner N, number of episodes M to obtain π'_t.
5: $\pi_t = (1 - \eta_{1,t})\pi_{t-1} + \eta_{1,t}\pi'_t$.
6: **end for**
7: **return** π_T.

Algorithm 6 PSDP Boosting

1: **input**: Exploration distribution $\{\rho\}_{h=1}^H$, reward function r, number of weak learner N, number of episodes M
2: **for** $h = H, H-1, \cdots, 1$ **do**
3: Run Algorithm 7 with distribution ρ_h, policy $\pi_{h+1:H}$, number of iterations N, number of episodes M to obtain π_h.
4: **end for**
5: **return** $\pi = (\pi_1, \pi_2, \cdots \pi_H)$.

Each optimization inside the for loop of PSDP Boosting algorithm, i.e., Algorithm 7, share a similar flavor with the Internal Boost in Brukhim et al. (2022), where it specifically solves a contextual

optimization problem. PSDP Boosting builds the policy from the last layer H to the first layer. At each layer h, it calls a subroutine Layer Boosting (Algorithm 7) to obtain a policy π_h for that layer. Layer Boosting runs for N iterations. At each iteration, it collects M episodes by executing $\pi_{h+1:H}$ with the exploration distribution ρ_h. The collected dataset is then modified to form a new dataset with linear losses (Line 5), which is then fed into the weak learner to obtain a new policy. The new policy is then aggregated with the previous policy $\tilde{\pi}_{n-1}$ to form the new policy $\tilde{\pi}_n$. The final output of Layer Boosting is then projected to be a valid policy via the projection operator $\Gamma(\cdot)$ to form the policy π_h for layer h. We provide the guarantee for the boosting procedure in Algorithm 7 as follow,

Lemma H.1. *Let $\beta = \sqrt{\frac{1}{\alpha N}}$ and $\hat{\pi}_h$ be the boosted policy obtained from Layer Boosting (Algorithm 7).*

$$\sup_{\overline{\pi}_h \in \Pi} \mathbb{E}_{\rho_h} \left[\sum_a Q(x, a; \pi_{h+1:H})(\overline{\pi}_h(a \mid x) - \hat{\pi}_h(a \mid x) \right] \leq \frac{2|\mathcal{A}|R_{\max}}{\alpha}(\varepsilon_{\mathsf{w}} + \frac{2}{\sqrt{N}})$$

Algorithm 7 Layer Boosting

1: **input:** Sample state distribution ρ_h, policy $\pi_{h+1:H}$, number of iterations N, number of episodes M.
2: Initialize $\tilde{\pi}_0$ arbitrarily.
3: **for** $n = 1, \cdots, N$ **do**
4: Execute $\pi_{h+1:H}$ with the sample distribution ρ_h via Algorithm 8 for M episodes, and obtain the dataset $D_n = \{(s_i, \widehat{Q_i})\}_{i=1}^M$.
5: Modify dataset D_n to a new dataset $D'_n = \{(s_i, f_i)\}_{i=1}^M$, such that for all $i \in [M]$:
6:

$$f_i = \frac{1}{\beta}\left(y_i - \tilde{\pi}_n\left(\cdot \mid s_i\right)\right),$$

$$y_i = \underset{y \in \mathbb{R}^{|A|}}{\operatorname{argmin}}\{ -\widehat{Q}_i^\top y + G \min_{z \in \Delta_A} \|z - y\| + \frac{1}{2\beta}\left\|\tilde{\pi}_n\left(\cdot \mid s_i\right) - y\right\|^2\},$$

where $G = H|\mathcal{A}|, \beta = \sqrt{\frac{1}{\alpha N}}$, and $f_i, \widehat{Q_i} \in \mathbb{R}^{|\mathcal{A}|}$.
7: Let $\mathcal{W}_n$ be the policy chosen by the weak learning oracle when given data set D_n.
8: Update $\tilde{\pi}_n = \left(1 - \eta_{2,n}\right)\tilde{\pi}_{n-1} + \frac{\eta_{2,n}}{\alpha}\mathcal{W}_n$.
9: **end for**
10: **return** $\Gamma(\tilde{\pi}_N)$.

Algorithm 8 Trajectory sampler

1: **input:** Sample state distribution ρ_h, policy $\pi_{h+1:H}$.
2: Sample $s_h \sim \rho_h$, and action $a' \sim \mathcal{U}(\mathcal{A})$ uniformly.
3: Take action a' at state s_h, then continue to execute $\pi_{h+1:H}$. Upon terminating, set $R(s_h, a)$ as the sum of the reward from layer $h+1$ onwards.
4: Define the vector $\widehat{Q^\pi_{s_h}}$, such that for all $a \in \mathcal{A}, \widehat{Q^\pi_{s_h}}(a) = |A| \cdot R\left(s_h, a'\right) \cdot \mathbb{I}_{a=a'}$.
5: **return** $\left(s_h, \widehat{Q^\pi_{s_h}}\right)$.

The final output of PSDP Boosting is then a concatenation of the policies from layer 1 to layer H, i.e., $\pi = (\pi_1, \pi_2, \cdots \pi_H)$. These policy from Algorithm 6 will be aggregated once again to form a final policy in Algorithm 5. Next, we provide the following result for Algorithm 5.

Theorem H.2. *Algorithm 5 samples TNM episodes with $T = O\left(\frac{H^2 C_\infty^2 R_{\max}}{\varepsilon}\right), N = O\left(\frac{HC_\infty|\mathcal{A}|R_{\max}}{\alpha\varepsilon}\right)^2$, $M = m\left(\frac{\alpha\varepsilon}{C_\infty H|\mathcal{A}|R_{\max}}, \frac{\delta}{NT}\right)$, set $\eta_{1,t} = \min\left\{1, \frac{2C_\infty}{t}\right\}$, with probability $1-\delta$, $V^{\pi^*} - V^\pi \leq HC_\infty\mathcal{E}(\mathbb{\Pi}, \Pi) + \varepsilon$.*

Sample Complexities: *If* $m(\varepsilon, \delta) = \frac{\log|\mathcal{W}|}{\varepsilon^2}\log(\frac{1}{\delta})$ *for some measure of weak learning complexity* $|\mathcal{W}|$*, then the algorithm samples* $O\left(\frac{H^6 C_\infty^6 R_{\max}^5 \log(|\mathcal{W}|)}{\alpha^4\varepsilon^5}\right)$ *episodes.*

Remark H.3. For the discounted MDPs, we can set the effective horizon $H \approx \frac{1}{1-\gamma}$. Note that the total reward $R_{\max}$ is at most $\frac{1}{1-\gamma}$ we recover the sample complexity $\tilde{O}\left(\frac{C_\infty^6 |A|^4 \log|\mathcal{W}|}{(1-\gamma)^{11}\alpha^4\varepsilon^5}\right)$ in the episodic model (Brukhim et al., 2022, Theorem 7).

I Policy Cover

In this section, we show how a policy cover can be used to bound the occupancy mismatch term C_∞ in Theorem H.2. The key idea is to construct, via the boosting procedure in Appendix H, a set of policies that collectively achieve a good coverage over the state space. This guarantees that the induced occupancy measure adequately covers the state space, thereby bounding the mismatch ratio C_∞.

A policy cover is a collection of policies whose occupancy measures together cover the state space sufficiently well. Policy cover are defined over an ensemble of policies, i.e., a distribution $p \in \Delta(\overline{\Pi})$, for a class of interest policy $\overline{\Pi} \subset \Pi_{\texttt{rns}}$. The goal is to find a distribution p such that the induced occupancy measure $d_h^p(x) = \mathbb{E}_{\pi\sim p}[d_h^\pi(x)]$ covers the state space well for all $h \in [H]$. Prior analyses of policy covers (Misra et al., 2020; Amortila et al., 2024) rely on such ensembles, where the expected occupancy measure can be decoupled across policies, e.g., let $p = \mathrm{Unif}(\pi^1, \ldots, \pi^T)$ be the ensemble then the occupancy $d_h^p(x)$ can be decoupled as $d_h^p(x) = \frac{1}{T}\sum_{t=1}^T d_h^{\pi^t}(x)$.

Our analysis relies on the extended MDPs and the truncated policy class, the contruction is the same as discounted MDP, with some modification in the context of episodic setting.

Extended MDP and truncated policy class. Similarly to the discounted MDP, we construct an extended MDP $\overline{\mathcal{W}}$ by augmenting the original state space and action space with $\overline{\mathcal{A}} = \mathcal{A} \cup \{\mathfrak{t}\}$ and $\overline{\mathcal{X}} = \mathcal{X} \cup \{\mathfrak{t}\}$. The transition in the extended MDP is the same as the orginal MDP, except that the terminal state is a self-looping state, i.e., $P_h(\mathfrak{t} \mid \mathfrak{t}, a) = 1 \forall a \in \overline{\mathcal{A}}$. For every policy π in the original class Π, we define a corresponding truncated policy $\overline{\pi} \in \overline{\Pi}$ as:

$$\overline{\pi}(x_h) = \begin{cases} \pi(x_h), & x_h \in \mathcal{B} \\ \mathfrak{t}, & \text{otherwise}, \end{cases}$$

for some "well-explore" set $\mathcal{B}$ that we will defined later depend on which notion of the coverability that we used. Intuitively, policy acts the same when in well-explored set, otherwise, it picks the terminated action $\mathfrak{t}$.

I.1 RL Boosting with L_∞ coverability

We proceed with the assumption on the access to a distribution μ such that the L_∞ concentration is bounded, and subsequently define our bonus exploration and the policy transformation.

Assumption I.1 (L_∞-concentrability coefficient)**.** We assumes access to a distribution $\mu \in \Delta(\mathcal{X}\times\mathcal{A})$ for which the L_∞-concentrability coefficient $\widetilde{C}_\infty(\mu) := \sup_\pi \sup_{(x,a)\in\mathcal{X}\times\mathcal{A}} \left\{\frac{d_h^\pi(x,a)}{\mu(x,a)}\right\}$ is bounded.

We abbreviate $C_\infty \equiv C_{\infty;h}^M(\mu)$, and define our exploration bonus as:

$$\begin{aligned} w_h^k(x,a) &= \frac{\widetilde{C}_\infty \mu(x,a)}{\sum_{i<t} d_h^{\pi^i}(x,a) + \widetilde{C}_\infty \mu(x,a)}, \\ b_h^k(x,a) &= 2w_h^k(x,a). \end{aligned} \tag{23}$$

Policy transformation $\mathcal{T}(\Pi, b^k)$**.** Given the well-explored set $\mathcal{B}^k := \left\{x_h : w_h^k(x,a) \le \frac{1}{2}, \forall a \in \mathcal{A}, h \in [H]\right\}$, the transformed policy $\widehat{\pi}$ is defined as follows:

$$\widehat{\pi}(x) = \begin{cases} \pi(x_h), & \text{if } x_h \in \mathcal{B}^k \\ \mathrm{Unif}\left(\left\{a \in \mathcal{A} : w_h^k(x_h,a) > \frac{1}{2}\right\}\right), & \text{otherwise} \end{cases}$$

Algorithm 9 Boosting with L_∞ coverability

input: number of epoch K, number of iterations T, number of weak learner N, number of sample episodes M, policy class Π.
Initialize policy π^1 arbitrarily.
for $k = 1, \cdots, K$ **do**
    Contruct new bonus $b_h^k(x,a)$ with policy cover $d_h^{p^k}$ (Eq. (23)).
    Update policy class $\widehat{\Pi} \leftarrow \mathcal{T}\left(\Pi, b^k\right)$.
    Run (Algorithm 6) with reward function $r + b^k$, policy class $\widehat{\Pi}$, initial distribution d^{p^k} to obtain π^k.
    Update the policy cover $p^{k+1} = \mathrm{Unif}(\pi_1, \ldots, \pi_k)$.
end for
return $\pi := \mathrm{argmax}_{\pi \in \{\pi^1, \ldots, \pi^K\}} V^\pi$.

