Provable Benefits of Representational Transfer in Reinforcement Learning

Alekh Agarwal, Yuda Song, Wen Sun, Kaiwen Wang, Mengdi Wang, Xuezhou Zhang *

Abstract

We study the problem of representational transfer in RL, where an agent first pretrains in a number of source tasks to discover a shared representation, which is subsequently used to learn a good policy in a target task. We propose a new notion of task relatedness between source and target tasks, and develop a novel approach for representational transfer under this assumption. Concretely, we show that given a generative access to source tasks, we can discover a representation, using which subsequent linear RL techniques quickly converge to a near-optimal policy, with only online access to the target task. The sample complexity is close to knowing the ground truth features in the target task, and comparable to prior representation learning results in the source tasks. We complement our positive results with lower bounds without generative access, and validate our findings with empirical evaluation on rich observation MDPs that require deep exploration.

1 Introduction

Leveraging historical experiences acquired in learning past skills to accelerate the learning of a new skill is a hallmark of intelligent behavior. In this paper, we study this question in the context of reinforcement learning (RL). Specifically, we consider a setting where the learner is exposed to multiple tasks and ask the following question:

Can we accelerate RL by sharing representations across multiple related tasks?

There is rich empirical literature which studies multiple approaches to this question and various paradigms for instantiating it. For instance, in a multi-task learning scenario, the learner has simultaneous access to different tasks and tries to improve the sample complexity by sharing data across them (Caruana, 1997). Other works study a transfer learning setting, where the learner has access to multiple source tasks during a *pre-training* phase, followed by a target task (Pan and Yang, 2009). The goal is to learn features and/or a policy which can be quickly adapted to succeed in the target task. More generally, the paradigms of meta-learning (Finn et al., 2017), lifelong learning (Parisi et al., 2019) and curriculum learning (Bengio et al., 2009) also consider related questions.

On the theoretical side, questions of representation learning have received an increased recent emphasis owing to their practical significance, both in supervised learning and RL settings. In RL, a limited form of transfer learning across multiple downstream reward functions is enabled by several recent reward-free representation learning approaches (Jin et al., 2020a; Zhang et al., 2020; Wang et al., 2020; Du et al., 2019; Misra et al., 2020; Agarwal et al., 2020; Modi et al., 2021). Inspired by recent treatments of representation transfer in supervised (Maurer et al., 2016; Du et al., 2020) and imitation learning (Arora et al., 2020), some works also study more general task collections in bandits (Hu et al., 2021; Yang et al., 2020, 2022) and RL (Hu et al., 2021; Lu et al., 2021). Almost all these works study settings where the representation is *frozen* after pre-training in the source tasks, and a linear policy or optimal value function approximation is trained in the target task using these learned features. This setting, which we call *representational transfer*, is the main focus of our paper.

A crucial question in formalizing representational transfer settings is the notion of similarity between source and target tasks. Prior works in supervised learning make the stringent assumption that the covariates x follow the same underlying distribution in all the tasks, and only the conditional $P(y|x)$ can vary across tasks (Du et al., 2020). This assumption does not nicely generalize to RL settings, where state distributions are typically policy dependent, and prior attempts to extend this assumption to RL (Lu et al., 2021) result in strong assumptions on the learning setup. Other works (Hu et al.,

*Alphabetical Order. alekhagarwal@google.com, yudas@andrew.cmu.edu, {ws455,kw437}@cornell.edu, {mengdiw,xz7392}@princeton.edu.

Offline Reinforcement Learning Workshop at Neural Information Processing Systems (NeurIPS 2022).

2021; Yang et al., 2020, 2022) focus on linear representations only, which limits the expressivity of the feature maps, and does not adequately represent the empirical literature in the field.

Our contributions. In this context, our work makes the following contributions:

- We propose a new linear span assumption of task relatedness for representational transfer, where the target task dynamics can be expressed as a (state-dependent) linear span of source task dynamics, in addition to the dynamics being low-rank under a shared representation. We give examples captured by this assumption, and it generalizes all prior settings for representational transfer in RL. We do not make any linearity assumptions on our feature maps.
- When we have *generative access* to source tasks, we provide a novel algorithm REPTRANSFER that successfully pretrains a representation for downstream *online learning in any target task* (i.e., no generative access in target task) satisfying the linear span assumption, when the source tasks satisfy a common latent reachability assumption. The regret bound of learning in the target task is close to that of learning in a linear MDP equipped with the ground truth features, the strongest possible yardstick in our setup. The additional terms in our regret largely arise out of the distributional mismatch between source and target tasks, which is expected. We complement the theory with an empirical validation of REPTRANSFER on the challenging rich observation combination lock benchmarks (Misra et al., 2020), confirming our theoretical findings.
- Without generative access to source tasks, we show the statistical hardness of representational transfer under the linear span assumption, and confirm this hardness in our empirical evaluation. We show that an additional assumption that every *observed state* is reachable in every source task is sufficient for allowing fully online learning in source tasks.

The new task relatedness assumption, reward-free learning result for low-rank MDPs and our analysis of LSVI-UCB under average case misspecification may be of independent interest.

2 Related Work

Multi-task and Transfer Learning in Supervised Learning. The theoretical benefit of representation learning are well studied under conditions such as the i.i.d. task assumption (Maurer et al., 2016) and the diversity assumption (Du et al., 2020; Tripuraneni et al., 2020). Many works below successfully adopt the frameworks and assumptions to sequential decision making problems.

Multi-task and Transfer Learning in Bandit and small-size MDPs. A sequence of recent works study multi-task linear bandit problems with linear representation functions ($\phi(s) = A\,s$) (Hu et al., 2021; Yang et al., 2020, 2022). The techniques developed in these works crucially rely on the linear representation function structure and can not be applied to nonlinear function classes. Lazaric et al. (2013) study spectral techniques for online sequential transfer learning. Brunskill and Li (2013) consider multi-task RL under a fixed distribution over finitely many diverse MDPs, while Brunskill and Li (2014) considers transfer learning in semi-MDPs by learning options (temporally extended actions). All these works consider small size tabular models while we focus on large-scale MDPs.

Multi-task and Transfer Learning in RL via representation learning. Beyond tabular MDPs, Arora et al. (2020) and D'Eramo et al. (2019) show benefits of representation learning in imitation learning and planning, but do not address exploration. Lu et al. (2021) study transfer learning in low-rank MDPs with general nonlinear representations, but make a generative model assumption on both the source tasks and the target task, along with other distributional and structural assumptions. We do not require generative access to the target task and make much weaker structural assumptions on the source-target relatedness. Recently and independently, Cheng et al. (2022) also studied transfer learning in low-rank MDPs in the online learning setting, identical to the setting we study in Section 5. However, their analysis relies on an additional assumption that bounds the point-wise TV error with the population TV error, which we show is in fact not necessary. We refer readers to Appendix B for a more detailed comparison with the above two closely related works.

Efficient Representation Learning in RL. Even in the single task setting, efficient representation learning is an active area of study that has seen many recent advances with exploration (Agarwal et al., 2020; Modi et al., 2021; Uehara et al., 2021; Zhang et al., 2022) or without (Ren et al., 2021). Other papers study feature selection (e.g. Farahmand and Szepesvári, 2011; Jiang et al., 2015; Pacchiano et al., 2020; Cutkosky et al., 2021; Lee et al., 2021; Zhang et al., 2021), or focus on learning sparse models (Hao et al., 2021a,b).

3 Preliminaries

Notations: We denote total variation distance of P_1 and P_2 by $\|P_1 - P_2\|_{TV}$. Given a vector a, we define $\|a\|_B = \sqrt{a^\top B a}$. $c_0, c_1, \ldots$ are universal constants. We use $a \lesssim b$ to mean $a \le Cb$ for some universal constant C. Also, $[K] = \{1, \ldots, K\}$ and $\lambda_{\min}(A)$ is the smallest eigenvalue of matrix A. Please see Table 1 for a full list of notations and Table 2 for a list of algorithms.

Low-rank Markov Decision Processes: We consider a finite-horizon episodic Markov Decision Process $M = \{H, S, A, \{P_h^\star\}_{0:H-1}, \{r_h\}_{0:H-1}, d_0\}$, specified by the episode length H, state space S, discrete action space A of size A, transition models $P_h^\star : S \times A \to \Delta(S)$, reward functions $r_h : S \times A \to [0,1]$, and an initial distribution $d_0 \in \Delta(S)$. Occasionally, we use $P^\star$ to denote $\{P_h^\star\}_{0:H-1}$ and r to denote $\{r_h\}_{0:H-1}$. Starting from an initial state $s_0 \sim d_0$, an agent chooses action a_h at step h upon observing the state s_h, receives a reward $r_h(s_h, a_h)$, and transitions to $s_{h+1} \sim P_h^\star(\cdot|s_h, a_h)$. We assume that r and d_0 are known. Given a policy $\pi : S \to \Delta(A)$ and transition kernel P, we use the notation $\mathbb{E}_{\pi,P}[\cdot]$ to denote expectation under the distribution of trajectories when π is executed in P. The value function $V_{P,r,h}^\pi(s) = \mathbb{E}_{\pi,P}\left[\sum_{h'=h}^{H-1} r(s_{h'}, a_{h'}) \,|\, s_h = s\right]$ gives the expected reward-to-go of π under the MDP with transitions P, rewards r, starting at state s in step h. Similarly, we define the Q function $Q_{P,r,h}^\pi(s,a) := r_h(s,a) + \mathbb{E}_{s' \sim P_h(\cdot|s,a)} V_{P,r,h+1}^\pi(s')$. The expected total reward of a policy π under transitions P and rewards r is denoted as $V_{P,r}^\pi := \mathbb{E}_{s_0 \sim d_0} V_{P,r,0}^\pi(s_0)$. We define the state-action occupancy distribution $d_{P,h}^\pi(s,a)$ as the probability of π visiting (s,a) at time step h under π and P. $d_{P,h}^\pi(s) = \sum_{a \in A} d_{P,h}^\pi(s,a)$ is the marginal state visitation, which is equal to $\int_A d_{P,h}^\pi(s,a)$.

We study low-rank MDPs defined as follows (Jiang et al., 2017; Agarwal et al., 2020). The conditions on the upper bounds of the norm of $\phi^\star, \mu^\star$ are just for normalization.

Definition 3.1 (Low-rank MDP) A transition model $P_h^\star : S \times A \to \Delta(S)$ admits a low rank decomposition with rank $d \in \mathbb{N}$ if there exist two unknown embedding functions $\phi_h^\star : S \times A \mapsto \mathbb{R}^d$, $\mu_h^\star : S \mapsto \mathbb{R}^d$ such that $\forall s, s' \in S, a \in A$: $P_h^\star(s'|s,a) = \phi_h^\star(s')^\top \psi_h^\star(s,a)$, where $\|\psi_h(s,a)\|_2 \le \sqrt{d}$ for all (s,a) and for any function $g : S \to [0,1]$, $\|\int g(s) d\mu_h^\star(s)\|_2 \le \sqrt{d}$. An MDP is a low rank MDP if $P_h^\star$ admits such a low rank decomposition for all $h = 0, 1, \ldots, H-1$.

Low-rank MDPs capture the latent variable model (Agarwal et al., 2020) where $\phi^\star(s,a)$ is a distribution over a discrete latent state space, and the block-MDP model (Du et al., 2019) where $\phi^\star(s,a)$ is a one-hot encoding vector. Note that the linear MDP model (Yang and Wang, 2020; Jin et al., 2020b) assumes known $\phi^\star$, which significantly simplifies the algorithm design.

Transfer Learning: In contrast to the classic single-task learning setting, in this paper, we explore the setting of transfer learning, where learning consists of two phases: (1) the pre-training phase where the agent interacts with $K-1$ source tasks with dynamics $P_k^\star$, and (2) the deployment phase where the agent is deployed into the K-th target task and no longer has access to the source tasks. The performance is measured mainly by the regret incurred in the target task upon deployment, while we also desire small sample complexity in the source tasks. We denote $d_{k,h}^\pi = d_{P_k^\star,h}^\pi$.

In order for the pre-training phase to help with learning in the target task, we must make assumptions on the connections between tasks. In this work, we make the following fundamental structural assumption on all the tasks at hand, namely that they share the same underlying representation.

Assumption 3.1 (Common representation) We assume that all tasks $P_k^\star$ are low-rank MDPs with a shared representation $\phi_h^\star(s,a)$ but distinct $\mu_{k,h}^\star(s')$, that is $P_{k,h}^\star(s'|s,a) = \phi_h^\star(s,a)^\top \mu_{k,h}^\star(s')$.

Assumption 3.2 (Realizability) For any source task $k \in [K-1]$ and any $h \in [H]$, we assume that the agent has access to realizable function classes Φ_h and $\Psi_{k,h}$, such that $\phi_h^\star \in \Phi_h$ and $\mu_{k,h}^\star \in \Psi_{k,h}$. For normalization, we assume that for all h, all $\phi \in \Phi_h$ satisfy $\|\phi(s,a)\|_2 \le \sqrt{d}$, and for all $\psi \in \Psi_{k,h}$ and any function $g : S \to [0,1]$, $\|\int g(s) d\psi(s)\|_2 \le \sqrt{d}$.

Assumption 3.2 is standard realizability condition in function class which is made in almost all prior works on RL with linear and nonlinear function approximation (e.g., Jiang et al. (2017); Sun et al. (2019); Jin et al. (2020b); Agarwal et al. (2020); Du et al. (2021); Uehara et al. (2021)).

Since all the tasks share a common representation, we expect the agent to learn a good representation during its initial interactions with the source tasks, and then learn a good policy for the target task using this representation in the deployment phase. The availability of multiple source tasks is

bene cial as the agent might not be able to learn a good representation for all parts of the observation space of the target task from one source only. In order to realize this intuition though, we need two additional assumptions which we describe next.

Assumptions for representational transfer. In addition to standard assumptions in the source tasks, we make the following structural and relatedness assumptions on the source and target tasks.

Assumption 3.3 (Feature reachability in the source tasks). We assume that $\nu := \min_{k \in [K-1]; h \in [H]} \max_{\pi} \min_{s_h : P^\star_k(s_h) > 0} \mathbb{E}_{\pi, P^\star_k}[\phi^\star_h(s_h, a_h)\, \phi^\star_h(s_h, a_h)^\top]$ is strictly positive.

Assumption 3.3 intuitively requires that no subspace of $\mathbb{R}^d$ is unreachable in the source tasks, as otherwise it's impossible to guarantee the quality of the learned representation in that subspace which may contain high rewards in the target task. Note that no reachability is required in the target task.

The next assumption quanti es the relatedness of the target task and the source tasks.

Assumption 3.4 (Relatedness: Point-wise Linear span). For any $h \in [H]$ and $s' \in S_{h+1}$, there is a vector $\omega_h(s') \in \mathbb{R}^{K-1}$ such that $\phi^\star_{K,h}(s') = \sum_{k=1}^{K-1} \omega_{k,h}(s')\, \phi^\star_{k,h}(s')$, with $\omega_{\max} = \max_{h,k,s' \in S} |\omega_{k,h}(s')|$ and $\Omega = \max_h \sum_{k=1}^{K-1} \max_{s' \in S} |\omega_{k,h}(s')|$.

Assumption 3.4 ensures that if s' is reachable from an (s, a) pair in the target task, then it must be reachable from the same (s, a) pair in at least one of the source tasks. This intuitively is also necessary for transfer learning, as s' could be a high rewarding state in the target task. A special case is the convex combination, i.e., for any $s' \in S, h$, $\phi^\star_{K,h}(s') = \sum_k p_k \phi^\star_{k,h}(s')$ with $p_k \geq 0$, $\sum_k p_k = 1$, which implies $\omega_{\max} \leq 1$ and $\Omega = 1$. On the other hand, if the target task largely focuses on observations quite rare under the source tasks, then Ω can grow large, and this is unavoidable in a transfer learning setting.

We remark that unlike prior work (Du et al., 2020; Tripuraneni et al., 2020; Lu et al., 2021), we do not make any assumption on the data generating distribution in either the source or the target tasks. Instead, our approach is end-to-end, i.e., we collect our own data from scratch for representation learning by doing strategic exploration in the source tasks (see Section B for a detailed discussion). In fact, in Theorem C.1 we show that the above assumptions do not permit successful transfer in the supervised learning setting, thus establishing an interesting separation between supervised and reinforcement learning.

We conclude this section with a couple of examples where our assumptions are satis ed.

Example 3.1 (Mixture of source tasks). Perhaps the simplest example of our assumptions is where $P^\star_K(s'|s, a) = \sum_{k=1}^{K-1} \omega_k P^\star_k(s'|s, a)$ with the coef cients independent of s', $\omega_k \geq 0$ and $\sum_{k=1}^{K-1} \omega_k = 1$. Such mixtures of base models have been considered in several prior works (Modi et al., 2020; Ayoub et al., 2020). While prior works study the case of arbitrary, known base models, we instead allow structured, unknown base models with a shared representation $\phi^\star$.

Example 3.2 (Block MDPs with shared latent dynamics). In this example, each MDP $P^\star_k$ is a Block MDP (Du et al., 2019) with a shared latent space $\mathcal{Z}$ and a shared decoder $\phi^\star : S \to \mathcal{Z}$. In a block MDP, given state action pair (s, a), the decoder $\phi^\star$ maps s to a latent state z, the next latent state is sampled from the latent transition $z' \sim P(\cdot|z, a)$, and the next state is generated from an emission distribution $s' \sim o(\cdot|z')$. Recall that $o(s'|z') > 0$ at only one $z' \in \mathcal{Z}$ for any $s' \in S$ for a block MDP. We assume that the latent transition model $P(z'|z, a)$ is shared across all the tasks, but the emission process differs across the MDPs. For instance, in a typical navigation example used to motivate Block MDPs, the latent dynamics might correspond to navigating in a shared 2-D map, while emission distributions capture different wall colors or lighting conditions across multiple rooms. Then Assumption 3.3 requires that the agent can visit the entire 2-D map, while Assumption 3.4 requires that the color/lighting conditions of the target task resemble that of at least one source task. The coef cients ω for any s' are non-zero on the source tasks which can generate that observation.

4 Transfer Learning with Generative Access to Source Tasks

We rst study a setting where we assume generative model access to the source tasks, while having only online access to the target task

Assumption 4.1 (Generative access to the source tasks). We assume that we have access to generative models for the $K-1$ source tasks. Speci cally, for any $P^\star_k$ with $k \in [K-1]$, we can query any (s_h, a_h) pair, and the generative model will return a next state sample $s_{h+1} \sim P^\star_{k,h}(s_h, a_h)$.

Algorithm 1 Transfer learning with generative access (REPTRANSFER)

PRE-TRAINING PHASE

Input: exploratory policies $\{\pi_k\}_{k=1}^{K-1}$, size of cross-sampled datasets n, failure probability δ.

1: **for** task pairs (i,j), i.e. for all $i,j \in [K-1]$ s.t. $i \neq j$ **do** . cross sampling procedure
2: For each $h \in [H-1]$, sample dataset $\mathcal{D}_{ij,h}$ containing n i.i.d. (s,a,s') tuples sampled as:

$$(s,a) \sim d^{\pi_i}_{i,h-1}, \; s \sim P^{\star}_{j,h-1}(s,a), \; a \sim U(\mathcal{A}), \; s' \sim P^{\star}_{i,h}(s,a). \quad (1)$$

3: For all $h \in [H-1]$, learn $\hat{\phi}_h$ = Multi-task REPLEARN$\left(\{\bigcup_{j \in [K-1]} \mathcal{D}_{kj,h}\}_{k \in [K-1]}\right)$. (Algorithm 3)

DEPLOYMENT PHASE

Additional Input: number of deployment episodes T

1: Set $\beta = H\sqrt{d} + dH\sqrt{\log(dHT/\delta)}$.
2: Run LSVI-UCB$\left(\{\hat{\phi}_h\}_{h=0}^{H-1}, r = r_K, T, \delta\right)$ in the target task $P^{\star}_K$ (Algorithm 6).

Having generative model access is not unrealistic, especially in applications where a high-quality simulation environment is available. Generative access also does not trivialize the challenge of efficient exploration in the source tasks, since there are a potentially infinite number of states and the ground truth representation $\phi^{\star}$ is unknown. Prior works using generative access typically require either a known representation $\phi^{\star}$ (so that one can perform D-optimal design to construct an exploratory state-action distribution (Agarwal et al., 2019)), or directly assume access to a diverse state-action distribution which provides coverage and from which one can sample. Neither such a diverse sampling distribution is given in our case. We also note that in Section 5 we will show that without any additional assumptions, generative access in source tasks is necessary.

Algorithm overview. Given the above setup, now we present our algorithm REPTRANSFER, detailed in Algorithm 1. REPTRANSFER takes a representation learning approach to the transfer learning problem and operates in two phases. During the pre-training phase, REPTRANSFER performs reward-free exploration in each of the source tasks to learn task-specific policies π_k which satisfies for all $h = 0, 1, \ldots, H-1$, that $\mathbb{E}_{s_h \sim P^{\star}_k, a_h \sim \pi_k(s_h)}\left[\phi^{\star}_h(s_h,a_h)\phi^{\star}_h(s_h,a_h)^{\top}\right] \succeq \sigma_{\min} I$ for some $\sigma_{\min} > 0$. We call such policies $\sigma_{\min}$-exploratory and will use this definition in Lemma 4.1 and Theorem 4.1. In Section 4.1 we present one particular algorithm that finds such π_k's with a specific $\sigma_{\min}$, but our main analysis below is modular to the choice of the reward-free exploration algorithm.

Given the exploratory policies $\{\pi_k\}_{k \in [K-1]}$, REPTRANSFER collects a joint dataset across the source tasks using these policies and cross-sampling across pairs of tasks, and then learns a single representation $\hat{\phi}$. In the deployment phase, REPTRANSFER runs optimistic least squares value iteration (Algorithm 6) using the learned representation $\hat{\phi}$ in the target task.[3]

The cross sampling procedure and learning a shared representation. We now describe the cross sampling procedure in detail. Given an exploratory policy π_k for each source task k, the next step is to sample fresh data under cross-sampling procedure using the generative model. Consider a particular (k,j) pair of source environments. For each $h \in [H]$, we first sample $(s_{h-1}, a_{h-1}) \sim d^{\pi_k}_{k,h-1}$. Then, in the simulator of task k, we reset to (s_{h-1}, a_{h-1}) and perform a transition step to s_h, i.e., $s_h \sim P_{j,h-1}(s_{h-1}, a_{h-1})$. Then we reset the simulator for task k to state s_h, sample a uniformly random action a_h and then perform another transition to s_{h+1}, i.e., $s_{h+1} \sim P^{\star}_k(s_h, a_h)$. Such a procedure is only possible under the generative model setting. Given datasets, each of size n, collected using the cross-sampling procedure, we perform an MLE-based representation learning procedure (Algorithm 3) jointly on all source tasks. An adaptation of the result of Agarwal et al. (2020, Theorem 21) to the multi-task setting shows that Multi-task REPLEARN (Algorithm 3) achieves the following total variation guarantee with probability at least $1-\delta$:

$$\sum_{k=1}^{K-1} \mathbb{E}_{s,a \sim \mathcal{D}_{k,h}} \left\| \hat{\phi}_h(s,a)^{\top} \hat{\phi}_{k,h}(\cdot) - \phi^{\star}_h(s,a)^{\top} \phi^{\star}_{k,h}(\cdot) \right\|_{TV}^2 \leq \varepsilon_N := O\left((\log(|\Phi|/\delta) + K\log(|\Psi|))/N\right). \quad (2)$$

[2] Given any dataset $\{s, a, r, s'\}$ feature ϕ, and reward r, LSVI learns a Q function backward, i.e., at step h via $w_h = \arg\min_w \sum_{s,a,s'} (w^{\top}\phi(s,a) - \hat{V}_{h+1}(s'))^2 + \|w\|^2$ and sets $\hat{V}_h(s) = \max_a (r(s,a) + w_h^{\top}\phi(s,a)), \forall s$. UCB, short for Upper Confidence Bound, refers to an exploration bonus added to basic LSVI.

