Proceedings of Machine Learning Research vol 195:1–60, 2023 36th Annual Conference on Learning Theory

Provable Benefits of Representational Transfer in Reinforcement Learning

Alekh Agarwal ALEKHAGARWAL@GOOGLE.COM
Google

Yuda Song YUDAS@ANDREW.CMU.EDU
Carnegie Mellon University

Wen Sun WS455@CORNELL.EDU
Cornell University

Kaiwen Wang KW437@CORNELL.EDU
Cornell University

Mengdi Wang MENGDI@PRINCETON.EDU
Princeton University

Xuezhou Zhang XZ7392@PRINCETON.EDU
Princeton University

Editors: Gergely Neu and Lorenzo Rosasco

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 in 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. In our experiments, we observe speed up in learning in the target by pre-training, and also validate the need for generative access in source tasks.

Keywords: Transfer Learning, Low-Rank MDPs, Reinforcement Learning Theory.

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?

© 2023 A. Agarwal, Y. Song, W. Sun, K. Wang, M. Wang & X. Zhang.

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 $P(yjx)$ 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 extensions to RL (Lu et al., 2021; Cheng et al., 2022) resulted in strong assumptions during the learning setup.
With this context, we summarize our main contributions below.

- **Task relatedness:** We propose a new *state-dependent* linear span assumption of task relatedness and give examples captured by this setting. Our formulation *generalizes all prior settings for representational transfer in RL*, *e.g.*, Cheng et al. (2022).

- **Transfer guarantees under weaker assumptions:** We propose a transfer RL algorithm REPTRANSFER and prove that it pre-trains a representation for downstream online learning in any target task satisfying the linear span assumption. Our algorithm employs a novel *cross-sampling procedure* made possible by generative access in the source tasks. Our key result is that the target task regret almost matches (up to a task-relatedness constant) that of learning in a linear MDP with *known* features, the strongest possible benchmark to compete with. Our regret bounds for REPTRANSFER hold under *significantly weaker* coverage assumptions than prior works, and we do not require any generalization assumptions. We highlight one key technical contribution is a novel analysis of LSVI-UCB (Jin et al., 2020b) attains *regret under an average-case misspecified linear MDP*.

- **Lower bound without generative access:** We further show a counter-example where representational transfer fails without generative access under our assumptions. As a partial remedy, we posit that every *observed state* is reachable in each source task, and show a modification of REPTRANSFER is still sufficient for transfer learning with only online access. While strong,

	Task-relatedness	Access type (source/target)	Reachability	Generalization (assumption)
Lu et al. (2021)	Full rank LSVI weight matrix from source	Gen/Gen	Distribution q (**given**) covers observations	$\mathrm{err}(s;a) \le C\mathsf{E}_q[\mathrm{err}]\forall s;a$
Cheng et al. (2022)	$P^?_{\text{target}}(s'|s;a) = \sum_{i=1}^K {}_i P^?_i(s'|s;a)$	On/On	Observational reachability	$\mathrm{err}(s;a) \le C\mathsf{E}_{\text{Unif}}[\mathrm{err}]\forall s;a$
Theorem 3 (this paper)	$P^?_{\text{target}}(s'|s;a) = \sum_{i=1}^K {}_i(s) P^?_i(s'|s;a)$	Gen/On	Feature reachability ($^?$)	None
Theorem 7 (this paper)	$P^?_{\text{target}}(s'|s;a) = \sum_{i=1}^K {}_i(s) P^?_i(s'|s;a)$	On/On	Observational reachability	None
Theorem 6 (lower bound)	$P^?_{\text{target}}(s'|s;a) = \sum_{i=1}^K {}_i(s) P^?_i(s'|s;a)$	On/On	Feature reachability ($^?$)	None

Table 1: Assumptions for representational transfer in low-rank MDPs. "Gen" and "On" refer to generative or online access to source and target tasks. Feature reachability means that each source task a policy with a full rank covariance under the features $^?$ (Assumption 3.2). Observational reachability requires each high-dimensional raw observation to be reachability with some lower bounded probability (Assumption 4.1). The last row is a lower bound which precludes learning under the assumptions of Theorem 3 without generative access in the source tasks.

this observational reachability assumption still generalizes prior results in transfer RL, *e.g.*, Cheng et al. (2022).

- **Empirical validation:** We empirically validate REPTRANSFER on a challenging benchmark (Misra et al., 2020), and show that REPTRANSFER saves an order of magnitude of target samples compared to training from scratch using the SOTA Block MDP algorithm BRIEE.

Our intermediate results may also be of independent interest: (1) to pre-train a representation, we developed an oracle-efficient reward-free exploration algorithm for low-rank MDPs, (2) to transfer the pre-trained representation to the target task, we develop a new analysis for linear MDP under an average case model misspecification extending prior work which relies on a much stronger ℓ_1 style model misspecification (Jin et al., 2020b).

1.1. Related Work

Transfer Learning in Low-rank MDPs. The closest work to ours is Cheng et al. (2022), which also performs reward-free exploration in the source tasks for representation learning, and use the learned representation in the target task to perform online learning. Cheng et al. (2022) proposed a linear span assumption with globally fixed coefficients, which is generalized by our *state-dependent* linear span assumption. However, despite the more stringent relatedness condition, their work still makes stronger assumptions to enable transfer. First, their Assumption 5.3 sidesteps the need to handle generalization by assuming point-wise error is bounded by average-case error (this allows them to directly use the result from Jin et al. (2020b)), whereas our analysis only relies on standard in-distribution generalization and indeed one of our key technical contributions is showing that LSVI-UCB succeeds even with average-case misspecification. Our Theorem 7 generalizes the result of Cheng et al. (2022). Second, their Assumption 5.1 assumes reachability in the high-dimensional

observation space, whereas we show that with our novel cross-sampling, it is possible to require the much more realistic spectral coverage in the ground truth (unknown) feature space (Assumption 3.2).

Another work on transfer learning in low-rank MDPs is Lu et al. (2021), which also makes much stronger assumptions than our work. First, they require generative access in *both* source and target tasks. Second, they require the covariance of *any pair* of features (in feature class) to be full rank, while we only require this reachability condition for the true feature ?. Third, they assume a given distribution q on which the learned representation can extrapolate, whereas we explicitly construct such a data distribution using the novel cross-sampling procedure. In sum, compared to the prior works Cheng et al. (2022); Lu et al. (2021), we leverage a novel cross-sampling procedure to enable transfer under significantly weaker assumptions. Furthermore, we prove a lower bound showing that generative access is necessary unless stronger assumptions, *e.g.*, those in Cheng et al. (2022), are made. We summarize the comparision to these prior works in Table 1.

Transfer Learning in Bandit and small-size MDPs. Lazaric et al. (2013) study spectral techniques for online sequential transfer learning in multi-arm bandits. Brunskill and Li (2014) study transfer in semi-MDPs by learning options. Lecarpentier et al. (2021) consider lifelong learning in Lipschitz MDP. All these works consider tabular models while we focus on large-state MDPs.

Multi-task learning. While the multi-task setting also deals with multiple tasks, it is different from the transfer learning setting in its objective. The goal of multi-task learning is to perform well over all tasks (typically the average performance of tasks), while transfer learning cares exclusively about performance in the target task. Thus, the results from multi-task learning are not directly comparable to the transfer learning results that we focus on in this paper. We survey some multi-task literature for completeness. For multi-task learning in low-rank MDPs, Huang et al. (2023) only assumed ? to be shared (? can be arbitrarily different between tasks), and showed that the sample complexity of a multi-task variant of BiLin-UCB (Du et al., 2021) does not scale as Kj j but only as j j. However, like BiLin-UCB, the algorithm is not computationally efficient. Several recent works study multi-task linear bandits with linear representations ($(s) = As$ with unknown A) (Hu et al., 2021; Yang et al., 2020, 2022). The techniques developed in these works crucially rely on the linear structure and can not be applied to nonlinear function classes.

For a discussion of the empirical transfer literature, as well as more detailed comparisons to related works, please see Section C.

2. Preliminaries

In this paper, we study transfer learning in finite-horizon, episodic Markov Decision Processes (MDPs), $M = hH; S; A; fP_h^? g_{0:H\ 1}; fr_h g_{0:H\ 1}; d_0 i$, specified by the episode length H, state space S, discrete action space A of size A, *unknown* transition dynamics $P_h^? : S\ A\ !\ (S)$, *known* reward functions $r_h : S\ A\ !\ [0; 1]$, and a *known* initial distribution $d_0\ 2\ (S)$. We now define the value, Q functions and visitation distribution, where we make the dependence on the transition dynamics $P^? = fP_h^? g_{0:H\ 1}$ and reward functions $r = fr_h g_{0:H\ 1}$ explicit. For any Markov policy $: S\ !\ (A)$, let $\mathsf{E}_{;P^?}[\ ]$ denote the expectation under the trajectory distribution of executing in an MDP with transitions $P^?$, *i.e.*, start at an initial state $s_0\ d_0$, then for all $h\ 2\ [0 : H\ 1]$[1], $a_h\ h(s_h); s_{h+1}\ P_h^?(s_h; a_h)$. If $P^?$ is clear from context, we use $\mathsf{E}[\ ]$ instead. The value function is the expected reward-to-go of start-

1. For $1 \le a \le b$, we denote $[a : b] = a, a+1, \ldots, b-1, b$, and $[b] = [1 : b]$.

ing at state s in step h, *i.e.*,$V^{P^?;r;h}(s) = \mathsf{E}_{;P^?}\left[\sum_{=h}^{H} {}^{1} r\ (s\ ;a\ )\ j\ s_h = s\right]$. The Q function is $Q^{P^?;r;h}(s;a) := r_h(s;a) + \mathsf{E}_{s^0\ \ P_h^?(\,js;a)} V^{P^?;r;h+1}(s^0)$. We denote the expected total reward of a policy as $V^{P;r} := \mathsf{E}_{s_0\ \ d_0} V^{P;r;0}(s_0)$. We define the state-action occupancy distribution $d^{P^?;h}(s;a)$ as the probability of visiting $(s;a)$ at time step h.

The *transfer learning* problem consists of two phases: (1) the *pre-training phase* where the agent interacts with K source tasks with transition dynamics $\{P_k^?\}_{k2[K]}$, and (2) the *deployment phase* where the agent is deployed into the target task with transition dynamics $P^?_{\text{target}}$ and can no longer access the source tasks. The performance of a transfer learning algorithm is measured by (1) the sample complexity in the source tasks during pre-training, and (2) the regret in the target task during deployment, which is defined as $\mathsf{Regret}(T) = \sum_{t=1}^{T} V^?_{\text{target}} \quad V^{\ t}_{P^?_{\text{target}};r;}$ where $V^?_{\text{target}}$ is the optimal value that can be obtained in the target task, and ^{t} is the policy played in the t-th episode of deployment. For notation, we let $d_{k;h}$ be short-hand for $d^{P_k^?;h}$.

We begin formalizing our problem with the low-rank MDP model.

Definition 1 (Low-rank MDP (Jiang et al., 2017; Agarwal et al., 2020)) *A transition model $P_h^? : S \quad A \ ! \quad (S)$ is low rank with rank d 2 N if there exist two unknown embedding functions $^?_h : S \quad A \ 7! \ \mathsf{R}^d$, $^?_h : S \ 7! \ \mathsf{R}^d$ such that $8s;s^0$ 2 $S;a$ 2 $A : P_h^?(s^0\ j\ s;a) = {}^?_h(s^0)^{>}\ {}^?_h(s;a)$, where $k\ ^?_h(s;a)k_2 \quad 1$ for all $(s;a)$ and for any function $g : S \ ! \ [0;1]$, $\left\|\int g(s)\mathrm{d}\ {}^?_h(s)\right\|_2 \quad \overline{d}$. An MDP is a low rank MDP if $P_h^?$ admits such a low rank decomposition for all h 2 $[0 : H \quad 1]$.*

Low-rank MDPs capture the latent variable model (Agarwal et al., 2020) where $^?(s;a)$ is a distribution over a discrete latent state space Z, and the block-MDP model (Du et al., 2019) where $^?(s;a)$ is a one-hot encoding vector. Note that can be a non-linear, flexible function class, so the low-rank framework generalizes prior works with linear representations (Hu et al., 2021; Yang et al., 2020, 2022). Next, we define what it means for a policy to be exploratory in a low-rank MDP.

Definition 2 (Feature coverage) *For 2 R_+, a policy is -exploratory in an MDP with transition dynamics $P^?$ if for all h 2 $[0 : H \quad 1]$, we have $\min(\mathsf{E}_{;P^?}[\ {}^?_h(s_h;a_h)\ {}^{?>}_h(s_h;a_h)^{>}])$.*

An exploratory intuitively ensures that the whole R^d feature space is well-explored in a spectral sense. Note this generalizes the notion of "Policy Cover" in Block MDPs from Misra et al. (2019).

We now make two mild structural assumptions on the tasks to enable representational transfer.

Assumption 2.1 (Common $^?$) *All tasks are low-rank MDPs with a shared representation $^?_h(s;a)$.*

Assumption 2.2 (Point-wise Linear span) *For any h 2 $[0 : H \quad 1]$ and s^0 2 S_{h+1}, there is a vector $_h(s^0)$ 2 R^K such that $^?_{target;h}(s^0) = \sum_{k=1}^{K} \ _{k;h}(s^0)\ {}^?_{k;h}(s^0)$.*

The motivation for Assumption 2.2 is: if s^0 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. Intuitively, this is necessary for transfer learning to succeed, as s^0 could be a high rewarding state in the target. Based on Assumption 2.2, we define, $_{\max} = \max_{h;k;s^0 2 S} j\ _{k;h}(s^0)j$ and $= \max_h \sum_{k=1}^{K} \max_{s^0 2 S} j\ _{k;h}(s^0)j$. Note that $_{\max}$, which we assume is bounded. We conclude the section with a couple of examples where these assumptions are satisfied.

Example 1 (Mixture of source tasks) *The mixture model posits that the target task's transition dynamics is a mixture of the source tasks, i.e., $P^?_{target}(s^0 j s;a) = \sum_{k=1}^{K} \ _k P_k^?(s^0 j s;a)$. This maps to*

Algorithm 1 Exploratory Policy Search (EPS)

1: **Input:** MDP M with online access, num. LSVI-UCB episodes $N_{\text{LSVI-UCB}}$, num. model-learning episodes $N_{\text{RewardFree}}$, failure probability .
2: Learn model $f\widehat{P}_h = (\widehat{\ }_h; \widehat{\ }_h)g_{h=0}^{H\ 1}$ by running RewardFree Rep-UCB (Algorithm 3) in M for $N_{\text{RewardFree}}$ episodes.
3: Set $= dH\sqrt{\log(dHN_{\text{LSVI-UCB}}=)}$.
4: Return $= \text{LSVI-UCB}\left(f\widehat{\ }_h g_{h=0}^{H\ 1}; r = 0; N_{\text{LSVI-UCB}};\ ; \text{UniformActions} = \text{True}\right)$ by simulating in the learned model $\widehat{P}$ (Algorithm 5). Note this step requires no samples from $P^?$.

Assumption 2.2 with $_{k;h}(s^0) = p_k$ *where* $fp_k g_{k2[K]}$ *is a probability distribution, so* $= 1$. *These mixture models have been studied in the context of known source models (Modi et al., 2020; Ayoub et al., 2020), and, corresponding to our Assumption 2.1, unknown low-rank source models with the same* $^?$ *(Cheng et al., 2022). Our linear span Assumption 2.2 strictly generalizes the mixture model by allowing linear span coefficients to flexibly depend on* s^0*, which is more realistic in practice.*

In Example 1, (and hence $_{\max}$) was nicely bounded by 1. However, if the target task largely focuses on observations quite rare under the source tasks, then can grow large.

Example 2 (Block MDPs with shared latent dynamics) *Here, each MDP* $P_k^?$ *is a Block MDP (Du et al., 2019) with a shared latent space* Z *and a shared decoder* $^? : S\ !\ Z$. *In a block MDP, given state action pair* $(s; a)$, *the decoder* $^?$ *maps* s *to a latent state* z, *the next latent state is sampled from the latent transition* $z^0\ \ P(\ jz; a)$, *and the next state is generated from an emission distribution* $s^0\ \ o(\ jz^0)$. *Recall that* $o(s^0jz^0) > 0$ *at only one* $z^0\ 2\ Z$ *for any* $s^0\ 2\ S$ *for a block MDP. We assume that the latent transition model* $P(z^0jz; a)$ *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 Definition 2 posits that the agent can visit the entire 2-D map, while Assumption 2.2 requires that the color/lighting conditions of the target task resemble that of at least one source task. The coefficients for any* s^0 *are non-zero on the source tasks which can generate that observation.*

3. Representational Transfer with Generative Access in Source Tasks

In this section, we study transfer learning assuming *generative access* to the source tasks.

Assumption 3.1 (Generative access in the source tasks) *For any* $k\ 2\ [K]; h\ 2\ [0 : H\ \ 1]$ *and* $s; a\ 2\ S\ \ A$, *we can query independent samples from* $P_{k;h}^?(s; a)$.

The generative model access is not unrealistic, especially in applications where a high-quality simulation environment is available. Perhaps surprisingly, we will also show (in Section 4) that generative access in source tasks is *necessary* assuming only feature coverage, as in Definition 2.

3.1. The Algorithm

We first describe the helper algorithm **E**xploratory **P**olicy **S**earch (EPS) (Algorithm 1) to discover exploratory policies in low-rank MDPs. EPS has two steps. First, it runs a reward-free variant of

Algorithm 2 Transfer learning with generative access (REPTRANSFER)

PRE-TRAINING PHASE

Input: exploratory policies in the source tasks $\{\ _k\}_{1:K}$, function classes $\ ; \{\ _k\}_{1:K}$, size of cross-sampled datasets n, failure probability $\ $.

1: **for** *task pairs* $i; j$, *i.e.*, for all $i; j \in [K]$ **do** {cross sampling procedure}

2: For each $h \in [H \ \ 1]$, sample dataset $D_{ij;h}$ containing n *i.i.d.* $(s; a; s^0)$ tuples sampled as:

$$(s; a) \quad d^{\ i}_{i;h\ \ 1}; s \quad P^?_{j;h\ \ 1}(\ js; a); a \quad \text{unif}(A); s^0 \quad P^?_{i;h}(\ js; a): \tag{1}$$

For $h = 0$, use $s \quad P^?_{j;h\ \ 1}(\ js; a); a \quad \text{unif}(A); s^0 \quad P^?_{i;h}(\ js; a)$.