With the definition of the bonus function and the policy transformation, we provide the boosting for episodic MDP with the L_∞ coverability as follows.

Theorem I.2. *For any $\varepsilon, \delta \in (0,1)$, suppose Assumption I.1 is satisfied, Algorithm 9 is instantiated with the following parameters: $K = O(\frac{\widetilde{C}_\infty H \iota}{\varepsilon})$, $N = O(\frac{\widetilde{C}_\infty^2 H^4 |\mathcal{A}|^2 \iota^2}{\alpha^2 \varepsilon^4})$, $M = m\left(\frac{\alpha \varepsilon^2}{\widetilde{C}_\infty H^2 |\mathcal{A}|}, \frac{\delta}{KHN}\right)$, where $\iota = \log(\frac{\widetilde{C}_\infty H}{\varepsilon})$. Then Algorithm 2 produces policy π, such that with probability at least $1-\delta$,*

$$V^* - V^\pi \le \frac{\widetilde{C}_\infty H \iota}{\varepsilon} \mathcal{E}(\mathbb{\Pi}, \Pi) + \varepsilon.$$

Sample Complexity: *If $m(\varepsilon, \delta) = \frac{\log |\mathcal{W}|}{\varepsilon^2} \log \frac{1}{\delta}$ for some measure of weak learning complexity $|\mathcal{W}|$, then the algorithm use at most $KHNM \le \widetilde{O}\left(\frac{\widetilde{C}_\infty^5 H^{10} |\mathcal{A}|^4 \log |\mathcal{W}|}{\alpha^4 \varepsilon^9}\right)$ episodes.*

Proof. For any episode $k \in [K]$, we define a well-explored set $\mathcal{B}_h^k := \{x, a : \frac{\widetilde{C}_\infty \mu(x,a)}{d^{p^k}(x,a)} \le k-1\}$, we say $x_{\in} \mathcal{B}_h^k$ if $(x,a) \in \mathcal{B}^k, \forall a \in \mathcal{A}$. For any policy $\overline{\pi} \in \overline{\Pi}_k$, and $\pi \in \mathcal{T}(\Pi, \mathcal{B}^k)$, for any $x \in \mathcal{B}^k$, we have

$$d^{\overline{\pi}}(x) \le (k-1) d^{p^k}(x)$$

$$d_h^\pi(x_h) = \sum_{a \in \mathcal{A}} d_h^\pi(x_h, a) \le \sum_{a \in \mathcal{A}} \widetilde{C}_\infty \mu(x_h, a) \le \sum_{a \in \mathcal{A}} (k-1) d^{p^k}(x_h, a) \le (k-1) d^{p^k}(x_h),$$

Let $\pi = \mathrm{argmin}_{\pi \in \{\pi^1, \ldots, \pi^K\}} V^{\pi^k}$ be the output of the Algorithm 9, we have the following:

$$\le \frac{1}{K} \sum_{k=1}^K V^{\pi^*} - V^{\pi^k}$$

We proceed to bound the RHS by decomposing the summation into the following three terms:

$$\frac{1}{K} \sum_{k=1}^K V^{\pi^*} - V^{\pi^k} = \underbrace{\frac{1}{K} \sum_{k=1}^K V^{\pi^*} - \overline{V}_{b^k}^{\overline{\pi^*}}}_{I_1 \le 0} + \underbrace{\frac{1}{K} \sum_{k=1}^K \overline{V}_{b^k}^{\overline{\pi^*}} - \overline{V}_{b^k}^{\pi^k}}_{I_2} + \underbrace{\frac{1}{K} \sum_{k=1}^K \overline{V}_{b^k}^{\pi^k} - V^{\pi^k}}_{I_3}.$$

For I_2, we have

$$\begin{aligned}
\overline{V}_{b^k}^{\overline{\pi^*}} - \overline{V}_{b^k}^{\pi^k} &= \sum_{h=1}^{H} \mathbb{E}_{x_h \sim d_h^{\overline{\pi}}} \left[\overline{Q}_h \left(x_h; \overline{\pi}_h \circ \pi_{h+1:H}^k \right) - \overline{V}_h \left(x_h; \pi_{h:H}^k \right) \right] \\
&= \sum_{h=1}^{H} \mathbb{E}_{x_h \sim d_h^{\overline{\pi}}} \left[\overline{Q}_h \left(x_h; \overline{\pi}_h \circ \pi_{h+1:H}^k \right) - \overline{V}_h \left(x_h; \pi_{h:H}^k \right) \mathbb{1}\{x \in \mathcal{B}^k\} \right] \\
&\quad + \sum_{h=1}^{H} \mathbb{E}_{x_h \sim d_h^{\overline{\pi}}} \left[\overline{Q}_h \left(x_h; \overline{\pi}_h \circ \pi_{h+1:H}^k \right) - \overline{V}_h \left(x_h; \pi_{h:H}^k \right) \mathbb{1}\{x \notin \mathcal{B}^k\} \right]
\end{aligned}$$

The first term in the equation above can be upper bounded by:

$$\begin{aligned}
&\sum_{h=1}^{H} \mathbb{E}_{x_h \sim d_h^{\overline{\pi}}} \left[\overline{Q}_h \left(x_h; \overline{\pi}_h \circ \pi_{h+1:H}^k \right) - \overline{V}_h \left(x_h; \pi_{h:H}^k \right) \right] \\
&\leq (k-1) \sum_{h=1}^{H} \mathbb{E}_{x_h \sim d_h^{p^k}} \left[\overline{Q}_h \left(x_h; \overline{\pi}_h \circ \pi_{h+1:H}^k \right) - \overline{V}_h \left(x_h; \pi_{h:H}^k \right) \right] \\
&\leq (k-1) \sum_{h=1}^{H} \mathbb{E}_{x_h \sim d_h^{p^k}} \left[\overline{Q}_h \left(x_h; \overline{\pi}_h \circ \pi_{h+1:H}^k \right) - \max_a \overline{Q}_h(x_h, a; \overline{\pi}_h \circ \pi_{h+1:H}^k) \right] \\
&+ (k-1) \sum_{h=1}^{H} \mathbb{E}_{x_h \sim d_h^{p^k}} \left[\max_a \overline{Q}_h(x_h, a; \pi_{h+1:H}^k) - \max_{\pi_h \in \Pi} \overline{Q}_h(x_h, a; \pi_h \circ \pi_{h+1:H}^k) \right] \\
&+ (k-1) \sum_{h=1}^{H} \mathbb{E}_{x_h \sim d_h^{p^k}} \left[\max_{\pi_h \in \Pi} \overline{Q}_h(x_h, a; \pi_h \circ \pi_{h+1:H}^k) - \overline{V}_h \left(x_h; \pi_{h:H}^k \right) \right] \\
&\leq (k-1) H \mathcal{E}(\mathbb{\Pi}, \Pi) + \frac{2(k-1)H|\mathcal{A}|}{\alpha} (\varepsilon_{\mathsf{w}} + \frac{2}{\sqrt{N}}),
\end{aligned}$$

where the last inequality follows by Lemma H.1.

For the second term, we have $\overline{\pi}_h$ plays $\mathfrak{t}$ when $x_h \notin \mathcal{B}^k$, this implies $\overline{Q}_h \left(x_h; \overline{\pi}_h \circ \pi_{h+1:H}^k \right) = 1$. By the policy transformation construction, we have $\overline{V}_h \left(x_h; \pi_{h:H}^k \right) \geq 1$, therefore, we can safely ignore this second term.

For I_3, we have:

$$\begin{aligned}
\sum_{k=1}^{K} \overline{V}_{b^k}^{\pi^k} - V^{\pi^k} &= \sum_{k=1}^{K} \sum_{h=1}^{H} \mathbb{E}^{\pi^k} [b_h^k(x, a)] \\
&= \sum_{h=1}^{H} \sum_{k=1}^{K} \mathbb{E}^{\pi^k} \left[\frac{\widetilde{C}_\infty \mu(x_h, a_h)}{\sum_{i<k} d_h^{\pi^i}(x) + \widetilde{C}_\infty \mu(x_h, a_h)} \right] \\
&= 2\widetilde{C}_\infty \sum_{x \in \mathcal{X}, a \in \mathcal{A}} \sum_{k=1}^{K} \mu(x, a) \frac{d_h^{\pi^k}(x, a)}{\sum_{i<k} d^{\pi^i}(x, a) + \widetilde{C}_\infty \mu(x, a)} \\
&\leq 2H\widetilde{C}_\infty \log(2K)
\end{aligned}$$

Put everything together, we have:

$$V^{\pi^*} - V^{\pi} \leq KH\mathcal{E}(\mathbb{\Pi}, \Pi) + \frac{2KH|\mathcal{A}|}{\alpha} (\varepsilon_{\mathsf{w}} + \frac{2}{\sqrt{N}}) + \frac{2H\widetilde{C}_\infty \log(2K)}{K}$$

□

I.2 RL boosting with pushforward coverability

The exploration bonus that uses L_∞ does require the knowledge of the reference distribution μ. The pushforward coverability aim to remove this assumption, however, we would get a looser upper bound.

This pushforward coverability is a more general notion that can be applied to various structured MDPs. The main idea is to learn a new policy that can reach the states that are not well covered by the previous policy cover. Instead of using a reference distribution μ in L_∞ coverability, we use the transition instead. Specifically, we use the following weight function as an exploration objective:

$$w_h^k(x_h \mid x_{h-1}, a_{h-1}) := \frac{P_{h-1}(x_h \mid x_{h-1}, a_{h-1})}{\sum_{i<k} d_h^{\pi^{h,i}}(x_h) + P_{h-1}(x_h \mid x_{h-1}, a_{h-1})},$$

Intuitively, if a state x_h is poorly covered by previous policies, the sum $\sum_{i<k} d_h^{\pi^{h,i}}(x_h)$ will be small, and the resulting weight will be higher, thereby encouraging exploration of that state. Similar to L_∞ coverability, we use the following exploration bonus: $b_h^k(x_h \mid x_{h-1}, a_{h-1}) := w_h^k(x_h \mid x_{h-1}, a_{h-1})$

Since the reward construction requires the knowledge of the transition, which is unknown in general, we can efficiently estimate the weight function via contrastive learning (Amortila et al., 2024, Algorithm 4).