Algorithm 2 REWARDFREE

1: Input: MDP $P^?$ with online access, num. LSVI-UCB episodes $N_{\text{LSVI-UCB}}$, num. model-learning episodes $N_{\text{REWARDFREE}}$, failure probability δ.
2: Learn model $\widehat{P}_h = (\widehat{b}_h, \widehat{b}_h)_{h=0}^{H-1}$ by running REWARDFREE REP-UCB (Algorithm 4) in $P^?$ for $N_{\text{REWARDFREE}}$ episodes.
3: Set $\beta = dH\sqrt{\log(dHN_{\text{LSVI-UCB}}/\delta)}$.
4: Return $\pi = \text{LSVI-UCB}(\widehat{b}_h, \widehat{b}_h)_{h=0}^{H-1}; r = 0; N_{\text{LSVI-UCB}}; \beta;$ UNIFORMACTIONS $=$ TRUE$)$ by simulating in the learned model $\widehat{P}$ (Algorithm 6). Note this step requires no samples from $P^?$.

where $_{k;h}$ denotes the generating distribution of inputted datasets $D_{k;h}$, each with size N. For example, in REPTRANSFER we have $D_{k;h} = [_{j=1}^{K-1} D_{kj;h}$ and $N = n(K-1)$.

Representational transfer. Next we transfer this guarantee to the target task via Assumption 3.4, a key insight of our work. Specifically, we show that under cross sampling and Assumption 3.4, $\widehat{b}_h$ also linearly approximates the true transition in the target task well, under the occupancy measure of any policy. Remarkably, this holds before the agent has ever interacted with the target task.

Lemma 4.1 Suppose Assumption 3.4 and that for all source tasks $k \in [K-1]$ we have a $_{\min}$-exploratory policy $_k$. Then, for any $\in (0,1)$, learning features $\widehat{b}$ using the cross-sampling procedure of Algorithm 1 satisfies (2) w.p. $1-$. Furthermore, for any $h = 0, 1, \ldots, H-1$, there exists $_h : S \to \mathbb{R}^d$ such that for any function $g : S \to [0,1]$, $\|\int g(s)d_h(s)\|_2 \le \sqrt{d}$ and

$$\sup_{,P_k^?} \left\| \widehat{b}_h(s_h,a_h)^\top \int e_h(\,) - {}_h^?(s_h,a_h)^\top {}_{K;h}^?(\,) \right\|_{TV} \le "_{TV} := \sqrt{|A|\,{}_{\max}^3 K n/{}_{\min}}.$$

Lemma 4.1 implies that $\widehat{b}$ is a feature such that $R_K^?$ is an approximatly linear MDP in $\widehat{b}$. Learning in approximately linear MDPs has been studied in Jin et al. (2020b), but under a much stronger ℓ_1 error bound rather than the average misspecification here. Our result for downstream task learning under such an average model misspecification case presents the strongest result for learning in an approximately linear MDP, and the result might be of independent interest.

Regret bound for REPTRANSFER. Putting things together, we obtain the following regret bound for online learning in the target task using our learned representation $\widehat{b}$.

Theorem 4.1 (Regret under generative source access) Suppose Assumptions 3.1-3.4 and 4.1, and suppose the input policies $_k$ are $_{\min}$-exploratory. Then, for any $\in (0,1)$, w.p. $1-$, REPTRANSFER when deployed in the target task has regret at most $\widetilde{O}\big(H^2 d^{1.5}\sqrt{T\log(1/)}\big)$, with at most Kn generative accesses per source task, with $n = O\big(\frac{1}{{}_{\min}} A\,{}_{\max}^3 KT\,\log\frac{|\,|}{} + K\log|\,|\big)$.

Remarkably, Theorem 4.1 shows that with the pre-trained features, we achieve the same regret bound on the target task to the setting of linear MDP with known $?$ (Jin et al., 2020b), up to the additional factor. The scaling factor only depends on itself and captures the hardness of transfer learning. For special cases such as convex combination, i.e., state-independent and $_h \in (K)$, then $= 1$. In the worst-case, some dependence on the scale of seems unavoidable as we can have a state s^0 such that $_K(s^0) = 1$ and $_1(s^0) \approx 1$ with $_1(s^0) \approx 1$. This corresponds to a rarely observed state for the source task encountered often in the target, and our estimates of transitions involving this state can be highly unreliable if it is not seen in any other source, roughly scaling the error between target and source tasks as (s^0). Obtaining formal lower bounds that capture a matching dependence on structural properties is an interesting question for future research.

4.1 Reward-free exploration in the source environments.

So far we have assumed that we have a reward-free exploration black box algorithm that can provide an exploratory policy $_k$ for each source task k. In this section, we give a detailed algorithm that achieves this goal.

Recall that our transfer learning algorithm, for source task k, relies on an exploratory policy $_k$ such that the empirical covariance with respect to $?_h$ is lower bounded. This ensures good exploration in the underlying ground truth feature space in $P_K^?$. REWARDFREE Algorithm 2 achieves this goal

by first invoking the RewardFree Rep-UCB algorithm (Algorithm 4) to learn an estimated linear MDP model $\widehat{P}_k$ for $P_k^?$. Subsequently, we perform reward-free exploration (e.g., using LSVI-UCB with zero reward) within each $\widehat{P}_k$ which involves no further environment interactions.

Lemma 4.2 (Reward-free Exploration) Fix any source task $k \in [K-1]$. Suppose Assumptions 3.2 and 3.3. Then, for any $\in (0,1)$, w.p. $1-$, RewardFree (Algorithm 2) with $N_{\text{LSVI-UCB}} = \widetilde{\Theta}\big(A^3 d^6 H^8 / ^2\big)$ and $N_{\text{RewardFree}} = \widetilde{\Theta}\big(A^3 d^4 H^6 \log(1/) + N^2_{\text{LSVI-UCB}}\big)$ returns a $_{\min}$-exploratory policy $_k$ where $_{\min} = \widetilde{\Theta}\big(A^{-3} d^{-5} H^{-7} ^2\big)$: The sample complexity here is $N_{\text{RewardFree}}$ episodes in the source task.

To the best of our knowledge, Lemma 4.2 is the first result that finds a full-rank policy cover in the low-rank MDP setting, and might be of independent interest. Wagenmaker et al. (2022) recently obtained a related guarantee in the linear MDP setting with known features. Incorporating Lemma 4.2 and Theorem 4.1, we get our final sample complexity bound.

Theorem 4.2 (Regret for RepTransfer with RewardFree subroutine) Suppose Assumptions 3.1-3.4 and 4.1 hold, and let $\in (0,1)$. Let $\{_k\}_{k=1}^{K-1}$ be exploratory policies learned from running RewardFree on each source task with $N_{\text{LSVI-UCB}}$ and $N_{\text{RewardFree}}$ set as in Lemma 4.2. Then, w.p. $1-$, running RepTransfer with $\{_k\}_{k=1}^{K-1}$ has regret in the target task of $\widetilde{O}\big(H^2 d^{1.5}\sqrt{T\log(1/)}\big)$, with at most $\widetilde{O}\big(A^4 {}^3_{\max} d^5 H^7 K^2 T ^2 (\log(|/) + K\log|)\big)$ generative accesses per source task.

5 Transfer Learning with Online Access to Source Tasks

In the previous section, we show that efficient transfer learning is possible under very weak structural assumptions, assuming the help of generative access to the source tasks. One natural question is whether transfer learning is possible with only online access to the source tasks. In that case, the cross-sampling procedure no longer works. Somewhat surprisingly, we show that this is impossible without significantly stronger assumption.

Theorem 5.1 (Impossibility Result) Let M_K be the set of K-task multi-set that satisfies (1) all tasks are Block MDPs; (2) all tasks satisfy Assumption 3.3 and Assumption 3.4; (3) the latent dynamics are exactly the same for all source and target tasks. For any pre-training algorithm which outputs a feature $\hat{}$ by interacting with the source tasks $\{P_k^?\}_{k\in[K-1]} \in M_K$, such that with probability at least $1-1/2$, A will output a feature $\hat{}$, such that for any policy taking the functional form of $(s) = f(\hat{}(s,a))_{a\in A}; f \in (s,a))_{a\in A}$, we have $V_K^? - V_K \geq 1/2$.

The above theorem implies that a representation learned only from online access to source tasks does not enable learning in downstream tasks if the downstream task algorithm is restricted to use the representation as the only information of the state-action pairs (e.g., running LSVI-UCB with $\hat{}$).

We briefly explain the intuition behind the above lower bound. For a Block MDP, for (s,a), we can model the ground-truth $^?$ as a one-hot encoding (z,a) corresponding to the latent state-action pair (z,a) with $z = {}^?(s)$ being the encoded latent state. The key observation here is that any permutation of $^?$ will also be a perfect feature in terms of characterizing the Block MDPs, since it corresponds to simply permuting the indices of the latent states. Therefore, without cross referencing, the agent could potentially learn different permutations in different source tasks, which would collapse in the target task. A precise constructive proof of Theorem 5.1 can be found in Section C.

Part of the reason that the above example fails is that each source task has its own observed subset of raw states, which permits such permutation to happen. In what follows, we show that, indeed, under an additional assumption on the reachability of raw states, a slight variant of the same algorithm (Algorithm 5 in Section F) can achieve the same regret with only online access to the source tasks.

Assumption 5.1 (Reachability in the raw states) For all source tasks $k \in [K-1]$, any policy and $h = 0, 1, \ldots, H-1$, we have $\inf_{s\in S, a\in A} d_{k,h}(s,a) \geq {}_{\text{raw}} \min \mathbb{E}_{, P_k^?}\big[{}^?_h(s_h,a_h)\,{}^?_h(s_h,a_h)^\top\big]$.

Assumption 5.1 implies that for each source task, any policy that achieves a full-rank covariance matrix also achieves global coverage across the raw state-action space. In addition, in order to apply importance sampling (IS) to transfer the TV error from source task to target task, we need to assume

	Source	O-REPTRANSFER	G-REPTRANSFER	Oracle	Target
Comblock	1	8006 (294.7)	7790 (267.6)	7048 (164.8)	181450 (147600.2)
Comblock PO	1	1	7764 (145.7)	7336 (270.7)	85000 (14469.8)

(a) (b)

Figure 1:(a): A visualization of the rich observation comblock environment. Latent states (white and black) emit continuous high-dimensional observations. The reward is sparse (only in white states at H). Each white state has 10 actions where one of them is a good action that leads to the next two white states while the other 9 lead to black states (good action for each white state is different). Once the agent is in black states, it stays stuck in black until the end of the episode. Thus, a random exploration strategy has an exponentially small probability of hitting the goal (i.e., roughly $(1/10)^H$). (b) Top: Number of episodes required to solve the target environment under the setting from Sec. 6.1. Bottom: Number of episodes required to solve the target environment under the setting from Sec. 6.2. An algorithm solves the target task if it can achieve the optimal return (i.e., 1) for 5 consecutive iterations with 50 evaluation runs each. We include the mean and standard deviation (in the brackets) for 5 random seeds. ∞ denotes that an algorithm can not solve the target task within a fixed sample budget.

that the target task distribution has bounded density. This is true, for example, when $\mathcal{S}$ is discrete or when $\mathcal{S}$ is compact and has bounded measure.

Assumption 5.2(Bounded density)For all $(\cdot; h; s; a)$, we have $d_{K;h}(s; a) \leq 1$.

Theorem 5.2 (Regret with online access)Suppose Assumptions 3.1-3.4,5.1,5.2 hold. For any $\in (0; 1)$, w.p. $1 -$, Algorithm 5 with appropriately set parameters achieves a regret in the target task at most $\tilde{O}\left(d^{1:5} H^2 \sqrt{T \log(1/)}\right)$, with at most $\mathrm{poly}\left(A; {}_{\max}; d; H; K; T; {}^{-1}; {}_{\mathrm{raw}}^{-1}; \log(|\mathcal{Z}||\mathcal{S}|/)\right)$ online queries in the source tasks.

Assumption 5.1 is satisfied in a Block MDP, when, for example, the emission function $o(s|z)$ satisfies that $\forall s, \exists z, \text{ s.t. } o(s|z) \geq c$. That is, for any source task, any state in the state space can be generated by at least one latent state.

6 Experiments

For the empirical study, we investigate the benefit of transfer learning under the Block MDP setting, on the challenging Rich Observation Combination Lock (comblock) benchmark. We choose this environment because one must recover the correct feature from rich observations and perform strategic exploration or otherwise pay for exponential sample complexity. We include a visual overview in Fig. 1(a). This environment is uniquely challenging since it requires strategic exploration and latent state discovery at the same time, which results the failures of common deep RL methods (Misra et al., 2020), and theoretical RL approaches based on linear function approximation (and reproducing kernel Hilbert space) (Zhang et al., 2022).

In particular, in this section we study the following questions: i) what are the benefits of representational transfer using multiple source tasks, and ii) whether generative access to source tasks is needed. We design two sets of experiments with various source and target environment configurations. We defer the details of the experiments (the design of vanilla comblock, hyperparameters, etc.) in Appendix I and include the essential design information in the following two sections.

Baselines.We denote *Source* as the smallest sample complexity of LSVI-UCB using learned features from any of the source tasks; O-REPTRANSFER as REPTRANSFER with only online access to the source tasks; G-REPTRANSFER as REPTRANSFER with generative access to the source tasks; *Oracle* as learning in the target task with ground truth features; and *Target* as running BRIEE (Zhang et al., 2022) — the SOTA Block MDP algorithm, in the target task with no pretraining.[3]

[3]Importantly, BRIEE differs in how it learns features, but uses the same LSVI-UCB method at each step for policy learning with the learned features.

Figure 2: (a): Visualization of the decoder source (top) and REPTRANSFER (bottom). (b): Visualization of the decoder O-REPTRANSFER (top) and G-REPTRANSFER (bottom). For each baseline, The h-th column in the i-th image denotes the averaged decoded states from the 30 observations generated by latent state h, for $i \in \{0, 1, 2\}$ and $h \in [25]$, from the corresponding target environment. The optimal decoder should recover the latent states up to a permutation. In Fig a (top), note that the learned features in source task fail to solve the target because of the collapse at timestep 5: both observations from state 0 and 1 are mapped to state 0. Note that in the source task where this feature is trained, such collapse can happen when state 0 and 1 have identical latent transition (for detailed discussion we refer to Misra et al. (2020); Zhang et al. (2022)). In Fig b (top), REPTRANSFER with only online access learns an incorrect decoder when the source tasks' observation spaces are disjoint. This is because the learned feature can decode each source task with a different permutation.

6.1 Comblock without partitioned observations

In this section we start off with an easier setting: we use 5 source tasks, each with $H = 25$, 3 latent states per level and 10 actions. The latent transition dynamics are different for each source task, but notice that for comblock, the reachability assumption (c.r. Assumption 3.3) is always satisfied. The emission distribution of all the source and target tasks is identical, so Assumption 5.1 holds here. For the construction of the target task, for each timestep h, we choose the latent transition dynamics from one of the sources uniformly at random (thus Assumption 3.4 is satisfied). We record the number of episodes in the target environment that each method takes to solve the target environment in Table. 1(b). We first observe that REPTRANSFER with either online or generative access can solve the target task (since. Assumption 5.1 holds). Second, we observe that directly applying the learned feature from any single source task does not suffice to solve the target environment. This is because the representation learned from a single source task may collapse two latent states into a single one during encoding (e.g., if two latent states at the same time step have exactly identical latent transitions). See the visualization of comparisons between the learned decoders from a single source task and REPTRANSFER in Fig. 2(a). Third, the result shows that REPTRANSFER saves order of magnitude of target samples compared with training in the target environment from scratch using the SOTA Block MDPs algorithm BRIEE. This set of results verifies the empirical benefits of representation learning from multiple tasks, i.e., resolves ambiguity and speeds up downstream task learning

6.2 Comblock with partitioned observation space (Comblock-PO)

In this section, following the intuition of our lower bound (Theorem 5.1), we construct a setting where the supports of the emission distributions from each task are completely disjoint, while the emission distribution in the target task is a mixture of all source emissions and the latent dynamics are identical across tasks. Hence Assumption 3.4 holds while Assumption 5.1 fails. So we expect that an algorithm without generative access to source tasks will fail based on Theorem 5.1. Specifically, under disjoint emission supports, a representation can decode the latent state correctly for each source, but permute latent state labels across sources, causing decoding errors on the target. We record the number of target episodes for each method to solve the target task in Table. 1(b). We observe that indeed the online version fails while the generative version still succeeds. We show the visualizations of O-REPTRANSFER and G-REPTRANSFER in Fig. 2(b), and we note that the empirical results exactly match our theoretical results. This ablation verifies that source generative model access is needed, unless one has additional stronger distributional assumptions.

References

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

Alekh Agarwal, Sham Kakade, Akshay Krishnamurthy, and Wen Sun. Flambe: Structural complexity and representation learning of low rank mdps. Advances in Neural Information Processing Systems, volume 33, pages 20095–20107, 2020.

Sanjeev Arora, Simon Du, Sham Kakade, Yuping Luo, and Nikunj Saunshi. Provable representation learning for imitation learning via bi-level optimization. In International Conference on Machine Learning, pages 367–376. PMLR, 2020.

Alex Ayoub, Zeyu Jia, Csaba Szepesvari, Mengdi Wang, and Lin Yang. Model-based reinforcement learning with value-targeted regression. In International Conference on Machine Learning, pages 463–474. PMLR, 2020.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In Proceedings of the 26th annual international conference on machine learning, pages 41–48, 2009.

Emma Brunskill and Lihong Li. Sample complexity of multi-task reinforcement learning. arXiv preprint arXiv:1309.6821, 2013.

Emma Brunskill and Lihong Li. Pac-inspired option discovery in lifelong reinforcement learning. In International conference on machine learning, pages 316–324. PMLR, 2014.

Rich Caruana. Multitask learning. Machine learning, 28(1):41–75, 1997.

Yuan Cheng, Songtao Feng, Jing Yang, Hong Zhang, and Yingbin Liang. Provable benefit of multitask representation learning in reinforcement learning. arXiv preprint arXiv:2206.05900, 2022.

Ashok Cutkosky, Christoph Dann, Abhimanyu Das, Claudio Gentile, Aldo Pacchiano, and Manish Purohit. Dynamic balancing for model selection in bandits and rl. In International Conference on Machine Learning, pages 2276–2285. PMLR, 2021.

Carlo D'Eramo, Davide Tateo, Andrea Bonarini, Marcello Restelli, and Jan Peters. Sharing knowledge in multi-task deep reinforcement learning. In International Conference on Learning Representations, 2019.

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.

Simon S Du, Wei Hu, Sham M Kakade, Jason D Lee, and Qi Lei. Few-shot learning via learning the representation, provably. arXiv preprint arXiv:2002.09434, 2020.

Simon S Du, Sham M Kakade, Jason D Lee, Shachar Lovett, Gaurav Mahajan, Wen Sun, and Ruosong Wang. Bilinear classes: A structural framework for provable generalization in rl. In International Conference on Machine Learning, 2021.

Amir-massoud Farahmand and Csaba Szepesvári. Model selection in reinforcement learning. Machine learning, 85(3):299–332, 2011.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In International conference on machine learning, pages 1126–1135. PMLR, 2017.

Botao Hao, Yaqi Duan, Tor Lattimore, Csaba Szepesvári, and Mengdi Wang. Sparse feature selection makes batch reinforcement learning more sample efficient. In International Conference on Machine Learning pages 4063–4073. PMLR, 2021a.

Botao Hao, Tor Lattimore, Csaba Szepesvári, and Mengdi Wang. Online sparse reinforcement learning. In International Conference on Artificial Intelligence and Statistics, pages 316–324. PMLR, 2021b.

Jiachen Hu, Xiaoyu Chen, Chi Jin, Lihong Li, and Liwei Wang. Near-optimal representation learning for linear bandits and linear rl. In *International Conference on Machine Learning*, pages 4349–4358. PMLR, 2021.

Nan Jiang, Alex Kulesza, and Satinder Singh. Abstraction selection in model-based reinforcement learning. In *International Conference on Machine Learning*, pages 179–188. PMLR, 2015.

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, Akshay Krishnamurthy, Max Simchowitz, and Tiancheng Yu. Reward-free exploration for reinforcement learning. In *International Conference on Machine Learning*, pages 4870–4879. PMLR, 2020a.

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

Alessandro Lazaric, Emma Brunskill, et al. Sequential transfer in multi-armed bandit with nite set of models. *Advances in Neural Information Processing Systems*, 26, 2013.

Jonathan Lee, Aldo Pacchiano, Vidya Muthukumar, Weihao Kong, and Emma Brunskill. Online model selection for reinforcement learning with function approximation. In *International Conference on Arti cial Intelligence and Statistics*, pages 3340–3348. PMLR, 2021.

Rui Lu, Gao Huang, and Simon S Du. On the power of multitask representation learning in linear mdp. *arXiv preprint arXiv:2106.08053*, 2021.

Thodoris Lykouris, Max Simchowitz, Alex Slivkins, and Wen Sun. Corruption-robust exploration in episodic reinforcement learning. In *Conference on Learning Theory*, pages 3242–3245. PMLR, 2021.

Andreas Maurer, Massimiliano Pontil, and Bernardino Romera-Paredes. The bene t of multitask representation learning. *Journal of Machine Learning Research*, 17(81):1–32, 2016.

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

Aditya Modi, Nan Jiang, Ambuj Tewari, and Satinder Singh. Sample complexity of reinforcement learning using linearly combined model ensembles. In *International Conference on Arti cial Intelligence and Statistics*, pages 2010–2020. PMLR, 2020.