3: For each $h \in [0 : H \ \ 1]$, learn features with MLE, *i.e.*, "Multi-task REPLEARN",

$$\widehat{\ }_h; \widehat{\ }_{1:K} = \underset{2\ ;\ _k 2\ \ _k\ i;j 2[K]}{\text{argmax}} \sum \mathsf{E}_{D_{ij;h}}\left[\log\ \ (s; a)^{>}\ \ _k(s^0)\right]: \tag{2}$$

DEPLOYMENT PHASE

Additional Input: number of deployment episodes T.

1: Set $\ = H\sqrt{d} + \ \ dH\sqrt{\log(dHT=\ )}$.

2: Run LSVI-UCB$\left(\widehat{f\ }_h g_{h=0}^{H\ \ 1}; r = r_{\text{target}}; T; \ \ \right)$ in the target task $P^?_{\text{target}}$ (Algorithm 5).

REP-UCB(Uehara et al., 2021) in each source task k, to learn a linear MDP which approximates the true low-rank MDP $P^?_k$. Then, an exploratory policy is learned via reward-free exploration in the learned linear MDP (e.g., using LSVI-UCB with zero reward), which involves no further interactions with the true environment. Intuitively, the policy $\ _k$ is trained to fit Definition 2 in the source task k.

We now present our main algorithm REPTRANSFER (Algorithm 2), which takes as input exploratory policies in each source task that can be obtained from EPS. During the pre-training phase, REPTRANSFER collects a dataset via a novel *cross-sampling* procedure across all pairs of source tasks. Note this step is only possible due to generative access in the source tasks. Concretely, fix any $h \in [H \ \ 1]$ and let $\ _i; \ _j$ be exploratory policies from source tasks $i; j \in [K]$. We first sample from the visitation distribution of $\ _i$ in task i, *i.e.*, $s_{h\ \ 1}; a_{h\ \ 1} \quad d^{\ i}_{i;h\ \ 1}$. Then, in the simulator of task j, we *reset* to $(s_{h\ \ 1}; a_{h\ \ 1})$ and perform a transition step to s_h, *i.e.*, $s_h \quad P_{j;h\ \ 1}(s_{h\ \ 1}; a_{h\ \ 1})$. Next, we uniformly sample an action a_h, reset the simulator of task k to state $s_h; a_h$, and transition to s_{h+1}, *i.e.*, $s_{h+1} \quad P^?_k(s_h; a_h)$. We then perform Maximum Likelihood Estimation (MLE) representation learning in Eq. (2) using the union of the cross-sampled datasets across all pairs of source tasks. In sum, REPTRANSFER learns a *single* representation $\widehat{\ }$ in the pre-training phase using MLE on the cross-sampled datasets from exploratory policies across tasks. In the deployment phase, REPTRANSFER runs optimistic least squares value iteration (LSVI-UCB) in the target task with the learned representation. First proposed by Jin et al. (2020b), LSVI-UCB is displayed in Algorithm 5, which at a high level is as follows. Given any dataset, $fs; a; r; s^0g$ feature $\ $, and reward r, LSVI learns a Q function backward, *i.e.*, at step h via $\hat{w}_h = \arg\min_w \sum_{s;a;s^0}(w^{>}\ \ (s; a) \quad \hat{V}_{h+1}(s^0))^2 + \ \ kwk^2$ and sets $\hat{V}_h(s) = \max_a(r(s; a) + \hat{w}_h^{>}\ \ (s; a)); 8s$. UCB, short for Upper Confidence Bound, refers to an exploration bonus added to basic LSVI.

3.2. Main Result

In this section, we prove our main transfer learning result, which shows that REPTRANSFER achieves near optimal regret in the target task with nice pre-training sample complexity in the source tasks. Our main result requires two assumptions. First, we need to ensure that EPS can successfully discover an exploratory policy in the source tasks, *i.e.*, there should exist a policy that non-trivially reaches the whole $\mathbb{R}^d$ in the feature space. Without exploratory policies in the source tasks, it may be possible that the optimal target policy visits subspaces unexplorable in any source task, in which case, pre-training will not have any benefits.

Assumption 3.2 (Reachability in source tasks) *There exists a* $\quad 2\ \mathbb{R}_+$ *such that for all* $k\ 2$ $[K]; h\ 2\ [0 : H \quad 1]$, *there exists a policy* $\quad$ *such that* $\min\left(\mathbb{E}_{;P_k^?}[\ {}_h^?(s_h; a_h)\ {}_h^?(s_h; a_h)^>]\right) \quad .$

Note that this low-rank reachability assumption generalizes the reachability assumption in latent variable and block MDPs, e.g. (Modi et al., 2021; Misra et al., 2020).

Second, For the MLE in Eq. (2) to succeed, we need to assume the standard realizability assumption, which is made in almost all prior works in low-rank MDPs.

Assumption 3.3 (Realizability) *For any source task* $k\ 2\ [K]$ *and any* $h\ 2\ [H]$, ${}_h^?\ 2 \quad$ *and* ${}_{k;h}^?\ 2 \quad {}_k$. *For normalization, we assume that for all* $\quad 2 \quad ; \quad 2 \quad {}_k; g : S \ ! \ [0; 1]$, *we have* $k\ (s; a)k_2 \quad 1$ *and* $\left\| \int g(s)\mathrm{d}\ {}_h^?(s) \right\|_2 \quad \overline{pd}.$

This leads to our main theorem.

Theorem 3 (Regret under generative source access) *Suppose Assumptions 2.1,2.2,3.1,3.2,3.3 hold, and fix any* $\quad 2\ (0; 1)$. *Then, running* REPTRANSFER *with policies from* EPS *(parameters set as in Lemma 3.1) has regret in the target task of* $\widetilde{O}\left(\quad H^2 d^{1:5} \sqrt{T \log(1= \ )} \right)$, *with at most* $\widetilde{O}\left(A^4 \ {}^3_{\max} d^5 H^7 K^2 T \quad {}^2(\log(j \ \ j= \ ) + K \log j \ \ j)\right)$ *generative accesses per source task.*

Remarkably, Theorem 3 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 $\quad$ factor that depends on the linear span coefficients and captures the intrinsic hardness of transfer learning. For special cases such as convex combination, i.e., $\quad$ is state-independent and ${}_h\ 2 \quad (K)$, then $\quad = 1$. In the worst-case, some dependence on the scale of $\quad$ seems unavoidable as we can have a state s^0 such that ${}_{\text{target}}(s^0) = 1$ and ${}_k(s^0) \quad 1$ with ${}_k(s^0) \quad 1$. 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 $j \ {}_k(s^0)j$. Obtaining formal lower bounds that capture a matching dependence on structural properties of $\quad$ is an interesting question for future research.

3.3. Proof Sketch

The proof can be broken down into three parts. First, under reachability, we show in Lemma 3.1 that EPS can indeed identify an exploratory policy. Second, we show in Lemma 3.2 that our novel cross-sampling procedure with MLE can learn a representation that linearly approximates $P^?_{\text{target}}$, in an average-case sense. Third, we prove that even under average-case misspecification, LSVI-UCB succeeds with low regret. We start by showing that EPS can identify an exploratory policy.

Lemma 3.1 (Source task exploration) *Suppose Assumptions 3.2,3.3 hold. Then, for any* $\in (0;1)$, *w.p.* 1 , *running* EPS *in any source task with* $N_{\text{LSVI-UCB}} = \widetilde{}(A^3d^6H^8 \ ^2)$ *and* $N_{\text{REWARDFREE}} = \widetilde{O}\left(A^3d^4H^6 \log\left(\frac{i \ jj \ j}{}\right) N^2_{\text{LSVI-UCB}}\right)$ *returns a* $_{\min}$*-exploratory policy where* $_{\min} = \widetilde{}(A \ ^3d \ ^5H \ ^7 \ ^2)$*: The sample complexity in the source task is* $N_{\text{REWARDFREE}}$ *episodes.*

To the best of our knowledge, Lemma 3.1 is the first result that finds an exploratory policy in low-rank MDPs, and might be of independent interest. Wagenmaker et al. (2022) recently obtained a related guarantee in the linear MDP setting with known features $^?$. Cheng et al. (2022, REFUEL) is also a reward-free modification of Rep-UCB, but the algorithm proceeds jointly over all tasks while we run REWARDFREE REP-UCB in each task independently. We note that REFUEL involves optimizing the Pseudo-Cumulative Value Function (PCV), which may be computationally hard in the planning step. Our REWARDFREE REP-UCB's planning step is the same as Rep-UCB (i.e., planning in a known linear MDP model), and is computationally efficient. We also remark that this step of identifying exploratory policies is modular and one could also directly use the reward-free algorithm FLAMBE (Agarwal et al., 2020), despite having a worse sample complexity in source.

We now analyze our novel cross-sampling procedure using the MLE generalization analysis of Agarwal et al. (2020). Under realizability, running multi-task MLE in (2) with these datasets satisfies the following w.p. at least 1 ,

$$\sum_{i;j \in [K]} \mathsf{E}_{\ ij;h} \left\| \widehat{}_h(s;a)^\top \widehat{}_{k;h}(\ ) \quad {}^?_h(s;a)^\top \ {}^?_{k;h}(\ ) \right\|^2_{TV} \quad _N := O((\log(j \ \models \ ) + K \log(j \ j))\models_N); \quad (3)$$

where $k \ k_{TV}$ denotes the total variation (TV) norm, and $_{ij;h}$ is the distribution from which we sampled $D_{ij;h}$. That is, $s;a \ _{ij;h}$ is equivalent to $(\widetilde{s};\widetilde{a}) \ d^{\ i}_{i;h \ 1};s \ P^?_{j;h \ 1}(\widetilde{s};\widetilde{a});a \ \text{unif}(A)$. Then, by the one-step back lemma (Lemma F.2, which is valid due to the low-rank structure of the target), followed by the linear span assumption (Assumption 2.2), we can prove the following lemma.

Lemma 3.2 (Target model error) *Suppose Assumption 2.2 holds and* $_k$ *is* $_{\min}$*-exploratory for each source task* k. *For any* $\in (0;1)$, *w.p.* 1 , $8\, h \in [0:H \ \ 1]$, $9\, \widetilde{}_h : S \ ! \ \mathsf{R}^d$ *such that*

$$\sup \mathsf{E}_{\ ;P^?_{target}} \left\| \widehat{}_h(s_h;a_h)^\top \widetilde{}_h(\ ) \quad {}^?_h(s_h;a_h)^\top \ {}^?_{target;h}(\ ) \right\|_{TV} \quad "_{TV} := \sqrt{jAj \ ^3_{\max} K \ _n =} \ _{\min}; \quad (4)$$

and, for any function $g : S \ ! \ [0;1]$, $k \int g(s) \mathrm{d}\widetilde{}_h(s) k_2 \quad p_{\overline{d}}$.

Lemma 3.2 implies that the learned $\widehat{}$ is a feature such that $P^?_{\text{target}}$ is *approximately linear* in $\widehat{}$, under the occupancy distribution induced by *any* policy. Remarkably, this guarantee holds before the agent has ever interacted with the target task! Intuitively, this is because cross-sampling ensures that our training data contains all possible states that can be encountered in the target task. Failure modes without this can be found in the discussion following Theorem 6.

The final step is to show that the deployment phase, which runs LSVI-UCB in an approximately linear MDP of the target task, achieves low regret. Note that we face an approximately linear MDP, as Lemma 3.2 shows, due to the use of learned features $\widehat{}$, even though $P^?_{\text{target}}$ is linear in the unknown features $^?$. Online learning in approximately linear MDPs has been studied in Jin et al. (2020b), but under a much stronger $`_1$ error bound. Instead, we work under the weaker, and more realistic, average-case misspecification in Eq. (4). Indeed, it is possible that some states are unlikely to be

visited by any policy, so we should not impose strong misspecification restrictions on these parts of the state space. We now state our novel LSVI-UCB regret bound.

Theorem 4 (LSVI-UCB under average-misspecification) *Under Eq. (4), for any* $2\ (0;1)$*, w.p.* 1 *,* LSVI-UCB *in the deployment phase has regret* $\widetilde{O}\left(dH^2T\varepsilon_{TV} + d^{1:5}H^2\sqrt{T\log(1=)}\right)$*:*

The key step in proving Theorem 4 is showing *almost-optimism under the occupancy distribution of the optimal policy* $^?$. As a technical remark, we also employ a novel trajectory-wise indicator to deal with the clipping of the value estimates. Note the in the leading term comes from the scaling of the $\widetilde{}$ in Lemma 3.2. The full proof and general LSVI-UCB result is in Theorem 15. To the best of our knowledge, this is the strongest result for learning in an approximately linear MDP, which may be of independent interest.

We arrive at the final regret bound by collecting enough samples in the source tasks to make $\varepsilon_{TV} = 1=\sqrt{T}$, which makes the first linear-in-T term lower order. This gives the REPTRANSFER guarantee. Note that the guarantee holds independent of the mechanism used for obtain exploratory policies in the source tasks.

Theorem 5 (REPTRANSFER) *Suppose Assumptions 2.1,2.2,3.1,3.3, and* $_k$ *is* $_{\min}$*-exploratory for each source task* k*. Then, for any* $2\ (0;1)$*, w.p.* 1 *,* REPTRANSFER *when deployed in the target task has regret at most* $\widetilde{O}\left(H^2d^{1:5}\sqrt{T\log(1=)}\right)$*, with at most* Kn *generative accesses per source task, with* $n = O\left(\frac{1}{\min}A \frac{3}{\max}KT\left(\log\frac{j\ j}{} + K\log j\ \ j\right)\right)$.

Combining with the $_{\min}$ specified in Lemma 3.1, we conclude the proof sketch.

4. Failure of transfer learning without generative access to source tasks

In the previous section, we show that efficient transfer learning is possible under very weak structural assumptions, but requires generative access to the source tasks. One natural question is whether transfer learning is possible with only online access to the source tasks. Somewhat surprisingly, we show that this is impossible without significantly stronger assumption.

Theorem 6 (Lower bound for online access to source tasks) *Let* $M = f(P_1^?;:::;P_K^?;P_{target}^?)g$ *be a set of* $K + 1$ *tasks that satisfies (1) all tasks are Block MDPs; (2) all tasks satisfy Assumption 3.2 and Assumption 2.2; (3) the latent dynamics are exactly the same for all source and target tasks. For any pre-training algorithm* A *which outputs a feature* $\hat{}$ *by interacting with the source tasks* $k\ 2\ [K]$*, there exists* $(P_1^?;:::;P_K^?;P_{target}^?)\ 2\ M$*, such that with probability at least* $1=2$*,* A *will output a feature* $\hat{}$*, such that for any policy taking the functional form of* $(s) = f\left(f\hat{}(s;a)g_{a2A};fr(s;a)g_{a2A}\right)$*, we have* $V_{target}^? \quad V_{target} \quad 1=2$*:*

Here, the particular functional form f is defined so that the policy cannot distinguish between two state-action pair with the same feature embedding. Theorem 6 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. In a Block MDP, for any $(s;a)$, we can model the ground-truth $^?$ as a one-hot encoding $e_{(z;a)}$ 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 6 can be found in Section D. 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 a permutation to happen.

4.1. Representational Transfer under Observational Reachability

To complement the impossibility result, we next show that under an additional assumption on the reachability of raw states, a slight variant of the same algorithm (Algorithm 4 in Section G) can achieve the same regret with only online access to the source tasks. The main difference in Algorithm 4 is that it performs sampling directly from the occupancy distribution of $\ _{k}$ in source task k (in an online, episodic manner without needing generative access) instead of the cross-sampling procedure used in Algorithm 2.

Assumption 4.1 (Reachability in the raw states) *For all source tasks $k \ 2 \ [K]$, any policy $\ $ and $h \ 2 \ [0 : H \ \ 1]$, we have $\inf_{s2S;a2A} d_{k;h}(s; a) \ \ _{raw} \ \min\Big(\mathsf{E}_{\ ;P_k^{?}}\big[\ _h^{?}(s_h; a_h) \ \ _h^{?}(s_h; a_h)^{>}\big]\Big)$.*

Assumption 4.1 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 that the target task distribution has bounded density. This is true, for example, when S is discrete.

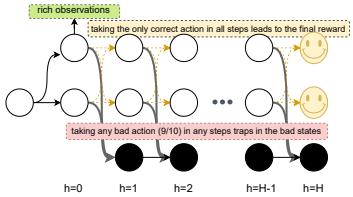
Assumption 4.2 (Bounded density) *For all $(\ \ ; h; s; a)$, we have $d_{target;h}(s; a) \ \ 1$.*

Theorem 7 (Regret with online access) *Suppose Assumptions 2.1-2.2,4.1,4.2 hold. W.p. $1 \ \ $, Algorithm 4 with appropriate parameters achieves a regret in the target $\widetilde{O}\Big(\ \ d^{1:5}H^2\sqrt{T \log(1= \ )}\Big)$, with $\mathrm{poly}\big(A; \ \ _{\max}; d; H; K; T; \ \ ^{1}; \ \ _{raw}^{1}; \log(j \ \ jj \ \ j= \ )\big)$ online queries in the source tasks.*

Assumption 4.1 is satisfied in a Block MDP, when, for example, the emission function $o(sjz)$ satisfies that $8s; 9z;$ s.t., $o(sjz) \ \ c$. That is, for any source task, any state in the state space can be generated by at least one latent state. However, we believe such a covering condition is generally too strong to hold in practice. Furthermore, the parameter $\ _{raw}$ will typically scale with the number of observed states, which we expect to be prohibitively large in most interesting problems, and view the this result has mainly to quantify the degree of applicability of Theorem 6.

5. Experiments

In this section we empirically study the following questions: i) the effectiveness of pretraining with REPTRANSFER, under the linear span and feature/observation coverage assumptions. ii) the hardness of representational transfer under the linear span assumption without the generative model access. Our experiments are under the Block MDP setting, with the challenging Rich Observation Combination Lock (comblock) benchmark (Fig. 1(a)). We design two sets of experiments to investigate the above questions respectively. We defer the details of the experiments in Appendix J.