Next, we start with the definition of pushforward coverability as follow

Definition I.3 (Pushforward coverability)**.**

$$C_{\mathsf{push};h} = \inf_{\mu \in \Delta(\mathcal{X})} \sup_{(x,a,x') \in \mathcal{X}\times\mathcal{A}\times\mathcal{X}} \left\{ \frac{P_{h-1}(x' \mid x, a)}{\mu(x')} \right\},$$

with $C_{\mathsf{push}} = \max_{h\in[H]} C_{\mathsf{push};h}$.

In constrast to the distribution mismatch C_∞, the pushforward coverability C_{push} is bounded by the intrinsic complexity of the MDP; this is general notion that can be applied to various structured MDPs. For example, tabular MDPs with state space $\mathcal{X}$ have $C_{\mathsf{push}} \le |\mathcal{X}|$, block MDPs with latent space $\mathcal{S}$ have $C_{\mathsf{push}} \le |\mathcal{S}|$, and low-rank MDPs of dimension d have $C_{\mathsf{push}} \le d$.

Algorithm 10 Boosting with push forward coverability

1: **input**: number of iterations T, number of weak learner N, number of sample episodes M, policy class Π.
2: Initialize policy π^1 arbitrarily, $p^1 = \{\pi^1\}$.
3: **for** $k = 1, \cdots, K$ **do**
4: Run Algorithm 11 to estimate $\widehat{w}_h^k(x \mid x', a')$ with policy cover p^k and then contruct new bonus $\widehat{b}_h^k(x \mid x', a')$ for each layer h.
5: Run Algorithm 6 with reward function $r + \widehat{b}^k$, initial distribution d^{p^k} to obtain π^k.
6: Update the policy cover $p^{k+1} = \mathrm{Unif}(\pi_1, \ldots, \pi_k)$.
7: **end for**
8: **return** $\pi := \mathrm{argmax}_{\pi \in \{\pi^1,\ldots,\pi^K\}} V^\pi$.

To perform the weight estimation, we assume access to a weight function class $\mathcal{W} = \mathcal{W}_{1:H}$, with $\mathcal{W}_h \subseteq (\mathcal{X} \times \mathcal{A} \times \mathcal{X} \to \mathbb{R}_+)$ which we restate the following assumption from Amortila et al. (2024) as follows.

Assumption I.4 (Weight function realizability)**.** For all $h \ge 2$ and all $\pi \in \Pi$

$$w_h^\pi\left(x' \mid x, a\right) := \frac{P_{h-1}\left(x' \mid x, a\right)}{d_h^\pi\left(x'\right)} \in \overline{\mathcal{W}}_h$$

Next, we provide the result for Algorithm 10 in the following theorem.

Theorem I.5. *For any $\varepsilon, \delta \in (0,1)$, suppose Assumption I.4 is satisfied. Algorithm 10 is instantiated with the following parameters: $K = O(\frac{C_{\mathsf{push}} H \iota}{\varepsilon})$, $N = O(\frac{C_{\mathsf{push}}^2 H^6 |\mathcal{A}|^2 \iota^2}{\alpha^2 \epsilon^6})$, $M = m\left(\frac{\alpha\epsilon^3}{C_{\mathsf{push}} H^3 |\mathcal{A}|}, \frac{\delta}{KHNN_{\mathsf{weight}}}\right)$, $N_{\mathsf{weight}} = O(\frac{C_{\mathsf{push}} H^4 |\mathcal{A}| \log(|\overline{\mathcal{W}}|\delta^{-1})\iota}{\varepsilon^4})$, where $\iota = \log(\frac{C_{\mathsf{push}} H}{\varepsilon})$. Then Algorithm 10 produces policy π, such that with probability at least $1-\delta$,*

$$V^* - V^\pi \le \frac{C_{\mathsf{push}} H^3 \iota}{\varepsilon^2} \mathcal{E}(\Pi, \Pi) + \varepsilon.$$

Algorithm 11 Weight estimation

Let $\frac{1}{(t-1)}\sum_{i<t}\pi^i \circ_{h-1} \pi_{\mathsf{unif}}$ if $t \geq 1$ and $q := p_{h-1}$ otherwise.
Let $\mathcal{D}_1 = \mathcal{D}_2 = \varnothing$.
For each $j \in [n]$, draw $\pi \sim q$ and sample $\left(x^j_{h-1}, a^j_{h-1}, x^j_h\right) \sim \pi$. Add $\left(x^j_{h-1}, a^j_{h-1}, x^j_h\right)$ to both $\mathcal{D}_1$ and $\mathcal{D}_2$.
for $i < t$ **do**
    Draw n samples $\left\{\left(x^j_{h-1}, a^j_{h-1}, x^j_h\right)\right\}_{j\in[n]}$ independently by drawing $\pi \sim q$ and $\left(x^j_{h-1}, a^j_{h-1}, x^j_h\right) \sim \pi$.
    Draw n samples $\left\{\left(\tilde{x}^j_{h-1}, \tilde{a}^j_{h-1}, \tilde{x}^j_h\right)\right\}_{j\in[n]}$ by sampling $\left(\tilde{x}^j_{h-1}, \tilde{a}^j_{h-1}, \tilde{x}^j_h\right) \sim \pi^i$.
    Add $\left\{\left(x^j_{h-1}, a^j_{h-1}, x^j_h\right)\right\}_{j\in[n]}$ to $\mathcal{D}_1$ and add $\left\{\left(x^j_{h-1}, a^j_{h-1}, \tilde{x}^j_h\right)\right\}_{j\in[n]}$ to $\mathcal{D}_2$.
end for
Set $\widehat{w} := \arg\max_{w\in\mathcal{W}_h} \widehat{\mathbb{E}}_{\mathcal{D}_1}\left[\log\left(w\left(x_h \mid x_{h-1}, a_{h-1}\right)\right)\right] - t\cdot\widehat{\mathbb{E}}_{\mathcal{D}_2}\left[w\left(x_h \mid x_{h-1}, a_{h-1}\right)\right]$.

Assuming access to an efficient weak learning oracle, our result implies that both computational and statistical complexities scale as $\mathrm{poly}(C_{\mathsf{push}}, H, |\mathcal{A}|, \log|\mathcal{W}|, \log(|\overline{\mathcal{W}}|)\alpha^{-1}, \varepsilon^{-1})$.

Specifically,

Proof of Theorem I.5. We follow the same proof procedure in L_∞ cover, we have the following:

$$V^{\pi^*} - V^{\pi} \leq \frac{1}{K}\sum_{k=1}^{K} V^{\pi^*} - V^{\pi^k}$$

We proceed to bound the RHS by decomposing the summation into the following three terms:

$$\sum_{k=1}^{K} V^{\pi^*} - V^{\pi^k} = \underbrace{\frac{1}{K}\sum_{k=1}^{K} V^{\pi^*} - \overline{V^{\pi^*}}_{\widehat{b}^k}}_{I_1} + \underbrace{\frac{1}{K}\sum_{k=1}^{K} \overline{V^{\pi^*}}_{\widehat{b}^k} - \overline{V^{\pi^k}}_{\widehat{b}^k}}_{I_2} + \underbrace{\frac{1}{K}\sum_{k=1}^{K} \overline{V^{\pi^k}}_{\widehat{b}^k} - V^{\pi^k}}_{I_3},$$

We have $I_1 \leq 0$ by the dynamics of the extended MDP.

For I_2 with $x \in \mathcal{B}^k$ can be upper bounded by $\frac{k-1}{\zeta}H\mathcal{E}(\mathbb{\Pi}, \Pi) + \frac{(k-1)H|\mathcal{A}|}{\zeta\alpha}(\varepsilon_{\mathsf{w}} + \frac{2}{\sqrt{N}})+$, follow by a similar argument in L_∞.

For I_2 with $x \notin \mathcal{B}^k$

$$\begin{aligned}
&\sum_{h=1}^{H} \mathbb{E}_{x_h\sim d^{\overline{\pi}}_h}\left[\overline{Q}_{\widehat{b}^k}\left(x_h; \overline{\pi}_h \circ \pi^k_{h+1:H}\right) - \overline{V}_{\widehat{b}^k}\left(x_h; \pi^k_{h:H}\right)\mathbb{1}\{x \notin \mathcal{B}^k, x \neq \mathfrak{t}\}\right] \\
&\leq \sum_{h=1}^{H} \mathbb{E}_{x_h\sim d^{\overline{\pi}}_h}\left[1 - 2\mathbb{E}^{\pi}_{x',a'}[\widehat{w}^k_h(x \mid x', a')]\mathbb{1}\{x \notin \mathcal{B}^k, x \neq \mathfrak{t}\}\right] \\
&\leq \sum_{h=1}^{H} \mathbb{E}_{x_h\sim d^{\overline{\pi}}_h}\left[1 - 2\mathbb{E}^{\pi}_{x',a'}[w^k_h(x \mid x', a')]\mathbb{1}\{x \notin \mathcal{B}^k, x \neq \mathfrak{t}\}\right] \\
&\qquad + 2\mathbb{E}_{x_h\sim d^{\overline{\pi}}_h}\left[\mathbb{E}^{\pi}_{x',a'}[w^k_h(x \mid x', a') - \widehat{w}^k_h(x \mid x', a')]\mathbb{1}\{x \notin \mathcal{B}^k, x \neq \mathfrak{t}\}\right]
\end{aligned}$$

We further bound the first term as follows:

$$\sum_{h=1}^{H} \mathbb{E}_{x_h\sim d^{\overline{\pi}}_h}\left[\sum_{x',a'\in\mathcal{X}\times\mathcal{A}} d^{\pi}(x', a')P_{h-1}(x_h \mid x', a')\widehat{w}^k_h(x_h \mid x', a')\mathbb{1}\{x \notin \mathcal{B}^k, x \neq \mathfrak{t}, P_{h-1}(x_h \mid x', a') < \zeta\}\right] < H\zeta$$