Aditya Modi, Jinglin Chen, Akshay Krishnamurthy, Nan Jiang, and Alekh Agarwal. Model-free representation learning and exploration in low-rank mdps. *arXiv preprint arXiv:2102.07035*, 2021.

Aldo Pacchiano, Christoph Dann, Claudio Gentile, and Peter Bartlett. Regret bound balancing and elimination for model selection in bandits and rl. *arXiv preprint arXiv:2012.13045*, 2020.

Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2009.

German I Parisi, Ronald Kemker, Jose L Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural Networks*, 113:54–71, 2019.

Tongzheng Ren, Tianjun Zhang, Csaba Szepesvári, and Bo Dai. A free lunch from the noise: Provable and practical exploration for representation learning. *arXiv preprint arXiv:2111.11485*, 2021.

Wen Sun, Nan Jiang, Akshay Krishnamurthy, Alekh Agarwal, and John Langford. Model-based rl in contextual decision processes: Pac bounds and exponential improvements over model-free approaches. In *Conference on learning theory*, pages 2898–2933. PMLR, 2019.

Nilesh Tripuraneni, Michael Jordan, and Chi Jin. On the theory of transfer learning: The importance of task diversity. *Advances in Neural Information Processing Systems*, 33:7852–7862, 2020.

Masatoshi Uehara, Xuezhou Zhang, and Wen Sun. Representation learning for online and of ine rl in low-rank mdps. In International Conference on Learning Representations, 2021.

Andrew Wagenmaker, Yifang Chen, Max Simchowitz, Simon S Du, and Kevin Jamieson. Reward-free rl is no harder than reward-aware rl in linear markov decision processes. arXiv preprint arXiv:2201.11206, 2022.

Ruosong Wang, Simon S Du, Lin Yang, and Russ R Salakhutdinov. On reward-free reinforcement learning with linear function approximation. Advances in neural information processing systems, 33:17816–17826, 2020.

Jiaqi Yang, Wei Hu, Jason D Lee, and Simon Shaolei Du. Impact of representation learning in linear bandits. In International Conference on Learning Representations, 2020.

Jiaqi Yang, Qi Lei, Jason D Lee, and Simon S Du. Nearly minimax algorithms for linear bandits with shared representation. arXiv preprint arXiv:2203.15664, 2022.

Lin Yang and Mengdi Wang. Reinforcement learning in feature space: Matrix bandit, kernels, and regret bound. In Proceedings of the 37th International Conference on Machine Learning, pages 10746–10756, 2020.

Andrea Zanette, Ching-An Cheng, and Alekh Agarwal. Cautiously optimistic policy optimization and exploration with linear function approximation. COLT, 2021.

Weitong Zhang, Jiafan He, Dongruo Zhou, Amy Zhang, and Quanquan Gu. Provably ef cient representation learning in low-rank markov decision processes. arXiv preprint arXiv:2106.11935, 2021.

Xuezhou Zhang, Yuzhe Ma, and Adish Singla. Task-agnostic exploration in reinforcement learning. Advances in Neural Information Processing Systems, 33:11734–11743, 2020.

Xuezhou Zhang, Yuda Song, Masatoshi Uehara, Mengdi Wang, Wen Sun, and Alekh Agarwal. Ef cient reinforcement learning in block mdps: A model-free representation learning approach. arXiv preprint arXiv:2202.00063, 2022.

Appendices

A Notations

Table 1: List of Notations

$S; A; A$	State and action spaces, and $A = \lvert A \rvert$.
(S)	The set of distributions supported by S.
$_{\min}(A)$	Smallest eigenvalue of matrix A.
e_j	One-hot encoding of, i.e. 0 at each index except the one corresponding to.
	Length of vector implied from context.
$(x)_y$	$\min\{x; y\}$.
H	Episode length of MDPs, a.k.a. time horizon. We index steps as $h = 0; 1; :::; H - 1$.
K	There are a total of K tasks. The rst $k = 1; :::; K - 1$ are source tasks, and the K-th task is the target task.
d	dimension of the low-rank MDP, i.e. dimension of.
$P^?_{k;h}$	Ground truth transition at time for task.
r_h	Reward function of the target task (i.e. task).
$E_{;P}[\,]$	Expectation under the distribution of trajectories when is executed in. We sometimes omit when the MDP is clear from context.
$d_{P;h}$	Occupancy distribution of under transitions at time.
$d_{k;h}$	Occupancy distribution for the -th task, i.e. $d_{P^?_{k;h}}$.
$^?_h(s; a)$	Embedding function for $(s; a)$ at time.
$_h$	Realizable function class for $^?_h$.
$\lvert\ \rvert$	De ned as $\max_h \lvert\ _h \rvert$.
$^?_{k;h}(s')$	Emission embedding function for at time for environment.
$_{k;h}$	Realizable function class for $^?_{k;h}$.
$\lvert\ \rvert$	De ned as $\max_{k;h} \lvert\ _{k;h} \rvert$.
	Feature reachability in the source task (Assumption 3.3).
$_{\max}$;	Constants de ned in point-wise linear span assumption (Assumption 3.4).
$_{\text{raw}}$	Raw states reachability parameter (Assumption 5.1)

Table 2: List of Algorithms

REPTRANSFER (Algorithm 1)	Transfer learning with generative access.
REWARDFREE (Algorithm 2)	Reward Free exploration in low-rank MDP.
Multi-task REPLEARN (Algorithm 3)	Multi-task Maximum Likelihood (MLE) Representation Learning.
REWARDFREE REP-UCB (Algorithm 4)	Modi ed REP-UCB for reward-free model learning.
REPTRANSFER for online access (Algorithm 5)	Simpli ed version of REPTRANSFER if we have reachability under raw states (see Theorem 5.2).
LSVI-UCB (Algorithm 6)	Optimistmic Least Squares Value Iteration (with bonus).

B Comparisons to closely related works

Recall that in addition to the commonly made shared representation assumption (Assumption 3.1), we made two additional structural assumptions: reachability (Assumption 3.3) and linear span on $\Psi_?$ (Assumption 3.4). The reachability assumption is commonly made in prior works even in single agent RL, e.g. (Modi et al., 2021; Misra et al., 2020). It ensures that there is no redundant dimensions in the ground-truth representation, which is a reasonable requirement. The linear span assumption is closely related to the diversity assumptions made in prior works of transfer learning in both supervised learning (Tripuraneni et al., 2020) and reinforcement learning (Lu et al., 2021).

Lu et al. (2021): In the prior work of Lu et al. (2021), which also studies transfer learning in low-rank MDPs with nonlinear function approximations, they need to make the following assumptions:

1. shared representation (identical to our Assumption 3.1).
2. task diversity (similar to our Assumption 3.4).
3. generative model access to both the source and the target tasks. In contrast, we only require generative model access to the source tasks and allow online learning in the target task.
4. a somewhat strong coverage assumption saying that the data covariance matrix (under the generative data distribution) between arbitrary pairs of features $\phi, \phi' \in \Phi$ must be full rank. In contrast, our analysis only requires coverage in the true feature $\phi^\star$ in the source tasks.
5. the existence of an ideal distribution q on which the learned representation can extrapolate. We do not require an assumption of a similar nature. Instead, we show that the data collected from our strategic reward-free exploration phase suffices for successful transfer.
6. the uniqueness for each ϕ in the sense of linear-transform equivalence. Two representation functions ϕ and ϕ' can yield similar estimation result if and only if they differ by just an invertible linear transformation. In contrast, we do not make any additional structural assumptions on the function class Φ beyond realizability.

In summary, our work present a theoretical framework that permits successful representation transfer based on significantly weaker assumptions. We believe that this is a solid step towards understanding transfer learning in RL.

Cheng et al. (2022): A concurrent work of Cheng et al. (2022) also studies the exact same problem as ours. Both works study the setting where the agent performs reward-free exploration in the source tasks for representation learning and used the learned representation for the target task. Both works achieves similar sample complexity in source and target tasks, albeit using very different algorithms. However, in addition to the assumptions that we made, Cheng et al. (2022) has to make an additional assumption.[4]

Assumption B.1 For any two different models in the model class $\mathcal{M}$, say $P^1(s'|s,a) = \langle \phi^1(s,a), \mu^1(s') \rangle$ and $P^2(s'|s,a) = \langle \phi^2(s,a), \mu^2(s') \rangle$, there exists a constant C_R such that for all $(s,a) \in \mathcal{S} \times \mathcal{A}$ and $h \in [H]$,

$$\|P_h^1(\cdot|s,a) - P_h^2(\cdot|s,a)\|_{TV} \le C_R \mathbb{E}_{(s,a) \sim \mathcal{U}(\mathcal{S} \times \mathcal{A})} \|P_h^1(\cdot|s,a) - P_h^2(\cdot|s,a)\|_{TV}$$

This assumption ensures that the point-wise TV error is bounded, as long as the population-level TV error is bounded. This assumption is used to transfer the MLE error from the source tasks to the target task. This type of assumption is strong in the sense that we typically expect C_R to scale with $|\mathcal{S}|$. In contrast, our analysis (Lemma F.1) shows that this assumption is in fact not necessary assuming online access only to source tasks. The generative access to source task studied here, which enables transfer under weaker reachability assumptions is not studied in their work.

It is worth noting that Cheng et al. (2022) also study offline RL in the target task which we do not cover, while we mainly focus on the setting of generative models in the source tasks and demonstrating a more complete picture by proving generative model access in source tasks is needed without

[4] with a the minor difference that they directly assume the state space is compact with bounded measure

additional assumptions. Comparing to (Cheng et al., 2022), we also further implement and perform experimental evaluations of our algorithm.

C Impossibility Results

Here, we present an interesting result showing that the above assumptions we make are so weak that they do not even permit ef cient transfer in supervised learning:

Theorem C.1 (Counter-example in supervised learning). Assume that we want to perform conditional density estimation, where $P_k^?(y|x) = \phi^?(x)^\top \psi_k^?(y)$. Under Assumption 3.1 (shared representation) and Assumption 3.4 (linear span), and assume that in each source task, one have access to a data generating distribution $\rho_k(x)$ such that $\min \mathbb{E}_{\rho_k}[\phi^?(x)\phi^?(x)^\top] \geq \alpha$ (reachability). No algorithm can consistently achieve $\mathbb{E}_{\rho_K}[\|\hat{P}_K^?(y|x) - P_K^?(y|x)\|_{TV}] \leq 1/2$ on the target task using the feature learned from the source tasks with probability more than $1/2$.

Proof of Theorem C.1. Consider the following example. $\mathcal{X} = \mathbb{R}^2$ and we have the following 3 sets.

$$S_1 = B_{1/2}((-1, -1))$$
$$S_2 = B_{1/2}((2, 2))$$
$$S_3 = B_{1/2}((0, 1))$$

where $B_a((x, y))$ stands for the ball with radius a centered at (x, y). These will be the support of 3 tasks: task 1 and 2 are two source tasks, task 3 is the target task. Let's assume $P_k^?(x)$ are uniform distribution on S_k.

Suppose that the feature class only contains two functions:

$$\phi_1 : \mathbb{R}^2 \to \{0, 1\}, \quad \phi_1(x) = 0 \text{ if } x_1 \leq x_2 \text{ else } 1, \text{ and } (1, 0), (0, 1) \text{ otherwise}$$
$$\phi_2 : \mathbb{R}^2 \to \{0, 1\}, \quad \phi_2(x) = 0 \text{ if } x_2 \leq x_1 \text{ else } 1, \text{ and } (0, 1), (1, 0) \text{ otherwise}$$

That is, the feature maps from $\mathbb{R}^2$ to the set of binary encoding of dimension 2, i.e. $\{(1, 0), (0, 1)\}$. We further assume that $\psi_k^? = (p_1(y), p_2(y))$ for some distributions p_1, p_2, which is identical for all task k, where $\|p_1, p_2\|_{TV} = 1$. We also assume that $\psi_k^?$ is known to the learner a prior, i.e. $\Psi_k = \{\psi_k^?\}$ for all $k \in [K]$, so all the learner needs to do is to pick the correct ϕ out of two candidates.

Given the above setup, it's easy to verify that both Assumption 3.1 and Assumption 3.4 are satis ed, because the decision boundary of both ϕ_1 and ϕ_2 passes through the support of the source tasks, and all $\psi_k^?$'s are identical. However, ϕ_1 and ϕ_2 are equivalent in S_1 and S_2 in terms of their representation power, therefore no algorithm can always pick the correct feature function with probability more than $1/2$, regardless of the number of samples. Suppose ϕ_1 is the true feature and the algorithm incorrectly chooses ϕ_2. Then, for $x \in S_3 \cap \{x_1 \leq 0\}$ which has probability mass $1/2$, $\hat{P}_3(y|x) = p_2$ whereas $P_3^?(y|x) = p_1$. Thus, the expected total variation distance between $\hat{P}_K$ and $P_K^?$ is $1/2$. □

The above construction shows that our assumption are not suf cient to permit reliable representation transfer, even in the supervised learning setting. Yet, surprisingly, these assumptions are sufficient in the RL setting, implying somehow that transfer learning in RL in easier than transfer learning in SL. To understand this phenomenon, observe that in RL, the marginal distribution $\rho(s, a)$ is not independent from the conditional density $P(s'|s, a)$ we desire to estimate. In particular, if one collects data in the source tasks in an online fashion via running a policy, $\rho(s, a)$ is structurally restricted to be an occupancy distribution generated by the ground-truth transition $P^?(s'|s, a)$. Such a connection can only exist in Markov chains, and our analysis elegantly utilizes this additional structure to establish the soundness of the learned representation. Also note, crucially, that we never learn a representation to capture ρ_k, which would suffer from similar issues as the supervised learning setting, but is not necessary for sample-ef cient RL.

Next, we prove the impossibility result in Theorem 5.1, restated below as Theorem C.3. This result shows that one can not achieve online learning in the source tasks without signi cantly stronger

assumptions such as Assumption 5.1. Before that, we provide a preliminary version, showing that the learned $\hat{}$ is not suf cient to t the transition model in the target task, which motivates the construction in Theorem C.3.

Theorem C.2 (Impossibility Result: Model Learning) Let M_K be the set of K-task multi-set that satis es

1. all tasks are Block MDPs;
2. all tasks satisfy Assumption 3.3 and Assumption 3.4;
3. the latent dynamics are exactly the same for all source and target tasks.

For any pre-training algorithm A, there exists $\{P_k^?\}_{k=1:K} \in M_K$ and an occupancy distribution $_K$ on the target task, such that with probability at least $1=2$, A will output a feature $\hat{}$ and for any

$$E_{\ K}\ \|\hat{}(s;a)^{>}\ (\ )\quad P_K^?(\ js;a)\|_{TV}\quad 1=2:$$

Proof of Theorem C.2 Consider a tabular MDP with 2 latent states z_1, z_2 and an observation state space $S = R_1 \ R_2 \ B_1 \ B_2$, where in task 1 one can only observe $R_1 \ R_2$ and in task 2 one can only observe $B_1 \ B_2$. Correspondingly $o_1(sjz)$ is only supported on $R_1 \ R_2$ (i.e., $o_1(R_i jz_i) = 1$) and similar for task 2. Let the latent state transition be such that $P(z_1 jz_1; a) = 1$ and $P(z_2 jz_2; a) = 1$, i.e. only self-transition regardless of the actions.

Now, consider a 2-element feature class $= \{\ _1;\ _2\}$ such that

$$\begin{aligned}
_1 &= \{R_1 \ ! \ \ 1; R_2 \ ! \ \ 2; B_1 \ ! \ \ 1; B_2 \ ! \ \ 2\} \\
_2 &= \{R_1 \ ! \ \ 1; R_2 \ ! \ \ 2; B_1 \ ! \ \ 2; B_2 \ ! \ \ 1\}
\end{aligned}$$

Denote $_i(s;a) = e_{(\ _i(s);a)}$ for $i \in [1;2]$. Consider for each task k, a 2-element $_k$ class in the form of $_k = \{(o_k(sjz_1); o_k(sjz_2)); (o_k(sjz_2); o_k(sjz_1))\}$.

Notice that $_1$ and $_2$ are merely permutations of one another and so given any single task data, the two hypothesis will not be distinguishable by any means. Therefore, for any algorithm, there is at least probability $1=2$ that it will choose the wrong hypothesis if the ground truth is sampled between $_1$ and $_2$ uniformly at random. Suppose $_1$ is the correct hypothesis and $_2$ is the one that the algorithm picks (i.e., $\hat{} = \ _2$). Let task 3 be such that any state emits to $R_1 \ R_2$ and $B_1 \ B_2$ each with probability $1=2$ (i.e., $o_3(R_i jz_i) = o_3(B_i jz_i) = 0:5$). This construction satis es Assumption 3.3 and Assumption 3.4.

Then, within task 3, one would encounter observations from both $R_1 \ R_2$ and $B_1 \ B_2$ which should be mapped to latent state z_1 and z_2 respectively by the true decoder $_1$, but are instead both mapped to latent state z_1 by the learned decoder $_2$, and thus z_1 and z_2 become indistinguishable. Suppose $_K(z_1) = \ _K(z_2) = 1=2$, then

$$\begin{aligned}
&E_{\ \kappa}[\|\hat{}(s;a)^{>}\ (\ )\quad P^?(\ js;a)\|_{TV}] \\
&= \frac{1}{4}\|\hat{}(R_1)^{>}\ (\ )\quad (R_1)^{?>}\ ^?(\ )\|_{TV} + \frac{1}{4}\|\hat{}(B_1)^{>}\ (\ )\quad (B_1)^{?>}\ ^?(\ )\|_{TV} \\
&\quad + \frac{1}{4}\|\hat{}(R_2)^{>}\ (\ )\quad (R_2)^{?>}\ ^?(\ )\|_{TV} + \frac{1}{4}\|\hat{}(B_2)^{>}\ (\ )\quad (B_2)^{?>}\ ^?(\ )\|_{TV} \\
&= \|o_1\quad o_1^?\|_{TV}=4 + \|o_2\quad o_1^?\|_{TV}=4 + \|o_2\quad o_2^?\|_{TV}=4 + \|o_1\quad o_2^?\|_{TV}=4 \\
&\quad \frac{1}{4}\|o_1^?\quad o_2^?\|_{TV} + \frac{1}{4}\|o_1^?\quad o_2^?\|_{TV} \\
&= \frac{1}{2};
\end{aligned}$$

where the last second inequality uses triangle inequality, and the last equality comes from the fact that $o_3(\ jz_1)$ and $o_3(\ jz_2)$ have disjoint support which implies that $\|p_1^?\quad p_2^?\|_{TV} = 1$. □

Now, we are ready to restate and prove Theorem 5.1.

Theorem C.3 (Impossibility Result: Optimal Policy Identification). Let $\mathcal{M}_K$ be the set of K-task multi-set that satisfies

1. all tasks are Block MDPs;
2. all tasks satisfy Assumption 3.3 and Assumption 3.4;
3. the latent dynamics are exactly the same for all source and target tasks.

For any pre-training algorithm $\mathcal{A}$, there exists $\{P_k^?\}_{k=1:K} \in \mathcal{M}_K$, such that with probability at least $1/2$, $\mathcal{A}$ will output a feature $\hat{}$, such that for any policy taking the functional form of $(s) = f(\hat{}(s,a))_{a\in\mathcal{A}}; r(s,a))_{a\in\mathcal{A}}$, we have

$$V^? \quad V \quad 1/2.$$

Proof of Theorem C.3. Consider a tabular MDP with $H = 2$, two latent states z_1, z_2 for $h = 1$ and two latent states z_3, z_4 for $h = 2$.

- For $h = 1$, let there be two actions a_1, a_2. Let the observation state space be $\mathcal{S} = \{R_1, R_2, B_1, B_2\}$, where in task 1 one can only observe R_1, R_2 and in task 2 one can only observe B_1, B_2. Correspondingly, $o_1(s|z)$ is only supported on R_1, R_2 (i.e., $o_1(R_i|z_i) = 1$) and similar for task 2. Let the latent state transition be such that $P(z_3|z_1,a_1) = P(z_3|z_2,a_2) = 1$, and $P(z_4|z_1,a_2) = P(z_4|z_2,a_1) = 1$. All rewards are 0 for $h = 1$.
- For $h = 2$, in state z_3, all actions have reward 1, and in state z_4 all actions have reward 0.
- The initial state distribution is $d_0(z_1) = d_0(z_2) = 1/2$.

Now, consider a 2-element feature class $= \{\ _1, \ _2\}$ for $h = 1$, such that

$$\ _1 = \{R_1 \to 1, R_2 \to 2, B_1 \to 1, B_2 \to 2\}$$
$$\ _2 = \{R_1 \to 1, R_2 \to 2, B_1 \to 2, B_2 \to 1\}$$

Denote $\ _i(s,a) = e_{(\ _i(s),a)}$ for $i \in [1,2]$. In addition, define $= \{\ _1, \ _2\}$ where

$$\ _1 = \{z_3 \to (1,0), z_4 \to (0,1)\}$$
$$\ _2 = \{z_4 \to (1,0), z_3 \to (0,1)\}$$

Notice that $\ _1$ and $\ _2$ are merely permutations of one another and so given any single task data, the two hypothesis will not be distinguishable by any means. Therefore, for any algorithm, there is at least probability $1/2$ that it will choose the wrong hypothesis. Suppose $\ _1$ is the correct hypothesis and $\ _2$ is the one that the algorithm picks (i.e., $\hat{} = \ _2$). Let task 3 be such that any state emits to R_1, R_2 and B_1, B_2 each with probability $1/2$ (i.e., $o_3(R_i|z_i) = o_3(B_i|z_i) = 0.5$). This construction satisfies Assumption 3.3 and Assumption 3.4.

Then, for any policy that only make decision based on $\hat{}(s,a)$ and $r(s,a)$, would output the same action for observations in R_1 and B_2, or for B_1 and R_2. However, notice that the optimal policy, which would try to go to z_3 from either z_1 or z_2, will pick a_1 at R_1 and B_1 while picking a_2 at R_2 and B_2, which means that the optimal policy will not agree on R_1 and B_2, and it also will not agree on R_2 and B_1. Thus clearly, no such policy as defined above is capable of capturing the optimal policy. From the reward perspective, notice that $d(z_1) = d(z_2) = 1/2$ and $d(R_1) = d(R_2) = d(B_1) = d(B_1) = 1/4$. Since $(R_1) = (B_2)$, the agent will only be able to collect reward at one of the R_1 and B_2 (but not at both). Similarly, since $(R_2) = (B_1)$, the agent will only be able to collect reward at one of the R_2 and B_1 by reaching z_3 (but not at both). This means that will have average reward $1/2$. Since the optimal policy will be able to collect reward at all R_1, R_2, B_1, B_2, it will have average reward 1. This concludes the proof. □

Algorithm 3 Multi-task REPLEARN

Input: Datasets $\{D_k\}_{k=1:K_1}$, model class $\Phi, \Psi_k, k = 1, \ldots, K_1$.