(a)

	Source	O-REPTRANSFER	G-REPTRANSFER	Oracle	Target
Observational and Feature Coverage	*1*	8006 (294.7)	7790 (267.6)	7048 (164.8)	181450 (147600.2)
Feature Coverage Only	*1*	*1*	7764 (145.7)	7336 (270.7)	85000 (14469.8)

(b)

Figure 1: **(a):** A visualization of the rich observation comblock environment (see Appendix J.1 for details). **(b) Top:** Number of episodes required to solve the target environment under the observational coverage setting. **(b) Bottom:** Number of episodes required to solve the target environment under the feature coverage setting. 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. *1* denotes that an algorithm can not solve the target task within a fixed sample budget. The sample efficiency of REPTRANSFER under feature and observational coverage verifies the benefit of representational transfer, and the failure of O-REPTRANSFER without observational coverage suggests the necessity of generative assumption during representational transfer. In Figure 2 in the appendix, we further provide the visualization of the representations that is learned in both settings.

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.

Effectiveness of REPTRANSFER. We first analyze the how representational transfer benefits where both feature coverage (Theorem 3) and observational coverage (Theorem 7) assumptions are met. We use 5 source tasks (horizon $H = 25$), with different latent dynamics. To ensure linear span assumption (Assumption 2.2), for each timestep h, we make the target latent transition dynamics from one of the sources uniformly at random. For the coverage assumptions, note that for comblock, the feature coverage assumption (Assumption 3.2) is always satisfied. We also guarantee the observational coverage (Assumption 4.1) by equipping all environments with the same emission distribution on a compact observational space. 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 4.1 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). 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**.

Hardness without the generative access. In this section, following the intuition of our lower bound (Theorem 6), 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 the latent coverage (Assumption 2.2) holds while observational coverage (Assumption 4.1) fails. So we expect that an algorithm without generative access to source tasks will fail based on Theorem 6. 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. **This ablation verifies that source generative model access is needed without the observational coverage**.

6. Conclusion

We study representational transfer among low rank MDPs which share the same unknown representation. Under a reasonably flexible linear span task relatedness assumption, we propose an algorithm that provably transfers the representation learned from source tasks to the target task. The regret in target task matches the bound obtained with oracle access to the true representation, using only polynomial number of samples from source tasks. Our approach relies on the generative model access in source tasks, which we prove is not avoidable in the worst case under the linear span assumption. To complement the lower bound, we propose a stronger assumption on the conditions of the reachability in raw states, under which online access to source tasks suffices for provably efficient representation transfer. Finding modalities other than generative access which avoid the lower bound, and a more extensive empirical evaluation beyond the proof-of-concept experiments here are important directions for future research

Acknowledgements: This material is based upon work supported by the National Science Foundation under Grant No. IIS-2154711.

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. In *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.

André Barreto, Will Dabney, Rémi Munos, Jonathan J Hunt, Tom Schaul, Hado P van Hasselt, and David Silver. Successor features for transfer in reinforcement learning. *Advances in neural information processing systems*, 30, 2017.

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.

Léonard Blier, Corentin Tallec, and Yann Ollivier. Learning successor states and goal-dependent values: A mathematical viewpoint. *arXiv preprint arXiv:2101.07123*, 2021.

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.

Peter Dayan. Improving generalization for temporal difference learning: The successor representation. *Neural computation*, 5(4):613–624, 1993.

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. *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.

Baihe Huang, Jason D. Lee, Zhaoran Wang, and Zhuoran Yang. Provably efficient multi-task reinforcement learning in large state spaces, 2023. URL https://openreview.net/forum?id=p6wiThIOS5m.

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 efficient 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 finite set of models. *Advances in Neural Information Processing Systems*, 26, 2013.

Erwan Lecarpentier, David Abel, Kavosh Asadi, Yuu Jinnai, Emmanuel Rachelson, and Michael L Littman. Lipschitz lifelong reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 8270–8278, 2021.

Jonathan Lee, Aldo Pacchiano, Vidya Muthukumar, Weihao Kong, and Emma Brunskill. Online model selection for reinforcement learning with function approximation. In *International Conference on Artificial 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 benefit 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 efficient rich-observation reinforcement learning. *arXiv preprint arXiv:1911.05815*, 2019.

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

Aditya Modi, Nan Jiang, Ambuj Tewari, and Satinder Singh. Sample complexity of reinforcement learning using linearly combined model ensembles. In *International Conference on Artificial 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.

Emilio Parisotto, Jimmy Lei Ba, and Ruslan Salakhutdinov. Actor-mimic: Deep multitask and transfer reinforcement learning. *arXiv preprint arXiv:1511.06342*, 2015.

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.

Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016.

Ahmed Touati and Yann Ollivier. Learning one representation to optimize all rewards. *Advances in Neural Information Processing Systems*, 34:13–23, 2021.

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 offline 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.

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 efficient 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. Efficient reinforcement learning in block mdps: A model-free representation learning approach. *arXiv preprint arXiv:2202.00063*, 2022.

Zhuangdi Zhu, Kaixiang Lin, and Jiayu Zhou. Transfer learning in deep reinforcement learning: A survey. *arXiv preprint arXiv:2009.07888*, 2020.

Appendices

Appendix A. Notations

Table 2: List of Notations

$S; A; A$	State and action spaces, and $A = jAj$.
(S)	The set of distributions supported by S.
$_{\min}(A)$	Smallest eigenvalue of matrix A.
e_j	One-hot encoding of j, i.e. 0 at each index except the one corresponding to j.
	Length of vector implied from context.
$(x)\ _y$	$\min fx; yg$.
H	Episode length of MDPs, a.k.a. time horizon. We index steps as $h = 0; 1; :::; H \quad 1$.
K	The number of source tasks.
d	dimension of the low-rank MDP, i.e. dimension of $^?$.
$P^?_{k;h}$	Ground truth transition at time h for source task k.
$P^?_{\text{target};h}$	Ground truth transition at time h for target task.
$r_{\text{target};h}$	Reward function of the target task.
$\mathsf{E}\ _{;P}[\,]$	Expectation under the distribution of trajectories when $\ $ is executed in P.
	We sometimes omit P when the MDP is clear from context.
$d_{P;h}$	Occupancy distribution of $\ $ under transitions P at time h.
$d_{k;h}$	Occupancy distribution for the k-th task, i.e. $d_{P^?_k;h}$.
$^?_h(s; a)$	Embedding function for $(s; a)$ at time h.
$_h$	Realizable function class for $^?_h$.
$j\ \ j$	Defined as $\max_h j\ \ _h j$.
$^?_{k;h}(s^0)$	Emission embedding function for s^0 at time h for environment k.
$_{k;h}$	Realizable function class for $^?_{k;h}$.
$j\ \ j$	Defined as $\max_{k;h} j\ \ _{k;h} j$.
$_{\max}$	$\max_{h;k;s^0 2S} j\ \ _{k;h}(s^0)j$ (based on Assumption 2.2).
	$\max_h \sum_{k=1}^K \max_{s^0 2S} j\ \ _{k;h}(s^0)j$ (based on Assumption 2.2).
	Feature reachability in the source task (Assumption 3.2).
$_{raw}$	Raw states reachability parameter (Assumption 4.1)

Appendix B. Omitted Algorithms

Algorithm 3 REWARDFREE REP-UCB

Input: Regularizer $\_n$, bonus scaling $\_n$, model class $M = \quad$, number of episodes N.
Initialize $\widehat{\ }_0$ as random and $D_{h;0}; D^0_{h;0} = ;$.
for episode $n = 1; 2; :::; N$ **do**
  Data collection from $\widehat{\ }_{n\ 1}$: for $h = 1; 2; :::; H\ \ 1$,

$$s\quad \widehat{d^{n\ 1}_h}; a\quad \mathrm{Unif}(A); s^0\quad P^?_h(s; a);$$
$$\tilde{s}\quad \widehat{d^{n\ 1}_{h\ 1}}; \tilde{a}\quad \mathrm{Unif}(A); \tilde{s}^0\quad P^?_{h\ 1}(\tilde{s}; \tilde{a}); \tilde{a}^0\quad \mathrm{Unif}(A); \tilde{s}^{00}\quad P^?_h(\tilde{s}^0; \tilde{a}^0);$$
$$D_{h;n} = D_{h;n\ 1} [\{(s; a; s^0)\}; D^0_{h;n} = D^0_{h;n\ 1} [\{(\tilde{s}^0; \tilde{a}^0; \tilde{s}^{00})\}:$$

  For $h = 0$, only collect $D_{0;n}$.
  Learn model via MLE: for all $h = 0; 1; :::; H\ \ 1$,

$$\widehat{P}_{h;n} = (\widehat{\ }_{h;n}; \widehat{\ }_{h;n}) = \operatorname*{argmax}_{\ h;\ \ h 2 M_h} \mathrm{E}_{D_{h;n}[D^0_{h;n}} \left[\log\ \ _h(s; a)^T\ \ _h(s^0)\right]:$$

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

$$\widehat{b}_{h;n}(s; a) = \ \ _n \left\| \widehat{\ }_{h;n}(s; a) \right\|_{\widehat{\ }^{\ 1}_{h;n}}$$
$$\widehat{\ }_{h;n} = \sum_{(s;a;\_) 2 D_{h;n}} \widehat{\ }_{h;n}(s; a) \widehat{\ }_{h;n}(s; a)^T + \ \ _n I:$$

  Learn policy $\widehat{\ }_n = \mathrm{argmax}\ \ V_{\widehat{P}_n; \widehat{b}_n}$ and let $\widehat{V}_n$ be its value.
Let $\widehat{n} = \mathrm{argmin}_{n\ \ N=2} \widehat{V}_n$.
Output: $\widehat{n}; \widehat{P}_{\widehat{n}}$.

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

$$\widehat{}_h; \widehat{}_{1:K} = \underset{2 \; ; \; k2}{\text{argmax}} \sum_{k2[K]} {}_k \mathsf{E}_{D_{k;h}} \left[\log \quad (s; a)^{>} \quad {}_k(s^0) \right]:$$

DEPLOYMENT PHASE

Additional Input: number of deployment episodes T.

1: Set $= H \sqrt{d} + dH \sqrt{\log(dHT=)}$.
2: Run LSVI-UCB$\left(f \widehat{}_h g_{h=0}^{H} {}^{1}; r = r_K; T; \right)$ in the target task $P^?_{\text{target}}$ (Algorithm 5).

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 5 LSVI-UCB

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

$$_{h;e} = \sum_{k=1}^{e\ \ 1} \widehat{\ }_h(s^k_h; a^k_h)\,\widehat{\ }_h(s^k_h; a^k_h)^{>} + I;$$

$$\widehat{w}_{h;e} = \ \ {}^{1}_{h;e} \sum_{k=1}^{e\ \ 1} \widehat{\ }_h(s^k_h; a^k_h)\,\widehat{V}_{h+1;e}(s^k_{h+1}):$$

    Set bonus and value functions,

$$b_{h;e}(s; a) = \left\| \widehat{\ }_h(s; a) \right\|_{\ \ {}^{1}_{h;e}};$$

$$\widehat{Q}_{h;e}(s; a) = \widehat{w}^{>}_{h;e}\,\widehat{\ }_h(s; a) + r_h(s; a) + \ \ b_{h;e}(s; a);$$

$$\widehat{V}_{h;e}(s) = \left(\max_a \left\{ \widehat{Q}_{h;e}(s; a) \right\}\right)_{\ \ M_V}:$$

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

Appendix C. More discussion on related works

C.1. Empirical works in transfer learning

The idea of learning transferable representation has been extensively explored in the empirical literature. Here we don't intend to provide a comprehensive survey of all existing works on this topic. Instead, we discuss a few representative approach that may be of interest.

Towards transfer learning across different environments, progressive neural network (Rusu et al., 2016) is among the first neural-based attempt to learning a transferable representation for a sequence of downstream tasks that tries to overcome the challenge of catastrophic forgetting. It maintains the learned neural models for all previous tasks and introduce additional connections between the network of the current tasks to those of prior tasks to allow information reuse. However, a drawback common to such an approach is that the network size grows linearly with the number of tasks. Other approaches include directly learning a multi-task policy that can perform well on a set of source tasks, with the hope that it will generalize to future tasks (Parisotto et al., 2015). Such an approach requires the tasks to be similar in their optimal policy, which is a much stronger assumption than ours.

Slightly off-topic are the works about "transfer learning" inside the same environment but across different reward functions, which is more restricted than the setting considered in this paper. Several prior works design representation learning algorithms that aim to learn a representation that generalize across multiple reward function/goals (Dayan, 1993; Barreto et al., 2017; Touati and Ollivier, 2021; Blier et al., 2021). These are related to the REWARDFREE REP-UCB we developped in Section E. The key difference is that we concern representation learning along with efficient exploration to derive an end-to-end polynomial sample complexity bound. These prior works do not consider exploration and do not come with provable sample complexity bounds. We refer interested readers to a recent survey (Zhu et al., 2020) for a comprehensive discussion of other empirical approaches.

C.2. Comparison to 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 2.1).
2. task diversity (similar to our Assumption 2.2).
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 $;\ ^{0}2$ must be full rank. In contrast, our analysis only requires coverage in the true feature $^{?}$ 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 in the sense of linear-transform equivalence. Two representation functions and ^{0} 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 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.

C.3. Comparison to Cheng et al. (2022)

Cheng et al. (2022) studies representational transfer in low-rank MDPs, with not only a weaker notion of task relatedness (with global coefficients in the linear span) but also stronger assumptions. Particularly, we restate the following strong assumption from Cheng et al. (2022, Assumption 5.3).

Assumption C.1 *For any two different models in the model class , say* $P^1(s^0|s;a) = h\ ^1(s;a);\ ^1(s^0)i$ *and* $P^2(s^0|s;a) = h\ ^2(s;a);\ ^2(s^0)i$*, there exists a constant* C_R *such that for all* $(s;a)\ 2\ S\quad A$ *and* $h\ 2\ [H]$*,*

$$kP^1(\ |s;a)\quad P^2(\ |s;a)k_{TV}\quad C_R\mathsf{E}_{(s;a)\ \ U(S;A)}kP^1(\ |s;a)\quad P^2(\ |s;a)k_{TV};$$

where U *is the uniform distribution.*

This assumption ensures that the point-wise TV error is bounded, as long as the population-level TV error is bounded. Cheng et al. (2022) used this 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 $|S|$. In contrast, our analysis (Lemma G.1) shows that this assumption is in fact not necessary, even 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 additional assumptions. Comparing to (Cheng et al., 2022), we also further implement and perform experimental evaluations of our algorithm.

C.4. Works in multi-task learning

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. Several recent works study multi-task linear bandits with linear representations ($(s) = A s$ with unknown A) (Hu et al., 2021; Yang et al., 2020, 2022). The techniques developed in these works crucially rely on the linear 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) study multi-task RL under a fixed distribution over finitely many MDPs, while Brunskill and Li (2014) consider transfer in semi-MDPs by learning options. Lecarpentier et al. (2021) consider lifelong learning in Lipschitz MDP. 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 4. 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.

Efficient Representation Learning in RL. Even in the single task setting, efficient representation learning is an active area witnessing 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 sparse models (Hao et al., 2021a,b).

Appendix D. Impossibility Results

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

Theorem 8 (Counter-example in supervised learning) *Assume that we want to perform conditional density estimation, where* $P_k^?(y|x) = \ ^?(x)^> \ ^?_k(y)$. *Under Assumption 2.1 (shared representation) and Assumption 2.2 (linear span), and assume that in each source task, one have access to a data generating distribution* $_k(x)$ *such that* $_{\min}\mathsf{E}_{\ k}[\ ^?(x)\ \ ^?(x)^>]$ *(reachability). No algorithm can consistently achieve* $\mathsf{E}_{\ \kappa}[k\hat{P}^?_{target}(y|x) \quad P^?_{target}(y|x)k_{TV}] \quad 1=2$ *on the target task using the feature learned from the source tasks with probability more than* $1=2$.

Proof [Proof of Theorem 8] Consider the following example. $X = \mathsf{R}^2$ 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 that $P_k^?(x)$ are uniform distribution on S_k.

Suppose that the feature class $\ $ only contains two functions:

$$_1 : fx_1 \quad 0 \ \& \ x_2 \quad x_1g \ ! \ (1;0), \text{ and } (0;1) \text{ otherwise}$$
$$_2 : fx_2 \quad 0 \ \& \ x_1 \quad x_2g \ ! \ (0;1), \text{ and } (1;0) \text{ otherwise}$$

That is, the feature maps from R^2 to the set of binary encoding of dimension 2, i.e. $f(1;0);(0;1)g$. We further assume that $\ ^?_k = (p_1(y);p_2(y))$ for some distributions $p_1;p_2$, which is identical for all task k, where $kp_1;p_2k_{TV} = 1$. We also assume that $\ ^?_k$ is known to the learner a prior, i.e. $_k = f\ ^?_kg$ for all $k \ 2 \ [K]$, so all the learner needs to do is to pick the correct $\ $ out of two candidates.

Given the above setup, it's easy to verify that both Assumption 2.1 and Assumption 2.2 are satisfied, because the decision boundary of both $_1$ and $_2$ passes through the support of the source tasks, and all $\ ^?_k$'s are identical. However, $_1$ and $_2$ 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 $_1$ is the true feature and the algorithm incorrectly chooses $_2$. Then, for $x \ 2 \ S_3 \bigcap fx_1 \quad 0g$ 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 $\hat{P}_{\text{target}}$ and $P^?_{\text{target}}$ is $1=2$. ■

The above construction shows that our assumption are not sufficient 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 on $(s; a)$ is not independent from the conditional density $P(s^0 | s; a)$ we desire to estimate. In particular, if one collects data in the source tasks in an online fashion via running a policy, $(s; a)$ is structurally restricted to be an occupancy distribution generated by the ground-truth transition $P^?(s^0 | s; a)$. 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 d_0, which would suffer from similar issues as the supervised learning setting, but is not necessary for sample-efficient RL.

Next, we prove the impossibility result in Theorem 6, restated below as Theorem 10. This result shows that one can not achieve online learning in the source tasks without significantly stronger assumptions such as Assumption 4.1. Before that, we provide a preliminary version, showing that the learned $\hat{}$ is not sufficient to fit the transition model in the target task, which motivates the construction in Theorem 10.