For the second term,

$$
\begin{aligned}
&\mathbb{E}_{x_h\sim d_h^{\overline{\pi}}}\left[\mathbb{E}^{\pi}_{x_{h-1},a_{h-1}}[w_h^k(x_h\mid x_{h-1},a_{h-1})-\widehat{w}_h^k(x\mid x_{h-1},a_{h-1})]\mathbb{1}\{x_h\notin\mathcal{B}^k,x_h\neq\mathfrak{t}\}\right]\\
&\le 2\mathbb{E}_{x_h\sim d_h^{\overline{\pi}}}\left[\sqrt{\mathbb{E}^{\pi}_{x_{h-1},a_{h-1}}\left[\left(\sqrt{w_h^k(x_h\mid x_{h-1},a_{h-1})}-\sqrt{\widehat{w}_h^k(x_h\mid x_{h-1},a_{h-1})}\right)^2\mathbb{1}\{x_h\notin\mathcal{B}^k,x_h\neq\mathfrak{t}\}\right]}\right]\\
&\le 2\sqrt{\mathbb{E}_{x_h\sim d_h^{\overline{\pi}}}\mathbb{E}^{\pi}_{x_{h-1},a_{h-1}}\left[\left(\sqrt{w_h^k(x_h\mid x_{h-1},a_{h-1})}-\sqrt{\widehat{w}_h^k(x_h\mid x_{h-1},a_{h-1})}\right)^2\mathbb{1}\{x_h\notin\mathcal{B}^k,x_h\neq\mathfrak{t}\}\right]}\\
&\le 2\sqrt{\sum_{\substack{x_h\in\mathcal{X}:\\x\neq\mathfrak{t}}}d_h^{\overline{\pi}}(x_h)\sum_{\substack{x',a'\in\mathcal{X}\times\mathcal{A}:\\x'\neq\mathfrak{t},a'\neq\mathfrak{t}}}d^{\pi}(x')\pi(a'\mid x')P_{h-1}(x_h\mid x',a')\left(\sqrt{w_h^k(x_h\mid x',a')}-\sqrt{\widehat{w}_h^k(x_h\mid x',a')}\right)^2}\\
&\le 2\sqrt{\sum_{\substack{x_h\in\mathcal{X}:\\x\neq\mathfrak{t}}}d_h^{\overline{\pi}}(x_h)\sum_{\substack{x',a'\in\mathcal{X}\times\mathcal{A}:\\x'\neq\mathfrak{t},a'\neq\mathfrak{t}}}(k-1)d^{p^k}(x')\pi(a'\mid x')P_{h-1}(x_h\mid x',a')\left(\sqrt{w_h^k(x_h\mid x',a')}-\sqrt{\widehat{w}_h^k(x_h\mid x',a')}\right)^2}\\
&\le 2\sqrt{|\mathcal{A}|\sum_{\substack{x_h\in\mathcal{X}:\\x\neq\mathfrak{t}}}\sum_{\substack{x',a'\in\mathcal{X}\times\mathcal{A}:\\x'\neq\mathfrak{t},a'\neq\mathfrak{t}}}(k-1)d^{p^k}(x')\pi_{\mathrm{unif}}(a'\mid x')P_{h-1}(x_h\mid x',a')\left(\sqrt{w_h^k(x_h\mid x',a')}-\sqrt{\widehat{w}_h^k(x_h\mid x',a')}\right)^2}\\
&\le 2\sqrt{\frac{(k-1)|\mathcal{A}|}{\zeta(k-1)}\sum_{i<t}\mathbb{E}^{\pi^i\circ_{h-1}\pi_{\mathrm{unif}}}\left[\left(\sqrt{w_h^k(x_h\mid x',a')}-\sqrt{\widehat{w}_h^k(x_h\mid x',a')}\right)^2\right]}\\
&\le\sqrt{\frac{(k-1)|\mathcal{A}|\varepsilon^2_{\mathsf{weight}}}{\zeta}}
\end{aligned}
$$

We have

$$
I_2\le\frac{k-1}{\zeta}H\mathcal{E}(\mathbb{\Gamma},\Pi)+\frac{(k-1)H|\mathcal{A}|}{\zeta\alpha}(\varepsilon_{\mathsf{w}}+\frac{2}{\sqrt{N}})+\sqrt{\frac{(k-1)|\mathcal{A}|\varepsilon^2_{\mathsf{weight}}}{\zeta}}+H\zeta
$$

For I_3,

$$
\begin{aligned}
I_3&=\frac{1}{K}\sum_{k=1}^{K}\mathbb{E}^{\pi^k}[\hat{w}_h^k(x_h\mid x_{h-1},x_{h-1})]\\
&\le\frac{1}{K}\sum_{h=1}^{H}\sum_{k=1}^{K}\left(3\mathbb{E}_h^{\pi^k}\left[w_h^k(x_h\mid x_{h-1},x_{h-1})\right]+2\mathbb{E}_h^{\pi^k}\left[\left(\sqrt{\hat{w}_h^k(x_h\mid x_{h-1},x_{h-1})}-\sqrt{w_h^k(x_h\mid x_{h-1},x_{h-1})}\right)^2\right]\right)
\end{aligned}
$$

Applying Lemma L.6, we have

$$
\begin{aligned}
&\sum_{k=1}^{K}\mathbb{E}_h^{\pi^k}[(\sqrt{\widehat{w}_h^k}-\sqrt{w_h^k})^2]\\
&\le\sqrt{2C_{\mathsf{push};h}\log(K)\sum_{\substack{k\in[K]\\i<k}}\mathbb{E}^{\pi^i}\left[(\mathbb{E}[(\sqrt{\widehat{w}_h^k(x_h\mid x_{h-1},a_{h-1})}-\sqrt{w_h^k(x_h\mid x_{h-1},a_{h-1})})^2\mid x_{h-1},a_{h-1}=\pi^k(x_{h-1})])^2\right]}\\
&\quad+4C_{\mathsf{push};h}\\
&\le\sqrt{8C_{\mathsf{push};h}|\mathcal{A}|\log(K)\sum_{k=1}^{K}\sum_{i<k}\mathbb{E}^{\pi^i\circ_{h-1}\pi_{\mathrm{unif}}}\left[\left(\sqrt{\widehat{w}_h^k\left(x_h\mid x_{h-1},a_{h-1}\right)}-\sqrt{w_h^k\left(x_h\mid x_{h-1},a_{h-1}\right)}\right)^2\right]}+4C_{\mathsf{push};h}\\
&\le\sqrt{8|\mathcal{A}|C_{\mathsf{push};h}\log(K)\varepsilon^2_{\mathsf{weight}}}+4C_{\mathsf{push}},
\end{aligned}
$$

where we use Lemma E.5 in the last inequality. The bound relate to the weight can be control by the following lemma:

Lemma I.6.

$$\sum_{k=1}^{K} \mathbb{E}_h^{\pi^k}\left[w_h^k(x_h \mid x_{h-1}, x_{h-1})\right] \leq C_{\mathsf{push}} \log(2K).$$

Therefore, we can bound I_3 as follows,

$$I_3 \leq \frac{4HC_{\mathsf{push}}\log(2K)}{K} + H\sqrt{8|\mathcal{A}|C_{\mathsf{push}}\log(K)\varepsilon_{\mathsf{weight}}^2} + \frac{4HC_{\mathsf{push}}}{K},$$

Put everything together, we have

$$\begin{aligned} V^{\pi^*} - V^{\pi} \leq \frac{KH}{\zeta}\mathcal{E}(\mathbb{\Pi}, \Pi) + \frac{KH|\mathcal{A}|}{\zeta\alpha}(\varepsilon_{\mathsf{w}} + \frac{2}{\sqrt{N}}) + H\sqrt{\zeta^{-1}K|\mathcal{A}|\varepsilon_{\mathsf{weight}}^2} \\ + \frac{8HC_{\mathsf{push}}\log(2K)}{K} + H\sqrt{8|\mathcal{A}|C_{\mathsf{push}}\log(K)\varepsilon_{\mathsf{weight}}^2} + H\zeta \end{aligned}$$

We complete the proof by choosing $\zeta = H^{-1}\varepsilon$. □

Proof of Lemma I.6. Let $\widetilde{d_h^k}(x_h) = \sum_{i=1}^{k-1} d_h^{\pi^i}(x_h)$, for any $h \in [H]$, let $\mu \in \Delta(\mathcal{X})$ attain the value of $C_{\mathsf{push};h}$, with $\epsilon = 1$, and $\delta = C_{\mathsf{push};h}$, applying Lemma L.4

$$\begin{aligned} \mathbb{E}^{\pi^k}\left[\frac{P_{h-1}(x_h \mid x_{h-1}, a_{h-1})}{\widetilde{d_h^k}(x_h) + P(x_h \mid x_{h-1}, a_{h-1})}\right] &\leq \mathbb{E}^{\pi^k}\left[\frac{P_{h-1}\left(x_h \mid x_{h-1}, a_{h-1}\right)}{\widetilde{d_h^k}(x_h) + C_{\mathsf{push};h}^M \mu\left(x_h\right)}\right] \\ &\quad + C_{\mathsf{push};h} \cdot \mathbb{E}^{\pi^k}\left[\frac{\mu\left(x_h\right)}{\widetilde{d_h^k}(x_h) + C_{\mathsf{push};h}\mu\left(x_h\right)}\right] \\ &\leq 2C_{\mathsf{push};h} \cdot \mathbb{E}^{\pi^k}\left[\frac{\mu\left(x_h\right)}{\widetilde{d_h^k}(x_h) + C_{\mathsf{push};h}^M \mu\left(x_h\right)}\right]. \end{aligned}$$

Since $\sup_{\pi\in\Pi} d_h^{\pi}(x) \leq \sup_{x'\in\mathcal{X}, a\in\mathcal{A}} P_{h-1}\left(x \mid x', a\right) \leq C_{\mathsf{push};h}^M \mu(x)$ for all $x \in \mathcal{X}$, Lemma L.3 implies that

$$\sum_{x\in\mathcal{X}}\sum_{k=1}^{K} \mu(x)\frac{d^{\pi^k}(x)}{\widetilde{d_h^k}(x) + C_{\mathsf{push};h}\mu(x)} \leq 2\log(2K),$$

which concludes the proof. □

I.3 Weight Function Estimation.

Lemma I.7. *For any $k \in [K]$, for any $\varepsilon_{\mathsf{weight}}, \delta \in (0,1)$, distribution $p^k \in \Delta(\mathbb{\Pi})$ and $\pi^1, \ldots, \pi^1 \in \mathbb{\Pi}$, Algorithm 11 ensures that with probability at least $1-\delta$, the output $\widehat{w}_h^k$ satisfies*

$$\frac{1}{k-1}\sum_{i<t} \mathbb{E}^{M^\star, \pi_h^i \circ_{h-1} \pi_{\mathrm{unif}}}\left[\left(\sqrt{\widehat{w}_h^i\left(x_h \mid x_{h-1}, a_{h-1}\right)} - \sqrt{w_h^i\left(x_h \mid x_{h-1}, a_{h-1}\right)}\right)^2\right] \leq \epsilon^2,$$

and does so using at most $N_{\mathsf{weight}} = 40\frac{\log(|\overline{\mathcal{W}}|\delta^{-1})}{\varepsilon_{\mathsf{weight}}^2}$ episodes.