Compute MLE $(\hat{\Phi}, \hat{\Psi}_{1:K_1}) := \underset{\Phi \in \Phi, \Psi_k \in \Psi_k}{\operatorname{argmax}} \sum_{k=1}^{K_1} \mathbb{E}_{D_k} \log \mathbb{P}(s,a)^\top \Psi_k(s')$.

Return $\hat{\Phi}$

Algorithm 4 REWARDFREE REP-UCB

Input: Regularizer λ_n, bonus scaling β_n, model class $\mathcal{M} = \Phi$, number of episodes N.

Initialize π_0 as random and $D_{h,0}, D'_{h,0} = \emptyset$.

for episode $n = 1, 2, \ldots, N$ do

Data collection from π_{n-1}: for $h = 1, 2, \ldots, H-1$,

$$s \sim d_h^{\pi_{n-1}}; a \sim \mathrm{Unif}(A); s' \sim P_h^\star(s,a);$$
$$s \sim d_h^{\pi_{n-1}}; a \sim \mathrm{Unif}(A); s' \sim P_{h-1}^\star(s,a); a' \sim \mathrm{Unif}(A); s'' \sim P_h^\star(s',a');$$
$$D_{h,n} = D_{h,n-1} \cup \{(s,a,s')\}; D'_{h,n} = D'_{h,n-1} \cup \{(s',a',s'')\}.$$

For $h = 0$, only collect $D_{0,n}$.

Learn model via MLE: for all $h = 0, 1, \ldots, H-1$,

$$\hat{\Phi}_{h,n} = (\hat{\Phi}_{h,n}, \hat{\Psi}_{h,n}) = \underset{\Phi_h \in \mathcal{M}_h, \Psi_h}{\operatorname{argmax}} \mathbb{E}_{D_{h,n}} [D'_{h,n} \log \Phi_h(s,a)^\top \Psi_h(s')].$$

Update exploration bonus: for all $h = 0, 1, \ldots, H-1$,

$$\beta_{h,n}(s,a) = \beta_n \left\| \hat{\Phi}_{h,n}(s,a) \right\|_{\Lambda_{h,n}^{-1}}$$
$$\Lambda_{h,n} = \sum_{(s,a,\_) \in D_{h,n}} \hat{\Phi}_{h,n}(s,a) \hat{\Phi}_{h,n}(s,a)^\top + \lambda_n I.$$

Learn policy $\hat{\pi}_n = \operatorname{argmax}_\pi V^{\pi, \hat{\Phi}_n, \beta_n}$ and let $\hat{V}_n$ be its value.

Let $\hat{n} = \operatorname{argmin}_{n \in [N/2]} \hat{V}_n$.

Output: $\hat{\pi} = \hat{\pi}_{\hat{n}}$, $\hat{\Phi} = \hat{\Phi}_{\hat{n}}$.

Theorem C.2 and Theorem C.3 show that it's impossible to allow online learning in the source tasks without much stronger assumptions. In our paper, we show that our Assumption 5.1, which ensures reachability in the raw states, is sufficient to establish an end-to-ending online transfer learning result. However, it is unclear if Assumption 5.1 is necessary for online learning. We leave this as an important direction of future work.

D Reward-free Rep-UCB

In this section, we adapt the Rep-UCB algorithm (Uehara et al., 2021) for reward-free exploration in a single task. We drop all task subscripts as this section is for a single task only, i.e. think about the task as being each source task. The original Rep-UCB algorithm was for infinite-horizon discounted MDPs, so we modify it to work for our undiscounted and finite-horizon setting. Our goal is to prove that Rep-UCB can learn a model that satisfies strong TV guarantees, i.e. Theorem D.1 and (5). Note that FLAMBE (Agarwal et al., 2020, Theorem 2) can be used for this directly, but at a worse (polynomial) sample complexity. Thus, we do a bit more work to derive a new model-learning algorithm for low-rank MDPs, based on Rep-UCB, that is more sample efficient in the source tasks.

A finite-horizon analysis of Rep-UCB was done in BRIEE (Zhang et al., 2022), so here we just need to replace BRIEE's RepLearn$_h$ with that of the MLE, which is how we learn $\hat{\Phi}$ and $\hat{b}$, as in

Rep-UCB. Recall the notation of (Zhang et al., 2022),

$$\pi_{h;n}(s;a) = \frac{1}{n}\sum_{i=0}^{n-1} d_h^{b_i}(s)\,\mathrm{Unif}(a)$$

$$\pi_{h;n}(s;a) = \frac{1}{n}\sum_{i=0}^{n-1} \mathbb{E}_{s\sim d_h^{b_i}(\cdot),a\sim\mathrm{Unif}(A)}[P_h^?(s'\mid s,a)\,\mathrm{Unif}(a)]$$

$$\pi_{h;n}(s;a) = \frac{1}{n}\sum_{i=0}^{n-1} d_h^{b_i}(s;a)$$

$$\Sigma_{\pi;n} = n\mathbb{E}_{\pi}[\phi(s;a)\,\phi(s;a)^T] + \lambda_n I.$$

By using MLE (Uehara et al., 2021, Lemma 18) to learn models, with probability at least $1-\delta$, for any $n = 1;2;\ldots;N$ and $h = 0;1;\ldots;H-1$, we have

$$\max\left\{\mathbb{E}_{\pi_{h;n}}\left[\left\|\widehat{P}_{h;n}(s;a) - P_h^?(s;a)\right\|_{TV}^2\right], \mathbb{E}_{\pi_{h;n}}\left[\left\|\widehat{P}_{h;n}(s;a) - P_h^?(s;a)\right\|_{TV}^2\right]\right\} \le \beta_n; \quad (3)$$

where

$$\beta_n = O\left(\frac{\log(|\mathcal{M}|nH/\delta)}{n}\right);$$

and $|\mathcal{M}| = \max_{h\in[H]}|\mathcal{P}_h||\mathcal{R}_h|$. We also adopt the same choice of λ_n parameters as BRIEE, which we assume from now on.

$$\lambda_n = (d\log(|\mathcal{M}|nH/\delta))$$

$$\lambda_n = p\frac{}{nA^2\beta_n + \beta_n d}:$$

As in Rep-UCB, we posit standard assumptions about realizability and normalization, on the (source) task of interest.

Assumption D.1 For any $h = 0;1;\ldots;H-1$, we have $P_h^? \in \mathcal{P}_h$ and $R_h^? \in \mathcal{R}_h$. For any $\phi \in \Phi_h$, $\|\phi(s;a)\|_2 \le 1$. For all $\phi \in \Phi_h$ and any function $g : S \to \mathbb{R}$, we have $\|\int_s g(s)d\phi(s)\|_2 \le \|g\|_1\sqrt{d}$.

Lemma D.1 Let r be any reward function. Suppose we ran Algorithm 4 with line 7 having reward $r + b_n$ instead of just b_n. Then, for any $\delta \in (0;1)$, w.p. at least $1-\delta$, we have

$$\sum_{n=0}^{N-1} V_{\widehat{P}_n;r+b_n}^{b_n} - V_{P^?;r}^{b_n} \le O\left(H^2d^2A^{1.5}\sqrt{N\log(|\mathcal{M}|NH/\delta)}\right)$$

Proof. Start from the third equation of Zhang et al. (2022, Theorem A.4). Following their proof until the last page of their proof, we arrive at the following: for any $n = 1;2;\ldots;N$,

$$V_{\widehat{P}_n;r+b_n}^{b_n} - V_{P^?;r}^{b_n}$$

$$\le \sum_{h=0}^{H-2}\mathbb{E}_{s,a\sim d_{P^?;h}^{b_n}}\left[\|\phi_h^?(s;a)\|_{\Sigma_{\pi_{h;n};\lambda_n}^{-1}}\right]\sqrt{\frac{\beta_n}{A\beta_n + \lambda_n d + \lambda_n d}} + \sqrt{\frac{\beta_n}{A\beta_1 d/n}}$$

$$+ (2H+1)\sum_{h=0}^{H-2}\mathbb{E}_{s,a\sim d_{P^?;h}^{b_n}}\left[\|\phi_h^?(s;a)\|_{\Sigma_{\pi_{h;n};\lambda_n}^{-1}}\right]\sqrt{\frac{\beta_n}{nA\beta_n + \lambda_n d}} + (2H+1)\sqrt{\frac{\beta_n}{A\lambda_n}}.$$

By elliptical potential arguments, we have

$$\sum_{n=0}^{N-1}\mathbb{E}_{s,a\sim d_{P^?;h}^{b_n}}\left[\|\phi_h^?(s;a)\|_{\Sigma_{\pi_{h;n};\lambda_n}^{-1}}\right] \le \sqrt{dN\log\left(1 + \frac{N}{d\lambda_1}\right)}:$$

Thus, summing over n, noting that $\lambda_n; \beta_n; \lambda_n$ are increasing in n, we can combine the above to get,

$$\sum_{n=0}^{N-1} V_{\widehat{P}_n;r+b_n}^{b_n} - V_{P^?;r}^{b_n}$$

$$\cdot \quad dN \log\left(1 + \frac{N}{d}\right)_{1} \quad H \sqrt{jAj\,{}^{2}_{N}d + {}_{N}d} + H^{2}\sqrt{NjAj\,{}_{N} + {}_{N}d}$$

$$\cdot \quad dN \log\left(1 + \frac{N}{d}\right)_{1} \quad H\sqrt{NjAj^{3}\,{}_{N}d + {}_{N}d^{2}} + H^{2}\sqrt{NjAj\,{}_{N} + {}_{N}d}$$

$$\cdot \quad dN \log\left(1 + \frac{N}{d}\right)_{1} \quad H^{2}\sqrt{djAj^{3}\log(jMj\ NH=\ ) + d^{3}\log(jMj\ NH=\ )}$$

$$2\ O\quad H^{2}d^{2}jAj^{1:5}\sqrt{N\log(jMj\ NH=\ )}\ :$$

□

This gives the following useful corollary for reward free exploration.

Lemma D.2 For any $\ 2\ (0;1)$ w.p. at least $1\ $ we have

$$\forall \hat{b}_{\hat{b}}\quad O\quad H^{2}d^{2}jAj^{1:5}\sqrt{\frac{\log(jMj\ NH=\ )}{N}}\ :$$

Proof. By de nition of $\forall \hat{b}_{\hat{b}}$, we have

$$\frac{N}{2}\forall \hat{b}_{\hat{b}} \quad \sum_{n=N=2}^{N-1} V^{\hat{b}_{\hat{b}}}_{\hat{p}_{n};\hat{b}_{n}} \quad \sum_{n=0}^{N-1} V^{\hat{b}_{\hat{b}}}_{\hat{p}_{n};\hat{b}_{n}};$$

which is bounded by the previous lemma and the fact that $V^{\hat{b}_{n}}_{P^{?};r=0} = 0$, since in Algorithm 4, the reward function is zero. □

Conditioning on this, we now show that the environment $\hat{b}_{\hat{b}}$ has low TV error for any policy-induced distribution.

Theorem D.1 For any policy $\ $, we have

$$\sum_{h=0}^{H-1} E_{d_{P^{?};h}}\quad P^{?}_{h}(s;a)\quad \hat{p}_{h;\hat{b}}(s;a)\ _{TV}\quad O\quad H^{3}d^{2}jAj^{1:5}\sqrt{\frac{\log(jMj\ NH=\ )}{N}}\ :=\ "_{TV}:$$

Proof. In this proof, let $\hat{b} = \hat{b}_{\hat{b}}$, which is the returned environment from the algorithm. Let $r(s;a) = \ P^{?}_{h}(s;a)\quad \hat{p}_{h}(s;a)\ _{TV}\ 2\ [0;2]$. Then,

$$\begin{aligned}
&\sum_{h=0}^{H-1} E_{d_{P^{?};h}}\quad E_{d_{\hat{p}\hat{b};h}}\ [r(s;a)] \\
&= V_{P^{?};r}\quad V_{\hat{p}\hat{b};r} \\
&= \sum_{h=0}^{H-1} E_{d_{\hat{p}\hat{b};h}}\quad E_{P^{?}_{h}(s;a)}\quad E_{\hat{p}\hat{b}_{h}(s;a)}\quad V_{P^{?};r;h+1}(s^{0}) \qquad \text{(Simulation lemma)} \\
&\quad 2H\sum_{h=0}^{H-1} E_{d_{\hat{p}\hat{b};h}}\quad P^{?}_{h}(s;a)\quad \hat{p}_{h}(s;a)\ _{TV}:
\end{aligned}$$

Thus,

$$\sum_{h=0}^{H-1} E_{d_{P^{?};h}}\quad P^{?}_{h}(s;a)\quad \hat{p}_{h}(s;a)\ _{TV}$$

$$(2H+1)\sum_{h=0}^{H-1} \mathbb{E}_{d^{\hat{\pi},h}}\left\| P_h^?(s,a) - \hat{P}_h(s,a)\right\|_{TV} \qquad \text{(by (Zhang et al., 2022, Lemma A.1))}$$

$$\le H\sum_{h=0}^{H-2}\mathbb{E}_{d^{\hat{\pi},h}}\left[\beta_{h,b_h}(s,a) + \left\|\hat{P}_h - P_h^?\right\|_1\right] + \sqrt{\frac{|\mathcal{S}|}{N}}$$

$$\le H\,V^{\hat{\pi},b_{th}}_{\hat{P},\beta_{b_{th}}} + 2|\mathcal{A}|\sqrt{\frac{\log(|\mathcal{S}||\mathcal{A}|H/\delta)}{N}}$$

$$\le H\,V^{b_{th}}_{\hat{P},\beta_{b_{th}}} + 2|\mathcal{A}|\sqrt{\frac{\log(|\mathcal{S}||\mathcal{A}|H/\delta)}{N}}$$

$$\le H^2 d^2|\mathcal{A}|\sqrt{\frac{\log(|\mathcal{S}||\mathcal{A}|H/\delta)}{N}} + |\mathcal{A}|\sqrt{\frac{\log(|\mathcal{S}||\mathcal{A}|H/\delta)}{N}} \qquad \text{(by Lemma D.2)}$$

$$\le 2\,H^3 d^2|\mathcal{A}|\sqrt{\frac{\log(|\mathcal{S}||\mathcal{A}|H/\delta)}{N}}\,.$$

□

This also gives us a guarantee on the TV distance between the visitation distributions induced by $\hat{P}$ vs. by $P^?$.

Lemma D.3 Suppose $\hat{\pi}$ satisfies the following for all $h = 0, 1, \ldots, H-1$,

$$\forall s : \left\| d^{\hat{P}^?,h} - \hat{d}^{\hat{\pi},h}\right\|_{TV} \le \varepsilon_h. \tag{4}$$

Then, for any $h = 0, 1, \ldots, H-1$, we have

$$\forall s : \left\| d^{\hat{P},h} - d^{\hat{P}^?,h}\right\|_{TV} \le \sum_{t=0}^{h-1}\varepsilon_t.$$

Note, for $h = 0$, the sum is empty so the right hand side is 0.

Proof. We proceed by induction for $h = 0, 1, \ldots, H-1$. For the base case of $h = 0$, no transition has been taken, so that $d^{\hat{P},0} = d^{\hat{P}^?,0}$. Now let $h \in \{0, 1, \ldots, H-2\}$ be arbitrary, and suppose that the claim is true for h (IH). We want to show the claim holds for $h+1$. One key fact we'll use is that, for any measure, we have $\|\cdot\|_{TV} = \sup_{\|f\|_\infty \le 1} |\mathbb{E}[f]|$. Below we use the notation that $f(s,\pi) = \mathbb{E}_{a\sim\pi(s)} f(s,a)$.

$$\begin{aligned}
&\left\| d^{\hat{P},h+1} - d^{h+1}\right\|_{TV} \\
&= \sup_{\|f\|_\infty \le 1} \mathbb{E}_{d^{\hat{P},h+1}}[f(s,a)] - \mathbb{E}_{d^{h+1}}[f(s,a)] \\
&= \sup_{\|f\|_\infty \le 1} \mathbb{E}_{(s,a)\sim d^{\hat{P},h};(s,a),\hat{P}_h(s,a)}[f(s,\pi^{h+1})] - \mathbb{E}_{(s,a)\sim d^{h};(s,a),P_h^?(s,a)}[f(s,\pi^{h+1})] \\
&\le \sup_{\|f\|_\infty \le 1} \mathbb{E}_{(s,a)\sim d^{\hat{P},h}}\mathbb{E}_{\hat{P}_h(s,a)} f(s,\pi^{h+1}) - \mathbb{E}_{(s,a)\sim d^{h}}\mathbb{E}_{\hat{P}_h(s,a)} f(s,\pi^{h+1}) \\
&\quad + \sup_{\|f\|_\infty \le 1} \mathbb{E}_{(s,a)\sim d^{h}}\left[\mathbb{E}_{\hat{P}_h(s,a)} f(s,a) - \mathbb{E}_{P_h^?(s,a)} f(s,\pi^{h+1})\right] \\
&\le \sum_{t=0}^{h-1}\varepsilon_t + \mathbb{E}_{(s,a)\sim d^{h}}\sup_{\|f\|_\infty \le 1}\left[\mathbb{E}_{\hat{P}_h(s,a)} - \mathbb{E}_{P_h^?(s,a)}\right] f(s,\pi^{h+1}) \qquad \text{(by IH and Jensen)} \\
&\le \sum_{t=0}^{h-1}\varepsilon_t + \varepsilon_h; \qquad \text{(by (4) and } \|f(s,\pi^{h+1})\|_\infty \le 1)
\end{aligned}$$

as desired. □

Thus, when combined with Theorem D.1, we have for $h = 0, 1, \ldots, H-1$ and any policy π,

$$\|\hat{P}_{\hat{M},h} - P^{\star}_h\|_{TV} \le O\left(H^3 d^2 |A|^{1.5} \sqrt{\frac{\log(|\mathcal{M}| NH/\delta)}{N}}\right) = \varepsilon_{TV}. \tag{5}$$

In other words, the sample complexity needed for a model-error of ε_{TV} is

$$O\left(\frac{H^6 d^4 |A|^3 \log(|\mathcal{M}| NH/\delta)}{\varepsilon_{TV}^2}\right).$$

Note this is much better than FLAMBE's guarantee (Agarwal et al., 2020, Theorem 2) which requires,

$$O\left(\frac{H^{22} d^7 |A|^9 \log(|\mathcal{M}| NH/\delta)}{\varepsilon_{TV}^{10}}\right).$$

E Reward-free Exploration

In this section, we show that the mixture policy returned by Algorithm 2 has good coverage. Recall that Algorithm 2 contains two main steps:

Step 1 Learn a model $\hat{P}$. This was the focus of the previous section, where our model-based method obtained a strong TV guarantee ((5)) by requiring number of episodes at most,

$$N_{\text{REWARDFREE}} = O\left(\frac{H^6 d^4 |A|^3 \log(|\mathcal{M}| NH/\delta)}{\varepsilon_{TV}^2}\right).$$

Step 2 Run LSVI-UCB (Algorithm 6) in the learned model $\hat{P}$ with reward at the e-th episode being $b_{h;e}$ and UNIFORMACTIONS = TRUE. The optimistic bonus pushes the algorithm to explore directions that are not well-covered yet by the mixture policy up to this point. With elliptical potential, we can establish that this process will terminate in polynomial number of steps.

We now focus on Step 2. Let π_h^{+1} denote rolling-in π for h steps and taking uniform actions on the $h+1$ step, thus inducing a distribution over s_{h+1}, a_{h+1}. We abuse the notation a little and use π_1^{+1} for a policy that just takes one uniform action from the initial distribution d_0.

Lemma E.1 Let $\delta \in (0,1)$ and run REWARDFREE (Algorithm 2). Let $\Lambda_{h;N}$ be the empirical covariance at the N-th iteration of LSVI-UCB (Algorithm 6). Then, w.p. at least $1-\delta$ we have,

$$\sup_{h=0}^{H-1} \mathbb{E}_{s_h, a_h \sim \pi_1^{+1}, \hat{P}} \|b_h(s_h, a_h)\|_{\Lambda_{h;N}^{-1}} \le O\left(d|A|^{1.5} H^3 \sqrt{\log(d|\mathcal{M}|NH/\delta)/N}\right).$$

Proof. In this proof, we'll treat the empirical MDP as $P^{\star}$, as that is the environment we're running in. Thus, we abuse notation and $\hat{M}_{h;e}$ is the model-based perpsective of the linear MDP, $\langle \hat{r}_h, \hat{P}_{h;e} \rangle$ where $\hat{P}_{h;e}$ is $\frac{1}{h;e} \sum_{k=1}^{e-1} b_h(s_h^k, a_h^k)(s_{h+1}^k)$. Also, in Algorithm 2, we set reward to be zero, but for the purpose of this analysis, suppose the reward function is precisely the (unscaled) bonus in LSVI-UCB, i.e. $r_{h;e}(s_h, a_h) = b_{h;e}(s_h, a_h)$. This does not change the algorithm at all since the β-scaling of the bonus dominates this reward in the definition of $\hat{Q}_{h;e}$, but thinking about the reward in this way will make our analysis simpler.

Recall the high-level proof structure of reward free guarantee of linear MDP (with known features b_h) (Wang et al., 2020, Lemma 3.2).

Step 1 Show that $\hat{Q}_{h;e} \in \mathcal{V}_h$ and w.p.1 $1-\delta$, for all h, e,

$$\forall s_h, a_h : \sup_{f \in \mathcal{V}_h} \hat{P}_{h;e}(s_h, a_h) f - P_h^{\star}(s_h, a_h) f \le b_{h;e}(s_h, a_h).$$

This step only uses self-normalized martingale bounds. So, line 9 can use any martingale sequence of states and actions, and this claim still holds, with $b_{h;e}$ using the appropriate covariance under the data.

Step 2 Show optimism conditioned on Step 1. Specifically, for all $h = 1, 2, \ldots, H$, we have $\mathbb{E}_{d_0}[V_0^?(s_0, r_e) - \hat{V}_{0,e}(s_0)] \geq 0$. To show this, we need that $\hat{b}_{h,e}(s_h) = \hat{Q}_{h,e}(s_h, \pi_h^e(s_h)) - \hat{Q}_{h,e}(s_h, \pi_h^?(s_h))$ (this is for the unclipped case of optimism), which we have satisfied in the algorithm, i.e. π_h^e is greedy w.r.t. $\hat{Q}_{h,e}$.

Step 3 Bound the sum $\sum_e \hat{V}_{h,e}$, where we decompose it as a sum of expected bonuses with the expectation is under π^e.

Step 3 is the only place where we use the fact that s_h^k, a_h^k are data sampled from rolling out π^e. For Step 1 and 2, please refer to existing proofs in (Agarwal et al., 2019; Jin et al., 2020b; Wang et al., 2020).