Theorem 9 (Impossibility Result: Model Learning) *Let* $M = \{P_1; :::; P_K; P_{target}\}$ *be a set of* $K + 1$ *tasks that satisfies*

1. *all tasks are Block MDPs;*
2. *all tasks satisfy Assumption 3.2 and Assumption 2.2;*
3. *the latent dynamics are exactly the same for all source and target tasks.*

For any pre-training algorithm A*, there exists* $(P_1; :::; P_K; P_{target}) \in M$ *and an occupancy distribution* $_{target}$ *on the target task, such that with probability at least* $1=2$*,* A *will output a feature* $\hat{}$ *and for any*

$$\mathbb{E}_{_{target}} k\hat{}(s; a)^{>} \quad ( ) \quad P^?_{target}(\, | s; a)k_{TV} \quad 1=2:$$

Proof [Proof of Theorem 9] Consider a tabular MDP with 2 latent states $z_1; z_2$ and an observation state space $S = R_1 \bigcup R_2 \bigcup B_1 \bigcup B_2$, where in task 1 one can only observe $R_1 \bigcup R_2$ and in task 2 one can only observe $B_1 \bigcup B_2$. Correspondingly, $o_1(s|z)$ is only supported on $R_1 \bigcup R_2$ (i.e., $o_1(R_i|z_i) = 1$) and similar for task 2. Let the latent state transition be such that $P(z_1|z_1; a) = 1$ and $P(z_2|z_2; a) = 1$, i.e. only self-transition regardless of the actions.

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

$$_1 = \{R_1 \ ! \ 1; R_2 \ ! \ 2; B_1 \ ! \ 1; B_2 \ ! \ 2\}$$
$$_2 = \{R_1 \ ! \ 1; R_2 \ ! \ 2; B_1 \ ! \ 2; B_2 \ ! \ 1\}$$

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

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., $^\wedge = \ _2$). Let task 3 be such that any state emits to $R_1 \bigcup R_2$ and $B_1 \bigcup B_2$ each with probability 1=2 (i.e., $o_3(R_ijz_i) = o_3(B_ijz_i) = 0{:}5$). This construction satisfies Assumption 3.2 and Assumption 2.2.

Then, within task 3, one would encounter observations from both R_1 and 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 $_{\text{target}}(z_1) = \ _{\text{target}}(z_2) = 1{=}2$, then

$$
\begin{aligned}
&\mathsf{E}_{\ _{\text{target}}}[k^\wedge(s;a)^{>}\ (\ )\quad P^{?}(\ js;a)k_{TV}] \\
&= \frac{1}{4}k^\wedge(R_1)^{>}\ (\ )\quad (R_1)^{?>}\ ^?(\ )k_{TV} + \frac{1}{4}k^\wedge(B_1)^{>}\ (\ )\quad (B_1)^{?>}\ ^?(\ )k_{TV} \\
&\quad + \frac{1}{4}k^\wedge(R_2)^{>}\ (\ )\quad (R_2)^{?>}\ ^?(\ )k_{TV} + \frac{1}{4}k^\wedge(B_2)^{>}\ (\ )\quad (B_2)^{?>}\ ^?(\ )k_{TV} \\
&= ko_1\quad o_1^?k_{TV}{=}4 + ko_2\quad o_1^?k_{TV}{=}4 + ko_2\quad o_2^?k_{TV}{=}4 + ko_1\quad o_2^?k_{TV}{=}4 \\
&\quad \frac{1}{4}ko_1^?\quad o_2^?k_{TV} + \frac{1}{4}ko_1^?\quad o_2^?k_{TV} \\
&= \frac{1}{2};
\end{aligned}
$$

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 $kp_1^?\quad p_2^?k_{TV} = 1$. ■

Now, we are ready to restate and prove Theorem 6.

Theorem 10 (Impossibility Result: Optimal Policy Identification) *Let* $M = f(P_1;::::;P_K;P_{target})g$ *be a set of* $K + 1$ *tasks that satisfies*

1. *all tasks are Block MDPs;*
2. *all tasks satisfy Assumption 3.2 and Assumption 2.2;*
3. *the latent dynamics are exactly the same for all source and target tasks.*

For any pre-training algorithm A*, there exists* $(P_1;::::;P_K;P_{target})\ 2\ M$*, such that with probability at least* $1{=}2$*,* A *will output a feature* $^\wedge$*, such that for any policy taking the functional form of* $(s) = f\left(f^\wedge(s;a)g_{a2A};fr(s;a)g_{a2A}\right)$*, we have*

$$V^?\quad V\qquad 1{=}2{:}$$

Proof [Proof of Theorem 10] 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 $S = R_1 \bigcup R_2 \bigcup B_1 \bigcup B_2$, where in task 1 one can only observe $R_1 \bigcup R_2$ and in task 2 one can only observe $B_1 \bigcup B_2$. Correspondingly, $o_1(sjz)$ is only supported on $R_1 \bigcup R_2$ (i.e., $o_1(R_ijz_i) = 1$) and similar for task 2. Let the latent state transition be such that $P(z_3jz_1; a_1) = P(z_3jz_2; a_2) = 1$, and $P(z_4jz_1; a_2) = P(z_4jz_2; a_1) = 1$. 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 $= f_1; 2g$ for $h = 1$, such that

$$_1 = fR_1 ! 1; R_2 ! 2; B_1 ! 1; B_2 ! 2g$$
$$_2 = fR_1 ! 1; R_2 ! 2; B_1 ! 2; B_2 ! 1g$$

Denote $_i(s; a) = e_{(_i(s);a)}$ for $i 2 [1; 2]$. In addition, define $= f_1; 2g$ where

$$_1 = fz_3 ! (1; 0); z_4 ! (0; 1)g$$
$$_2 = fz_4 ! (1; 0); z_3 ! (0; 1)g$$

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., $^ = _2$). Let task 3 be such that any state emits to $R_1 \bigcup R_2$ and $B_1 \bigcup B_2$ each with probability 1=2 (i.e., $o_3(R_ijz_i) = o_3(B_ijz_i) = 0:5$). This construction satisfies Assumption 3.2 and Assumption 2.2.

Then, for any policy that only make decision based on $^(s; a)$ 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. ■

Theorem 9 and Theorem 10 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 4.1, 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 4.1 is necessary for online learning. We leave this as an important direction of future work.

Appendix E. 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 11 and (7). 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 $_n$ with that of the MLE, which is how we learn $\widehat{\ }$ and $\widehat{\ }$, as in Rep-UCB. Recall the notation of (Zhang et al., 2022),

$$
\begin{aligned}
{}_{h;n}(s;a) &= \frac{1}{n}\sum_{i=0}^{n}{}^{1}\widehat{d_h^{\,i}}(s)\,\mathrm{Unif}(a)\\
{}_{h;n}(s;a) &= \frac{1}{n}\sum_{i=0}^{n}{}^{1}\mathsf{E}_{\tilde{s}\ \widehat{d_h^{\,i}}\ {}_{1};\tilde{a}\ \ \mathrm{Unif}(A)}[P_h^{?}(s\,j\,\tilde{s};\tilde{a})\,\mathrm{Unif}(a)]\\
{}_{h;n}(s;a) &= \frac{1}{n}\sum_{i=0}^{n}{}^{1}\widehat{d_h^{\,i}}(s;a)\\
{}_{;\ ;n} &= n\mathsf{E}\left[\ (s;a)\ \ (s;a)^{T}\right] +\ \ {}_{n}I:
\end{aligned}
$$

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

$$
\max\left\{\mathsf{E}_{\ h;n}\left\|\widehat{P}_{h;n}(s;a)\ \ P_h^{?}(s;a)\right\|_{TV}^{2};\mathsf{E}_{\ h;n}\left\|\widehat{P}_{h;n}(s;a)\ \ P_h^{?}(s;a)\right\|_{TV}^{2}\right\}\ \ {}_{n}; \tag{5}
$$

where

$$
{}_{n} = O\left(\frac{\log(j\,M\,jnH=\ )}{n}\right);
$$

and $j\,M\,j = \max_{h2[H]}j\ \ {}_{h}jj\ \ {}_{h}j$. We also adopt the same choice of $\ {}_{n};\ \ {}_{n}$ parameters as BRIEE, which we assume from now on.

$$
\begin{aligned}
{}_{n} &= \ \ (d\log(j\,M\,jnH=\ ))\\
{}_{n} &= \ \ \left(\sqrt{njAj^{2}\ \ {}_{n}+\ \ {}_{n}d}\right):
\end{aligned}
$$

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

Assumption E.1 *For any* $h = 0; 1; :::; H \quad 1$, *we have* $\overset{?}{h} 2 \quad h$ *and* $\overset{?}{h} 2 \quad h$. *For any* $\quad 2$ $k \ (s; a)k_2 \quad 1$. *For all* $\quad 2 \quad h$ *and any function* $g : S \ ! \ \mathsf{R}$, *we have* $k \int_s g(s)\mathrm{d} \ (s)k_2 \quad kgk_1 \ p\frac{h}{d}$.

Lemma E.1 *Let* r *be any reward function. Suppose we ran Algorithm 3 with line 7 having reward* $r + \widehat{b}_n$ *instead of just* $\widehat{b}_n$. *Then, for any* $\quad 2 \ (0; 1)$, *w.p. at least* $1 \quad$, *we have*

$$\sum_{n=0}^{N \ 1} V_{\widehat{P}_n; r+\widehat{b}_n}^{\widehat{n}} \quad V_{P^{\overset{?}{}}; r}^{\widehat{n}} \quad O\left(H^2 d^2 jAj^{1:5} \sqrt{N \log(j M jN H =)}\right)$$

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 $n = 1; 2; :::; N$,

$$\begin{aligned} & V_{\widehat{P}_n; r+\widehat{b}_n}^{\widehat{n}} \quad V_{P^{\overset{?}{}}; r}^{\widehat{n}} \\ & . \sum_{h=0}^{H \ 2} \mathsf{E}_{\widetilde{s};\widetilde{a}} \ d_{P^{\overset{?}{}}; h}^{\widehat{n}} k \ \overset{?}{h}(\widetilde{s}; \widetilde{a})k \ {}_{h;n; \ \overset{?}{h}}^{1} \sqrt{jAj \ {}_n^2 d + \ {}_n d} + \sqrt{jAj \ {}_1^2 d=n} \\ & + (2H + 1) \sum_{h=0}^{H \ 2} \mathsf{E}_{\widetilde{s};\widetilde{a}} \ d_{P^{\overset{?}{}}; h}^{\widehat{n}} k \ \overset{?}{h}(\widetilde{s}; \widetilde{a})k \ {}_{h;n; \ \overset{?}{h}}^{1} \sqrt{njAj \ {}_n + \ {}_n d} + (2H + 1)\sqrt{jAj \ {}_n}: \end{aligned}$$

By elliptical potential arguments, we have

$$\sum_{n=0}^{N \ 1} \mathsf{E}_{\widetilde{s};\widetilde{a}} \ d_{P^{\overset{?}{}}; h}^{\widehat{n}} k \ \overset{?}{h}(\widetilde{s}; \widetilde{a})k \ {}_{h;n; \ \overset{?}{h}}^{1} \quad \sqrt{dN \log\left(1 + \frac{N}{d \ 1}\right)}:$$

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

$$\begin{aligned} & \sum_{n=0}^{N \ 1} V_{\widehat{P}_n; r+\widehat{b}_n}^{\widehat{n}} \quad V_{P^{\overset{?}{}}; r}^{\widehat{n}} \\ & . \sqrt{dN \log\left(1 + \frac{N}{d \ 1}\right)} \left(H \sqrt{jAj \ {}_N^2 d + \ {}_N d} + H^2 \sqrt{NjAj \ {}_N + \ {}_N d}\right) \\ & . \sqrt{dN \log\left(1 + \frac{N}{d \ 1}\right)} \left(H \sqrt{NjAj^3 \ {}_N d + \ {}_N d^2} + H^2 \sqrt{NjAj \ {}_N + \ {}_N d}\right) \\ & . \sqrt{dN \log\left(1 + \frac{N}{d \ 1}\right)} \left(H^2 \sqrt{djAj^3 \log(j M jN H =) + d^3 \log(j M jN H =)}\right) \\ & 2 \ O\left(H^2 d^2 jAj^{1:5} \sqrt{N \log(j M jN H =)}\right): \end{aligned}$$

■

This gives the following useful corollary for reward free exploration.

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

$$\widehat{V}_{\widehat{n}} \quad O\left(H^2 d^2 j A j^{1:5} \sqrt{\frac{\log(j M j N H =)}{N}}\right):$$

Proof By definition of $\widehat{V}_{\widehat{n}}$, we have

$$\frac{N}{2}\widehat{V}_{\widehat{n}} \quad \sum_{n=N=2}^{N} V^{\widehat{n}}_{\widehat{P}_n;\widehat{b}_n} \quad \sum_{n=0}^{N} V^{\widehat{n}}_{\widehat{P}_n;\widehat{b}_n};$$

which is bounded by the previous lemma and the fact that $V^{\widehat{n}}_{P^?;r=0} = 0$, since in Algorithm 3, the reward function is zero. ■

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

Theorem 11 *For any policy , we have*

$$\sum_{h=0}^{H} \mathrm{E}_{d_{P^?;h}} \left\| P^?_h(s;a) \quad \widehat{P}_{h;\widehat{n}}(s;a) \right\|_{TV} \quad O\left(H^3 d^2 j A j^{1:5} \sqrt{\frac{\log(j M j N H =)}{N}}\right) := "_{TV}:$$

Proof In this proof, let $\widehat{P} = \widehat{P}_{\widehat{n}}$, which is the returned environment from the algorithm. Let $r(s;a) = \left\| P^?_h(s;a) \quad \widehat{P}_h(s;a) \right\|_{TV} \; 2 [0;2]$. Then,

$$\begin{aligned}
&\sum_{h=0}^{H} \left(\mathrm{E}_{d_{P^?;h}} \quad \mathrm{E}_{d_{\widehat{P};h}}\right)[r(s;a)] \\
&= V_{P^?;r} \quad V_{\widehat{P};r} \\
&= \sum_{h=0}^{H} \mathrm{E}_{d_{\widehat{P};h}} \left[\left(\mathrm{E}_{P^?_h(s;a)} \quad \mathrm{E}_{\widehat{P}_h(s;a)}\right) V_{P^?;r;h+1}(s^0)\right] && \text{(Simulation lemma)} \\
&\quad 2H \sum_{h=0}^{H} \mathrm{E}_{d_{\widehat{P};h}} \left\| P^?_h(s;a) \quad \widehat{P}_h(s;a) \right\|_{TV}:
\end{aligned}$$

Thus,

$$\begin{aligned}
&\sum_{h=0}^{H} \mathrm{E}_{d_{P^?;h}} \left\| P^?_h(s;a) \quad \widehat{P}_h(s;a) \right\|_{TV} \\
&\quad (2H + 1) \sum_{h=0}^{H} \mathrm{E}_{d_{\widehat{P};h}} \left\| P^?_h(s;a) \quad \widehat{P}_h(s;a) \right\|_{TV} && \text{(by (Zhang et al., 2022, Lemma A.1))}
\end{aligned}$$

$$
\begin{aligned}
&. \; H\left(\sum_{h=0}^{H} 2\,\mathrm{E}_{d_{\widehat{P};h}}\left[\widehat{b}_{h;\widehat{n}}(s;a)\right] + \sqrt{jAj\ _{N=2}}\right) \\
&\quad H\left(V_{\widehat{P};\widehat{b}_{\widehat{n}}} + \sqrt{2jAj\frac{\log(j\,M\,jN\,H=\ )}{N}}\right) \\
&\quad H\left(V^{\widehat{n}}_{\widehat{P};\widehat{b}_{\widehat{n}}} + \sqrt{2jAj\frac{\log(j\,M\,jN\,H=\ )}{N}}\right) \\
&. \; H\left(H^2d^2jAj^{1:5}\sqrt{\frac{\log(j\,M\,jN\,H=\ )}{N}} + \sqrt{jAj\frac{\log(j\,M\,jN\,H=\ )}{N}}\right) \qquad \text{(by Lemma E.2)} \\
&2\,O\left(H^3d^2jAj^{1:5}\sqrt{\frac{\log(j\,M\,jN\,H=\ )}{N}}\right):
\end{aligned}
$$

■

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

Lemma E.3 *Suppose $\widehat{P}$ satisfies the following for all $h = 0; 1; :::; H \quad 1$,*

$$8 \quad : \mathrm{E}_{d_{P^{?};h}}\left\|\widehat{P}_h(s;a) \quad P^{?}_h(s;a)\right\|_{TV} \quad "_h: \tag{6}$$

Then, for any $h = 0; 1; :::; H \quad 1$, we have

$$8 \quad : \left\|d_{\widehat{P};h} \quad d_{P^{?};h}\right\|_{TV} \quad \sum_{t=0}^{h\ \ 1} "_t:$$

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

Proof We proceed by induction for $h = 0; 1; :::; H \quad 1$. For the base case of $h = 0$, no transition has been taken, so that $d_{\widehat{P};0} = d_{P^{?};0}$. Now let $h\ 2\ f0; 1; :::; H \quad 2g$ 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 $k\ \ k_{TV} = \sup_{kfk_1\ \ 1} j\mathrm{E}\ [f]j$. Below we use the notation that $f(s;\ \ ) = \mathrm{E}_{a\ \ (s)} f(s;a)$.

$$
\begin{aligned}
&kd_{\widehat{P};h+1} \quad d_{h+1}k_{TV} \\
&= \sup_{kfk_1\ \ 1}\left|\mathrm{E}_{d_{\widehat{P};h+1}}[f(s;a)] \quad \mathrm{E}_{d_{h+1}}[f(s;a)]\right| \\
&= \sup_{kfk_1\ \ 1}\left|\mathrm{E}_{(\tilde{s};\tilde{a})\ \ d_{\widehat{P};h};(s;a)\ \ \widehat{P}_h(\tilde{s};\tilde{a})}[f(s;\ \ _{h+1})] \quad \mathrm{E}_{(\tilde{s};\tilde{a})\ \ d_h;(s;a)\ \ P^{?}_h(\tilde{s};\tilde{a})}[f(s;\ \ _{h+1})]\right| \\
&\quad \sup_{kfk_1\ \ 1}\left|\left(\mathrm{E}_{(\tilde{s};\tilde{a})\ \ d_{\widehat{P};h}} \quad \mathrm{E}_{(\tilde{s};\tilde{a})\ \ d_h}\right)\mathrm{E}_{\widehat{P}_h(\tilde{s};\tilde{a})} f(s;\ \ _{h+1})\right|
\end{aligned}
$$

$$+ \sup_{kfk_1 \quad 1} \left| \mathsf{E}_{(\tilde{s};\tilde{a})} \quad d_h \left[\mathsf{E}_{\widehat{P}_h(\tilde{s};\tilde{a})} f(s;a) \quad \mathsf{E}_{P_h^?(\tilde{s};\tilde{a})} f(s; \quad {}_{h+1}) \right] \right|$$

$$\sum_{t=0}^{h \quad 1} {}''_t + \mathsf{E}_{(\tilde{s};\tilde{a})} \quad d_h \left[\sup_{kfk_1 \quad 1} \left| \left(\mathsf{E}_{\widehat{P}_h(\tilde{s};\tilde{a})} \quad \mathsf{E}_{P_h^?(\tilde{s};\tilde{a})} \right) f(s; \quad {}_{h+1}) \right| \right] \qquad \text{(by IH and Jensen)}$$

$$\sum_{t=0}^{h \quad 1} {}''_t + {}''_h; \qquad \text{(by (6) and } kf(s; \quad {}_{h+1})k \quad 1)$$

as desired. ■

Thus, when combined with Theorem 11, we have for $h = 0; 1; :::; H \quad 1$ and any policy $\quad$,

$$kd_{\widehat{P};h} \quad d_{P^?;h}k_{TV} \quad O\left(H^3 d^2 jAj^{1:5} \sqrt{\frac{\log(j\,M\,jN\,H=\ )}{N}} \right) = {}''_{TV}: \tag{7}$$

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

$$O\left(\frac{H^6 d^4 jAj^3 \log(j\,M\,jN\,H=\ )}{{}''^2_{TV}} \right):$$

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

$$O\left(\frac{H^{22} d^7 jAj^9 \log(j\,M\,jN\,H=\ )}{{}''^{10}_{TV}} \right):$$

Appendix F. Reward-free Exploration

In this section, we show that the mixture policy returned by Algorithm 1 is exploratory. Recall that Algorithm 1 contains two main steps:

Step 1 Learn a model $\widehat{P}$. This was the focus of the previous section, where our modified REP-UCB method obtained a strong TV guarantee ((7)) by requiring number of episodes at most,

$$N_{\text{REWARDFREE}} = O\left(\frac{H^6 d^4 jAj^3 \log(j\,M\,jN\,H=\ )}{{}''^2_{TV}} \right):$$

Step 2 Run LSVI-UCB (Algorithm 5) in the *learned* model $\widehat{P}$ 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 ${}^{+1}_{h}$ denote rolling-in $\quad$ 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 ${}^{+1}_{1}$ for a policy that just takes one uniform action from the initial distribution d_0.