Proof. This proof follows similarly to Lemma J.7 of Amortila et al. (2024), Let $\bar{w}_h^t := t \cdot w_h^t$ and let $\check{w}_h^t := t \cdot \widehat{w}_h^t$. Here, we solve the optimization problem over the class $t \cdot \mathcal{W}_h$, which has $\left\|w'\right\|_\infty \leq t$ for all $w' \in \mathcal{W}_h$. Similar to the proof in Amortila et al. (2024), we have

$$\mu\left(x' \mid x, a\right) = P_{h-1}\left(x' \mid x, a\right), \quad \nu\left(x' \mid x, a\right) = \frac{1}{t}\left(\sum_{i<t} d_h^{M^\star, \pi^{h,i}}\left(x'\right) + P_{h-1}\left(x' \mid x, a\right)\right),$$

and

$$\omega(x, a) = \frac{1}{t-1}\sum_{i<t} d_{h-1}\pi^{h,i} \circ_{h-1} \pi_{\text{unif}}(x, a).$$

Under the weight assumption, we have

$$\frac{\mu\left(x' \mid x, a\right)}{\nu\left(x' \mid x, a\right)} = \bar{w}_h^t\left(x' \mid x, a\right) \in t \cdot \mathcal{W}_h,$$

so Lemma E.5 imply that

$$\mathbb{E}_{\left(x_{h-1}, a_{h-1}\right)\sim\omega}\left[\left(\sqrt{\check{w}_h^t\left(x_h \mid x_{h-1}, a_{h-1}\right)} - \sqrt{\bar{w}_h^t\left(x_h \mid x_{h-1}, a_{h-1}\right)}\right)^2\right] \leq \frac{20t\log\left(|\mathcal{W}|\delta^{-1}\right)}{n},$$

or equivalently,

$$\mathbb{E}_{\left(x_{h-1}, a_{h-1}\right)\sim\omega}\left[\left(\sqrt{\widehat{w}_h^t\left(x_h \mid x_{h-1}, a_{h-1}\right)} - \sqrt{w_h^t\left(x_h \mid x_{h-1}, a_{h-1}\right)}\right)^2\right] \leq \frac{20\log\left(|\mathcal{W}|\delta^{-1}\right)}{n}.$$

□

I.4 Proof of Appendix H

Proof of Theorem H.2. We define $\|\cdot\|_{\infty,1} : \mathbb{R}^{|\mathcal{X}|\times|\mathcal{A}|} \to \mathbb{R}$ as $\|z\|_{\infty,1} = \max_{x\in\mathcal{X}}\sum_a |z_{x,a}|$, and the following lemma give the result of the smoothness of V function for this norm

Lemma I.8 (Smoothness). *For any two policies* $\pi^{(1)}, \pi^{(2)}$

$$\left|V^{\pi^{(1)}} - V^{\pi^{(2)}} - \langle\nabla_\pi V^{\pi^{(2)}}, \pi^{(1)} - \pi^{(2)}\rangle\right| \leq H^2 \cdot R_{\max}\left\|\pi^{(1)} - \pi^{(2)}\right\|_{\infty,1}^2,$$

in other words, V^π *is* $H^2 R_{\max}$ *in* $\|\cdot\|_{\infty,1}$ *norm*

Similar to Brukhim et al. (2022), the rest of the proof essentially following the Frank-Wolfe style optimization, we apply the gradient dominance for the V function, we state such result in the next lemma.

Lemma I.9 (Gradient domination). $V_1(\pi^*) - V_1(\pi) \leq C_\infty(H\mathcal{E}(\mathbb{\Pi}, \Pi) + \max_{\pi'\in\Pi^H}\langle\nabla_\pi V^\pi, \pi' - \pi\rangle)$

Remind that, the FW algorithm in Brukhim et al. (2022) define the boosting Oracle $\mathcal{O}$ as an $(\epsilon_0, \mathcal{K})$-approximate linear optimizer over a convex set $\mathcal{K} \subseteq \mathbb{R}^d$ such that for any given $v \in \mathbb{R}^d$, we have $v^\top\mathcal{O}(v) \geq \max_{u\in\mathcal{K}} v^\top u - \epsilon_{\mathcal{O}}$. In the context of boosting RL, $v = \nabla_\pi V^\pi$, and let $\hat{\pi} \leftarrow \mathcal{O}(\nabla_w V^\pi)$ be policy obtained from the Oracle. This Oracle will guarantee that $\langle\nabla_\pi V^\pi, \hat{\pi}\rangle \geq \sup_{\bar{\pi}}\langle\nabla_\pi V^\pi, \bar{\pi}\rangle - \varepsilon_{\mathcal{O}}$. Next, let $h_t = V_1(\pi^*) - V_1(\pi_t)$, we have

$$\begin{aligned}
h_t &\leq h_{t-1} - \eta_t\langle\nabla_\pi V^{\pi_{t-1}}, \hat{\pi}_t - \pi_{t-1}\rangle + \eta_t^2 H^2 R_{\max}\|\hat{\pi}_t - \pi_{t-1}\|_{\infty,1}^2 && \text{(Lemma L.9)}\\
&\leq h_{t-1} - \sup_{\bar{\pi}}\eta_t\langle\nabla_\pi V^{\pi_{t-1}}, \bar{\pi} - \pi_{t-1}\rangle + \eta_t^2 H^2 R_{\max}\|\hat{\pi}_t - \pi_{t-1}\|_{\infty,1}^2 + \eta_t\varepsilon_{\mathcal{O}} && \text{(Oracle guarantee)}\\
&\leq (1 - \frac{\eta_t}{C_\infty})h_{t-1} + \eta_t^2 H^2 R_{\max}\|\hat{\pi}_t - \pi_{t-1}\|_{\infty,1}^2 + \eta_t(\varepsilon_{\mathcal{O}} + H\mathcal{E}) && \text{(Gradient dominance, Lemma I.9)}
\end{aligned} \tag{24}$$

The Oracle for our method is PSDP Boosting, let $\hat{\pi}$ be the policy obtain from this algorithm, and $\bar{\pi} = \sup_{\pi'} \langle \nabla_w V^{\pi}, \pi' \rangle$, we proceed to bound $\langle \nabla_\pi V^{\pi}, \bar{\pi} - \pi \rangle$ by applying Lemma I.10:

$$\begin{aligned}\langle \nabla_\pi V^{\pi_{t-1}}, \bar{\pi} - \pi_{t-1} \rangle &= \sum_{h=1}^{H} \mathbb{E}_{x_h \sim d_h^{\pi_{t-1}}} \left[\sum_{a \in \mathcal{A}} Q_h^{\pi}(x_h, a) \left(\bar{\pi}_h(a \mid x_h) - \pi_h(a \mid x_h) \right) \right] \\ &\leq \frac{2H|\mathcal{A}|R_{\max}}{\alpha} \left(\varepsilon_{\mathsf{w}} + \frac{2}{\sqrt{N}} \right),\end{aligned}$$

where the last inequality comes from the Layer Boosting algorithm, which we state the following lemma.

Put everything together, we have

$$h_t \leq (1 - \frac{\eta_t}{C_\infty}) h_{t-1} + \eta_t H C_\infty (\mathcal{E} + \frac{2|\mathcal{A}|R_{\max}}{\alpha} (\varepsilon_{\mathsf{w}} + \frac{2}{\sqrt{N}})) + \eta_t^2 H^2 R_{\max}$$

Applying the following result of a bounded positive sequences

Claim 3 (Claim 21, (Brukhim et al., 2022)). *Let $C \geq 1$, let g_t be the B-bounded positive sequence such that*

$$h_t \leq (1 - \frac{\sigma_t}{C}) h_{t-1} + \sigma_t E + \sigma_t^2 D,$$

then choosing $\sigma_t = \min\{1, \frac{2C}{t}\}$ implies $g_t \leq \frac{2C^2 \max\{2D, B\}}{t} + E$

We can bound $V^{\pi^*} - V^{\pi_T}$ as follows

$$V^{\pi^*} - V^{\pi_T} \leq \frac{2C_\infty^2 H^2 R_{\max}}{T} + C_\infty H \mathcal{E} + \frac{2HC_\infty |\mathcal{A}| R_{\max}}{\alpha} \varepsilon_{\mathsf{w}} + \frac{4HC_\infty |\mathcal{A}| R_{\max}}{\alpha \sqrt{N}},$$

which completes the proof. □

I.5 Proof of supporting claims

Proof of Lemma H.1. Similarly to Brukhim et al. (2022, Claim 10), the result is a direct application of Hazan and Singh (2021, Theorem 13) where we use the fact that V function is $|\mathcal{A}|R_{\max}$-lipschitz. In Brukhim et al. (2022), the internal boosting give the following guarantee: $\max_{\pi \in \Pi} \mathbb{E}_{s \sim d^{\pi_{t-1}}} [Q^{\pi_{t-1}}(s, \cdot)^\top \pi(\cdot \mid s) - Q^{\pi_{t-1}}(s, \cdot)^\top \hat{\pi}(\cdot \mid s)] \leq (2|\mathcal{A}|/(1-\gamma)\alpha)(\varepsilon + 2/\sqrt{N})$, where N is the number of weak learner, and $Q(\cdot, \cdots)$ is bounded by $\frac{1}{1-\gamma}$ □

Lemma I.10. *Policy gradient theorem for episodic MDPs*

$$\frac{\partial V_1^{\pi}}{\partial \pi_h(a_h \mid x_h)} = d_h^{\pi}(x_h) Q_h^{\pi}(x_h, a_h)$$

Proof of Lemma I.10.

$$\begin{aligned}V_1^{\pi} &= \mathbb{E}_{x_1 \sim \mu_1} \left[\sum_{h=1}^{H} r(x_h, a_h) \;\middle|\; \pi \right] \\ &= \mathbb{E}_{x_1 \sim \mu_1} \left[\sum_{h'=1}^{h-1} r(x_{h'}, a_{h'}) \;\middle|\; \pi_{1:h-1} \right] + \mathbb{E}_{x_h \sim d_h^{\pi}} \left[\sum_{h'=h}^{H} r(x_{h'}, a_{h'}) \;\middle|\; \pi_{h:H} \right] \\ &= \mathbb{E}_{x_1 \sim \mu_1} \left[\sum_{h'=1}^{h-1} r(x_{h'}, a_{h'}) \;\middle|\; \pi_{1:h-1} \right] + \mathbb{E}_{x_h \sim d^{\pi_{1:h-1}}} \left[\sum_{a \in \mathcal{A}} \pi_h(a \mid x_h) Q(x_h, a; \pi_{h+1:H}) \right]\end{aligned}$$

Since the first term has no dependence on π_h, taking derivative w.r.t. $\pi_h(a_h, x_h)$ will set it to zero. For the second term, we have d_h^π is the occupancy measure of executing policy $\pi_{1:h-1}$. Therefore, we can write the second term as

$$\mathbb{E}_{x_h \sim d_h^\pi}\left[\sum_{h'=h}^{H} r(x_{h'}, a_{h'}) \,\middle|\, \pi_{h:H}\right] = \mathbb{E}_{x_h \sim d^{\pi_{1:h-1}}}\left[\sum_{a\in\mathcal{A}} \pi_h(a \mid x_h) Q(x_h, a; \pi_{h+1:H})\right]$$