Now we show Step 3 for our modified algorithm with uniform actions. First, let us show a simulation lemma. For any episode $e = 1, 2, \ldots, N$, for any s, recalling definition of reward being $\hat{b}_{h,e}$, we have

$$\begin{aligned}\hat{V}_{0,e}(s_0) &\leq (1+\alpha)\hat{b}_{0,e}(s_0, \pi_0^e(s_0)) + \hat{P}_{0,e}(s_0, \pi^e(s_0))\hat{V}_{1,e} \\ &\leq (1+2\alpha)\hat{b}_{0,e}(s_0, \pi_0^e(s_0)) + P^?_{0,e}(s_0, \pi^e(s_0))\hat{V}_{1,e},\end{aligned}$$

where the first inequality is due to the thresholding of $\hat{b}_{h,e}$'s and the second inequality is due to Step 1. Continuing in this fashion, we have

$$\mathbb{E}_{d_0}\left[\hat{V}_{0,e}(s_0)\right] \leq (1+2\alpha)\sum_{h=0}^{H-1} \mathbb{E}_{\pi^e}[\hat{b}_{h,e}(s_h, a_h)].$$

Summing over $e = 1, 2, \ldots, N$, we have

$$\begin{aligned}\sum_{e=1}^N \mathbb{E}_{d_0}\left[\hat{V}_{0,e}(s_0)\right] &\leq \cdot \sum_{h=0}^{H-1}\sum_{e=1}^N \mathbb{E}_{\pi^e}[\hat{b}_{h,e}(s_h, a_h)] \\ &\leq \cdot \sum_{h=0}^{H-1}\sum_{e=1}^N \mathbb{E}_{(\pi_h^e, \pi_{h+1}^?)}[\hat{b}_{h,e}(s_h, a_h)]\end{aligned}$$

For each $h = 0, 1, \ldots, H-1$, apply Azuma's inequality to the martingale difference sequence $\Delta_e = \mathbb{E}_{(\pi_h^e, \pi_{h+1}^?)}[\hat{b}_{h,e}(s_h, a_h)] - \hat{b}_{h,e}(s_h^e, a_h^e)$. The envelope is at most 2α. So, w.p. $1-\delta$,

$$\leq \sum_{h=0}^{H-1}\sum_{e=1}^N \hat{b}_{h,e}(s_h^e, a_h^e) + \sqrt{N\log(H/\delta)}.$$

Now apply a self-normalized elliptical potential bound to the first term, giving that

$$\sum_{h=0}^{H-1}\sum_{e=1}^N \hat{b}_{h,e}(s_h^e, a_h^e) \leq \sum_{h=0}^{H-1} \sqrt{N\sum_{e=1}^N \hat{b}_{h,e}(s_h^e, a_h^e)^2} \leq H\sqrt{dN\log(N)}.$$

Thus, we finally have

$$\sum_{e=1}^N \mathbb{E}_{d_0}\left[\hat{V}_{0,e}(s_0)\right] \leq \cdot H\sqrt{dN\log(NH/\delta)}.$$

Consider any episode $e = 1, 2, \ldots, N$. By definition, $\hat{b}_{h,N} \geq \hat{b}_{h,e}$, so for all s, a we have pointwise that $\hat{b}_{h,N}(s, a) \geq \hat{b}_{h,e}(s, a)$. Hence, for all s, we have $V_0^?(s, r^N) \geq V_0^?(s, r^e)$, and further using optimism, we have

$$N\,\mathbb{E}_{d_0}[V_0^?(s_0, r_N)] \leq \sum_{e=1}^N \mathbb{E}_{d_0}[V_0^?(s_0, r_e)] \leq \sum_{e=1}^N \mathbb{E}_{d_0}\left[\hat{V}_{0,e}(s_0)\right] \cdot H\sqrt{dN\log(NH/\delta)}.$$

Now consider any h and policy π, and consider rolling it out for $h-1$ steps and taking a random action. Then we have

$$\mathbb{E}_{s_{h-1},a_{h-1}\sim d^{\pi}_{h-1}} b_h(s_h,a_h) \le \frac{1}{HN} \mathbb{E}_{d_0} V_0^{r+b^{\pi}_{h:N}}(s_0) \le AH\sqrt{d\log(NH/\delta)/N}.$$

Summing over h incurs an extra H factor on the right. This concludes the proof. □

Lemma E.2 (One-step back for Linear MDPs) Suppose $P_h = (\mu_h, \phi_h)$ is a linear MDP. Suppose π is any mixture of n policies, and let $\Lambda_h := n\mathbb{E}_{d^{\pi}_h}[\phi_h(s_h,a_h)\phi_h(s_h,a_h)^\top] + \lambda I$ denote the unnormalized covariance. For any $g : S \times A \to [0,R]$, policy π, and $h = 0,1,\ldots,H-2$, we have

$$\mathbb{E}[g(s_{h+1},a_{h+1})] \le \mathbb{E}\left[\|\phi_h(s_h,a_h)\|_{\Lambda_h^{-1}}\right] \sqrt{nA\,\mathbb{E}_{d^{\pi}_h}[g(s_{h+1},a_{h+1})^2] + d\lambda\|g\|_1^2}$$

Proof.

$$\mathbb{E}[g(s_{h+1},a_{h+1})] = \mathbb{E}\left[\langle \phi_h(s_h,a_h), \int_{s_{h+1}} g(s_{h+1},\pi_{h+1})\,d\mu_h(s_{h+1})\rangle\right]$$

$$\le \mathbb{E}\left[\|\phi_h(s_h,a_h)\|_{\Lambda_h^{-1}} \left\|\int_{s_{h+1}} g(s_{h+1},\pi_{h+1})\,d\mu_h(s_{h+1})\right\|_{\Lambda_h}\right],$$

where

$$\left\|\int_{s_{h+1}} g(s_{h+1},\pi_{h+1})\,d\mu_h(s_{h+1})\right\|_{\Lambda_h}$$

$$= \sqrt{n\mathbb{E}_{s_{h+1}\sim P_h(s_h,a_h)}[g(s_{h+1},\pi_{h+1})]^2 + \left\|\int_{s_{h+1}} g(s_{h+1},\pi_{h+1})\,d\mu_h(s_{h+1})\right\|_2^2}$$

$$\le \sqrt{nA\,\mathbb{E}_{d^{\pi}_h}[g(s_{h+1},a_{h+1})^2] + d\lambda\|g\|_1^2}.$$

□

Under reachability, we can show that small (squared) bonuses and spectral coverage, in the sense of having lower bounded eigenvalues, are somewhat equivalent.

Lemma E.3 Let Σ be a symmetric positive definite matrix and define the bonus $b_h(s,a) = \|\phi_h(s,a)\|_{\Sigma^{-1}}$. Then we have

1. For any policy π, $\mathbb{E}_{d^{\pi}_h}[b_h^2(s,a)] \le \frac{1}{\lambda_{\min}(\Sigma)}$. That is, coverage implies small squared bonus.
2. Suppose reachability under ρ (Assumption 3.3), then we have the converse: there exists for any policy π, $\mathbb{E}_{d^{\rho}_h}[b_h^2(s,a)] \ge \frac{\lambda}{\lambda_{\min}(\Sigma)}$. That is, small squared bonus implies coverage.

Proof. The first claim follows directly from Cauchy-Schwartz. Indeed, for any policy π, we have

$$\mathbb{E}_{d^{\pi}_h}[b_h^2(s,a)] \le \mathbb{E}_{d^{\pi}_h}[\|\phi_h(s,a)\|_2^2\|\Sigma^{-1}\|_2] \le \frac{1}{\lambda_{\min}(\Sigma)}.$$

For the second claim, Assumption 3.3 implies that there exist a policy π such that for all vectors $v \in \mathbb{R}^d$ with $\|v\|_2 = 1$, we have $\mathbb{E}_{d^{\pi}_h}[(\phi_h(s,a)^\top v)^2] \ge \cdot$. Now decompose $\Sigma = \sum_{i=1}^d \lambda_i v_i v_i^\top$, where λ_i, v_i are eigenvalue/vector pairs with $\|v_i\|_2 = 1$ and $\lambda_1 \le \lambda_2 \le \cdots \le \lambda_d$. Then substituting this into the definition of the bonus, we have

$$\mathbb{E}_{d^{\pi}_h}[b_h^2(s,a)] = \sum_{i=1}^d \frac{1}{\lambda_i}\mathbb{E}_{d^{\pi}_h}[(\phi_h(s,a)^T v_i)^2]$$

$$\frac{1}{d}\mathbb{E}_{d_h^e}\left(\widehat{\phi}_h(s,a)^\top v_d\right)^2$$

$$\frac{}{\min(\Sigma)}:$$

□

We now prove our main lemma for reward-free exploration, Lemma 4.2.

Lemma 4.2 (Reward-free Exploration) Fix any source task $k \in [K-1]$. Suppose Assumptions 3.2 and 3.3. Then, for any $\delta \in (0,1)$, w.p. $1-\delta$, RewardFree (Algorithm 2) with $N_{\text{LSVI-UCB}} = \widetilde{O}\left(A^3 d^6 H^8 \varepsilon^{-2}\right)$ and $N_{\text{RewardFree}} = \widetilde{O}\left(A^3 d^4 H^6 \log\frac{1}{\delta}\right) \le N_{\text{LSVI-UCB}}^2$ returns a $\varepsilon_{\min}$-exploratory policy π_k where $\varepsilon_{\min} = \widetilde{O}\left(A^3 d^5 H^7 \varepsilon^{-2}\right)$. The sample complexity here is $N_{\text{RewardFree}}$ episodes in the source task.

Proof of Lemma 4.2. In this proof, let

$$\Sigma_h^? = N_{\text{LSVI-UCB}}\mathbb{E}_{s_h^+,a_h^+}\left[\widehat{\phi}_h^?(s_h,a_h)\,\widehat{\phi}_h^?(s_h,a_h)^\top\right] + \lambda I,$$
$$\Sigma_h = N_{\text{LSVI-UCB}}\mathbb{E}_{s_h^+,a_h^+}\left[\widehat{\phi}_h(s_h,a_h)\,\widehat{\phi}_h(s_h,a_h)^\top\right] + \lambda I,$$

where $\lambda = dH\log(N_{\text{LSVI-UCB}}/\delta) \ge 1$. This setting of λ satisfies the precondition for the Concentration of Inverse Covariances Zanette et al. (2021, Lemma 39), which implies w.p. at least $1-\delta$ that

$$\Sigma_h^{-1} \le 2\left(\sum_{e=1}^{N_{\text{LSVI-UCB}}}\widehat{\phi}_h(s_h^e,a_h^e)\,\widehat{\phi}_h(s_h^e,a_h^e)^\top + \lambda I\right)^{-1} \le 2\,\Sigma_{h;N_{\text{LSVI-UCB}}}^{-1};$$

where we've also used the fact that $\lambda \ge 1$, so $(A+\lambda I)^{-1} \le (A+I)^{-1}$.

Under this event, for any π, we have,

$$\sum_{h=1}^H \mathbb{E}_{\pi;p}\left[\widehat{\phi}_h(s_h,a_h)\right]_{\Sigma_h^{-1}} \le \sum_{h=1}^H \mathbb{E}_{\pi;p}\left[\widehat{\phi}_h(s_h,a_h)\right]_{\Sigma_{h;N_{\text{LSVI-UCB}}}^{-1}}. \tag{6}$$

Now let $h = 0,1,\ldots,H-2$ be arbitrary. By Assumption 3.3 (there exists some policy with coverage) such that,

$$\begin{aligned}
&\frac{}{\min(\Sigma_{h+1}^?)} \\
&\mathbb{E}_{\pi_{h+1}^?}\left[\widehat{\phi}_{h+1}(s_{h+1},a_{h+1})^\top(\Sigma_{h+1}^?)^{-1}\right] && \text{(by Lemma E.3)}\\
&\mathbb{E}_{\pi_{h+1}^?}\left[\widehat{\phi}_{h+1}(s_{h+1},a_{h+1})^\top(\Sigma_{h+1}^?)^{-1}\right] && \text{(by } \Sigma \ge \Sigma^? - I\text{)}\\
&\mathbb{E}_{\pi;p}\left[\widehat{\phi}_h(s_h,a_h)\right]_{\Sigma_h^{-1}}\sqrt{A(2d+\varepsilon_{TV}N_{\text{LSVI-UCB}})} + \varepsilon_{TV} && \text{(by Corollary E.1)}\\
&\mathbb{E}_{\pi;p}\left[\widehat{\phi}_h(s,a)\right]_{\Sigma_h^{-1}}\sqrt{A(2d+1)} + 1 = N_{\text{LSVI-UCB}} && \text{(by } \varepsilon_{TV} = 1/N_{\text{LSVI-UCB}}\text{)}\\
&\le A^{1.5}d^2H^3\sqrt{\log(dHN_{\text{LSVI-UCB}}/\delta)/N_{\text{LSVI-UCB}}} + 1 = N_{\text{LSVI-UCB}} && \text{(by (6) and Lemma E.1)}\\
&\le A^{1.5}d^2H^3\sqrt{\log(dHN_{\text{LSVI-UCB}}/\delta)/N_{\text{LSVI-UCB}}}.
\end{aligned}$$

Recall that $\varepsilon = dH\log(N_{\text{LSVI-UCB}}/\delta)$, we have,

$$\frac{\min(\Sigma_{h+1}^?)}{N_{\text{LSVI-UCB}}} = \min\left(\mathbb{E}_{s_h^+,a_h^+}\left[\widehat{\phi}_{h+1}^?(s,a)\,\widehat{\phi}_{h+1}^?(s,a)^\top\right]\right)$$

$$\frac{1}{N_{\text{LSVI-UCB}}}\left(\frac{C}{A^{1:5}d^2H^3}\sqrt{\log(dHN_{\text{LSVI-UCB}}=\ )/N_{\text{LSVI-UCB}}}\right)dH\log(N_{\text{LSVI-UCB}}=\ )\Bigg)$$

$$\& \frac{C}{A^{1:5}d^2H^3}\sqrt{\frac{dH}{N_{\text{LSVI-UCB}}}};$$

where we've omitted the log terms for simplicity in the &. Now we optimize $N_{\text{LSVI-UCB}}$ to maximize this bound. For $a; b > 0$, to maximize a function of the form $f(x) = \sqrt{\frac{a}{x}} - \frac{b}{x}$, it's best to set $x^?$ such that $\sqrt{x^?} = \frac{2b}{a}$, resulting in value $f(x^?) = \frac{a^2}{4b}$. Setting,

$$x = N_{\text{LSVI-UCB}};$$
$$a = \frac{C}{A^{1:5}d^2H^3};$$
$$b = dH:$$

Hence, we need to set

$$N_{\text{LSVI-UCB}} = e\ b^2/a^2\ = e\ \frac{A^3d^6H^8}{2}\ ;$$

which results in a $_{\min}$ lower bound of

$$_{\min}\ E_{\ _h^{+1}}\ \ ^?_{h+1}(s_{h+1}; a_{h+1})\ ^?_{h+1}(s_{h+1}; a_{h+1})^T\ \ = e\ a^2/b = e\ \frac{^2}{A^3d^5H^7}\ :$$

Finally, we used the fact that $"^2_{TV} = 1/N_{\text{LSVI-UCB}}$, which is set by the choice of $N_{\text{REWARDFREE}}$ in the lemma statement to satisfy (5).

The above proves coverage of $^{+1}_h$ for $h = 0; 1; :::; H\ \ 2$. Finally to argue for $^{+1}_1$, which is simply taking a random action at time h, we can simply invoke Assumption 3.3 for $h = 0$ to get a policy $$ that

$$E_{\ _1^{+1}}\ \ ^?_0(s_0; a_0)\ \ ^?_0(s_0; a_0)^> \quad \frac{1}{A}E_e\ \ ^?_0(s_0; a_0)\ \ ^?_0(s_0; a_0)^> \quad \overline{A}:$$

□

Corollary E.1 Let $\ ; \ ^?_h; b_h$ be de ned as in the proof of Lemma 4.2. For any $h = 0; 1; :::; H\ \ 2$ and any policy $$, we have

$$E\left[\ ^?_{h+1}(s_{h+1}; a_{h+1})\ (\ ^?_{h+1})^{\ 1}\right]\ E_{\ ; p}\left[b_h(s_h; a_h)\ _{b_h^{\ 1}}\right]\ p\ \overline{jAj\ (2d + "_{TV}N_{\text{LSVI-UCB}})} + "_{TV}:$$

Intuitively, this means that coverage in the learned features implies coverage in the true features.

Proof. For shorthand, let $N = N_{\text{LSVI-UCB}}$. Apply Lemma E.2 (one-step back) to the learned model b and the function $(s; a)\ 7!\ k\ ^?_{h+1}(s; a)k_{(\ ^?_{h+1})^{\ 1}}$, which is bounded by $^{1=2}\ \ 1$. We have,

$$E_{\ ; p}\left[\ ^?_{h+1}(s_{h+1}; a_{h+1})\ (\ ^?_{h+1})^{\ 1}\right]$$
$$E_{\ ; p}\left[b_h(s_h; a_h)\ _{b_h^{\ 1}}\right]\sqrt{\frac{}{NjAjE_{\ _h^{+1}; p}k\ ^?_{h+1}(s_{h+1}; a_{h+1})k^2_{(\ ^?_{h+1})^{\ 1}} + d}}$$
$$E_{\ ; p}\left[b_h(s_h; a_h)\ _{b_h^{\ 1}}\right]\sqrt{\frac{}{NjAjE_{\ _h^{+1}}k\ ^?_{h+1}(s_{h+1}; a_{h+1})k^2_{(\ ^?_{h+1})^{\ 1}} + NjAj"_{TV} + d}}$$
$$E_{\ ; p}\left[b_h(s_h; a_h)\ _{b_h^{\ 1}}\right]\ p\ \overline{djAj\ + NjAj"_{TV} + d};$$

where we used the fact that

$$E_{\ _h^{+1}}k\ ^?_{h+1}(s_{h+1}; a_{h+1})k^2_{(\ ^?_{h+1})^{\ 1}}$$

$$= \mathrm{Tr}\left(\mathbb{E}_{s_h \sim \nu_h^{k+1}} \left[\phi_{h+1}(s_{h+1}, a_{h+1}) \phi_{h+1}(s_{h+1}, a_{h+1})^\top \right] - N \mathbb{E}_{s_h \sim \nu_h^{k+1}} \left[\phi_h(s_h, a_h) \phi_h(s_h, a_h)^\top \right] + I \right)^{-1}$$

$$= \frac{1}{N} \mathrm{Tr}\left(I - M \right) \leq \frac{d}{N},$$

where M is a positive definite matrix. Thus, doing an initial change from d_{m+1} to $d_{\mathrm{epoch},h+1}$ concludes the proof. □

F Representation Transfer

First, we prove Lemma 4.1, restated below.

Lemma 4.1 Suppose Assumption 3.4 and that for all source tasks $k \in [K-1]$ we have a $\nu_{\min}$-exploratory policy π_k. Then, for any $\delta \in (0,1)$, learning features $\hat{b}$ using the cross-sampling procedure of Algorithm 1 satisfies (2) w.p. $1-\delta$. Furthermore, for any $h = 0, 1, \ldots, H-1$, there exists $\theta_h : \mathcal{S} \to \mathbb{R}^d$ such that for any function $g : \mathcal{S} \to [0,1]$, $\int \|\hat{b}_h(s) - \theta_h(s)\|_2 g(s) d\mu(s) \leq \sqrt{d}\,\bar{\varepsilon}$ and

$$\sup_{\pi} \mathbb{E}_{\pi, P_K^?} \left\| \hat{b}_h(s_h, a_h)^\top \theta_h(\cdot) - \phi_h^?(s_h, a_h)^\top \psi_{K,h}^?(\cdot) \right\|_{TV} \leq \varepsilon_{TV} := \sqrt{|\mathcal{A}|}\, \frac{3 \max_k K_k}{\nu_{\min}} \bar{\varepsilon}.$$

Proof of Lemma 4.1 Fix an arbitrary g. Denote $\theta_h(s') = \sum_{k=0}^{K-1} \alpha_{k,h}(s') b_{k,h}(s')$. First, note that

$$\begin{aligned}
&\max_{g:\mathcal{S}\to[0,1]} \left\| \int \theta_h(s) g(s) d\mu(s) \right\|_2 \\
&\leq \max_{g:\mathcal{S}\to[0,1]} \sum_{k=0}^{K-1} \left\| \int b_{k,h}(s)\, \alpha_{k,h}(s) g(s) d\mu(s) \right\|_2 \\
&\leq \sum_{k=0}^{K-1} \max_s \alpha_{k,h}(s) \sqrt{d}\,\bar{\varepsilon} \qquad \left(\text{Since } \left\| \int b_{k,h}(s) g(s) d\mu(s) \right\|_2 \leq \sqrt{d}\,\bar{\varepsilon} \text{ by 3.2}\right) \\
&= \sqrt{d}\,\bar{\varepsilon}
\end{aligned}$$

For any $h = 0, 1, \ldots, H-1$, we have

$$\begin{aligned}
&\mathbb{E}_{\pi, P_K^?} \left\| \hat{b}_h(s_h, a_h)^\top \theta_h(\cdot) - \phi_h^?(s_h, a_h)^\top \psi_{K,h}^?(\cdot) \right\|_{TV} \\
&= \mathbb{E}_{\pi, P_K^?} \left[\sum_{s_{h+1}} \left| \sum_{k=1}^{K-1} \alpha_{k,h}(s_{h+1}) \hat{b}_h(s_h, a_h)^\top \hat{b}_{k,h}(s_{h+1}) - \phi_h^?(s_h, a_h)^\top \psi_{k,h}^?(s_{h+1}) \right| \right] \\
&\leq \mathbb{E}_{\pi, P_K^?} \left[\sum_{s_{h+1}} \sum_{k=1}^{K-1} |\alpha_{k,h}(s_{h+1})| \left| \hat{b}_h(s_h, a_h)^\top \hat{b}_{k,h}(s_{h+1}) - \phi_h^?(s_h, a_h)^\top \psi_{k,h}^?(s_{h+1}) \right| \right] \\
&\leq \max_k K_k \sum_{k=1}^{K-1} \mathbb{E}_{\pi, P_K^?} \left\| \hat{b}_h(s_h, a_h)^\top \hat{b}_{k,h}(\cdot) - \phi_h^?(s_h, a_h)^\top \psi_{k,h}^?(\cdot) \right\|_{TV}.
\end{aligned}$$

First consider the case when $h = 0$. At $h = 0$, the distribution under $P_K^?$ is the same as $\nu_{k,h}$, and so, we directly get that the above quantity is at most $\max_k K_k \bar{\varepsilon}^{1/2} \leq \varepsilon_{TV}$, which proves the $h = 0$ case.