Lemma F.1 *Let* $\in (0;1)$ *and run* REWARDFREE *(Algorithm 1). Let* $_{h;N}$ *be the empirical covariance at the* N*-th iteration of* LSVI-UCB *(Algorithm 5). Then, w.p. at least* 1 *we have,*

$$\sup \sum_{h=0}^{H} \mathbb{E}_{\ ^{+1}_{h}\ ^{1}_{1};\widehat{P}} \left\| \widehat{\ }_h(s_h;a_h) \right\|_{\ ^{1}_{h;N}} \ . \ Ad^{1:5}H^3\sqrt{\log(dNH=\ )=N}:$$

Proof In this proof, we'll treat the empirical MDP as $P^?$, as that is the environment we're running in. Thus, we abuse notation and $\widehat{P}_{h;e}$ is the model-based perpsective of the linear MDP, i.e. $\widehat{\ }_h\widehat{\ }_{h;e}$ where $\widehat{\ }_{h;e}$ is $\ ^{1}_{h;e}\sum_{k=1}^{e}\ ^{1}\widehat{\ }_h(s_h^k;a_h^k)\ (s_{h+1}^k)$. Also, in Algorithm 1, 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 -scaling of the bonus dominates this reward in the definition of $\widehat{Q}_{h;e}$, 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 $\widehat{\ }$) (Wang et al., 2020, Lemma 3.2).

Step 1 Show that $\widehat{V}_{h;e}\ 2\ V_h$ and w.p. 1 , for all $h;e$,

$$8s_h;a_h : \sup_{f\ 2V_h} \left|\widehat{P}_{h;e}(s_h;a_h) \quad P_h^?(s_h;a_h)\right| f \qquad b_{h;e}(s_h;a_h):$$

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 bonus $b_{h;e}$ using the appropriate covariance under the data.

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

Step 3 Bound the sum $\sum_e \widehat{V}_{h;e}$, where we decompose it as a sum of expected bonuses with the expectation is under ^{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 $\widehat{\ }^e$. 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 $e = 1;2;:::;N$, for any s, recalling definition of reward being $b_{h;e}$, we have

$$\begin{aligned}\widehat{V}_{0;e}(s_0) \quad & (1+\ )b_{0;e}(s_0;\ ^{e}_{0}(s_0)) + \widehat{P}_{0;e}(s_0;\ ^{e}(s_0))\widehat{V}_{1;e} \\ & (1+2\ )b_{0;e}(s_0;\ ^{e}_{0}(s_0)) + P^?_{0;e}(s_0;\ ^{e}(s_0))\widehat{V}_{1;e};\end{aligned}$$

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

$$\mathsf{E}_{d_0}\left[\widehat{V}_{0;e}(s_0)\right] \quad (1 + 2\ )\sum_{h=0}^{H\ 1}\mathsf{E}\ _e[b_{h;e}(s_h; a_h)]:$$

Summing over $e = 1; 2; :::; N$, we have

$$\sum_{e=1}^{N}\mathsf{E}_{d_0}\left[\widehat{V}_{0;e}(s_0)\right] \ . \quad \sum_{h=0}^{H\ 1}\sum_{e=1}^{N}\mathsf{E}\ _e[b_{h;e}(s_h; a_h)]$$
$$A\ \sum_{h=0}^{H\ 1}\sum_{e=1}^{N}\mathsf{E}_{(\ _h^e\ _1)^{+1}}[b_{h;e}(s_h; a_h)]$$

For each $h = 0; 1; :::; H\ \ 1$, apply Azuma's inequality to the martingale difference sequence $_e = \mathsf{E}_{(\ _h^e\ _1)^{+1}}[b_{h;e}(s_h; a_h)]\ \ b_{h;e}(s_h^e; a_h^e)$. The envelope is at most 2. So, w.p. 1 ,

$$A\ \sum_{h=0}^{H\ 1}\sum_{e=1}^{N}b_{h;e}(s_h^e; a_h^e) + A\ \sqrt{N\log(H=\ )}:$$

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

$$\sum_{h=0}^{H\ 1}\sum_{e=1}^{N}b_{h;e}(s_h^e; a_h^e) \quad \sum_{h=0}^{H\ 1}p\overline{N}\sqrt{\sum_{e=1}^{N}b_{h;e}(s_h^e; a_h^e)^2} \ . \ H\sqrt{dN\log(N)}:$$

Thus, we finally have

$$\sum_{e=1}^{N}\mathsf{E}_{d_0}\left[\widehat{V}_{0;e}(s_0)\right] \ . \ A\ \ H\sqrt{dN\log(NH=\ )}:$$

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

$$N\mathsf{E}_{d_0}[V_0^?(s_0; r_N)] \quad \sum_{e=1}^{N}\mathsf{E}_{d_0}[V_0^?(s_0; r_e)] \quad \sum_{e=1}^{N}\mathsf{E}_{d_0}\left[\widehat{V}_{0;e}(s_0)\right] \ . \ A\ \ H\sqrt{dN\log(NH=\ )}:$$

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

$$\mathsf{E}\ _{h\ \ 1;\widehat{P}}\left\|\widehat{\ }_h(s_h; a_h)\right\|\ _{\ _{h;N}^{1}} \quad \mathsf{E}_{d_0}\left[V_0\ ^{+1}_{h\ \ 1}(s_0; r_N)\right] \quad A\ \ H\sqrt{d\log(NH=\ )=N}:$$

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

Lemma F.2 (One-step back for Linear MDP) *Suppose $P_h = (\ _h;\ _h)$ is a linear MDP. Suppose is any mixture of n policies, and let $_h := n\mathsf{E}\left[\ _h(s_h;a_h)\ _h(s_h;a_h)^>\right] + \ I$ denote the unnormalized covariance. For any $g : S \quad A \ ! \ \mathsf{R}$, policy , and $h = 0;1;:::;H \quad 2$, we have*

$$\mathsf{E}\ [g(s_{h+1};a_{h+1})] \quad \mathsf{E}\left[k\ _h(s_h;a_h)k_{\ _h^{\ 1}}\right]\sqrt{nA\mathsf{E}_{\ _h^{+1}}[g(s_{h+1};a_{h+1})^2] + \ dkgk_1^2}$$

Proof

$$\mathsf{E}\ [g(s_{h+1};a_{h+1})] = \left\langle \mathsf{E}\ [\ _h(s_h;a_h)]; \int_{S_{h+1}} g(s_{h+1};\ _{h+1})\mathrm{d}\ _h(s_{h+1}) \right\rangle$$
$$\mathsf{E}\ k\ _h(s_h;a_h)k_{\ _h^{\ 1}} \left\| \int_{S_{h+1}} g(s_{h+1};\ _{h+1})\mathrm{d}\ _h(s_{h+1}) \right\|_{\ _h} ;$$

where

$$\left\| \int_{S_{h+1}} g(s_{h+1};\ _{h+1})\mathrm{d}\ _h(s_{h+1}) \right\|_{\ _h}$$
$$= n\mathsf{E}\left[\left(\mathsf{E}_{S_{h+1}\ \ P_h(s_h;a_h)}[g(s_{h+1};\ _{h+1})]\right)^2\right] + \left\| \int_{S_{h+1}} g(s_{h+1};\ _{h+1})\mathrm{d}\ _h(s_{h+1}) \right\|^2$$
$$njAj\mathsf{E}_{\ _h^{+1}}\left[g(s_{h+1};a_{h+1})^2\right] + \ dkgk_1^2 :$$

■

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

Lemma F.3 *Let be a symmetric positive definite matrix and define the bonus $b_h(s;a) = k\ _h^?(s;a)k_{\ 1}$. Then we have*

1. *For any policy , $\mathsf{E}_{d_h}\left[b_h^2(s;a)\right] \quad \frac{1}{\min(\ )}$. That is, coverage implies small squared bonus.*
2. *Suppose reachability under $^?$ (Assumption 3.2), then we have the converse: there exists $\widetilde{\ }$, for any policy , $\mathsf{E}_{d_h^{\widetilde{\ }}}\left[b_h^2(s;a)\right] \quad \frac{}{\min(\ )}$. That is, small squared bonus implies coverage.*

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

$$\mathsf{E}_{d_h}\left[b_h^2(s;a)\right] \quad \mathsf{E}_{d_h}\left[k\ _h^?(s;a)k_2^2k\quad^1k_2\right] \quad \frac{1}{\min(\ )}:$$

For the second claim, Assumption 3.2 implies that there exist a policy $\widetilde{\ }$ such that for all vectors $v \ 2 \ \mathsf{R}^d$ with $kvk_2 = 1$, we have $\mathsf{E}_{d_h^{\widetilde{\ }}}\left[(\ _h^?(s;a)^>v)^2\right] \quad$. Now decompose $\quad = \sum_{i=1}^d \ _iv_iv_i^>$,

where $_i; v_i$ are eigenvalue/vector pairs with $kv_i k_2 = 1$ and $_1 \quad _2 \quad ::: \quad _d$. Then substituting this into the definition of the bonus, we have

$$\mathsf{E}_{d_h^{\sim}}\left[b_h^2(s;a)\right] = \sum_{i=1}^{d} \frac{1}{i} \mathsf{E}_{d_h^{\sim}}\left[(\ \hat{}_h(s;a)^T v_i)^2\right]$$
$$\frac{1}{d} \mathsf{E}_{d_h^{\sim}}\left[(\ \hat{}_h(s;a)^T v_d)^2\right]$$
$$\overline{\min(\quad)}:$$

■

We now prove our main lemma for reward-free exploration, Lemma 3.1.

Lemma 3.1 (Source task exploration) *Suppose Assumptions 3.2,3.3 hold. Then, for any* $2\ (0;1)$, *w.p.* $1 \quad$, *running* EPS *in any source task with* $N_{\text{LSVI-UCB}} = \widetilde{}\left(A^3 d^6 H^8 \quad ^2\right)$ *and* $N_{\text{REWARDFREE}} = \widetilde{O}\left(A^3 d^4 H^6 \log\left(\frac{j \ jj \ j}{}\right) N^2_{\text{LSVI-UCB}}\right)$ *returns a* $_{\min}$*-exploratory policy where* $_{\min} = \widetilde{}\left(A \ ^3 d \ ^5 H \ ^7 \ ^2\right)$*: The sample complexity in the source task is* $N_{\text{REWARDFREE}}$ *episodes.*

Proof [Proof of Lemma 3.1] In this proof, let

$$\hat{}_h = N_{\text{LSVI-UCB}} \mathsf{E}_{\ h+1 \ 1}\left[\ \hat{}_h(s_h;a_h) \ \hat{}_h(s_h;a_h)^{>}\right] + \quad I;$$
$$\widehat{}_h = N_{\text{LSVI-UCB}} \mathsf{E}_{\ h+1 \ 1}\left[\widehat{}_h(s_h;a_h) \widehat{}_h(s_h;a_h)^{>}\right] + \quad I;$$

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

$$\widehat{}_h^{\ 1} \quad 2\left(\sum_{e=1}^{N_{\text{LSVI-UCB}}} \widehat{}_h(s_h^e;a_h^e) \widehat{}_h(s_h^e;a_h^e)^{>} + \quad I\right)^{\ 1} \quad 2 \ _{h;N_{\text{LSVI-UCB}}}^{1};$$

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

Under this event, for any , we have,

$$\sum_{h=1}^{H} \mathsf{E}_{\ ;\widehat{P}}\left\| \widehat{}_h(s_h;a_h)\right\|_{\widehat{}_h^{\ 1}} \cdot \sum_{h=1}^{H} \mathsf{E}_{\ ;\widehat{P}}\left\|\widehat{}_h(s_h;a_h)\right\|_{\widehat{}_{h;N_{\text{LSVI-UCB}}}^{1}} : \tag{8}$$

Now let $h = 0;1;:::;H \quad 2$ be arbitrary. By Assumption 3.2 (there exists some policy $\sim$ with coverage) such that,

$$\overline{\min\left(\ \hat{}_{h+1}\right)}$$

$$\mathbb{E}_{\sim}\left\| \widehat{}_{h+1}(s_{h+1};a_{h+1})\right\|^2_{(\widehat{}_{h+1})^{-1}} \qquad \text{(by Lemma F.3)}$$

$$\mathbb{E}_{\sim}\left\| \widehat{}_{h+1}(s_{h+1};a_{h+1})\right\|_{(\widehat{}_{h+1})^{-1}} \qquad \text{(by } \quad 1\text{)}$$

$$\mathbb{E}_{\sim;\widehat{P}}\left\| \widehat{}_h(s_h;a_h)\right\|_{\widehat{}_h^{-1}}\sqrt{A(2d+\varepsilon_{TV}N_{\text{LSVI-UCB}})}+\varepsilon_{TV} \qquad \text{(by Corollary 12)}$$

$$\mathbb{E}_{\sim;\widehat{P}}\left\| \widehat{}_h(s;a)\right\|_{\widehat{}_h^{-1}}\sqrt{A(2d+1)}+1=N_{\text{LSVI-UCB}} \qquad (\text{by } \varepsilon_{TV}=1=N_{\text{LSVI-UCB}})$$

$$. \quad A^{1:5}d^2H^3\sqrt{\log(dHN_{\text{LSVI-UCB}}=\ )=N_{\text{LSVI-UCB}}}+1=N_{\text{LSVI-UCB}} \qquad \text{(by (8) and Lemma F.1)}$$

$$. \quad A^{1:5}d^2H^3\sqrt{\log(dHN_{\text{LSVI-UCB}}=\ )=N_{\text{LSVI-UCB}}}:$$

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

$$\min\left(\mathbb{E}_{\ +1}\left[\ \widehat{}_{h+1}(s;a)\ \ \widehat{}_{h+1}(s;a)^T\right]\right)$$

$$= \frac{\min\left(\ \widehat{}_{h+1}\right)}{N_{\text{LSVI-UCB}}}$$

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

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

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 $f(x)=\sqrt{\frac{a}{x}}\quad\frac{b}{x}$, 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

$$N_{\text{LSVI-UCB}} = \tilde{}(b^2=a^2) = \tilde{}\left(\frac{A^3d^6H^8}{2}\right);$$

which results in a $\min$ lower bound of

$$\min\left(\mathbb{E}_{\ +1}\left[\ \widehat{}_{h+1}(s_{h+1};a_{h+1})\ \ \widehat{}_{h+1}(s_{h+1};a_{h+1})^T\right]\right) = \tilde{}(a^2=b) = \tilde{}\left(\frac{2}{A^3d^5H^7}\right):$$

Finally, we used the fact that $\varepsilon_{TV}=1=N_{\text{LSVI-UCB}}$, which is set by the choice of $N_{\text{REWARDFREE}}$ in the lemma statement to satisfy (7).