Take the partial derive w.r.t. $\pi_h(a \mid x)$ completes the proof. □

Proof of Lemma I.9. Apply performance different lemma for $V_1(\pi^*) - V_1(\pi)$, we have

$$\begin{aligned}
V_1(\pi^*) - V_1(\pi) &= \sum_{h=1}^{H} \mathbb{E}_{x_h \sim d_h^{\pi^*}}\left[\sum_{a\in\mathcal{A}} Q^\pi(x_h, a; \pi)(\pi^*(a \mid x_h) - \pi_h(a \mid x_h))\right] \\
&= \sum_{h=1}^{H} \mathbb{E}_{x_h \sim d_h^{\pi^*}}\Bigg[\underbrace{\sum_{a\in\mathcal{A}} Q_h^\pi(x_h, a)\pi^*(a \mid x_h) - \max_a Q_h^\pi(x_h, a)}_{<0} \\
&\qquad + \max_a Q^\pi(x_h, a) - \sum_{a\in\mathcal{A}} Q_h^\pi(x_h, a)\pi_h(a \mid x_h)\Bigg] \\
&\le \max_{h\in[H]} \max_{x_h \in \mathcal{X}_h} \frac{d_h^{\pi^*}(x_h)}{d_h^\pi(x_h)} \sum_{h=1}^{H} \mathbb{E}_{x_h \sim d_h^\pi}\left[\max_a Q_h^\pi(x_h, a) - \sum_{a\in\mathcal{A}} Q_h^\pi(x_h, a)\pi_h(a \mid x_h)\right],
\end{aligned} \tag{25}$$

in the last inequality we can take the mismatch ratio $C_\infty = \max_{h\in[H]} \max_{x_h\in\mathcal{X}_h} \frac{d_h^{\pi^*}(x_h)}{d_h^\pi(x_h)}$ outside since the term inside expectation is non-negative. Additionally, for any $h \in [H]$, the expectation can be written as,

$$\begin{aligned}
&\mathbb{E}_{x_h \sim d_h^\pi}\left[\max_a Q(x_h, a) - \sum_{a\in\mathcal{A}} Q(x_h, a)\pi_h(a \mid x_h)\right] \\
&\quad + \max_{\tilde{\pi}_h \in \Pi} \mathbb{E}_{x_h \sim d_h^\pi}\left[\sum_{a\in\mathcal{A}} Q(x_h, a)(\tilde{\pi}_h(a \mid x_h) - \pi_h(a \mid x_h))\right] \\
&\le \mathcal{E} + \mathbb{E}_{x_h \sim d_h^\pi}\left[\sum_{a\in\mathcal{A}} Q(x_h, a)(\pi'_h(a \mid x_h) - \pi(a \mid x_h))\right]
\end{aligned}$$

Apply Lemma I.10, we have

$$\langle \nabla_\pi V^\pi, \pi' - \pi\rangle = \sum_{h=1}^{H} \mathbb{E}_{x_h \sim d_h^\pi}\left[\sum_{a\in\mathcal{A}} Q_h^\pi(x_h, a)(\pi'_h(a \mid x_h) - \pi_h(a \mid x_h))\right]$$

which concludes the proof. □

Proof of Lemma I.8. Apply the performance difference Lemma L.2 and Lemma I.10, we have

$$
\begin{aligned}
&\left|V^{\pi^{(1)}} - V^{\pi^{(2)}} - \langle \nabla_\pi V^{\pi^{(2)}}, \pi^{(1)} - \pi^{(2)} \rangle\right| \\
&= \left| \sum_{h=1}^{H} \mathbb{E}_{x_h \sim d^{\pi^{(1)}_{1:h-1}}} \left[\sum_{a \in \mathcal{A}} Q(x_h, a; \pi_{h+1:H})(\pi_h^{(1)}(a \mid x_h) - \pi_h^{(2)}(a \mid x_h)) \right] \right. \\
&\quad \left. - \sum_{h=1}^{H} \mathbb{E}_{x_h \sim d^{\pi^{(2)}_{1:h-1}}} \left[\sum_{a \in \mathcal{A}} Q(x_h, a; \pi_{h+1:H})(\pi_h^{(1)}(a \mid x_h) - \pi_h^{(2)}(a \mid x_h)) \right] \right| \\
&\le \sum_{h=1}^{H} \left| \sum_{x_h \in \mathcal{X}_h} (d^{\pi^{(1)}_{1:h-1}}(x_h) - d^{\pi^{(2)}_{1:h-1}}(x_h)) \sum_{a \in \mathcal{A}} Q(x_h, a; \pi_{h+1:H})(\pi_h^{(1)}(a \mid x_h) - \pi_h^{(2)}(a \mid x_h)) \right| \\
&\le R_{\max} \sum_{h=1}^{H} \left| \sum_{x_h \in \mathcal{X}_h} d^{\pi^{(1)}_{1:h-1}}(x_h) - d^{\pi^{(2)}_{1:h-1}}(x_h) \right| \max_{h \in [H]} \max_{x_h \in \mathcal{X}_h} \left| \sum_{a \in \mathcal{A}} \pi_h^{(1)}(a \mid x_h) - \pi_h^{(2)}(a \mid x_h) \right| \\
&\le R_{\max} \sum_{h=1}^{H} \left| \sum_{x_h \in \mathcal{X}_h} d^{\pi^{(1)}_{1:h-1}}(x_h) - d^{\pi^{(2)}_{1:h-1}}(x_h) \right| \max_{h \in [H]} \left\| \pi^{(1)} - \pi^{(2)} \right\|_{\infty,1},
\end{aligned}
$$

where we use the fact that $Q(x_h, a; \pi_{h+1:H}) \le R_{\max}$. Now, for any given $h \in [H]$,

$$
\begin{aligned}
&\left| \sum_{x_h \in \mathcal{X}_h} d^{\pi^{(1)}_{1:h-1}}(x_h) - d^{\pi^{(2)}_{1:h-1}}(x_h) \right| \\
&= \left| \sum_{x_h \in \mathcal{X}_h} \sum_{x_{h-1} \in \mathcal{X}_{h-1}} P(x_h \mid x_{h-1}, a) \pi_h(a \mid x_{h-1}) d^{\pi^{(1)}_{1:h-1}}(x_{h-1}) \right. \\
&\qquad \left. - \sum_{x_h \in \mathcal{X}_h} \sum_{x_{h-1} \in \mathcal{X}_{h-1}} P(x_h \mid x_{h-1}, a) \pi_h(a \mid x_{h-1}) d^{\pi^{(2)}_{1:h-1}}(x_{h-1}) \right| \\
&\le H \left\| \pi_h^{(1)} - \pi_h^{(2)} \right\|_{\infty,1}
\end{aligned}
$$

Therefore,

$$
\sum_{h=1}^{H} \left| \sum_{x_h \in \mathcal{X}_h} d^{\pi^{(1)}_{1:h-1}}(x_h) - d^{\pi^{(2)}_{1:h-1}}(x_h) \right| \le H^2 \left\| \pi^{(1)} - \pi^{(2)} \right\|_{\infty,1},
$$

which completes the proof. □

J Discussion on Policy Cover

In this section, we provide a brief discussion on the policy cover with boosting. Note that boosting is also a type of ensemble method, where we aggregate many policies to form a final policy. It would be interesting to ask if the policy from boosting can be used as a policy cover that can well explored the state space. Our first observation is that if the policies share the same partial policy in the previous layer $h-1$, then we might be able to use boosting to decouple the occupancy measure h; we formalized this property in the following claim.

Claim 4. *Let $\pi_{1:h-1}$ denote the partial policy over the first $h-1$ layers, and consider the boosted policy at layer h defined as $\pi_h^t = \eta_t \pi_h^{t-1} + (1-\eta_t)\hat{\pi}_h^t$. Then, for any $x_h \in \mathcal{X}_h$ the corresponding occupancy measure satisfies $d_h^{\pi_{1:h-1} \circ_h \pi_{h,t}}(x) = (1-\eta_t) d_h^{\pi_{1:h-1} \circ_h \pi_h^{t-1}}(x) + \eta_t d_h^{\pi_{1:h-1} \circ_h \hat{\pi}_h^t}(x)$.*

While this property is promising and suggests that boosting can be applied in a layer-wise manner, however, the resulting boosted policy remains a *single policy*, which may not be as robust as a set of separate policy in a standard policy cover. Furthermore, ensemble method in boosting does not guarantee the occupancy decoupling as we shown in the claim below.

Claim 5. *Consider a set of non-stationary policies $\pi^1, \cdots, \pi^n$, and let $\{\alpha_i\}_{i=1}^n \in \Delta^{n-1}$ be a set of weight on the simplex. Then, for the following ensemble method:*

- ***Mixture policy ensemble*** *(Amortila et al., 2024). Let p be the distribution over $\pi^1, \ldots, \pi^n$ given by the weights $\{\alpha_i\}_{i=1}^n$, the induced occupancy sastisfies, $d^p(x) = \sum_{i\in[n]} \alpha_i d^{\pi^i}(x)$.*
- ***Boosting policy ensemble.*** *Let $\bar{\pi} = \sum_{i\in[n]} \alpha_i \pi^i$ be the boosted policy, this* does not *implies $d^{\bar{\pi}}(x) = \sum_{i\in[n]} \alpha_i d^{\pi^i}(x)$.*

The fundamental difference between the two ensemble methods is that the mixture policy ensemble maintains a set of policies, and samples a policy from this set to roll out the trajectory, whereas the boosted policy ensemble is a single policy that is the convex combination of the ensemble of many policies. Although both methods are ensemble methods, at first glance, the boosting approach seems more restricted since it only outputs a single valid policy at the end. As shown by Misra et al. (2020), a single policy might fail to achieve sufficient coverage using their minimum reachability argument. This suggests that we might still need to maintain a set of policies to ensure good coverage. Our analysis arrives at the same conclusion from the perspective of boosting.

Proof of Claim 4. We write $d^\pi(x) = P(x;\pi)$ to avoid the long superscript, then for any $x_h \in \mathcal{X}$,

$$\begin{aligned}
P(x_h; \pi_{1:h-1} \circ \pi_h^t) &= \sum_{x_{h-1}} \sum_{a\in\mathcal{A}} P(x_h \mid x_{h-1}, a) P(x_{h-1}; \pi_{1:h-1}) \pi_h^t(a \mid x_{h-1}) \\
&= \sum_{x_{h-1}} \sum_{a\in\mathcal{A}} P(x_h \mid x_{h-1}, a) P(x_{h-1}; \pi_{1:h-1}) \left((1-\eta_t)\pi_h^{t-1}(a \mid x_{h-1}) + \eta_t \hat{\pi}_h^t(a \mid x_{h-1})\right) \\
&= (1-\eta_t) \sum_{x_{h-1}} \sum_{a\in\mathcal{A}} P(x_h \mid x_{h-1}, a) P(x_{h-1}; \pi_{1:h-1}) \pi_h^{t-1}(a \mid x_{h-1}) \\
&\quad + \eta_t \sum_{x_{h-1}} \sum_{a\in\mathcal{A}} P(x_h \mid x_{h-1}, a) P(x_{h-1}; \pi_{1:h-1}) \hat{\pi}_h^t(a \mid x_{h-1}) \\
&= (1-\eta_t) P(x_h; \pi_{1:h-1} \circ \pi_h^{t-1}) + \eta_t P(x_h; \pi_{1:h-1} \circ \hat{\pi}_h^t),
\end{aligned}$$

which completes the proof. □

Proof of Claim 5.

Mixture policy ensemble. This can be easily verified since we pick a policy w.r.t. Δ^{n-1}.