Now consider any $h = 1, 2, \ldots, H-1$. To simplify notation, let us denote

$$\begin{aligned}
\mathrm{err}_{k,h}(s_h, a_h) &= \left\| \hat{b}_h(s_h, a_h)^\top \hat{b}_{k,h}(\cdot) - \phi_h^?(s_h, a_h)^\top \psi_{k,h}^?(\cdot) \right\|_{TV}, \\
w_{k,h} &= \int_{s_h} d_{K,h-1}^?(s_h) \mathbb{E}_{a_h \sim \pi_k(s_h)} \mathrm{err}_{k,h}(s_h, a_h),
\end{aligned}$$

$$_{k;h} = E_{\ k;P_k^?} \quad {}^?_h(s_h;a_h) \quad {}^?_h(s_h;a_h)^> \ :$$

Note that $_{\min}(\ _{k;h}) \quad _{\min}$ by assumption. Now continuing from where we left off, we take a one-step back as follows,

$$\begin{aligned}
&\max \sum_{k=1}^{K\ 1} E_{\ ;P_K^?}\ \mathrm{err}_{k;h}(s_h;a_h) \\
= &\max \sum_{k=1}^{K\ 1} E_{\ ;P_K^?} \quad {}^?_{h\ 1}(s_{h\ 1};a_{h\ 1});w_{k;h} \\
&\max \sum_{k=1}^{K\ 1} E_{\ ;P_K^?} \quad {}^?_{h\ 1}(s_{h\ 1};a_{h\ 1}) \quad {}^1_{k;h\ 1} \quad \|w_{k;h}\|_{\ k;h\ 1}
\end{aligned}$$

By $_{\min}$ guarantee of $_{k;h}$, and Jensen's inequality to push the square inside,

$$\begin{aligned}
&\frac{\max}{\min} \sum_{k=1}^{K\ 1} \sqrt{E_{s_{h\ 1};a_{h\ 1}\ \ k;P_k^?} E_{s_h\ \ P^?_{K;h\ 1}(s_{h\ 1};a_{h\ 1});a_h\ \ h}(s_h)} \mathrm{err}_{k;h}(s_h;a_h)^2 \\
&A^{1=2}\frac{\max}{\min} \sum_{k=1}^{K\ 1} \sqrt{E_{s_{h\ 1};a_{h\ 1}\ \ k;P_k^?} E_{s_h\ \ P^?_{K;h\ 1}(s_{h\ 1};a_{h\ 1});a_h\ U\ (A)}} \mathrm{err}_{k;h}(s_h;a_h)^2
\end{aligned}$$

By Assumption 3.4, the expectation over $P^?_{K;h\ 1}$ is a linear combination of expectations over $P^?_{j;h\ 1}$,

$$\begin{aligned}
&A^{1=2}\frac{3=2\ \max}{\min} \sum_{k=1}^{K\ 1} \sqrt{\sum_{j=1}^{K\ 1} E_{s_{h\ 1};a_{h\ 1}\ \ k;P_k^?} E_{s_h\ \ P^?_{j;h\ 1}(s_{h\ 1};a_{h\ 1});a_h\ U\ (A)}} \mathrm{err}_{k;h}(s_h;a_h)^2 \\
&A^{1=2}\frac{3=2\ \max}{\min} K^{1=2} \sum_{k=1}^{K\ 1} \sqrt{\sum_{j=1}^{K\ 1} E_{s_{h\ 1};a_{h\ 1}\ \ k;P_k^?} E_{s_h\ \ P^?_{j;h\ 1}(s_{h\ 1};a_{h\ 1});a_h\ U\ (A)}} \mathrm{err}_{k;h}(s_h;a_h)^2 \\
&A^{1=2}\frac{3=2\ \max}{\min} K^{1=2}\ {}^{1=2}_{n};
\end{aligned}$$

where we used the MLE guarantee (2) in the last step. □

Algorithm 5 Transfer learning with online access

PRE-TRAINING PHASE

Input: num. LSVI-UCB episodes $N_{\text{LSVI-UCB}}$, num. model-learning episodes $N_{\text{REWARDFREE}}$, size of cross-sampled datasets n, failure probability .

1: for source task $k = 1;:::;K \quad 1$ do
2: Find policy cover $_k = \text{REWARDFREE}(P_k^?; N_{\text{LSVI-UCB}}; N_{\text{REWARDFREE}}; \ )$. (Algorithm 2)
3: for source task $k = 1;:::;K \quad 1$ do
4: For each $h = 0;1;:::;H \quad 1$, sample D_k as n i.i.d. $(s_h;a_h;s_{h+1})$ tuples from $_k$.
5: For each $h = 0;1;:::;H \quad 1$, learn $b_h = \text{Multi-task REPLEARN}(fD_{k;h}g_{k2[K \quad 1]})$. (Algorithm 3)

DEPLOYMENT PHASE

Additional Input: number of deployment episodes T

1: Set $= H\sqrt{d} + dH\sqrt{\log(dHT=\ )}$.
2: Run LSVI-UCB $fb_h g_{h=0}^{H\ 1}; r = r_K; T;$ in the target task $P_K^?$ (Algorithm 6).

Next we state an analogous lemma for when we don't need generative access to the source task, but instead assume Assumption 5.1, and Assumption 5.2.

Lemma F.1 Suppose Assumption 5.1, and Assumption 5.2. Now take the setup of Lemma 4.1 with the only difference being that $\hat{b}$ is learned as in Algorithm 5. Then, the same guarantee of Lemma 4.1 holds with a slightly different right hand side for the bound on the TV-error,

$$\sup_{;P_K^?} \left\| \hat{b}_h(s_h,a_h)^\top \hat{b}_h(\cdot) \quad {}^?_h(s_h,a_h)^\top \; {}^?_{K;h}(\cdot) \right\|_{TV} \quad \frac{\max K^{1=2} \; n^{1=2}}{(\quad_{raw} \quad_{min})^{1=2}}.$$

Proof of Lemma F.1 Fix an arbitrary . Denote $\quad_h(s^0) = \sum_{k=0}^{K} {}^{1} \quad_{k;h}(s^0)\hat{}_{k;h}(s^0)$. Then, some algebra with importance sampling gives us the bound,

$$\begin{aligned}
&\mathbb{E}_{;P_K^?} \left\| \hat{b}_h(s_h,a_h)^\top \quad_h(\cdot) \quad {}^?_h(s_h,a_h)^\top \; {}^?_{K;h}(\cdot) \right\|_{TV} \\
&\mathbb{E}_{;P_K^?} \Big[\sum_{s_{h+1}} \sum_{k=1}^{K} {}^{1} \quad_{k;h}(s_{h+1}) \left\| \hat{b}_h(s_h,a_h)^\top b_{k;h}(s_{h+1}) \quad {}^?_h(s_h,a_h)^\top \; {}^?_{k;h}(s_{h+1}) \right\| \Big] \\
&\max \sum_{k=1}^{K} {}^{1} \mathbb{E}_{;P_K^?} \left\| \hat{b}_h(s_h,a_h)^\top b_{k;h}(\cdot) \quad {}^?_h(s_h,a_h)^\top \; {}^?_{k;h}(\cdot) \right\|_{TV} \\
&\max K^{1=2} \sqrt{\sum_{k=1}^{K} {}^{1} \mathbb{E}_{;P_K^?} \left\| \hat{b}_h(s_h,a_h)^\top b_{k;h}(\cdot) \quad {}^?_h(s_h,a_h)^\top \; {}^?_{k;h}(\cdot) \right\|_{TV}^2}
\end{aligned}$$

By Assumption 5.1, Assumption 5.2, for any $s;a$, we have $\frac{d_{K;h}(s;a)}{d^k_{k;h}(s;a)} \quad \frac{1}{\quad_{raw} \quad_{min} \mathbb{E}_{k;P_k^?}[\, {}^?_h(s_h;a_h) \; {}^?_h(s_h;a_h)^\top \,]} \quad \frac{1}{\quad_{raw} \quad_{min}}$, where we used the coverage-under-k assumption in the last inequality. In other words, for each $= 1;2;:::;K \quad 1$, we have $\frac{dd_{K;h}}{dd^k_{k;h}} \quad 1 \quad \frac{1}{\quad_{raw} \quad_{min}}$, hence we can importance sample,

$$\frac{\max K^{1=2}}{(\quad_{raw} \quad_{min})^{1=2}} \sqrt{\sum_{k=1}^{K} {}^{1} \mathbb{E}_{k;P_k^?} \left\| \hat{b}_h(s_h,a_h)^\top b_{k;h}(\cdot) \quad {}^?_h(s_h,a_h)^\top \; {}^?_{k;h}(\cdot) \right\|_{TV}^2} \quad \frac{\max K^{1=2} \; n^{1=2}}{(\quad_{raw} \quad_{min})^{1=2}}.$$

□

G Proofs for Deployment Phase

G.1 Online RL Lemmas

Lemma G.1 (Self-normalized Martingale) Consider $\ltrations f F_i g_{i=1;2;:::}$, so that $\mathbb{E}["_i \mid F_{i \quad 1}] = 0$ and $f "_i \mid F_{i \quad 1} g_{i=1;2;:::}$ are sub-Gaussian with parameter ^{2}. Let $f X_i g_{i=1;2;:::}$ be random variables in a hilbert space H. Suppose a linear operator $_0 : H \to H$ is positive de nite. For any t, de ne $_t = \quad_0 + \sum_{i=1}^{t} X_i X_i^T$. Then w.p. at least $1 \quad$, we have,

$$\forall t \quad 1 : \left\| \sum_{i=1}^{t} X_i "_i \right\|^2_{\quad_t \; 1} \quad {}^2 \log \frac{\det(\quad_t)^{\frac{1}{2}} \det(\quad_0)^{\frac{1}{2}}}{\quad}.$$

Proof. Lemma A.8 of (Agarwal et al., 2019). □

Lemma G.2 Let $_t = I + \sum_{i=1}^{t} x_i x_i^T$ for $x_i \in R^d$ and > 0. Then $\sum_{i=1}^{t} x_i^T (\quad_t)^{\quad 1} x_i \quad d$.

Proof. Lemma D.1 of (Jin et al., 2020b). □

G.2 Proof of main results

Let $(x)_y$ refer to the clamping operator, i.e. $(x)_y = \min\{x, y\}$. Let M_V be the maximum possible value in the MDP with the given reward function.

Algorithm 6 LSVI-UCB

1: Input: Features $\{b_h\}_{h=0;1;\ldots;H-1}$, reward $\{r_h\}_{h=0;1;\ldots;H-1}$, number of episodes N, bonus scaling parameter, UNIFORMACTIONS = FALSE.
2: **for** episode $e = 1; 2; \ldots; N$ **do**
3: Initialize $\hat{V}_{H;e}(s) = 0; \forall s$
4: **for** step $h = H-1; H-2; \ldots; 0$ **do**
5: Learn best predictor for $\hat{V}^e_{h+1}$,

$$\Lambda_{h;e} = \sum_{k=1}^{n} b_h(s^k_h; a^k_h)\, b_h(s^k_h; a^k_h)^\top + I;$$

$$w_{h;e} = \Lambda^{-1}_{h;e} \sum_{k=1}^{n} b_h(s^k_h; a^k_h)\, \hat{V}_{h+1;e}(s^k_{h+1}).$$

6: Set bonus and value functions,

$$b_{h;e}(s; a) = \beta \left\| b_h(s; a) \right\|_{\Lambda^{-1}_{h;e}};$$

$$\hat{Q}_{h;e}(s; a) = \left[w^\top_{h;e} b_h(s; a) + r_h(s; a) + b_{h;e}(s; a) \right]_{M_V};$$

$$\hat{V}_{h;e}(s) = \max_a \hat{Q}_{h;e}(s; a).$$

7: Set $\pi^e_h(s) = \operatorname{argmax}_a \hat{Q}_{h;e}(s; a)$.
8: Execute π^e to collect a trajectory $\{(s^e_h; a^e_h)\}_{h=0}^{H-1}$.
9: If UNIFORMACTIONS = TRUE, discard a^e_h and draw freshly sampled uniform actions independently for all h, i.e. $a^e_h \sim \text{Unif}(A)$.
10: Return: uniform mixture $\pi = \text{Uniform}(\{\pi^e\}_{e=1}^N)$.

Previously, Jin et al. (2020b) analyzed LSVI-UCB under point-wise model-misspecification. Here, we show that similar guarantees hold under a more general policy-distribution model-misspecification ε_{ms}, captured by Assumption G.1.

Assumption G.1 Suppose for every $h = 0; 1; \ldots; H-1$, there exists θ_h such that for any policy π,

$$\mathbb{E}_h \left[\left\| e_h(\cdot)^\top b_h(s_h; a_h) - P^\pi_h(\cdot \mid s_h; a_h) \right\|_{TV} \right] \le \varepsilon_{ms}.$$

We further assume that $\sup_{s;a;h} \|e_h(\cdot)^\top b(s; a)\|_{TV} \le M$ and $\|f^\top e_h\|_2 \le M\sqrt{d}\|f\|_1 \; \forall f : S \to \mathbb{R}$, for some positive constant M.

In other words, we only need the model to be accurate on average under the occupancy distributions realizable by policies. We also make a slight generalization on the regularization constant, which is set to 1 in the original linear MDP definition (Jin et al., 2020b). Later, we will later instantiate the above assumption with our transferred $e_h(s^0) = \sum_{k=1}^{K} \omega_{k;h}(s^0)\, b_{k;h}(s^0)$, then for any $s; a$, we have

$$\left\| e_h\, b_h(s; a) \right\|_{TV} = \sum_{s^0} \left| \sum_{k=1}^{K} \omega_{k;h}(s^0)\, b_{k;h}(s^0)^\top b_h(s; a) \right|$$

$$\le \sum_{s^0} \sum_{k=1}^{K} \left| \omega_{k;h}(s^0) \right| \left\| b_{k;h}(s^0)^\top b_h(s; a) \right\|$$

$$\leq \sum_{k=1}^{K} \max_{s'} |\alpha_{k;h}(s')| \qquad \text{(by } \|b_{k;h}\|_{TV} \leq 1\text{)}$$
$$\leq K.$$

Also,

$$\|f^T e_h\|_2 = \left\| \sum_{s'} \sum_{k=1}^{K} \alpha_{k;h}(s') b_{k;h}(s') f(s') \right\|_2^2$$
$$= \sum_{k=1}^{K} \max_{s'} |\alpha_{k;h}(s')| \left| \sum_{s'} b_{k;h}(s') f(s') \right|^2 \qquad \text{(by } \|f^T b_{k;h}\|_2^2 \leq \|f\|_1^2\text{)}$$
$$\leq K \|f\|_1^2.$$

So we will set $M = K$.

Note that we only need the existence of e_h here, and $e_h(\cdot)^T b_h(s,a)$ need not be a valid probability kernel. In fact, it may even be negative valued.

In this section, we make a model-based analysis of LSVI. Similar approaches have been used in prior works, e.g. Lykouris et al. (2021); Agarwal et al. (2019); Zhang et al. (2022). For simplicity, we suppose that S is finite, but may be exponentially large, as we suffer no dependence on S. The proof can be easily extended to in finite state spaces by replacing inner products with integrals.

Consider the following quantity,

$$b_{h;e} = \left(\sum_{k=1}^{K} b_h(s_h^k, a_h^k) b_h(s_h^k, a_h^k)^T \right)^{-1} \sum_{k=1}^{K} b_h(s_h^k, a_h^k) \left(V_{h+1;e}(s_{h+1}^k) + \|\alpha_{h;e}\|_2 \right);$$

where $\alpha_{h;e} = \operatorname{argmin}_{\alpha \in \mathbb{R}^d} \sum_{k=1}^{K} \|\alpha^T b_h(s_h^k, a_h^k) - V_{h+1;e}(s_{h+1}^k)\|_2^2 + \|\alpha\|_2^2$;

where $\phi(s)$ is a one-hot encoding of the state s. In words, this is the best choice for linearly (in $b_h(s,a)$) predicting $E_{s' \sim P_h^?(s,a)}[\phi(s')] = P_h^?(s' \mid s,a)$. We highlight that this is just a quantity for analysis and not computed in the algorithm. Finally, denote

$$\hat{P}_{h;e} = b_{h;e} b_h;$$
$$\hat{r}_h = e_h b_h.$$

We will also sometimes use the shorthand $Pf(s,a)$ for $E_{s' \sim P(\cdot \mid s,a)}[f(s')]$.

For each $h = 0, 1, \ldots, H-1$, let $\mathcal{V}_h$ denote the class of functions

$$s \mapsto \max_a \left\{ w^T b_h(s,a) + r_h(s,a) + \|e^T b_h(s,a)\|_{M_V} \right\} \quad \|w\|_2 \leq M_V, \|e\|_1 \leq M_V, e \in [0,B], \quad I \text{ symmetric}$$

The motivation behind this construction is that V_{h+1} satisfies the key property that all of the learned value functions $\hat{V}_{h;e}$ during Algorithm 6 are captured in this class.

Lemma G.3 For any $h = 0, 1, \ldots, H-1$,

1. $\sup_s \hat{V}_{h;e}(s) \leq M_V$.
2. For any $e = 1, 2, \ldots, N$, we have $\hat{V}_{h;e} \in \mathcal{V}_h$.
3. $\forall f \in \mathcal{V}_h$, we have $\sup_s |f(s)| \leq M_V$.

Proof. Recall that

$$\hat{V}_{h;e}(s) = \max_a \left\{ \hat{w}_{h;e}^T b_h(s,a) + r_h(s,a) + \|b_{h;e}(s,a)\|_{M_V} \right\}$$

$$\text{where } w_{h;e} = \sum_{k=1}^{\infty} \prod_{h;e}^{1} b_h(s_h^k, a_h^k) V_{h+1;e}(s_{h+1}^k):$$

From the thresholding, we have

$$V_{h;e}(s) \quad M_V :$$

We can bound the norm of $w_{h;e}$ as follows,

$$\|w_{h;e}\| \quad \prod_{h;e}^{1} \sum_{k=1}^{\infty} \prod_{2}^{1} V_{h+1;e}(s_{h+1}^k) \quad N \sup_s V_{h+1;e}(s) \quad N M_V :$$

We also required B, and we regularized the covariance with, so $_{min}$ is at least 1. Hence $b_{h;e}$ satisfies all the conditions to be in V_h. □

Now we control the metric entropy of V_h in $`_1$, i.e. $d(f_1; f_2) = \sup_s |f_1(s) \quad f_2(s)|$ for $f_i \in V_h$.

Lemma G.4 Let $" > 0$ be arbitrary and let $N_"$ be the smallest $"$-net with $`_1$ of V_h. Then,

$$\log |N_"| \quad d \log(1 + 6L/") + \log(1 + 6B/") + d^2 \log(1 + 18B^2 \sqrt{d}/"^2):$$

Proof. Let $f_1; f_2 \in V_h$. Then,

$$\begin{aligned}
&|f_1(s) \quad f_2(s)| \\
&\quad \max_a (w_1 \quad w_2)^T b_h(s;a) + \quad {}_1 \|b_h(s;a)\|_{1}^{-1} \quad {}_2 \|b_h(s;a)\|_{2}^{-1} \\
&\quad \|w_1 \quad w_2\|_2 + \max_a \quad ({}_1 \quad {}_2) \|b_h(s;a)\|_{1}^{-1} + {}_2 \max_a \quad \|b_h(s;a)\|_{1}^{-1} \quad \|b_h(s;a)\|_{2}^{-1} \\
&\quad \|w_1 \quad w_2\|_2 + |{}_1 \quad {}_2| B \max_a \sqrt{\|b_h(s;a)\|_{1}^{-1} \quad \|b_h(s;a)\|_{2}^{-1}} \qquad ({}_{min}({}_1) \quad 1) \\
&\quad \|w_1 \quad w_2\|_2 + |{}_1 \quad {}_2| B \sqrt{\quad {}_1^{-1} \quad {}_2^{-1}};
\end{aligned}$$

where we used for any $a; b \quad 0$, we have $\sqrt{a} \quad \sqrt{b} = \frac{|a \quad b|}{\sqrt{a}+\sqrt{b}} \quad \sqrt{|a \quad b|}$. Now proceeding like the Lemma 8.6 in the RL Theory Monograph (Agarwal et al., 2019), we have the result. □

In this section, we'll use the following bonus scaling parameter,

$$:= O \quad \sqrt{N d}"_{ms} M_V + M_V M \quad d \sqrt{\log(dNM_V/)} : \tag{7}$$

The following high probability event (E_{model}) is a key step in our proof. Essentially, Theorem G.1 guarantees that, for all functions in V_h, the model we learn is an accurate predictor of the expectation, up to a bonus and some vanishing terms.

For all the following lemmas and theorems, suppose Assumption G.1 and the bonus scaling is as in (7). Throughout the section, $_h(\ _h)$ refers to indicator functions of the trajectory $\ _h = (s_0; s_1; :::; s_h)$. As before, the expectations $E[g(\ _h)]$ are with respect to the distribution of trajectories when is executed in the environment $P^?$.