The above proves coverage of $\ ^{+1}_h$ for $h=0;1;:::;H\quad 2$. Finally to argue for $\ ^{+1}_1$, which is simply taking a random action at time h, we can simply invoke Assumption 3.2 for $h=0$ to get a policy $\tilde{}$ that

$$\mathbb{E}_{\ +1}\left[\ \widehat{}_0(s_0;a_0)\ \ \widehat{}_0(s_0;a_0)^>\right] \quad \frac{1}{A}\mathbb{E}_{\sim}\left[\ \widehat{}_0(s_0;a_0)\ \ \widehat{}_0(s_0;a_0)^>\right] \quad \overline{A}:$$

■

Corollary 12 *Let* $\ ; \ \widehat{\ }_h; \widehat{\ }_h$ *be defined as in the proof of Lemma 3.1. For any* $h = 0; 1; :::; H \quad 2$ *and any policy* $\ $, *we have*

$$\mathsf{E}\left[\left\| \widehat{\ }_{h+1}(s_{h+1}; a_{h+1})\right\|_{(\widehat{\ }_{h+1})^{\ 1}}\right] \quad \mathsf{E}_{;\widehat{P}}\left[\left\|\widehat{\ }_h(s_h; a_h)\right\|_{\widehat{\ }_h^{\ 1}}\right]\sqrt{jAj(2d + "_{TV}N_{\text{LSVI-UCB}})} + "_{TV}:$$

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 F.2 (one-step back) to the learned model $\widehat{P}$ and the function $(s; a) \ 7! \ k \ \widehat{\ }_{h+1}(s; a)k_{(\widehat{\ }_{h+1})^{\ 1}}$, which is bounded by $^{1=2} \quad 1$. We have,

$$\begin{aligned}
&\mathsf{E}_{;\widehat{P}}\left\| \widehat{\ }_{h+1}(s_{h+1}; a_{h+1})\right\|_{(\widehat{\ }_{h+1})^{\ 1}} \\
&\mathsf{E}_{;\widehat{P}}\left\|\widehat{\ }_h(s_h; a_h)\right\|_{\widehat{\ }_h^{\ 1}}\sqrt{NjAj\mathsf{E}_{\ h^{+1};\widehat{P}}k \ \widehat{\ }_{h+1}(s_{h+1}; a_{h+1})k^2_{(\widehat{\ }_{h+1})^{\ 1}} + d} \\
&\mathsf{E}_{;\widehat{P}}\left\|\widehat{\ }_h(s_h; a_h)\right\|_{\widehat{\ }_h^{\ 1}}\sqrt{NjAj\mathsf{E}_{\ h^{+1}}k \ \widehat{\ }_{h+1}(s_{h+1}; a_{h+1})k^2_{(\widehat{\ }_{h+1})^{\ 1}} + NjAj"_{TV} + d} \\
&\mathsf{E}_{;\widehat{P}}\left\|\widehat{\ }_h(s_h; a_h)\right\|_{\widehat{\ }_h^{\ 1}}\sqrt{djAj + NjAj"_{TV} + d};
\end{aligned}$$

where we used the fact that

$$\begin{aligned}
&\mathsf{E}_{\ h^{+1}}k \ \widehat{\ }_{h+1}(s_{h+1}; a_{h+1})k^2_{(\widehat{\ }_{h+1})^{\ 1}} \\
&= \mathrm{Tr}\left(\mathsf{E}_{\ h^{+1}}\left[\ \widehat{\ }_{h+1}(s_{h+1}; a_{h+1}) \ \widehat{\ }_{h+1}(s_{h+1}; a_{h+1})^{>}\right]\left(N\mathsf{E}_{\ h^{+1}}\left[\ \widehat{\ }_h(s_h; a_h) \ \widehat{\ }_h(s_h; a_h)^{>}\right] + \quad I\right)^{\ 1}\right) \\
&= \frac{1}{N}\mathrm{Tr}(I \quad M) \quad \frac{d}{N};
\end{aligned}$$

where M is a positive definite matrix. Thus, doing an initial change from d_{h+1} to $d_{\widehat{P};h+1}$ concludes the proof. ■

Appendix G. Representation Transfer

First, we prove Lemma 3.2, restated below.

Lemma 3.2 (Target model error) *Suppose Assumption 2.2 holds and* $_k$ *is* $_{\min}$*-exploratory for each source task* k. *For any* $\ 2 \ (0; 1)$, *w.p.* $1 \quad$, $8 \ h \ 2 \ [0 : H \quad 1]$, $9 \ \widetilde{\ }_h : S \ ! \ \mathsf{R}^d$ *such that*

$$\sup \mathsf{E}_{;P^{\ }_{target}}\left\|\widehat{\ }_h(s_h; a_h)^{>}\widetilde{\ }_h(\ ) \quad \widehat{\ }_h(s_h; a_h)^{>} \ \widehat{\ }_{target;h}(\ )\right\|_{TV} \quad "_{TV} := \sqrt{jAj \ \frac{3}{\max}K \ _{n=\ \min}}; \quad (4)$$

and, for any function $g : S \ ! \ [0; 1]$, $k\int g(s)\mathrm{d}\widetilde{\ }_h(s)k_2 \quad p\overline{d}$.

Proof [Proof of Lemma 3.2] Fix an arbitrary . Denote $_h(s^{\emptyset}) = \sum_{k=1}^{K}$ $_{k;h}(s^{\emptyset})\widehat{}_{k;h}(s^{\emptyset})$. First, note that

$$\max_{g:S\ !\ [0;1]} \left\| \int \ _h(s)g(s)\mathrm{d}(s) \right\|_2$$

$$\max_{g:S\ !\ [0;1]} \sum_{k=1}^{K} \left\| \int \widehat{}_{k;h}(s)\ \ _{k;h}(s)g(s)\mathrm{d}(s) \right\|_2$$

$$\sum_{k=1}^{K} \max_{s} \ _{k;h}(s) \quad p_{\overline{d}} \qquad \text{(Since } \int \widehat{}_{k;h}(s)g(s)\mathrm{d}(s) \quad p_{\overline{d}} \text{ by 3.3)}$$

$$= \quad p_{\overline{d}}$$

For any $h = 0; 1; :::; H$ 1, we have

$$\mathsf{E}\ _{;P^{?}_{\text{target}}} \left\| \widehat{}_h(s_h; a_h)^{>} \ \ _h(\ ) \quad \overset{?}{}_h(s_h; a_h)^{>} \ \ \overset{?}{}_{K;h}(\ ) \right\|_{TV}$$

$$= \mathsf{E}\ _{;P^{?}_{\text{target}}} \left[\sum_{s_{h+1}} \left| \sum_{k=1}^{K} \ _{k;h}(s_{h+1}) \left(\widehat{}_h(s_h; a_h)^{>} \widehat{}_{k;h}(s_{h+1}) \quad \overset{?}{}_h(s_h; a_h)^{>} \ \ \overset{?}{}_{k;h}(s_{h+1}) \right) \right| \right]$$

$$\mathsf{E}\ _{;P^{?}_{\text{target}}} \left[\sum_{s_{h+1}} \sum_{k=1}^{K} j \ _{k;h}(s_{h+1}) j \left| \widehat{}_h(s_h; a_h)^{>} \widehat{}_{k;h}(s_{h+1}) \quad \overset{?}{}_h(s_h; a_h)^{>} \ \ \overset{?}{}_{k;h}(s_{h+1}) \right| \right]$$

$$\max \sum_{k=1}^{K} \mathsf{E}\ _{;P^{?}_{\text{target}}} \left\| \widehat{}_h(s_h; a_h)^{>} \widehat{}_{k;h}(\ ) \quad \overset{?}{}_h(s_h; a_h)^{>} \ \ \overset{?}{}_{k;h}(\ ) \right\|_{TV} :$$

First consider the case when $h = 0$. At $h = 0$, the distribution under $P^{?}_{\text{target}}$ is the same as $_{k;h}$, and so, we directly get that the above quantity is at most $\max \ n^{1=2}$ ", which proves the $h = 0$ case.

Now consider any $h = 1; 2; :::; H$ 1. To simplify notation, let us denote

$$\mathrm{err}_{k;h}(s_h; a_h) = \left\| \widehat{}_h(s_h; a_h)^{>} \widehat{}_{k;h}(\ ) \quad \overset{?}{}_h(s_h; a_h)^{>} \ \ \overset{?}{}_{k;h}(\ ) \right\|_{TV};$$

$$w_{k;h} = \int_{s_h} \mathrm{d}\ \overset{?}{}_{\text{target};h\ \ 1}(s_h)\mathsf{E}_{a_h \quad _h(s_h)}\mathrm{err}_{k;h}(s_h; a_h);$$

$$_{k;h} = \mathsf{E}\ _{k;P^{?}_{k}} \left[\ \overset{?}{}_h(s_h; a_h) \ \ \overset{?}{}_h(s_h; a_h)^{>} \right] :$$

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

$$\max \sum_{k=1}^{K} \mathsf{E}\ _{;P^{?}_{\text{target}}} \mathrm{err}_{k;h}(s_h; a_h)$$

$$= \quad \max \sum_{k=1}^{K} \mathsf{E}\ _{;P^{?}_{\text{target}}} \left\langle \ \overset{?}{}_h\ _1(s_h\ \ _1; a_h\ \ _1); w_{k;h} \right\rangle$$

$$\max \sum_{k=1}^{K} \left(\mathsf{E}_{\ ;P^{?}_{\text{target}}} \left\| \ \hat{\ }^{?}_{h\ \ 1}(s_{h\ \ 1}; a_{h\ \ 1}) \right\|_{\ \ \frac{1}{k;h}\ \ 1} \right) k w_{k;h} k_{\ \ k;h\ \ 1}$$

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

$$\cancel{p} \frac{\max}{\min} \sum_{k=1}^{K} \sqrt{\mathsf{E}_{s_{h\ \ 1};a_{h\ \ 1}\ \ \ k;P^{?}_{k}} \mathsf{E}_{s_{h}\ \ P^{?}_{\text{target};h\ \ 1}(s_{h\ \ 1};a_{h\ \ 1});a_{h}\ \ \ h}(s_h) \text{err}_{k;h}(s_h;a_h)^2}$$

$$\cancel{A^{1=2}\ p} \frac{\max}{\min} \sum_{k=1}^{K} \sqrt{\mathsf{E}_{s_{h\ \ 1};a_{h\ \ 1}\ \ \ k;P^{?}_{k}} \mathsf{E}_{s_{h}\ \ P^{?}_{\text{target};h\ \ 1}(s_{h\ \ 1};a_{h\ \ 1});a_{h}\ \ \text{unif}(A)} \text{err}_{k;h}(s_h;a_h)^2}$$

By Assumption 2.2, the expectation over $P^{?}_{\text{target};h\ \ 1}$ is a linear combination of expectations over $P^{?}_{j;h\ \ 1}$,

$$\cancel{A^{1=2}\ p} \frac{3=2\ \max}{\min} \sum_{k=1}^{K} \sqrt{\sum_{j=1}^{K} \mathsf{E}_{s_{h\ \ 1};a_{h\ \ 1}\ \ \ k;P^{?}_{k}} \mathsf{E}_{s_{h}\ \ P^{?}_{j;h\ \ 1}(s_{h\ \ 1};a_{h\ \ 1});a_{h}\ \ \text{unif}(A)} \text{err}_{k;h}(s_h;a_h)^2}$$

$$\cancel{A^{1=2}\ p} \frac{3=2\ \max K^{1=2}}{\min} \sqrt{\sum_{k=1}^{K}\sum_{j=1}^{K} \mathsf{E}_{s_{h\ \ 1};a_{h\ \ 1}\ \ \ k;P^{?}_{k}} \mathsf{E}_{s_{h}\ \ P^{?}_{j;h\ \ 1}(s_{h\ \ 1};a_{h\ \ 1});a_{h}\ \ \text{unif}(A)} \text{err}_{k;h}(s_h;a_h)^2}$$

$$\cancel{A^{1=2}\ p} \frac{3=2\ \max K^{1=2}\ n^{1=2}}{\min};$$

where we used the MLE guarantee (3) in the last step. ■

Next we state an analogous lemma for when we don't need generative access to the source task, but instead assume Assumption 4.1, and Assumption 4.2.

Lemma G.1 *Suppose Assumption 4.1, and Assumption 4.2. Now take the setup of Lemma 3.2 with the only difference being that $\hat{\ }$ is learned as in Algorithm 4. Then, the same guarantee of Lemma 3.2 holds with a slightly different right hand side for the bound on the TV-error,*

$$\sup \mathsf{E}_{\ ;P^{?}_{target}} \left\| \hat{\ }_h(s_h;a_h)^{>} \hat{\ }_h(\ ) \quad \ ^{?}_{h}(s_h;a_h)^{>} \ ^{?}_{target;h}(\ ) \right\|_{TV} \quad \frac{\max K^{1=2}\ n^{1=2}}{(\ raw\ \ \min)^{1=2}};$$

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

$$\mathsf{E}_{\ ;P^{?}_{\text{target}}} \left\| \hat{\ }_h(s_h;a_h)^{>} \ _h(\ ) \quad \ ^{?}_{h}(s_h;a_h)^{>} \ ^{?}_{\text{target};h}(\ ) \right\|_{TV}$$

$$\mathsf{E}_{\ ;P^{?}_{\text{target}}} \left[\sum_{s_{h+1}} \left| \sum_{k=1}^{K} \ _{k;h}(s_{h+1}) \left(\hat{\ }_h(s_h;a_h)^{>} \hat{\ }_{k;h}(s_{h+1}) \quad \ ^{?}_{h}(s_h;a_h)^{>} \ ^{?}_{k;h}(s_{h+1}) \right) \right| \right]$$

$$\max \sum_{k=1}^{K} \mathsf{E}_{;P^{?}_{\text{target}}} \left\| \widehat{\ }_h(s_h;a_h)^{>} \widehat{\ }_{k;h}(\,) \quad {}^{?}_{h}(s_h;a_h)^{>} \; {}^{?}_{k;h}(\,) \right\|_{TV}$$

$$\max K^{1=2} \sqrt{\sum_{k=1}^{K} \mathsf{E}_{;P^{?}_{\text{target}}} \left\| \widehat{\ }_h(s_h;a_h)^{>} \widehat{\ }_{k;h}(\,) \quad {}^{?}_{h}(s_h;a_h)^{>} \; {}^{?}_{k;h}(\,) \right\|^2_{TV}}$$

By Assumption 4.1, Assumption 4.2, for any $s;a$, we have $\frac{d_{\text{target};h}(s;a)}{d^{k}_{k;h}(s;a)} \quad \frac{1}{raw \;\min\left(\mathsf{E}_{\;k;P^{?}_{k}}[\;{}^{?}_{h}(s_h;a_h) \;\; {}^{?}_{h}(s_h;a_h)^{>}]\right)}$ $\frac{1}{raw \;\min}$, where we used the coverage-under- $_k$ assumption in the last inequality. In other words, for each source task $k \; 2 \; [K]$, we have $\left\| \frac{\mathrm{d}d_{\text{target};h}}{\mathrm{d}d^{k}_{k;h}} \right\|_1 \quad \frac{1}{raw \;\min}$, hence we can use importance sampling,

$$\frac{\max K^{1=2}}{(\;raw \;\min)^{1=2}} \sqrt{\sum_{k=1}^{K} \mathsf{E}_{\;k;P^{?}_{k}} \left\| \widehat{\ }_h(s_h;a_h)^{>} \widehat{\ }_{k;h}(\,) \quad {}^{?}_{h}(s_h;a_h)^{>} \; {}^{?}_{k;h}(\,) \right\|^2_{TV}}$$

$$\frac{\max K^{1=2} \;\; {}^{1=2}_{n}}{(\;raw \;\min)^{1=2}}.$$

■

Appendix H. Proofs for LSVI-UCB under average-case misspecification

H.1. Auxiliary RL Lemmas

Lemma H.1 (Self-normalized Martingale) *Consider filtrations $fF_i g_{i=1;2;:::}$, so that $\mathsf{E}[''_i \; j \; F_{i \;\; 1}] = 0$ and $f''_i \; j \; F_{i \;\; 1} g_{i=1;2;:::}$ are sub-Gaussian with parameter ^{2}. Let $fX_i g_{i=1;2;:::}$ be random variables in a hilbert space H. Suppose a linear operator $_0 : H \;\; ! \;\; H$ is positive definite. For any t, define $_t = \;\; _0 + \sum_{i=1}^{t} X_i X_i^T$. Then w.p. at least $1 \quad$, we have,*

$$8t \quad 1 : \left\| \sum_{i=1}^{t} X_i ''_i \right\|^2_{\;\; t^{\;1}} \quad {}^2 \log\left(\frac{\det(\;\; _t) \det(\;\; _0)^{\;\; 1}}{^2} \right):$$

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

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

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

H.2. Proof of main result

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 $"_{ms}$, captured by Assumption H.1.

Assumption H.1 *Suppose for every* $h = 0; 1; :::; H \quad 1$, *there exist* $\tilde{}_h$ *such that for any policy* ,

$$\mathsf{E}\left[\left\|\tilde{}_h(\ )^T\hat{}_h(s_h; a_h) \quad P_h^?(\ j\ s_h; a_h)\right\|_{TV}\right] \quad "_{ms}:$$

We further assume that $\sup_{s;a;h}\left\|\tilde{}_h(\ )^T\hat{}(s; a)\right\|_{TV} \quad M$ *and* $kf^T\tilde{}_hk_2 \quad M \quad p\bar{d}kfk_1 \ 8f : S \ ! \ \mathsf{R}$, *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 M , 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 $\tilde{}_h(s^0) = \sum_{k=1}^K \ _{k;h}(s^0)\hat{}_{k;h}(s^0)$, then for any $s; a$, we have

$$\begin{aligned} k\tilde{}_h\hat{}_h(s; a)k_{TV} &= \sum_{s^0}\left|\sum_{k=1}^K \ _{k;h}(s^0)\hat{}_{k;h}(s^0)^T\hat{}_h(s; a)\right| \\ &\quad \sum_{s^0}\sum_{k=1}^K j \ _{k;h}(s^0)jj\hat{}_{k;h}(s^0)^T\hat{}_h(s; a)j \\ &\quad \sum_{k=1}^K \max_{s^0} j \ _{k;h}(s^0)j \qquad\qquad (\text{by } \left\|\hat{}_{k;h}\hat{}_h(s; a)\right\|_{TV} \quad 1) \\ &\quad : \end{aligned}$$

Also,

$$\begin{aligned} kf^T\tilde{}_hk_2 &= \left\|\sum_{s^0}\sum_{k=1}^K \ _{k;h}(s^0)\hat{}_{k;h}(s^0)f(s^0)\right\|_2 \\ &= \sum_{k=1}^K \max_{s^0} j \ _{k;h}(s^0)j\left\|\sum_{s^0}\hat{}_{k;h}(s^0)f(s^0)\right\|_2 \\ &\quad p\bar{d}kfk_1 : \qquad\qquad (\text{by } kf^T\hat{}_{k;h}k_2 \quad p\bar{d}kfk_1) \end{aligned}$$

So we will set $M = \ .$

Note that we only need the existence of $\tilde{}_h$ here, and $\tilde{}_h(\ )^T\hat{}_h(s; a)$ 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 infinite state spaces by replacing inner products with P by integrals.

Consider the following quantity,

$$\widehat{}_{h;e} = \left(\sum_{k=1}^{e-1} (s_{h+1}^k)\widehat{}_h(s_h^k, a_h^k)^T\right)(\quad_{h;e})^{-1} 2 \operatorname*{argmin}_{2R^S \quad d} \sum_{k=1}^{e-1} k \quad \widehat{}_h(s_h^i; a_h^i) \qquad (s_{h+1}^i)k_2 + k \quad k_F^2;$$

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

$$\widehat{P}_{h;e} = \widehat{}_{h;e}\widehat{}_h;$$
$$\widetilde{P}_h = \widetilde{}_h\widehat{}_h:$$

We will also sometimes use the shorthand $P f(s;a)$ for $\mathbb{E}_{s' \quad P(\cdot|s;a)}[f(s')]$.