Boosted (averaged) policy ensemble. To prove this second point, we show that there exist a MDP and and a set of policies $\{\pi_i\}_{i=1}^n$, that $d^{\bar{\pi}}(x) \neq \sum_{i\in[n]} \alpha_i d^{\pi_i}(x)$, where $\bar{\pi} = \sum_{i\in[n]} \alpha_i \pi_i$. Let consider an MDP with H=3, and two policies π_1 and π_2. Let $\alpha \in (0,1)$ and define the boosted policy as: $\pi = \alpha\pi_1 + (1-\alpha)\pi_2$. The first state is sampled from the initial distribution, $x_1 \sim P_0$. For any $x_2 \in \mathcal{X}_2$, we can show that the occupancy of the boosted policy can be decouple as $d^{\bar{\pi}}(x_2) = \alpha d^{\pi}(x_2) + (1-\alpha) d^{\pi_2}(x_2)$ by simply applying the definition of occupancy measure:

$$\begin{aligned}
d^{\bar{\pi}}(x_2) &= \sum_{x_1\in\mathcal{X}_1} P(x_2 \mid x_1, a) \sum_a (\alpha\pi_1(a \mid x_1) + (1-\alpha)\pi_2(a \mid x_1)) P(x_1) \\
&= \alpha \sum_{x_1\in\mathcal{X}_1} P(x_2 \mid x_1, a) \sum_a \pi_1(a \mid s_1) P(x-1) + (1-\alpha) \sum_{x_1\in\mathcal{X}_1} P(x_2 \mid x_1, a)\pi_2(a \mid s_1)) P(x_1) \\
&= \alpha d^{\pi_1}(x_2) + (1-\alpha) d^{\pi_2}(x_2)
\end{aligned}$$

Now, for $d^{\bar{\pi}}(x_3)$,

$$\begin{aligned}
d^{\bar{\pi}}(x_3) &= \sum_{x_2\in\mathcal{X}_2} \sum_a P(x_3 \mid x_2, a)(\alpha\pi_1(a \mid x_2) + (1-\alpha)\pi_2(a \mid x_2)) d^{\bar{\pi}}(x_2) \\
&= \alpha^2 d^{\pi_1}(x_3) + (1-\alpha)^2 d^{\pi_2}(x_3) \\
&\quad + \alpha(1-\alpha) \sum_{x_2\in\mathcal{X}_2} P(x_3 \mid x_2, a) \sum_a (\pi_1(a \mid x_2) d^{\pi_2}(x_2) + \pi_2(a \mid x_2) d^{\pi_1}(x_2)).
\end{aligned}$$

Let us consider the occupancy measure corresponding to $d^{\pi_1}(x_3)$:

$$\alpha^2 d^{\pi_1}(x_3) + \alpha(1-\alpha) \sum_{x_2 \in \mathcal{X}_2} P(x_3 \mid x_2, a) \sum_a \pi_2(a \mid x_2) d^{\pi_1}(x_2),$$

In general, this term is not equal to $\alpha d^{\pi_1}(x_3)$ since the summation over the action depends on π_2; this argument also completes the proof. □

We can see that even with two policies, the occupancy can be coupled and become very complicated as we move forward, with just a few more layers.

K Experiments

While the primary contribution of this work is theoretical, we include empirical evaluations to serve as a sanity check for our theory. We leave comprehensive experiments in a more complex environment for future work. We validate our algorithm on two classical control tasks from the OpenAI Gym benchmark: CartPole and MountainCar (Brockman et al., 2016). In CartPole, the agent applies a left or right force to a cart in order to balance an upright pole for as long as possible. In MountainCar, the agent must learn to first climb the left hill to build sufficient momentum to reach the goal at the top of the right hill.

We discretize the continuous state space of each environment into a finite number of bins along each dimension. Based on this discretization, we estimate the occupancy measure d^π using a count-based estimation over the discretized space, which is then used to construct the policy cover and the corresponding exploration bonus. We employ a depth-3 decision tree implemented using Scikit-learn (Pedregosa et al., 2011) as the weak learner. We use a total of 50 weak learners in our experiments. At each boosting iteration, the weak learner is trained to optimize the bonus-augmented objective induced by the current policy cover. We use the L_∞ coverability in the experiment, and the bonus is $b(x,a) = \frac{2}{1-\gamma} \frac{1}{\sum_{i<t} d^{\pi^i}(x,a)+1}$ since $\widetilde{C}_\infty = |\mathcal{X}||\mathcal{A}|$, and $\mu(x,a) = \frac{1}{|\mathcal{X}||\mathcal{A}|}$.

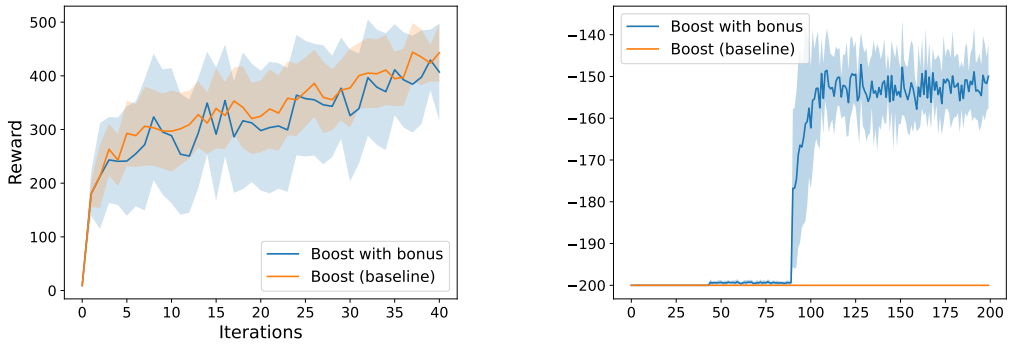

Figure 2: The total reward of cartpole (left) and mountain car (right). We compare our method with the baseline boosting of Brukhim et al. (2022). We run each experiment 10 times using boosting with 50 decision trees as weak learners, and report the total reward along with confidence intervals. Our method is competitive with the base-line on the dense-reward environment such as CartPole. For MountainCar, a sparse reward environment, our method succeeds while the baseline failed.

We also provide an example run of boosting MountainCar in Figure 3, along with a heatmap illustration of how the policy explores the state space. The MountainCar environment state consists of the car's position and velocity, which we discretize into 10 bins per dimension, resulting in 100 discrete states in total. As shown in the figure, the number of unique states increases steadily as more policies are added to the policy cover. The heat map highlights the regions of the state space visited by the policy, indicating that the policy gradually expands its coverage while simultaneously focusing on trajectories that lead toward the goal, balancing exploration and exploitation. All of our experiments run on a machine with 8 cores CPU and 32GB RAM.

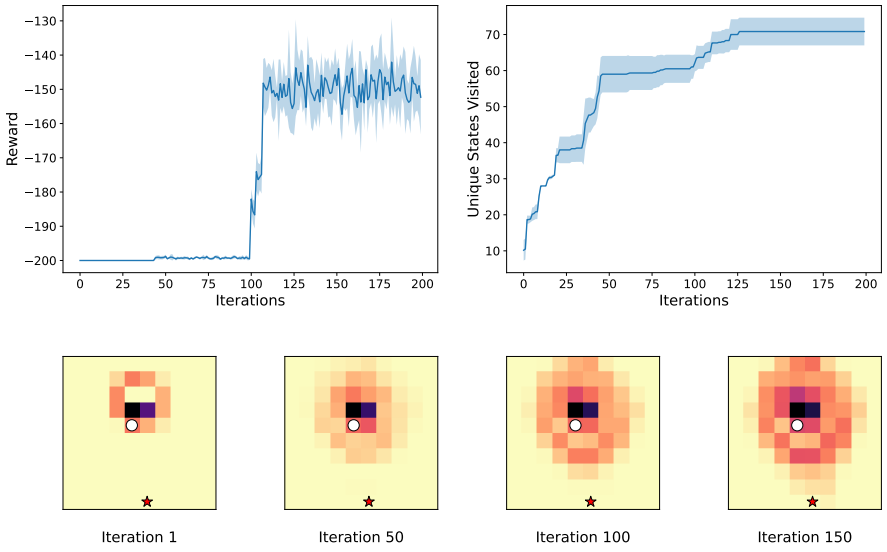

Figure 3: MountainCar results with $N = 50$ weak learners. **Top:** average episodic reward and number of unique states visited. **Bottom:** state visitation heatmaps, where the x-axis denotes velocity, and the y-axis denotes position. The start state is marked by $\circ$, and the goal state at zero velocity is marked by $\star$.

L Supporting lemmas

Lemma L.1 (The performance difference lemma for discounted MDPs, (Kakade and Langford, 2002)). *For all policies π, π' and states x_0,*

$$V^{\pi}(x_0) - V^{\pi'}(x_0) = \frac{1}{1-\gamma}\mathbb{E}_{x\sim d^{\pi}_{x_0}}\mathbb{E}_{a\sim\pi(\cdot|x)}\left[A^{\pi'}(x,a)\right]$$

Lemma L.2 (Performance Difference Lemma for episodic MDPs, Lemma 13 (Misra et al., 2020),). *For any episodic decision process with any reward function R, and any two non-stationary policies $\pi^{(1)}_{1:H}$ and $\pi^{(2)}_{1:H}$, let $d_h^{\pi^{(1)}} \in \Delta(\mathcal{X}_h)$ be the distribution at time h induced by policy $\pi^{(1)}_{1:H}$. Then we have*

$$V\left(\pi^{(1)}_{1:H}\right) - V\left(\pi^{(2)}_{1:H}\right) = \sum_{h=1}^{H}\mathbb{E}_{x_h\sim d_h^{\pi^{(1)}}}\left[V\left(x_h;\pi^{(1)}_h\circ\pi^{(2)}_{h+1:H}\right) - V\left(x_h;\pi^{(2)}_{h:H}\right)\right].$$

Lemma L.3 (Lemma 4 (Xie et al., 2023)). *Let $d^1, d^2, \ldots, d^K$ be an arbitrary sequence of distributions over a set $\mathcal{Z}$, and let $\mu \in \Delta(\mathcal{Z})$ be a distribution such that $d^k(z)/\mu(z) \le C$ for all $(z,t) \in \mathcal{Z}\times[T]$. Then for all $z \in \mathcal{Z}$, we have*

$$\sum_{t=1}^{K}\frac{d^t(z)}{\sum_{i<k}d^i(z) + C\cdot\mu(z)} \le 2\log(2K)$$

Lemma L.4 (Lemma C.11 Amortila et al. (2024)). *For any $\pi \in \Pi_{\mathrm{rns}}, d \in \mathbb{R}_+^{\mathcal{X}}, \mu \in \mathbb{R}_+^{\mathcal{X}}$, and $\varepsilon, \delta > 0$, we have that*

$$\mathbb{E}^{\pi}\left[\frac{P_{h-1}\left(x_h \mid x_{h-1}, a_{h-1}\right)}{d\left(x_h\right) + \varepsilon\cdot P_{h-1}\left(x_h \mid x_{h-1}, a_{h-1}\right)}\right] \le \mathbb{E}^{\pi}\left[\frac{P^M_{h-1}\left(x_h \mid x_{h-1}, a_{h-1}\right)}{d\left(x_h\right) + \delta\cdot\mu\left(x_h\right)}\right] + \frac{\delta}{\varepsilon}\cdot\mathbb{E}^{\pi}\left[\frac{\mu\left(x_h\right)}{d\left(x_h\right) + \delta\cdot\mu\left(x_h\right)}\right].$$