Theorem G.1 Let $2 (0; 1)$. Then, w.p. 1 , for any time h, episode e, indicator functions $_1; : : : ; _H$, and policy , we have

$$\sup_{f \in V} \left| E \left[b_{h;e}(s_h; a_h) \quad P_h^?(s_h; a_h) \quad f \ _h(\ _h) \right] \right| \quad E \left[b_h^e(s_h; a_h) \ _h(\ _h) \right] + \|V_h\|_1 "_{ms} :$$

(E_{model})

Proof. Condition on the outcome of Lemma G.5, which implies that w.p. 1, for any $h, e, \ldots, _h$, we have

$$\sup_{f \in V_h} E\left[\left(\widehat{P}_{h,e}(s_h,a_h) - \widehat{P}^e_h(s_h,a_h)\right) f_{h}(\ _h)\right] \quad E\left[\left(\widehat{P}^e_h(s_h,a_h) - P^?_h(s_h,a_h)\right) f_{h}(\ _h)\right]:$$

Also, for any $h, e, \ldots, _h$, by Assumption G.1, we have (w.p. 1) that

$$\begin{aligned}
&\sup_{f \in V_h} E\left[\left(\widehat{P}^e_h(s_h,a_h) - P^?_h(s_h,a_h)\right) f_{h}(\ _h)\right] \quad E\left[\sup_{f \in V_h}\left(\widehat{P}^e_h(s_h,a_h) - P^?_h(s_h,a_h)\right) f_{h}(\ _h)\right] \\
&\quad E\left[\sup_{f \in V_h}\left(\widehat{P}^e_h(s_h,a_h) - P^?_h(s_h,a_h)\right) f\right] \\
&\quad kV_h k_1 \ "_{ms}:
\end{aligned}$$

Combining these two yields the result, as

$$\begin{aligned}
&\sup_{f \in V_h} E\left[\left(\widehat{P}_{h,e}(s_h,a_h) - P^?_h(s_h,a_h)\right) f_{h}(\ _h)\right] \\
&\sup_{f \in V_h} E\left[\left(\widehat{P}^e_h(s_h,a_h) - P^?_h(s_h,a_h)\right) f_{h}(\ _h)\right] + \sup_{f \in V_h} E\left[\left(\widehat{P}_{h,e}(s_h,a_h) - \widehat{P}^e_h(s_h,a_h)\right) f_{h}(\ _h)\right]:
\end{aligned}$$

□

Lemma G.5 Suppose Assumption G.1 and the bonus scaling set as in (7). For any $\in (0,1)$, w.p. at least 1 , we have for any time h, episode e, and policy ,

$$8 s_h, a_h : \sup_{f \in V_h} \left(\widehat{P}_{h,e}(s_h,a_h) - \widehat{P}^e_h(s_h,a_h)\right) f \quad b_{h,e}(s_h,a_h):$$

Proof. Consider any $h, e, \ldots$. De ne $"^k_h := (s^k_{h+1}) + P^?_h(s^k_{h+1} j s^k_h, a^k_h)$, so that $E["^k_h \mid H_{k\ 1}] = 0$, where $H_{k\ 1}$ contains the states and actions before episode k. In what follows, we slightly abuse notation, as $P(s,a)\ b^T(s,a)$ will denote the outer product, and hence $R^{S\ d}$ quantity.

$$\begin{aligned}
b_{h,e}\ _{h,e} &= \sum_{k=1}^{n\ 1} (s^k_{h+1})\ b_h(s^k_h,a^k_h)^T \\
&= \sum_{k=1}^{n\ 1} P^?_h(s^k_h,a^k_h) \quad \widehat{P}^e_h(s^k_h,a^k_h)\ b_h(s^k_h,a^k_h)^T + \sum_{k=0}^{n\ 1} \widehat{P}^e_h(s^k_h,a^k_h) \quad "^k_h\ b_h(s^k_h,a^k_h)^T \\
&= \sum_{k=1}^{n\ 1} P^?_h(s^k_h,a^k_h) \quad \widehat{P}^e_h(s^k_h,a^k_h)\ b_h(s^k_h,a^k_h)^T + e_h(\ _{h,e} \quad I) \left(\sum_{k=0}^{n\ 1} "^k_h\ b_h(s^k_h,a^k_h)^T\right):
\end{aligned}$$

Rearranging, we have

$$\begin{aligned}
b_{h,e} \quad e_h = &\left(\sum_{k=0}^{n\ 1} P^?_h(s^k_h,a^k_h) \quad \widehat{P}^e_h(s^k_h,a^k_h)\ b_h(s^k_h,a^k_h)^T\right) (\ _{h,e})^{\ 1} \\
&e_h(\ _{h,e})^{\ 1} \left(\sum_{k=0}^{n\ 1} "^k_h\ b_h(s^k_h,a^k_h)^T\right) (\ _{h,e})^{\ 1}:
\end{aligned}$$

Now let $f \in V_h$ be arbitrary. For any s_h, a_h, multiply the above with $b_h(s_h,a_h)$ and multiply with f, we have

$$\begin{aligned}
&\left(\widehat{P}_{h,e}(s_h,a_h) - \widehat{P}^e_h(s_h,a_h)\right) f \\
&= f^T (b_{h,e} \quad e_h)\ b_h(s_h,a_h)
\end{aligned}$$

$$f^T \Big(\sum_{k=1}^{N-1} \big(P_h^?(s_h^k, a_h^k) - P_h(s_h^k, a_h^k) \big) b_h(s_h^k, a_h^k)^T \Big) \Lambda_{h,e}^{-1} b_h(s_h, a_h)$$
$$\underbrace{\qquad\qquad\qquad\qquad\qquad\qquad}_{\text{Term(a)}}$$
$$+ \underbrace{f^T e_h \Lambda_{h,e}^{-1} b_h(s_h, a_h)}_{\text{Term(b)}}$$
$$+ \underbrace{f^T \Big(\sum_{k=1}^{N-1} \eta_h^k b_h(s_h^k, a_h^k)^T \Big) \Lambda_{h,e}^{-1} b_h(s_h, a_h)}_{\text{Term(c)}} :$$

We can deterministically bound Term (b) as follows,

$$\begin{aligned}
&\sup_{f \in \mathcal{V}_h} f^T e_h \Lambda_{h,e}^{-1} b_h(s_h, a_h) \\
&= \sup_{f \in \mathcal{V}_h} (\Lambda_{h,e}^{-1/2} f^T e_h)^T \Lambda_{h,e}^{-1/2} b_h(s_h, a_h) \\
&\le \sup_{f \in \mathcal{V}_h} \|\Lambda_{h,e}^{-1/2}\|_2 \|f^T e_h\|_2 \|b_{h,e}(s_h, a_h)\| \\
&\le \|\mathcal{V}_h\|_1 M \sqrt{d} \|b_{h,e}(s_h, a_h)\|. \qquad \text{(by Assumption G.1)}
\end{aligned}$$

This term will be lower order compared to the other two.

We now derive the bound for Term (c) for any fixed $f \in \mathcal{V}_h$. Observe that

$$f^T \Big(\sum_{k=1}^{N-1} \eta_h^k b_h(s_h^k, a_h^k)^T \Big) \Lambda_{h,e}^{-1} b_h(s_h, a_h) = \Big(\Lambda_{h,e}^{-1/2} \sum_{k=1}^{N-1} b_h(s_h^k, a_h^k)(f^T \eta_h^k) \Big)^T \Lambda_{h,e}^{-1/2} b_h(s_h, a_h)$$
$$\le \Big\| \sum_{k=1}^{N-1} b_h(s_h^k, a_h^k)(f^T \eta_h^k) \Big\|_{\Lambda_{h,e}^{-1}} \|b_{h,e}(s_h, a_h)\|.$$

Now we argue w.p. $1-\delta$, for any e, h we have

$$\Big\| \sum_{k=1}^{N-1} b_h(s_h^k, a_h^k)(f^T \eta_h^k) \Big\|_{\Lambda_{h,e}^{-1}} \le 2\|\mathcal{V}_h\|_1 \sqrt{2\log(1/\delta) + d\log(N+1)},$$

which implies the claim about $\|b_{h,e}(s_h, a_h)\|$. Indeed, we can apply Lemma G.1. Checking the preconditions, $E_{P_h^?(s_h,a_h)}[f^T \eta_h^k \mid H_{k-1}] = 0$, $|f^T \eta_h^k| \le \|f\|_1 \|\eta_h^k\|_1 \le 2\|\mathcal{V}_h\|_1$, $\det(\Lambda_0) = \det(I) = 1$, and $\det(\Lambda_t) = \det(\Lambda_{h,e}) \le (e+1)^d$ since the largest eigenvalue is ≤ 1. So, w.p. at least $1-\delta$, for all e, we have the above inequality.

Thus, for any fixed $f \in \mathcal{V}_h$, w.p. $1-\delta$, for all e, h we have,

$$\begin{aligned}
&\|b_{h,e}(s_h, a_h)\| \le \|P_h^?(s_h, a_h) - P_h(s_h, a_h)\| \cdot \|f\| \cdot \\
&\quad \text{Term(a)} + \text{Term(b)} + \text{Term(c)} \\
&\le 4\|\mathcal{V}_h\|_1 (1 + M\sqrt{d}) \sqrt{\log(1/\delta) + d\log(N)} + \sqrt{dN}\|\mathcal{V}_h\|_1 \varepsilon_{ms} \|b_{h,e}(s_h, a_h)\| \\
&\quad + \|\mathcal{V}_h\|_1 M \sqrt{d} \|b_{h,e}(s_h, a_h)\| \\
&\quad + 4\|\mathcal{V}_h\|_1 \sqrt{\log(1/\delta) + d\log(N)} \|b_{h,e}(s_h, a_h)\| \\
&\le \sqrt{dN}\|\mathcal{V}_h\|_1 \varepsilon_{ms} + \|\mathcal{V}_h\|_1 M \sqrt{\log(1/\delta) + d\log(N)} \|b_{h,e}(s_h, a_h)\|.
\end{aligned}$$

Now we apply a covering argument. Namely, union bound the above argument to every element in an ε_{net}-net of $\mathcal{V}_h$. For any $f \in \mathcal{V}_h$, let f^e be its neighbor in the net such that $\|f - f^e\|_1 \le \varepsilon_{net}$, so we have

$$\widehat{P}_{h,e}(s_h,a_h) - P^e_h(s_h,a_h)\, f = \widehat{P}_{h,e}(s_h,a_h) - P^e_h(s_h,a_h)\, f^e + \widehat{P}_{h,e}(s_h,a_h) - P^e_h(s_h,a_h)\,(f^e - f)$$

and

$$\widehat{P}_{h,e}(s_h,a_h) - P^e_h(s_h,a_h)\,(f^e - f) \le \|f^e - f\|_1\,(N+1) \le \varepsilon_{net} N.$$

Setting $\varepsilon_{net} = N$, the metric entropy is of the order $d\log(N(M_V + B)) + \log(BN) + d^2\log(BdN)$. The error incurred with this epsilon net is a constant, which is lower order.

Thus, we have

$$\begin{aligned}
\forall s_h,a_h: \sup_{f\in\mathcal{V}_h} \widehat{P}_{h,e}(s_h,a_h) - P^e_h(s_h,a_h)\, f \\
&\le \sqrt{dN}\,\|\mathcal{V}_h\|_1\,\varepsilon_{ms} + \|\mathcal{V}_h\|_1 M \sqrt{\log(1/\delta) + d\log(M_V) + d^2\log(BdN)}\;\widehat{b}_{h,e}(s_h,a_h) \\
&\le \sqrt{dN}\,M_V\,\varepsilon_{ms} + M_V M \sqrt{\log(1/\delta) + d\log(M_V) + d^2\log(BdN)}\;\widehat{b}_{h,e}(s_h,a_h)
\end{aligned}$$

Note that ε_{ms} scales as $\sqrt{\log B}$, so one can find a valid B by solving $B \ge \varepsilon_{ms}$ for B.

□

Lemma G.6 Let $f \in \mathcal{V}_h$. For any $\delta \in (0,1)$, w.p. at least $1-\delta$, for any time h, episode e, we have

$$\forall s_h,a_h: \; f^T \Big(\sum_{k=1}^{n_e} P^?_h(s^k_h,a^k_h) - P^e_h(s^k_h,a^k_h)\; b_h(s^k_h,a^k_h)^T + \lambda_{h,e}\Big)^{-1} b_h(s_h,a_h) \le 4\|\mathcal{V}_h\|_1\,(1+M)\sqrt{\log(1/\delta) + d\log(N)} + \sqrt{dN}\,\|\mathcal{V}_h\|_1\,\varepsilon_{ms}\;\widehat{b}_{h,e}(s_h,a_h).$$

Proof. First observe that

$$\begin{aligned}
&f^T \Big(\sum_{k=1}^{n_e} (P^?_h(s^k_h,a^k_h) - e_h\, b_h(s^k_h,a^k_h))\, b_h(s^k_h,a^k_h)^T + \lambda_{h,e}\Big)^{-1} b_h(s_h,a_h) \\
&= \lambda_{h,e}^{-1/2} \Big(\sum_{k=1}^{n_e} b_h(s^k_h,a^k_h) f^T (P^?_h(s^k_h,a^k_h) - e_h\, b_h(s^k_h,a^k_h))\Big)^T \lambda_{h,e}^{-1/2} b_h(s_h,a_h) \\
&\quad \Big\| \sum_{k=1}^{n_e} b_h(s^k_h,a^k_h)\varepsilon_k \Big\|^{-1}_{\lambda_{h,e}} \|b_h(s_h,a_h)\|_{\lambda_{h,e}^{-1}};
\end{aligned}$$

where $\varepsilon_k = P^?_h(s^k_h,a^k_h) - P^e_h(s^k_h,a^k_h)\, f$.

Now we will argue that w.p. $1-\delta$, for all e,h,

$$\Big\| \sum_{k=1}^{n_e} b_h(s^k_h,a^k_h)\varepsilon_k \Big\|^{-1}_{\lambda_{h,e}} \le 4\|\mathcal{V}_h\|_1\,(1+M)\sqrt{\log(1/\delta) + d\log(N)} + \sqrt{dN}\,\|\mathcal{V}_h\|_1\,\varepsilon_{ms},$$

which will imply the claim for all s_h,a_h.

Apply self-normalized martingale concentration (Lemma G.1) with $X_i = b_h(s^i_h,a^i_h)$ and $\varepsilon_i = \varepsilon_i - E[\varepsilon_i \mid \mathcal{H}_{i-1}]$, where the expectation is over (s^i_h,a^i_h) in the definition of ε_i. To see sub-Gaussianity, bound the envelope $|\varepsilon_k| \le \|f\|_1 \|P^?_h(s^k_h,a^k_h) - e_h\, b_h(s^k_h,a^k_h)\|_{TV} \le \|\mathcal{V}_h\|_1\,(1+M)$, and thus

$\|\cdot\| \le 2kV_h k_1 (1 + M)$. Now compute the determinant: $\det(\Lambda_{h;0}) = 1$ and since $\lambda_{\max}(\Lambda_{h;e}) \le e + 1$, we have that $\log\det(\Lambda_{h;e}) \le d\log(e+1)$. Hence, w.p. at least $1-\delta$, we have

$$\forall e : \sum_{k=1}^{K} b_h(s_h^k, a_h^k)\,(\hat{\theta}_k - \mathbb{E}[\hat{\theta}_k \mid \mathcal{H}_{k-1}]) \le 2kV_h k_1 (1 + M)\sqrt{2\log(1/\delta) + d\log(N+1)}\,\|\cdot\|_{\Lambda_{h;e}^{-1}}.$$

By Assumption G.1 applied to π^k (the data-generating policy for episode k), we have $\|\mathbb{E}[\hat{\theta}_k \mid \mathcal{H}_{k-1}]\| \le \sqrt{\sum_i c_i^2}\sqrt{\sum_i \|x_i\|^2}$. Recall for any scalars c_i and vectors x_i, we have $\|\sum_i c_i x_i\| \le \sum_i |c_i|\,\|x_i\|$. Thus,

$$\begin{aligned}
&\sum_{k=1}^{K} b_h(s_h^k, a_h^k)\,\mathbb{E}[\hat{\theta}_k \mid \mathcal{H}_{k-1}] \\
&\le \sqrt{\sum_{k=1}^{K} b_h(s_h^k, a_h^k)^2\,\|\cdot\|_{\Lambda_{h;e}^{-1}}}\sqrt{\sum_{k=1}^{K} \mathbb{E}[\hat{\theta}_k \mid \mathcal{H}_{k-1}]^2} \\
&\le \sqrt{d\log\frac{(e+1)kV_h k_1}{\delta}}\,\|\cdot\|_{\Lambda_{h;e}^{-1}}. \qquad \text{(by Lemma G.2)}
\end{aligned}$$

Combining these two bounds concludes the proof.

□

Lemma G.7 (Optimism) Suppose (E_{model}) holds. Let $\beta = kV_h k_1 \varepsilon_{\text{ms}}$. Then, for any episode $e = 1, 2, \ldots, N$, we have

$$\forall h = 0, 1, \ldots, H-1 : \mathbb{E}_{s_h \sim d_h}\left[Q_h^{\star}(s_h, a_h) - \widehat{Q}_{h;e}(s_h, a_h)\right] \le \Gamma_h(\Lambda_h) \qquad (H - h),$$

and

$$\forall h = 0, 1, \ldots, H-1 : \mathbb{E}_{s_h \sim d_h}\left[V_h^{\star}(s_h) - \widehat{V}_{h;e}(s_h)\right] \le \Gamma_{h-1}(\Lambda_{h-1}) \qquad (H - h),$$

where

$$\Gamma_h(s_h) := \left[\widehat{Q}_{h;e}(s_h, b_h^e(s_h)) - M_V\right]_+$$

$$\Gamma_h(\Lambda_h) = \sum_{h'=0}^{h} \Gamma_{h'}(s_{h'}).$$

Abusing notation, $\Gamma_{-1}(\cdot)$ is the constant function 1.

In particular, we have that

$$\mathbb{E}_{d_0}\left[V_0^{\star}(s_0) - \widehat{V}_{0;e}(s_0)\right] \le H.$$

Proof. Fix any episode e. We prove both claims via induction on $h = H, H-1, H-2, \ldots, 1, 0$. The base case holds trivially since $\widehat{Q}_{H;e}$ and $V_H^{\star}$ are zero at every state by definition. Indeed, we have that for any π, including $\pi^{\star}$, that

$$\begin{aligned}
&\mathbb{E}\left[P_{H-1}^{\star}(s_{H-1}, a_{H-1})(V_H^{\star} - \widehat{V}_{H;e})\,\Gamma_{H-1}(\Lambda_{H-1})\right] \\
&= \mathbb{E}\left[(0 - 0)\,\Gamma_{H-1}(\Lambda_{H-1})\right] = 0.
\end{aligned}$$

Now let's show the inductive step. Let $h \in \{H-1, H-2, \ldots, 1, 0\}$ be arbitrary and suppose the inductive hypothesis. So suppose that optimism holds at $h+1$ (we don't even need Q-optimism in the future), i.e.

$$\mathbb{E}_{s_h \sim d_h}\left[P_h^{\star}(s_h, a_h)(V_{h+1}^{\star} - \widehat{V}_{h+1;e})\,\Gamma_h(\Lambda_h)\right] = \mathbb{E}_{s_{h+1} \sim d_{h+1}}\left[V_{h+1}^{\star}(s_{h+1}) - \widehat{V}_{h+1;e}(s_{h+1})\,\Gamma_h(\Lambda_h)\right]$$

$(H - h - 1)$ (IH)

Recalling that $\widehat{Q}_{h;e}(s_h;a_h) = r_h(s_h;a_h) + \widehat{P}_{h;e}(s_h;a_h)\widehat{V}_{h+1;e} + b_{h;e}(s_h;a_h)$, we have

$$\begin{aligned}
&\mathbb{E}_{?}\left[Q^{?}_h(s_h;a_h) - \widehat{Q}_{h;e}(s_h;a_h)\right]_{\pi_h(\cdot)} \\
&= \mathbb{E}_{?}\left[P^{?}_h(s_h;a_h)V^{?}_{h+1} - \widehat{P}_{h;e}(s_h;a_h)\widehat{V}_{h+1;e} - b_{h;e}(s_h;a_h)\right]_{\pi_h(\cdot)} \\
&\le \mathbb{E}_{?}\left[P^{?}_h(s_h;a_h) - \widehat{P}_{h;e}(s_h;a_h)\right]\widehat{V}_{h+1;e} - b_{h;e}(s_h;a_h)\Big]_{\pi_h(\cdot)} + (H-h-1) \\
&\qquad \text{(by (IH))} \\
&\le \mathbb{E}_{?}\left[\widehat{P}_{h;e}(s_h;a_h) - P^{?}_h(s_h;a_h)\right]\widehat{V}_{h+1;e}\Big]_{\pi_h(\cdot)} + \mathbb{E}_{?}[b_{h;e}(s_h;a_h)]_{\pi_h(\cdot)} + (H-h-1) \\
&\le (H-h-1) + (H-h) = (H-h);
\end{aligned}$$

(by (E_{model}) and $\widehat{V}_{h+1;e} \in 2\mathcal{V}_h$ (Lemma G.3))

which proves the Q-optimism claim.

Now let's prove V-optimism.

$$\begin{aligned}
&\mathbb{E}_{?}\left[V^{?}_h(s_h) - \widehat{V}_{h;e}(s_h)\right]_{\pi_{h-1}(\cdot)} \\
&= \mathbb{E}_{?}\left[Q^{?}_h(s_h;a_h) - \widehat{Q}_{h;e}(s_h;b^{e}_h(s_h))\right]_{M_V\,\pi_{h-1}(\cdot)} \\
&= \mathbb{E}_{?}[(Q^{?}_h(s_h;a_h) - M_V)\pi_{h-1}(\cdot)(1 - \phi_h(s_h))] \\
&\quad + \mathbb{E}_{?}\left[Q^{?}_h(s_h;a_h) - \widehat{Q}_{h;e}(s_h;b^{e}_h(s_h))\right]_{\pi_{h-1}(\cdot)\phi_h(s_h)} \\
&\le \mathbb{E}_{?}\left[Q^{?}_h(s_h;a_h) - \widehat{Q}_{h;e}(s_h;b^{e}_h(s_h))\right]_{\pi_h(\cdot)} \\
&\le \mathbb{E}_{?}\left[Q^{?}_h(s_h;a_h) - \widehat{Q}_{h;e}(s_h;\phi^{?}_h(s_h))\right]_{\pi_h(\cdot)} \\
&\le (H-h);
\end{aligned}$$

by Q-optimism. □

Remark G.1 We did not require $\widehat{P}_{h;e}$ to be a valid transition! It is in general unbounded and can even have negative entries!

Lemma G.8 (Simulation) For any episode $e = 1; 2; \ldots; N$, we have

$$\mathbb{E}_{d_0}\left[\widehat{V}_{0;e}(s_0) - V_0^{\pi_e}(s_0)\right] \le \sum_{h=0}^{H-1} \mathbb{E}_{b^{\pi_e}}\left[b_{h;e}(s_h;a_h) + (\widehat{P}_h(s_h;a_h) - P^{?}_h(s_h;a_h))\widehat{V}_{h+1;e}\right]$$

Proof. We progressively unravel the left hand side. For any s_0,

$$\begin{aligned}
&\widehat{V}_{0;e}(s_0) - V_0^{\pi_e}(s_0) \\
&\le \widehat{Q}_{0;e}(s_0;\pi_e(s_0)) - Q_0^{\pi_e}(s_0;\pi_e(s_0)) \\
&= b_{0;e}(s_0;\pi_e(s)) + \widehat{P}_{0;e}(s_0;\pi_e(s))\,P^{?}_0(s_0;\pi_e(s_0))\,\widehat{V}_{1;e} + P^{?}_0(s_0;\pi_e(s_0))\,\widehat{V}_{1;e} - V_1^{\pi_e};
\end{aligned}$$

where the inequality is due to the thresholding on the value function. Now, perform this recursively on the $P^{?}_0(s_0;\pi_e(s_0))\,\widehat{V}_{1;e} - V_1^{\pi_e}$ term. Doing this unravelling H times gives the result. □

Theorem G.2 Suppose Assumption G.1 and let α and bonus be defined as in (7). Let $\delta \in (0;1)$. Then w.p. at least $1-\delta$, we have that the regret of LSVI is sublinear,

$$\mathrm{NV}^{?} - \sum_{e=0}^{N-1} V^{b^{\pi_e}} \le \widetilde{\mathcal{O}}\left(dHNM_V\sqrt{\log(HN/\delta)}\,\iota_{ms} + d^{1.5}H\sqrt{N}M_V M\log(dHN/\delta)\right)$$

where $\widetilde{\mathcal{O}}$ hides $\log$ dependence.