For each $h = 0; 1; \ldots; H \quad 1$, let V_h denote the class of functions

$$\left\{ s \mapsto \left(\max_a \left\{ w^T \widehat{}_h(s;a) + r_h(s;a) + \widetilde{}k\widehat{}_h(s;a)k \quad_1 \right\}\right)_{M_V} \middle| kwk_2 \quad N, M_V; \widetilde{} 2 [0;B]; \quad I \text{ symmetric} \right\}$$

The motivation behind this construction is that V_h satisfies the key property that all of the learned value functions $\widehat{V}_{h;e}$ during Algorithm 5 are captured in this class.

Lemma H.3 *For any $h = 0; 1; \ldots; H \quad 1$,*

1. $\sup_s \left|\widehat{V}_{h;e}(s)\right| \quad M_V$.
2. *For any $e = 1; 2; \ldots; N$, we have $\widehat{V}_{h;e} 2 V_h$.*
3. *$8f 2 V_h$, we have $\sup_s |f(s)| \quad M_V$.*

Proof Recall that

$$\widehat{V}_{h;e}(s) = \left(\max_a \left\{ \widehat{w}_{h;e}^T \widehat{}_h(s;a) + r_h(s;a) + \quad b_{h;e}(s;a) \right\}\right)_{M_V}$$
$$\text{where } \widehat{w}_{h;e} = \quad {}^1_{h;e} \sum_{k=1}^{e-1} \widehat{}_h(s_h^k; a_h^k)\widehat{V}_{h+1;e}(s_{h+1}^k):$$

From the thresholding, we have

$$\left|\widehat{V}_{h;e}(s)\right| \quad M_V:$$

We can bound the norm of $\widehat{w}_{h;e}$ as follows,

$$k\widehat{w}_{h;e}k \quad \left\| \quad {}^{1}_{h;e} \right\|_2 \sum_{k=1}^{e} \left| \widehat{V}_{h+1;e}(s^k_{h+1}) \right| \quad N \sup_s \left| \widehat{V}_{h+1;e}(s) \right| \quad N M_V :$$

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

Now we control the metric entropy of V_h in $\grave{}_1$, i.e. $d(f_1; f_2) = \sup_s jf_1(s) \quad f_2(s)j$ for $f_i \; 2 \; V_h$.

Lemma H.4 *Let $" > 0$ be arbitrary and let $N_"$ be the smallest $"$-net with $\grave{}_1$ of V_h. Then,*

$$\log jN_"j \quad d \log(1 + 6L="") + \log(1 + 6B="") + d^2 \log(1 + 18B^2 \overline{p}\, d="^2"):$$

Proof Let $f_1; f_2 \; 2 \; V_h$. Then,

$$\begin{aligned}
& jf_1(s) \quad f_2(s)j \\
& \quad \max_a \left| (w_1 \quad w_2)^T \widehat{}_h(s;a) + \; {}_1 \left\| \widehat{}_h(s;a) \right\|_{ {}_1{}^1} \quad {}_2 \left\| \widehat{}_h(s;a) \right\|_{ {}_2{}^1} \right| \\
& \quad kw_1 \quad w_2k_2 + \max_a \left| (\; {}_1 \quad {}_2) \left\| \widehat{}_h(s;a) \right\|_{ {}_1{}^1} \right| + \; {}_2 \max_a \left| \left\| \widehat{}_h(s;a) \right\|_{ {}_1{}^1} \quad \left\| \widehat{}_h(s;a) \right\|_{ {}_2{}^1} \right| \\
& \quad kw_1 \quad w_2k_2 + j \; {}_1 \quad {}_2j + B \max_a \sqrt{ \left\| \widehat{}_h(s;a) \right\|_{ {}_1{}^1} \quad \left\| \widehat{}_h(s;a) \right\|_{ {}_2{}^1} } \qquad (\; {}_{\min}(\; {}_1) \quad 1) \\
& \quad kw_1 \quad w_2k_2 + j \; {}_1 \quad {}_2j + B \sqrt{ \left\| \; {}_1{}^1 \quad {}_2{}^1 \right\|_2 };
\end{aligned}$$

where we used for any $a; b \quad 0$, we have $\left| \overline{p}\, \overline{a} \quad \overline{p}\, \overline{b} \right| = \frac{\overline{ja \quad bj}}{\overline{p}\, \overline{a} + \overline{p}\, \overline{b}} \sqrt{ja \quad bj} \quad \sqrt{ja \quad bj}$. 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,

$$:= O\left(\overline{N d}"_{ms} M_V \; + M_V M \; d \sqrt{\log(dN M_V = \;)} \right): \tag{9}$$

The following high probability event (E_{model}) is a key step in our proof. Essentially, Theorem 13 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 H.1 and the bonus scaling is set as in (9). Throughout the section, $_h(\; _h)$ refers to indicator functions of the trajectory $_h$, where $_h = (s_0; s_1; :::; s_h)$. As before, the expectations $\mathsf{E}\,[g(\; _h)]$ are with respect to the distribution of trajectories when is executed in the environment $P^?$.

Theorem 13 *Let 2 (0; 1). Then, w.p. 1 , for any time h, episode e, indicator functions* $_1; : : : ; _H$*, and policy , we have*

$$\sup_{f \in V} \left| \mathbb{E}\left[\left(\widehat{P}_{h;e}(s_h; a_h) \quad P^?_h(s_h; a_h) \right) f \ _h(\ _h) \right] \right| \quad \mathbb{E}\left[b^e_h(s_h; a_h) \ _h(\ _h) \right] + kV_h k_1 \ "_{ms}: \tag{E_{model}}$$

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

$$\sup_{f \in V_h} \left| \mathbb{E}\left[\left(\widehat{P}_{h;e}(s_h; a_h) \quad \widetilde{P}_h(s_h; a_h) \right) f \ _h(\ _h) \right] \right| \quad \mathbb{E}\left[b^e_h(s_h; a_h) \ _h(\ _h) \right]:$$

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

$$\begin{aligned} \sup_{f \in V_h} \left| \mathbb{E}\left[\left(\widetilde{P}_h(s_h; a_h) \quad P^?_h(s_h; a_h) \right) f \ _h(\ _h) \right] \right| \quad & \mathbb{E}\left[\sup_{f \in V_h} \left| \left(\widetilde{P}_h(s_h; a_h) \quad P^?_h(s_h; a_h) \right) f \right| \ _h(\ _h) \right] \\ & \mathbb{E}\left[\sup_{f \in V_h} \left| \left(\widetilde{P}_h(s_h; a_h) \quad P^?_h(s_h; a_h) \right) f \right| \right] \\ & kV_h k_1 \ "_{ms}: \end{aligned}$$

Combining these two yields the result, as

$$\begin{aligned} & \sup_{f \in V_h} \left| \mathbb{E}\left[\left(\widehat{P}_{h;e}(s_h; a_h) \quad P^?_h(s_h; a_h) \right) f \ _h(\ _h) \right] \right| \\ & \sup_{f \in V_h} \left| \mathbb{E}\left[\left(\widetilde{P}_h(s_h; a_h) \quad P^?_h(s_h; a_h) \right) f \ _h(\ _h) \right] \right| + \sup_{f \in V_h} \left| \mathbb{E}\left[\left(\widehat{P}_{h;e}(s_h; a_h) \quad \widetilde{P}_h(s_h; a_h) \right) f \ _h(\ _h) \right] \right|: \end{aligned}$$

■

Lemma H.5 *Suppose Assumption H.1 and the bonus scaling is set as in (9). For any 2 (0; 1), w.p. at least 1 , we have for any time h, episode e, and policy ,*

$$8 s_h; a_h : \sup_{f \in V_h} \left| \left(\widehat{P}_{h;e}(s_h; a_h) \quad \widetilde{P}_h(s_h; a_h) \right) f \right| \quad b_{h;e}(s_h; a_h):$$

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

$$\widehat{}_{h;e} \ _{h;e} = \sum_{k=1}^{e \ 1} (s^k_{h+1})^{\widehat{}} _h(s^k_h; a^k_h)^T$$

$$= \sum_{k=1}^{e} \left(P_h^?(s_h^k; a_h^k) \quad \widetilde{P}_h(s_h^k; a_h^k) \right) \widehat{\ }_h(s_h^k; a_h^k)^T + \sum_{k=0}^{e} \left(\widetilde{P}_h(s_h^k; a_h^k) \quad \text{"}_h^k \right) \widehat{\ }_h(s_h^k; a_h^k)^T$$

$$= \sum_{k=1}^{e} \left(P_h^?(s_h^k; a_h^k) \quad \widetilde{P}_h(s_h^k; a_h^k) \right) \widehat{\ }_h(s_h^k; a_h^k)^T + \widetilde{\ }_h(\ _{h;e} \quad I) \quad \left(\sum_{k=0}^{e} \text{"}_h^k \widehat{\ }_h(s_h^k; a_h^k)^T \right):$$

Rearranging, we have

$$\widehat{\ }_{h;e} \quad \widetilde{\ }_h = \left(\sum_{k=0}^{e} \left(P_h^?(s_h^k; a_h^k) \quad \widetilde{P}_h(s_h^k; a_h^k) \right) \widehat{\ }_h(s_h^k; a_h^k)^T \right) (\ _{h;e})^{\ 1}$$
$$\widetilde{\ }_h(\ _{h;e})^{\ 1} \quad \left(\sum_{k=0}^{e} \text{"}_h^k \widehat{\ }_h(s_h^k; a_h^k)^T \right) (\ _{h;e})^{\ 1}:$$

Now let $f \ 2 \ V_h$ be arbitrary. For any $s_h; a_h$, multiply the above with $\widehat{\ }_h(s_h; a_h)$ and multiply with f, we have

$$\left| \left(\widehat{P}_{h;e}(s_h; a_h) \quad \widetilde{P}_h(s_h; a_h) \right) f \right|$$
$$= \left| f^T (\widehat{\ }_{h;e} \quad \widetilde{\ }_h) \widehat{\ }_h(s_h; a_h) \right|$$
$$\underbrace{\left| f^T \left(\sum_{k=1}^{e} \left(P_h^?(s_h^k; a_h^k) \quad \widetilde{P}_h(s_h^k; a_h^k) \right) \widehat{\ }_h(s_h^k; a_h^k)^T \right) \quad {}^{1}_{h;e} \widehat{\ }_h(s_h; a_h) \right|}_{\text{Term(a)}}$$
$$+ \underbrace{\left| f^T \widetilde{\ }_h \quad {}^{1}_{h;e} \widehat{\ }_h(s_h; a_h) \right|}_{\text{Term(b)}}$$
$$+ \underbrace{\left| f^T \left(\sum_{k=1}^{e} \text{"}_h^k \widehat{\ }_h(s_h^k; a_h^k)^T \right) \quad {}^{1}_{h;e} \widehat{\ }_h(s_h; a_h) \right|}_{\text{Term(c)}}:$$

We can deterministically bound Term (b) as follows,

$$\sup_{f \, 2 V_h} \left| f^T \widetilde{\ }_h \quad {}^{1}_{h;e} \widehat{\ }_h(s_h; a_h) \right|$$
$$= \sup_{f \, 2 V_h} \left| (\ {}^{1=2}_{h;e} f^T \widetilde{\ }_h)^T \left(\ {}^{1=2}_{h;e} \widehat{\ }_h(s_h; a_h) \right) \right|$$
$$\sup_{f \, 2 V_h} \left\| \ {}^{1=2}_{h;e} \right\|_2 k f^T \widetilde{\ }_h k_2 b_{h;e}(s_h; a_h)$$
$$k V_h k_1 M \ \overline{p} \, b_{h;e}(s_h; a_h): \qquad \text{(by Assumption H.1)}$$

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

We now derive the bound for Term (c) for any fixed $f \in V_h$. Observe that

$$\left| f^T \left(\sum_{k=1}^{e} \frac{1}{} \, "_h^k \, \widehat{} \, _h(s_h^k; a_h^k)^T \right) \quad {}^{1}_{h;e} \widehat{} \, _h(s_h; a_h) \right| = \left| \left(\quad {}^{1=2}_{h;e} \sum_{k=1}^{e} \frac{1}{} \, \widehat{} \, _h(s_h^k; a_h^k)(f^T "_h^k) \right)^T \left(\quad {}^{1=2}_{h;e} \widehat{} \, _h(s_h; a_h) \right) \right|$$
$$\left\| \sum_{k=1}^{e} \frac{1}{} \, \widehat{} \, _h(s_h^k; a_h^k)(f^T "_h^k) \right\| \quad {}_{h;e}^{1} \, b_{h;e}(s_h; a_h):$$

Now we argue w.p. $1 \quad$, for any $e; h$ we have

$$\left\| \sum_{k=1}^{e} \frac{1}{} \, \widehat{} \, _h(s_h^k; a_h^k)(f^T "_h^k) \right\| \quad {}_{h;e}^{1} \quad \left(2kV_h k_1 \sqrt{2 \log(1= \ ) + d \log(N + 1)} \right);$$

which implies the claim about all $s_h; a_h$. Indeed, we can apply Lemma H.1. Checking the preconditions, $\mathsf{E}_{P_h^?(s_h;a_h)} \left[f^T "_h^k \ j \ H_{k \ 1} \right] = 0, \quad jf^T "_h^k j \quad kfk_1 k"_h^k k_1 \quad 2kV_h k_1$, $\det(\ _0) = \det I = 1$, and $\det(\ _t) = \det(\ _{h;e}) \quad (e+1)^d$ since the largest eigenvalue is $e + 1$. So, w.p. at least $1 \quad$, for all e, we have the above inequality.

Thus, for any fixed $f \in V_h$, w.p. $1 \quad$, for all $e; h$ we have,

$$\begin{aligned}
&\left| \left(\widehat{P}_{h;e}(s_h; a_h) \quad \widetilde{P}_h(s_h; a_h) \right) f \right| \\
&\quad \mathsf{Term(a)} + \mathsf{Term(b)} + \mathsf{Term(c)} \\
&\quad \left(4kV_h k_1 (1 + M \ ) \sqrt{\log(1= \ ) + d \log(N)} + \sqrt{dN} kV_h k_1 "_{ms} \right) b_{h;e}(s_h; a_h) \\
&+ \left(kV_h k_1 M \quad \sqrt{d} \right) b_{h;e}(s_h; a_h) \\
&+ \left(4kV_h k_1 \sqrt{\log(1= \ ) + d \log(N)} \right) b_{h;e}(s_h; a_h) \\
&\quad . \quad \left(\sqrt{dN} kV_h k_1 "_{ms} + kV_h k_1 M \ \sqrt{\log(1= \ ) + d \log(N)} \right) b_{h;e}(s_h; a_h):
\end{aligned}$$

Now we apply a covering argument. Namely, union bound the above argument to every element in an $"_{net}$-net of V_h. For any $f \in V_h$, let $\widetilde{f}$ be its neighbor in the net s.t. $k\widetilde{f} \quad fk_1 \quad "_{net}$, so we have

$$\left| \left(\widehat{P}_{h;e}(s_h; a_h) \quad \widetilde{P}_h(s_h; a_h) \right) f \right| \quad \left| \left(\widehat{P}_{h;e}(s_h; a_h) \quad \widetilde{P}_h(s_h; a_h) \right) \widetilde{f} \right| + \left| \left(\widehat{P}_{h;e}(s_h; a_h) \quad \widetilde{P}_h(s_h; a_h) \right) (\widetilde{f} \quad f) \right|$$

and

$$\left| \left(\widehat{P}_{h;e}(s_h; a_h) \quad \widetilde{P}_h(s_h; a_h) \right) (\widetilde{f} \quad f) \right| \ . \ k\widetilde{f} \quad fk_1 (N + 1) \ . \ "_{net} N:$$

Setting $"_{net} = N$, 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

$$\forall s_h, a_h : \sup_{f \in V_h} \left| \left(\widehat{P}_{h,e}(s_h, a_h) \quad \widetilde{P}_h(s_h, a_h) \right) f \right| \\ \cdot \left(\sqrt{dN} k V_h k_1 \varepsilon_{ms} + k V_h k_1 M \sqrt{\log(1/\delta) + d \log(M_V) + d^2 \log(BdN)} \right) b_{h,e}(s_h, a_h) \\ \cdot \left(\sqrt{dN} M_V \varepsilon_{ms} + M_V M \sqrt{\log(1/\delta) + d \log(M_V) + d^2 \log(BdN)} \right) b_{h,e}(s_h, a_h)$$

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

■

Lemma H.6 *Let $f \in V_h$. For any $\in (0,1)$, w.p. at least 1 , for any time h, episode e, we have*

$$\forall s_h, a_h : \left| f^T \left(\sum_{k=1}^{e-1} P_h^?(s_h^k, a_h^k) \quad \widetilde{P}_h(s_h^k, a_h^k) \right) \widehat{}_h(s_h^k, a_h^k)^T \quad {}^{-1}_{h,e} \widehat{}_h(s_h, a_h) \right| \\ \left(4 k V_h k_1 (1 + M) \sqrt{\log(1/\delta) + d \log(N)} + \sqrt{dN} k V_h k_1 \varepsilon_{ms} \right) b_{h,e}(s_h, a_h).$$

Proof First observe that

$$\left| f^T \left(\sum_{k=1}^{e-1} (P_h^?(s_h^k, a_h^k) \quad \widetilde{}_h \widehat{}_h(s_h^k, a_h^k)) \widehat{}_h(s_h^k, a_h^k)^T \right) \quad {}^{-1}_{h,e} \widehat{}_h(s_h, a_h) \right| \\ = \left| \left({}^{-1/2}_{h,e} \sum_{k=1}^{e-1} \widehat{}_h(s_h^k, a_h^k) f^T (P_h^?(s_h^k, a_h^k) \quad \widetilde{}_h \widehat{}_h(s_h^k, a_h^k)) \right)^T \left({}^{-1/2}_{h,e} \widehat{}_h(s_h, a_h) \right) \right| \\ \left\| \sum_{k=1}^{e-1} \widehat{}_h(s_h^k, a_h^k) \eta_k \right\|_{{}^{-1}_{h,e}} b_{h,e}(s_h, a_h);$$

where $\eta_k = \left(P_h^?(s_h^k, a_h^k) \quad \widetilde{P}_h(s_h^k, a_h^k) \right) f$.

Now we will argue that w.p. 1 , for all e, h,

$$\left\| \sum_{k=1}^{e-1} \widehat{}_h(s_h^k, a_h^k) \eta_k \right\|_{{}^{-1}_{h,e}} \left(4 k V_h k_1 (1 + M) \sqrt{\log(1/\delta) + d \log(N)} + \sqrt{dN} k V_h k_1 \varepsilon_{ms} \right);$$

which will imply the claim for all s_h, a_h.