Lemma L.5 (Lemma J.2 Amortila et al. (2024)). *For any distribution $\omega \in \Delta(\mathcal{Z})$ and any pair of functions $w, w' : \mathcal{Z} \to \mathbb{R}_+$,*

$$\mathbb{E}_{\omega}[w] \le 3\mathbb{E}_{\omega}\left[w'\right] + 2\mathbb{E}_{\omega}\left[\left(\sqrt{w} - \sqrt{w'}\right)^2\right],$$

if $w(z), w'(z) \in [0, B], \forall z \in \mathcal{Z}$ then,

$$|\mathbb{E}_{\omega}[w] - \mathbb{E}_{\omega}[w']| \le 2B\sqrt{\mathbb{E}_{\omega}[(\sqrt{w} - \sqrt{w'})^2]}.$$

Lemma L.6 (Lemma J.3 (Amortila et al., 2024)). *Consider a set $\mathcal{Z}$ and a sequence of distributions $d^1, \ldots, d^K \in \Delta(\mathcal{Z})$ for which there exists a distribution $\mu \in \Delta(\mathcal{Z})$ such that $\sup_{z\in\mathcal{Z}}\left\{\frac{d^k(z)}{\mu(z)}\right\} \le C$ for all $k \in [K]$. For any sequence of functions $g^1, \ldots, g^K \subset (\mathcal{Z} \to [-B, B])$, it holds that*

$$\sum_{k=1}^{K}\mathbb{E}_{z\sim d^k}[g(z)] \le \sqrt{2C\log(2K)\sum_{k=1}^{K}\sum_{i<k}\mathbb{E}_{z\sim d^i}\left[\left(g^i(z)\right)^2\right]} + 2CB.$$

L.1 Boosting Framework (Brukhim et al., 2022)

In this section, we provide the subroutines from Brukhim et al. (2022) that were omitted from the main text. Internal Boost (Algorithm 12) instantiates Algorithm 3 from Hazan and Singh (2021). At a high level, Internal Boost can be viewed as an *offline stochastic contextual optimization* problem; it is a straightforward application of Theorem 13 from Hazan and Singh (2021), and we restate its guarantees in the following lemma:

Lemma L.7 (Claim 10 Brukhim et al. (2022)). *Let* $\beta = \sqrt{1/\alpha N}, \eta_{2,n} = \min\{2/n, 1\}, \|Q\| \in [0, B]$. π'_t *produced by Algorithm 1 satisfies* $\max_{\pi\in\Pi} \mathbb{E}_{(s,Q)\sim\mathcal{D}_t}\left[Q^\top \pi(s)\right] - \mathbb{E}_{(s,Q)\sim\mathcal{D}_t}\left[Q^\top \pi'_t(s)\right] \le \frac{2B}{\alpha}(\varepsilon_{\mathsf{w}} + 2/\sqrt{N})$.

Algorithm 12 Internal Boost (Algorithm 2 of Brukhim et al. (2022))

1: **Input**: number of iterations N, number of episodes M, initial policy π, initial state distribution μ.
2: Set $\tilde{\pi}_0$ to be an arbitrary policy in Π.
3: **for** $n = 1$ **to** N **do**
4: Execute π with μ via Algorithm 13 for M episodes, to get $D_n = \{(s_i, \widehat{Q}_i)_{i=1}^M\}$.
5: Modify D_n to produce a new dataset $D'_n = \{(s_i, f_i)\}_{i=1}^M$, such that for all $i \in [m]$:

$$f_i = \frac{1}{\beta}\left(y_i - \tilde{\pi}_n(\cdot|s_i)\right),$$

$$y_i = \operatorname*{argmin}_{y\in\mathbb{R}^{|\mathcal{A}|}} \left\{ -\widehat{Q}_i^\top y + G \min_{z\in\Delta_{\mathcal{A}}} \|z - y\| + \frac{\|\tilde{\pi}_n(\cdot|s_i) - y\|^2}{2\beta} \right\},$$

where $G = \frac{|\mathcal{A}|}{1-\gamma}, \beta = \frac{2\gamma}{(1-\gamma)^3}$ and $f_i, \widehat{Q}_i \in \mathbb{R}^{|\mathcal{A}|}$.
6: Let $\mathcal{L}_n$ be the policy chosen by the weak learning oracle when given data set D'_n.
7: Update

$$\tilde{\pi}_n = (1 - \eta_{2,n})\tilde{\pi}_{n-1} + \frac{\eta_{2,n}}{\alpha}\mathcal{L}_n.$$

8: **end for**
9: **return** $\Gamma\left[\tilde{\pi}_N\right]$.

Algorithm 13 Trajectory Sampler: samples a state $s \sim d^\pi$, and an unbiased estimate of Q^π_s

1: Sample state $s_0 \sim \mu$, action $a' \sim \mathcal{U}(\mathcal{A})$ uniformly.
2: Sample $s \sim d^\pi$ as follows: at every timestep h, with probability γ, act according to π; else, accept s_h as the sample and proceed to Step 3.
3: Take action a' at state s_h, then continue to execute π, and use a termination probability of $1 - \gamma$. Upon termination, set $R(s_h, a')$ as the *undiscounted* sum of rewards from time h onwards.
4: Define the vector $\widehat{Q^\pi_{s_h}}$, such that for all $a \in \mathcal{A}$, $\widehat{Q^\pi_{s_h}}(a) = |\mathcal{A}| \cdot R(s_h, a') \cdot \mathbb{I}_{a=a'}$.
5: **return** $(s_h, \widehat{Q^\pi_{s_h}})$.

An important generalization of the property of convexity is gradient domination, which we restate below.

Definition L.8 (Gradient Domination). A function $f : \mathcal{K} \to \mathbb{R}$ is said to be $(\kappa, \tau, \mathcal{K}_1, \mathcal{K}_2)$-locally gradient dominated (around $\mathcal{K}_1$ by $\mathcal{K}_2$) if for all $x \in \mathcal{K}_1$, it holds that

$$\max_{y\in\mathcal{K}} f(y) - f(x) \le \kappa \cdot \max_{y\in\mathcal{K}_2} \left\{ \nabla f(x)^\top (y - x) \right\} + \tau.$$

The following smoothness property of the V function is also helpful for our analysis that we use on Appendix D and Appendix E.

Lemma L.9 (Lemma 16 (Brukhim et al., 2022)). V^π *is* $\frac{2B\gamma}{(1-\gamma)^2}$*-smooth in the* $\|\cdot\|_{\infty,1}$ *norm, for* $V^\pi(x) \in [0, B], \forall x \in \mathcal{X}$.

The boosting in Algorithm 1 can be cast as a non-convex Frank-Wolfe algorithm, which we provide the pseudo-code in Algorithm 14, we denote by $\mathcal{O}$ a black-box oracle to an $(\epsilon_0, \mathcal{K}_2)$-approximate linear optimizer over a convex set $\mathcal{K} \subseteq \mathbb{R}^d$ such that for any given $v \in \mathbb{R}^d$, we have $v^\top \mathcal{O}(v) \geq \max_{u \in \mathcal{K}_2} v^\top u - \epsilon_0$.

Algorithm 14 Non-convex Frank-Wolfe

Input: $T > 0$, objective f, linear optimizer $\mathcal{O}$, rate η_t.
Choose $x_0 \in \mathcal{K}$ arbitrarily.
for $t = 1, \ldots, T$ **do**
    Call $z_t = \mathcal{O}(\nabla_{t-1})$, where $\nabla_{t-1} = \nabla f(x_{t-1})$. Set $x_t = (1 - \eta_t)x_{t-1} + \eta_t z_t$.
end for
return $\bar{x} := x_{t'}$ where $t' = \operatorname{argmin}_t \nabla_t^\top (z_t - x_t)$.

We provide the guarantee of Algorithm 14 as follows:

Lemma L.10 (Theorem 9 (Brukhim et al., 2022)). *Let $f : \mathcal{K} \to \mathbb{R}$ be L-smooth in some norm $\|\cdot\|_*$, bounded for all $x \in \mathcal{K}$, $|f(x)| \leq H$ for some $H > 0$, and let the diameter of $\mathcal{K}$ in $\|\cdot\|_*$ be D. Then, for a $(\epsilon_0, \mathcal{K}_2)$-linear optimization oracle $\mathcal{O}$, and $\eta_t = \eta = \sqrt{\frac{4H}{LD^2T}}$, the output $\bar{x}$ of Algorithm 14 satisfies*

$$\max_{x^* \in \mathcal{K}} f(x^*) - f(\bar{x}) \leq \frac{2\kappa^2 \max\{LD^2, H\}}{T} + \tau + \kappa\epsilon_0$$

Furthermore, if f is $(\kappa, \tau, \mathcal{K}_1, \mathcal{K}_2)$-locally gradient-dominated and $x_0, \ldots x_T \in \mathcal{K}_1$, then the output $\bar{x}$ of Algorithm 14 where $\eta_t = \min\{1, \frac{2\kappa}{t}\}$ satisfies the bound on the right.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The main theoretical results in Section 3 directly support the claims stated in the abstract and introduction. In particular, we provide guarantees that relax the bounded distribution mismatch assumption required in prior RL boosting work, which constitutes the central contribution of the paper.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: We discuss limitations and possibilities for future work in the second paragraph of the conclusion section.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

Answer: [Yes]

Justification: We explain the detailed setup in Section 2, provide all the assumptions along with the main results in Theorem 3.2 and Theorem 3.5, we provide complete proofs for each result in the Appendix section.

Guidelines:

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

4. **Experimental result reproducibility**

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

Answer: [Yes]

Justification: The experiment section provides the necessary details for replication, including the specific type of the weak learner and the construction of the exploration bonus.

Guidelines:

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

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

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

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

   Answer: [Yes]

   Justification: We include the script for the experiment in the supplemental material.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: We include the detail information in Appendix K.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: We report error bars for our method as well as the baseline from prior work to ensure statistically meaningful comparison.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: We report the computation resource in the Appendix K.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: We have reviewed the Code of Ethics and have made every effort to adhere to it.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [N/A]

    Justification: As this is a theoretical paper in nature, the societal impacts of this paper are somewhat limited in scope.

    Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [N/A]

    Justification: As this is a mainly theoretical work in nature, it does not present the high-risk profiles described in the guidelines.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: We use the OpenAI Gym in our experiments, which is cited in the experiment section, and we have complied with all terms of use.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [N/A]

    Justification: We do not have any new assets in this work.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: This work does not involve crowdsourcing or research with human subjects.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: This work does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

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

    Answer: [N/A]

    Justification: We only use LLM for editing assistance.

    Guidelines:

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