Proof. We first condition on the high-probability event $E(_{\text{model}})$, which occurs w.p. at least 1. Fix any arbitrary episode e. By optimism Lemma G.7 and the simulation lemma Lemma G.8,

$$\begin{aligned}
&\left[V_0^{?}(s_0) \quad V_0^{b^e}(s_0) \right] \quad \left[\hat{V}_0^{e}(s_0) \quad V_0^{b^e}(s_0) \right] + H \\
&\sum_{h=0}^{H-1} \left[E_{b^e} \left[\hat{b}_h^e(s_h, a_h) + \hat{P}_{h;e}(s_h, a_h) \quad P_h^{?}(s_h, a_h) \right] \hat{V}_{h+1;e} \right] + H
\end{aligned}$$

Applying (E_{model}) with no indicators, i.e. $_h(\ _h) = 1$ always, gives,

$$\sum_{h=0}^{H-1} E_{b^e} \left[2\, b_{h;e}(s_h, a_h) \right] + 2H :$$

Now, summing over $e = 1, 2, \ldots, N$, we have

$$\begin{aligned}
&\sum_{e=1}^{N} \left[E_{d_0} \left[V_0^{?}(s_0) \right] \quad V_0^{b^e}(s_0) \right] \\
&\quad 2HN + 2 \sum_{h=0}^{H-1} \sum_{e=1}^{N} E_{b^e} \left[b_{h;e}(s_h, a_h) \right]
\end{aligned}$$

By Azuma's inequality applied to the martingale difference $= E_{b^e}[b_{h;e}(s_h, a_h)] \quad b_{h;e}(s_h^e, a_h^e)$, which has envelope bounded by 2, implies w.p. 1 ,

$$2HN + 2 \sum_{h=0}^{H-1} \sum_{e=1}^{N} b_{h;e}(s_h^e, a_h^e) + 4 \sqrt{N \log(HN/)}.$$

It remains to bound the sum of expected bonuses. By Lemma G.2, we know that almost surely,

$$\sum_{h=0}^{H-1} \sum_{e=1}^{N} b_{h;e}(s_h^e, a_h^e) \quad H \sqrt{dN \log(N)}.$$

So, putting everything together,

$$\begin{aligned}
&\sum_{e=1}^{N} \left[E_{d_0} \left[V_0^{?}(s_0) \right] \quad V_0^{b^e}(s_0) \right] \\
&. \quad HN + H \sqrt{dN \log(N)} + \sqrt{N \log(HN/)} \\
&. \quad HN + \sqrt{dN} M_V \varepsilon_{ms} + M_V M \sqrt{d \log(dHN/)} \quad H \sqrt{dN \log(HN/)} \\
&= \quad HN + dHNM_V \sqrt{\log(HN/)} \varepsilon_{ms} + d^{1.5} H \sqrt{N} M_V M \log(dHN/).
\end{aligned}$$

Note that $H = H \| V \|_1 \varepsilon_{ms} = HM_V \varepsilon_{ms}$ is of lower order (with respect to N), we can simply drop it. This concludes the proof. □

Corollary G.1 By setting $= 1/N$, we have that expected regret also has the same rate as above.

Proof. The expected regret by law of total probability, since regret is at most NH,

$$\begin{aligned}
E[\mathrm{Reg}_N] \quad & E[\mathrm{Reg}_N \mid (E_{\text{model}})] + NH (1 \quad P((E_{\text{model}}))) \\
& E[\mathrm{Reg}_N \mid (E_{\text{model}})] + H.
\end{aligned}$$

Since H is lower-order, we have the same rate. □

H Proof of Main Theorems

First we prove Theorem 4.1 and Theorem 4.2.

Theorem 4.1 (Regret under generative source access) Suppose Assumptions 3.1-3.4 and 4.1, and suppose the input policies $\{\pi_k\}$ are $\alpha_{\min}$-exploratory. Then, for any $\delta \in (0,1)$, w.p. $1-\delta$, RepTransfer when deployed in the target task has regret at most $\tilde{O}(H^2 d^{1.5}\sqrt{T\log(1/\delta)})$, with at most Kn generative accesses per source task, with $n = O\left(\frac{1}{\alpha_{\min}} A_{\max}^3 KT \log\frac{|\mathcal{F}||\Pi|}{\delta} + K\log|\Pi|\right)$.

Theorem 4.2 (Regret for RepTransfer with RewardFree subroutine) Suppose Assumptions 3.1-3.4 and 4.1 hold, and let $\delta \in (0,1)$. Let $\{\pi_k\}_{k=1}^K$ be exploratory policies learned from running RewardFree on each source task with $N_{\text{LSVI-UCB}}$ and $N_{\text{RewardFree}}$ set as in Lemma 4.2. Then, w.p. $1-\delta$, running RepTransfer with $\{\pi_k\}_{k=1}^K$ has regret in the target task of $\tilde{O}(H^2 d^{1.5}\sqrt{T\log(1/\delta)})$, with at most $\tilde{O}(A^4 A_{\max}^3 d^5 H^7 K^2 T^2(\log(|\mathcal{F}|/\delta) + K\log|\Pi|))$ generative accesses per source task.

Proof of Theorem 4.1 and Theorem 4.2. For the regret bound, set $M_V = H$ and $M_\Phi = \mathcal{F}$ and apply Theorem G.2. This choice of M_Φ is valid by the argument following Assumption G.1. This gives us a regret bound of

$$\tilde{O}\left(dH^2 T\varepsilon_{\text{ms}} + d^{1.5}H^2\sqrt{T\log(1/\delta)}\right);$$

where ε_{ms} can be made smaller than $1/\sqrt{T}$, in which the second term dominates.

Now, we calculate the pre-training phase sample complexity in a source task.

First, let's calculate the reward-free model learning sample complexity, i.e. this is the number of samples required for learning $\hat{\mathcal{M}}_k$. Recall that we need this to be sufficiently large such that $\varepsilon_{TV} = 1/N_{\text{LSVI-UCB}}$. As required by Lemma 4.2, we need,

$$\begin{aligned} N_{\text{RewardFree}} &= \tilde{O}\left(A^3 d^4 H^6 \log(|\mathcal{F}||\Pi|/\delta)\right) N_{\text{LSVI-UCB}}^2 \\ &= \tilde{O}\left(A^3 d^4 H^6 \log(|\mathcal{F}||\Pi|/\delta)\right)\left(A^3 d^6 H^8\right)^2 \\ &= \tilde{O}\left(A^9 d^{16} H^{22}\right)\log(|\mathcal{F}||\Pi|/\delta). \end{aligned}$$

Second, we calculate the cross-sampling sample complexity. Recall that n is the number of samples in each pairwise dataset. In order to reduce ε_{ms} to $1/\sqrt{T}$, by Lemma 4.1, we need

$$\begin{aligned} 1/\sqrt{T} \quad &\varepsilon_{\text{ms}} \quad A_{\max}^3 K/\alpha_{\min}^{1/2}\sqrt{\frac{1}{n}\log\frac{|\mathcal{F}||\Pi|}{\delta} + K\log|\Pi|} \\ &A_{\max}^3 K/\alpha_{\min}^{1/2}\sqrt{\frac{1}{n}\log\frac{|\mathcal{F}||\Pi|}{\delta} + K\log|\Pi|} \qquad \text{(by (2))} \end{aligned}$$

which implies that we need

$$n \quad \frac{1}{\alpha_{\min}} A_{\max}^3 KT\left(\log\frac{|\mathcal{F}||\Pi|}{\delta} + K\log|\Pi|\right).$$

Incorporating the coverage result from Lemma 4.2 gives,

$$n \quad A^4 A_{\max}^3 d^5 H^7 KT^2\left(\log\frac{|\mathcal{F}||\Pi|}{\delta} + K\log|\Pi|\right).$$

Since each task is in at most K^2 pairwise datasets, each of size n, the total pre-training sample complexity per task is at most,

$$\begin{aligned} &N_{\text{RewardFree}} + (K \quad 2)\, n \\ &= \tilde{O}\left(A^9 d^{16} H^{22}\right)\log(|\mathcal{F}||\Pi|/\delta) + A^4 A_{\max}^3 d^5 H^7 K^2 T^2\left(\log\frac{|\mathcal{F}||\Pi|}{\delta} + K\log|\Pi|\right). \end{aligned}$$

□

Now we prove Theorem 5.2, restated below.

Theorem 5.2 (Regret with online access). Suppose Assumptions 3.1-3.4,5.1,5.2 hold. For any $\in (0;1)$, w.p. 1 , Algorithm 5 with appropriately set parameters achieves a regret in the target task at most $\tilde{O}\left(d^{1:5}H^{2}\sqrt{T\log(1=)}\right)$, with at most $\mathrm{poly}\left(A;\ _{\max};d;H;K;T;\ ^{1};\ _{\mathrm{raw}}^{1};\log(j\ jj\ j=)\right)$ online queries in the source tasks.

Proof of Theorem 5.2. We follows the same format as the proof of Theorem 4.1. The regret bound is identical.

Now let's compute the pre-training sample complexity. The regret bound requires us to set $"_{\mathrm{ms}}$ to $1=\sqrt{T}$. Here, our $"_{\mathrm{ms}}$ comes from Lemma F.1, so

$$1=\sqrt{T}\quad \frac{_{\max}K^{1=2}}{(\ _{\mathrm{raw}}\ _{\min})^{1=2}}\quad \sqrt{\frac{1}{n}\quad \log\frac{j\ j}{}+K\ \log j\ \ j\ };$$

which implies we need

$$n\quad \frac{^{2}_{\max}K}{\ _{\mathrm{raw}}\ _{\min}}\quad \log\frac{j\ j}{}+K\ \log j\ \ j\ :$$

Plugging in the coverage of Lemma 4.2,

$$n\quad \frac{^{2}_{\max}KT}{\ _{\mathrm{raw}}}\quad \log\frac{j\ j}{}+K\ \log j\ \ j\quad A^{\ 3}d^{\ 5}H^{\ 7\ \ 2\ \ 1}$$
$$\frac{A^{3}\ ^{2}_{\max}d^{5}H^{7}KT}{\ _{\mathrm{raw}}^{\ 2}}\quad \log\frac{j\ j}{}+K\ \log j\ \ j\ :$$

Here, we only collect one dataset, so the total pre-training sample complexity is

$$N_{\mathrm{REWARDFREE}}+n$$
$$=\tilde{O}\left(A^{9}d^{16}H^{22}\quad ^{4}\log(j\ jj\ j=)+A^{3}\ ^{2}_{\max}d^{5}H^{7}KT\ \ _{\mathrm{raw}}^{1}\quad ^{2}\quad \log\frac{j\ j}{}+K\ \log j\ \ j\right):$$

□

I Experiment Details

I.1 Construction of Comblock

In this section we first introduce the vanilla Combination lock (Comblock) environment that is widely used as the benchmark for algorithms for Block MDPs. We provide a visualization of the comblock environment in Fig. 1(a). Concretely, the environment has a horizon H, and 3 latent states $z_{i;h};i\ 2\ f\ 0;1;2g$ for each timestep h and 10 actions. Among the three latent states, we denote $z_{0;h}$ and $z_{1;h}$ as the good states which leads to the final reward and $z_{2;h}$ as the bad states. At the beginning of the task, the environment will uniformly and independent sample 1 out of the 10 actions for each good state $z_{0;h}$ and $z_{1;h}$ for each timestep h, and we denote these actions as $a_{0;h};a_{1;h}$ as the optimal actions (corresponding to each latent state). These optimal actions, along with the task itself, determines the dynamics of the environment. At each good latent state $s_{0;h}$ or $s_{1;h}$, if the agent takes the correct action, the environment transits to the either good state at the next timestep (i.e., $s_{0;h+1}$ or $s_{1;h+1}$) with equal probability. Otherwise, if the agent takes any 9 of the bad actions, the environment will transition to the bad state $z_{2;h+1}$ deterministicly, and the bad states transit to only bad states at the next timestep deterministicly. There are two situations where the agent receives a reward: one is uponing arriving the good states at the last timestep, the agent receives a reward of 1. The other is upon the first ever transition into the bad state, the agent receives an "anti-shaped" reward of 0.1 with

probability 0.5. Such design makes greedy algorithms without strategic exploration such as policy optimization methods easily fail. For the initial state distribution, the environment starts in $s_{0;0}$ or $s_{1;0}$ with equal probability. The dimension of the observation is $2^{d\log(H+|S|+1)}e$. For the emission distribution, given a latent state $s_{i;h}$, the observation is generated by first concatenate the one hot vectors of the state and the horizon, adding i.i.d. $N(0,0.1)$ noise at each entry, appending 0 at the end if necessary. Then finally we apply a linear transfermation on the observation with a Hadamard matrix. Note that without a good feature or strategic exploration, it takes $1/2^H$ actions to reach the final goal with random actions.

I.2 Construction of transfer setup in Section. 6.1

In this section we introduce the detailed construction of the experiment in Section. 6.1. For the source environment, we simply generate 5 random vanilla comblock environment described in Section. I.1. Note that in this way we ensure that the emission distribution shares across the sources, but the latent dynamics are different because the optimal actions are independently randomly selected. For the target environment, for each timestep h, we randomly acquire the optimal actions a_h from one of the sources and set it to be the optimal action of the target environment at time step h. If the selected optimal actions are different for the two good states. Otherwise we keep sampling until they are different. Note that under such construction, since we fix the emission distribution, Assumption 3.4 is satisfied if we set $\alpha = 1$ for the source environment where we select the optimal action and $\alpha = 0$ for the other sources, at each timestep. To see how Assumption 5.1 is satisfied, recall that comblock environment naturally satisfies Assumption 3.3, and identical emission implies that the conditional ratio of all observations between source and target is 1.

I.3 Construction of transfer setup in Section. 6.2

Now we introduce the construction of the Comblock with Partitioned Observation (Comblock-PO) environment. Comparing with the vanilla comblock environment, the major difference is in the observation space. In this setting, the size of the observation depends on the number of source environments K. Let the size of the original observation space be $O = |\mathcal{O}|$, the size of the observation for comblock-PO is KO. For the k-th source environment, where $k \in [K]$, the environment first generates the O-dimensional observation vector as in the original comblock, and then embed it to the $(k-1)O$-th to kO-th entries of the KO-dimensional observation vector, where it is 0 everywhere else. Thus we can see that the observation space for each source environment is disjoint (and thus the name partitioned observations). For the target enviornment, since the latent dynamcis are the same, we only need to design the emission distribution: for each latent state $s_{i,h}$, we assign the emission distribution uniformly at random from one of the sources.

I.4 Implementation details

Our implementation builds on BRIEE (Zhang et al., 2022)[5]. In the Multi-task REPLEARN stage, we requires our learned feature to predict the Bellman backup of all the sources simultaneously. Therefore, in each iteration we have k discriminators and k sets of linear weights (instead of 1 in BRIEE), where k is the number of source environments. For the deployment stage we implement LSVI following Algorithm 6.

To create the training dataset for Multi-task REPLEARN, for each (i, j) environment pairs where $i \neq j$, we collect 500 samples for each timestep. For each (i, i) environment pairs, we collect $500 \cdot (k-1) \cdot k$ samples for each timestep, where k denotes the number of sources. Thus we ensure that the number of samples from cross transition of different environments is the same as the number of samples from cross transition of the same environment. For the online setting, we simply sample $1000 \cdot (k-1) \cdot k$ samples for each (i, i) cross transition to ensure that the total number of samples is the same for GEN-TRANSFER and O-REP-TRANSFER.

[5] Code based on public repository: https://github.com/yudasong/briee .

To sample the initial state action pair (i.e.,(s,a) pair as in (1)), for90%of the samples, we follow the nal policy from each source environment trained using GEE. For the remaining10%, we follow the same policy to state$s$, and then take a uniform random policy to get With this sampling scheme we ensure that Assumption 3.3 is satis ed. In the setting of Section. 6.2, we follow a more simple procedure to ensure that the samples are more balanced among the three states: we skip the rst sampling step from environment(i.e., sample$s$ given(s,a)), and simply reset environment to$s$, where$s$ is one of the three states with equal probability, and generate the observation accordingly. Note that such visitation distribution is also possible in the online setting with a more nuanced sampling procedure, and in the experiment we use the same sampling procedure for both G-REPTRANSFERand O-REPTRANSFERfor a fair comparison.

I.5 Hyperparameters

In this section, we record the hyperparameters we try and the nal hyperparameter we use for each baselines. The hyperparameters for REPTRANSFERin Section. 6.1 is in Table. 3. The hyperparameters for REPTRANSFERin Section. 6.2 is in Table. 4. The hyperparameters for GEE is in Table. 5. We use the same set of hyperparameters for G-REPTRANSFERand O-REPTRANSFER.

Table 3: Hyperparameters for REPTRANSFERin Comblock.

	Value Considered	Final Value
Decoder learning rate	{1e-2}	1e-2
Discriminator$_f$ learning rate	{1e-2}	1e-2
Discriminator$_f$ hidden layer size	{256}	256
RepLearn Iteration$T$	{30}	30
Decoder number of gradient steps	{64}	64
Discriminator$_f$ number of gradient steps	{64}	64
Decoder batch size	{256}	256
Discriminator$_f$ batch size	{512}	512
RepLearn regularization coef cient	{0.01}	0.01
Decoder softmax temperature	{1}	1
Decoder$_0$ softmax temperature	{0.1}	0.1
LSVI bonus coef cient	$\{1, \frac{H}{5}\}$	1
LSVI regularization coef cient	{1}	1
Buffer size	{1e5}	1e5
Update frequency	{50}	50
Optimizer	{SGD}	SGD

Table 4: Hyperparameters for REPTRANSFER in Comblock-PO.

	Value Considered	Final Value
Decoder learning rate	{1e-2}	1e-2
Discriminator learning rate	{1e-2}	1e-2
Discriminator hidden layer size	{256,512}	256
Discriminator hidden layer number	{2,3}	3
RepLearn Iteration T	{30,40,50,100,150}	50
Decoder number of gradient steps	{64,80,128,256}	64
Discriminator number of gradient steps	{64,80,128,256}	64
Decoder batch size	{256,512}	512
Discriminator batch size	{256,512}	512
RepLearn regularization coefficient	{0.01}	0.01
Decoder softmax temperature	{1}	1
Decoder $_0$ softmax temperature	{0.1,1}	1
LSVI bonus coefficient	$\{1, \frac{H}{5}\}$	1
LSVI regularization coefficient	{1}	1
Buffer size	{1e5}	1e5
Update frequency	{50}	50
Optimizer	{SGD, Adam}	Adam

Table 5: Hyperparameters for BRIEE in Comblock and Comblock-PO.

	Value Considered	Final Value
Decoder learning rate	{1e-2}	1e-2
Discriminator learning rate	{1e-2}	1e-2
Discriminator hidden layer size	{256}	256
RepLearn Iteration T	{30}	30
Decoder number of gradient steps	{64}	64
Discriminator number of gradient steps	{64}	64
Decoder batch size	{512}	512
Discriminator batch size	{512}	512
RepLearn regularization coefficient	{0.01}	0.01
Decoder softmax temperature	{1}	1
Decoder $_0$ softmax temperature	{0.1}	0.1
LSVI bonus coefficient	$\{\frac{H}{5}\}$	$\frac{H}{5}$
LSVI regularization coefficient	{1}	1
Buffer size	{1e5}	1e5
Update frequency	{50}	50

I.6 Visualizations

In this section we provide a comprehensive visualization of the decoders for all baselines in the target environment. We observe that the behaviors of all baselines are similar across the 5 random seeds. Thus to avoid redundancy, we only show the visualization from 1 random seed.

I.6.1 Visualizations from Section. 6.1

We record the visualization of the 5 sources from Fig. 3 to Fig. 7; REPTRANSFER in Fig. 8; G-REPTRANSFER in Fig. 9; running BRIEE on target in Fig. 10.

I.6.2 Visualizations from Section. 6.2

We record the visualization of the 2 sources from Fig. 11 and Fig. 12; REPTRANSFER in Fig. 13; G-REPTRANSFER in Fig. 14; running BRIEE on target in Fig. 15. Note that the features collapse at some timesteps in Fig. 14 and Fig. 15, but this is acceptable because the optimal actions at those timesteps are the same for the collapsed states.

Figure 3: Visualization of decoders from source 1. Note the collapse happens at timestep 5, 9 and 17.

Figure 4: Visualization of decoders from source 2. Note the collapse happens at timestep 1 and 10.

Figure 5: Visualization of decoders from source 3. Note the collapse happens at timestep 14 and 15.

Figure 6: Visualization of decoders from source 4. Note the collapse happens at timestep 7, 16, 24.

Figure 7: Visualization of decoders from source 3. Note the collapse happens at timestep 13 and 16.

Figure 8: Visualization of decoders from CER TRANSFER

Figure 9: Visualization of decoders from GPT TRANSFER

Figure 10: Visualization of decoders from running GPT on target.

Figure 11: Visualization of decoders from source environment 1.

Figure 12: Visualization of decoders from source environment 2.

Figure 13: Visualization of decoders from GPT TRANSFER.

Figure 14: Visualization of decoders from GPT TRANSFER.

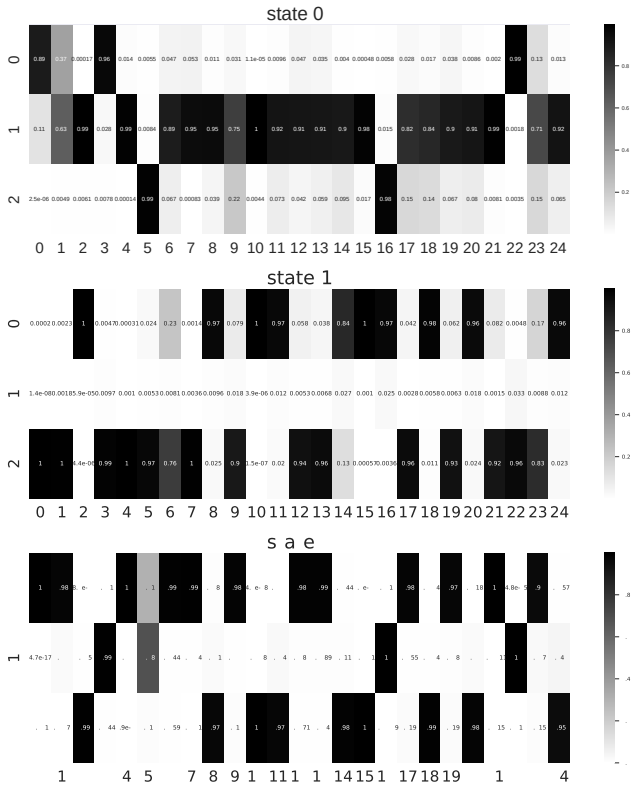

Figure 15: Visualization of decoders running BRIEE in the target.