Apply self-normalized martingale concentration (Lemma H.1) to $X_i = \widehat{}_h(s_h^i, a_h^i)$ and $\varepsilon_i = \eta_i \, \mathbb{E}[\eta_i \mid H_{i-1}]$, where the expectation is over (s_h^i, a_h^i) in the definition of η_i. To see sub-Gaussianity,

bound the envelope, $j\tilde{r}_k j \quad k f k_1 k P_h^?(s_h^k; a_h^k) \quad \tilde{}_h \hat{}_h(s_h^k; a_h^k) k_{TV} \quad k V_h k_1 (1 + M\ )$, and thus $j\tilde{\ }_k j \quad 2k V_h k_1 (1 + M\ )$. Now compute the determinants: $\det(\ _{h;0}) = 1$ and since $\max(\ _{h;e})$ $e + 1$, we have that $\log\det(\ _{h;e}) \quad d\log(e + 1)$. Hence, w.p. at least $1 \quad$, we have

$$8e : \left\| \sum_{k=1}^{e\ \ 1} \hat{}_h(s_h^k; a_h^k)(\tilde{r}_k \quad \mathsf{E}[\tilde{r}_k \, j \, H_{k\ \ 1}]) \right\|_{\ \ \frac{1}{h;e}} \quad 2k V_h k_1 (1 + M\ ) \sqrt{2\log(1= \ ) + d\log(N + 1)}:$$

By Assumption H.1 applied to ^{k} (the data-generating policy for episode k), we have $j\mathsf{E}[\tilde{r}_k \, j \, H_{k\ \ 1}] j$ $k V_h k_1 \ "_{ms}$. Recall for any scalars c_i and vectors x_i, we have $k\sum_i c_i x_i k \quad \sum_i j c_i j k x_i k$ $\sqrt{\sum_i c_i^2}\sqrt{\sum_i k x_i k^2}$. Thus,

$$\left\| \sum_{k=1}^{e\ \ 1} \hat{}_h(s_h^k; a_h^k)\mathsf{E}[\tilde{r}_k \, j \, H_{k\ \ 1}] \right\|_{\ \ \frac{1}{h;e}}$$

$$\sqrt{\sum_{k=1}^{e\ \ 1} k \hat{}_h(s_h^k; a_h^k) k^2}_{\ \ \frac{1}{h;e}} \sqrt{\sum_{k=1}^{e\ \ 1} \mathsf{E}[\tilde{r}_k \, j \, H_{k\ \ 1}]^2}$$

$$p\overline{d\sqrt{(e \quad 1)} k V_h k_1 \ "_{ms}}: \qquad \text{(by Lemma H.2)}$$

Combining these two bounds concludes the proof.

■

Lemma H.7 (Optimism) *Suppose (E_{model}) holds. Let* $= k V_h k_1 \ "_{ms}$. *Then, for any episode* $e = 1; 2; :::; N$, *we have*

$$8h = 0; 1; :::; H \quad 1 : \mathsf{E}_{\ ?}\left[\left(Q_h^?(s_h; a_h) \quad \widehat{Q}_{h;e}(s_h; a_h)\right)_{\ h}(\ _h)\right] \quad (H \quad h) \ ;$$

and

$$8h = 0; 1; :::; H \quad 1 : \mathsf{E}_{\ ?}\left[\left(V_h^?(s_h) \quad \widehat{V}_{h;e}(s_h)\right)_{\ h\ \ 1}(\ _{h\ \ 1})\right] \quad (H \quad h) \ ;$$

where

$$_h(s_h) := \mathsf{I}\left[\widehat{Q}_{h;e}(s_h; \hat{}_h^e(s_h)) \quad M_V\right]$$

$$_h(\ _h) = \prod_{h'=0}^{h} \ _{h'}(s_{h'}):$$

Abusing notation, $_1(\ )$ *is the constant function 1.*

In particular, we have that

$$\mathsf{E}_{d_0}\left[V_0^?(s_0) \quad \widehat{V}_{0;e}(s_0)\right] \quad H \ :$$

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

$$\begin{aligned} &\mathsf{E}\left[\left(P^?_{H\ \ 1}(s_{H\ \ 1}; a_{H\ \ 1})(V^?_H \quad \widehat{V}_{H;e})\right)_{H\ \ 1}(\ _{H\ \ 1})\right] \\ &= \mathsf{E}\left[(0 \quad 0)_{H\ \ 1}(\ _{H\ \ 1})\right] = 0. \end{aligned}$$

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

$$\mathsf{E}_{\ ?}\left[\left(P^?_h(s_h; a_h)(V^?_{h+1} \quad \widehat{V}_{h+1;e})\right)_{\ h}(\ _h)\right] = \mathsf{E}_{\ ?}\left[\left(V^?_{h+1}(s_{h+1}) \quad \widehat{V}_{h+1;e}(s_{h+1})\right)_{\ h}(\ _h)\right] \qquad (H \quad h \quad 1) \qquad \text{(IH)}$$

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

$$\begin{aligned} &\mathsf{E}_{\ ?}\left[\left(Q^?_h(s_h; a_h) \quad \widehat{Q}_{h;e}(s_h; a_h)\right)_{\ h}(\ _h)\right] \\ &= \mathsf{E}_{\ ?}\left[\left(P^?_h(s_h; a_h)V^?_{h+1} \quad \widehat{P}_{h;e}(s_h; a_h)\widehat{V}_{h+1;e} \quad b_{h;e}(s_h; a_h)\right)_{\ h}(\ _h)\right] \\ &\quad \mathsf{E}_{\ ?}\left[\left(\left(P^?_h(s_h; a_h) \quad \widehat{P}_{h;e}(s_h; a_h)\right)\widehat{V}_{h+1;e} \quad b_{h;e}(s_h; a_h)\right)_{\ h}(\ _h)\right] + (H \quad h \quad 1) \\ &\qquad\qquad \text{(by (IH))} \\ &\left|\mathsf{E}_{\ ?}\left[\left(\widehat{P}_{h;e}(s_h; a_h) \quad P^?_h(s_h; a_h)\right)\widehat{V}_{h+1;e\ \ h}(\ _h)\right]\right| \quad \mathsf{E}_{\ ?}[\ b_{h;e}(s_h; a_h)\ _h(\ _h)] + (H \quad h \quad 1) \\ &\quad + (H \quad h \quad 1) = (H \quad h)\ ; \qquad \text{(by } (E_{model}) \text{ and } \widehat{V}_{h+1;e}\ 2\ V_h \text{ (Lemma H.3))} \end{aligned}$$

which proves the Q-optimism claim.

Now let's prove V-optimism.

$$\begin{aligned} &\mathsf{E}_{\ ?}\left[\left(V^?_h(s_h) \quad \widehat{V}_{h;e}(s_h)\right)_{\ h\ \ 1}(\ _{h\ \ 1})\right] \\ &= \mathsf{E}_{\ ?}\left[\left(Q^?_h(s_h; a_h) \quad \left(\widehat{Q}_{h;e}(s_h; \widehat{\ }^e_h(s_h))\right)_{\ M_V}\right)_{\ h\ \ 1}(\ _{h\ \ 1})\right] \\ &= \mathsf{E}_{\ ?}[(Q^?_h(s_h; a_h) \quad M_V)_{\ h\ \ 1}(\ _{h\ \ 1})(1 \quad _h(s_h))] \\ &+ \mathsf{E}_{\ ?}\left[\left(Q^?_h(s_h; a_h) \quad \widehat{Q}_{h;e}(s_h; \widehat{\ }^e_h(s_h))\right)_{\ h\ \ 1}(\ _{h\ \ 1})\ _h(s_h)\right] \\ &\quad \mathsf{E}_{\ ?}\left[\left(Q^?_h(s_h; a_h) \quad \widehat{Q}_{h;e}(s_h; \widehat{\ }^e_h(s_h))\right)_{\ h}(\ _h)\right] \\ &\quad \mathsf{E}_{\ ?}\left[\left(Q^?_h(s_h; a_h) \quad \widehat{Q}_{h;e}(s_h; \ \widehat{\ }^?_h(s_h))\right)_{\ h}(\ _h)\right] \\ &\quad (H \quad h)\ ; \end{aligned}$$

by Q-optimism. ■

Remark 14 *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 H.8 (Simulation) *For any episode $e = 1; 2; :::; N$, we have*

$$\mathsf{E}_{d_0}\left[\widehat{V}_{0;e}(s_0) \quad V_0^{\ e}(s_0)\right] \quad \sum_{h=0}^{H\ 1} \mathsf{E}_{\sim e}\left[b_{h;e}(s_h; a_h) + (\widehat{P}_h(s_h; a_h) \quad P_h^?(s_h; a_h))\widehat{V}_{h+1;e}\right]$$

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

$$\begin{aligned}
&\widehat{V}_{0;e}(s_0) \quad V_0^{\ e}(s_0) \\
&\widehat{Q}_{0;e}(s_0;\ {}^{e}_{0}(s_0)) \quad Q_0^{\ e}(s_0;\ {}^{e}_{0}(s_0)) \\
&= b_{0;e}(s_0;\ {}^{e}_{0}(s)) + \left(\widehat{P}_{0;e}(s_0;\ {}^{e}_{0}(s)) \quad P_0^?(s_0;\ {}^{e}_{0}(s_0))\right)\widehat{V}_{1;e} + P_0^?(s_0;\ {}^{e}_{0}(s_0))\left(\widehat{V}_{1;e} \quad V_1^{\ e}\right);
\end{aligned}$$

where the inequality is due to the thresholding on the value function. Now, perform this recursively on the $P_0^?(s_0;\ {}^{e}_{0}(s_0))\left(\widehat{V}_{1;e} \quad V_1^{\ e}\right)$ term. Doing this unravelling h times gives the result. ■

Theorem 15 *Suppose Assumption H.1. Then, for any $2\ (0;1)$, w.p. at least 1 , we have that* LSVI-UCB *(with set to (9)) has regret at most,*

$$NV^? \quad \sum_{e=0}^{N\ 1} V^{\sim e} \quad \widetilde{O}\left(dHNM_V\sqrt{\log(HN=\ )}''_{ms} + d^{1:5}H^{\ p}\overline{N}M_V M \quad \log(dHN=\ )\right)$$

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

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

$$\begin{aligned}
\mathsf{E}_{d_0}\left[V_0^?(s_0) \quad V_0^{\sim e}(s_0)\right] \quad \mathsf{E}_{d_0}\left[\widehat{V}_0^e(s_0) \quad V_0^{\sim e}(s_0)\right] + H \\
\sum_{h=0}^{H\ 1} \mathsf{E}_{\sim e}\left[\ b_h^e(s_h; a_h) + \left(\widehat{P}_{h;e}(s_h; a_h) \quad P_h^?(s_h; a_h)\right)\widehat{V}_{h+1;e}\right] + H
\end{aligned}$$

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

$$\sum_{h=0}^{H\ 1} \mathsf{E}_{\sim e}[2\ \ b_{h;e}(s_h; a_h)] + 2H\ :$$

Now, summing over $e = 1; 2; :::; N$, we have

$$\sum_{e=1}^{N} \mathsf{E}_{d_0}\left[V_0^?(s_0) \quad V_0^{\sim e}(s_0)\right]$$

$$2HN + 2 \sum_{h=0}^{H} \sum_{e=1}^{1} \sum_{N} \mathrm{E}_{\sim^e}[b_{h;e}(s_h; a_h)]$$

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

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

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

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

So, putting everything together,

$$\begin{aligned}
&\sum_{e=1}^{N} \mathrm{E}_{d_0}\left[V_0^?(s_0) \quad V_0^{\sim^e}(s_0)\right] \\
&. HN + H\sqrt{dN \log(N)} + \sqrt{N \log(HN=)} \\
&. HN + \left(\sqrt{dN} M_V "_{ms} + M_V M \; d\sqrt{\log(dHN=)}\right) \; H\sqrt{dN \log(HN=)} \\
&= HN + dHN M_V \sqrt{\log(HN=)} "_{ms} + d^{1:5} H \sqrt{N} M_V M \; \log(dHN=):
\end{aligned}$$

Note that $H = HkVk_1 "_{ms} = HM_V "_{ms}$ is of lower order (with respect to N), we can simply drop it. This concludes the proof. ■

Corollary 16 *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,

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

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

Appendix I. Proof of Main Theorems

First we prove Theorem 5 and Theorem 3.

Theorem 5 (REPTRANSFER) *Suppose Assumptions 2.1,2.2,3.1,3.3, and $_k$ is $_{\min}$-exploratory for each source task k. Then, for any 2 $(0;1)$, w.p. 1 , REPTRANSFER when deployed in the target task has regret at most $\widetilde{O}\left(H^2 d^{1:5} \sqrt{T \log(1=)} \right)$, with at most Kn generative accesses per source task, with $n = O\left({}^{1}_{\min} A \ {}^{3}_{\max} KT \left(\log \frac{j \ j}{} + K \log j \ j \right) \right)$.*

Theorem 3 (Regret under generative source access) *Suppose Assumptions 2.1,2.2,3.1,3.2,3.3 hold, and fix any 2 $(0;1)$. Then, running REPTRANSFER with policies from EPS (parameters set as in Lemma 3.1) has regret in the target task of $\widetilde{O}\left(H^2 d^{1:5} \sqrt{T \log(1=)} \right)$, with at most $\widetilde{O}\left(A^4 \ {}^{3}_{\max} d^5 H^7 K^2 T \ {}^{2} (\log(j \ j=) + K \log j \ j) \right)$ generative accesses per source task.*

Proof [Proof of Theorem 5 and Theorem 3] For the regret bound, set $M_V = H$ and $M =$ and apply Theorem 15. This choice of M is valid by the argument following Assumption H.1. This gives us a regret bound of

$$\widetilde{O}\left(dH^2 T \, "_{ms} + \ d^{1:5} H^2 \sqrt{T} \log(1=) \right);$$

where $"_{ms}$ 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 $\widehat{P}_k$. Recall that we need this to be sufficiently large such that $"_{TV} = 1=N_{\text{LSVI-UCB}}$. As required by Lemma 3.1, we need,

$$\begin{aligned} N_{\text{REWARDFREE}} &= \widetilde{O}\left(A^3 d^4 H^6 \log(j \ jj \ j=) N^2_{\text{LSVI-UCB}} \right) \\ &= \widetilde{O}\left(A^3 d^4 H^6 \log(j \ jj \ j=) \left(A^3 d^6 H^8 \ {}^{2} \right)^2 \right) \\ &= \widetilde{O}\left(A^9 d^{16} H^{22} \ {}^{4} \log(j \ jj \ j=) \right): \end{aligned}$$

Second, we calculate the cross-sampling sample complexity. Recall that n is the number of samples in each pairwise dataset. In order to reduce $_{ms}$ to $1=\sqrt{T}$, by Lemma 3.2, we need

$$\begin{aligned} 1=\sqrt{T} \quad "_{ms} \quad & \left(A \ {}^{3}_{\max} K= \ {}_{\min} \right)^{1=2} \sqrt{n} \\ & \left(A \ {}^{3}_{\max} K= \ {}_{\min} \right)^{1=2} \sqrt{\frac{1}{n} \left(\log \frac{j \ j}{} + K \log j \ j \right)} \qquad \text{(by (3))} \end{aligned}$$

which implies that we need

$$n \quad {}^{1}_{\min} A \ {}^{3}_{\max} KT \left(\log \frac{j \ j}{} + K \log j \ j \right):$$

Incorporating the coverage result from Lemma 3.1 gives,

$$n \quad A^4 \ {}^3_{\max} d^5 H^7 K T \quad {}^2\left(\log \frac{j \ j}{} + K \log j \ \ j\right):$$

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

$$\begin{aligned} & N_{\text{REWARDFREE}} + (K \quad 2) \ \ n \\ & = \widetilde{O}\left(A^9 d^{16} H^{22} \quad {}^4 \log(j \ \ jj \ \ j= \ ) + A^4 \ {}^3_{\max} d^5 H^7 K^2 T \quad {}^2\left(\log \frac{j \ j}{} + K \log j \ \ j\right)\right): \end{aligned}$$

■

Now we prove Theorem 7, restated below.

Theorem 7 (Regret with online access) *Suppose Assumptions 2.1-2.2,4.1,4.2 hold. W.p.* $1 \quad$, *Algorithm 4 with appropriate parameters achieves a regret in the target* $\widetilde{O}\left(\ d^{1:5} H^2 \sqrt{T \log(1= \ )}\right)$, *with* $\text{poly}\left(A; \ \ {}_{\max}; d; H; K; T; \ \ {}^1; \ \ {}^1_{raw}; \log(j \ \ jj \ \ j= \ )\right)$ *online queries in the source tasks.*

Proof [Proof of Theorem 7] We follows the same format as the proof of Theorem 5. The regret bound is identical.

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

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

which implies we need

$$n \quad \frac{{}^2_{\max} K}{{}_{raw} \ {}_{\min}}\left(\log \frac{j \ j}{} + K \log j \ \ j\right):$$

Plugging in the coverage of Lemma 3.1,

$$\begin{aligned} n \quad & \frac{{}^2_{\max} K T}{{}_{raw}}\left(\log \frac{j \ j}{} + K \log j \ \ j\right)(A \ {}^3 d \ {}^5 H \ {}^7 \ {}^2) \ {}^1 \\ & \frac{A^3 \ {}^2_{\max} d^5 H^7 K T}{{}_{raw} \ {}^2}\left(\log \frac{j \ j}{} + K \log j \ \ j\right): \end{aligned}$$

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

$$N_{\text{REWARDFREE}} + n$$

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

■

Appendix J. Experiment Details

J.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 $z_{i;h}; i \ 2 \ f0; 1; 2g$ for each timestep h and 10 actions. Among the three latent states, we denote $z_0; z_1$ as the good states which leads to the final reward and z_2 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 $a_{0;h}; a_{1;h}$ 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., $s_{0;h+1}$, $s_{1;h+1}$) with equal probability. Otherwise, if the agent takes any 9 of the bad actions, the environment will transition to the bad state $s_{2;h+1}$ 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 $2^{d\log(H+jSj+1)e}$. 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 10^H actions to reach the final goal with random actions.

J.2. Construction of transfer setup in the observational coverage setting

In this section we introduce the detailed construction of our first experiment. For the source environment, we simply generate 5 random vanilla comblock environment described in Section.J.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 at h from one of

the sources and set it to be the optimal action of the target environment at timestep 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 2.2 is satisfied if we set = 1 for the source environment where we select the optimal action and = 0 for the other sources, at each timestep. To see how Assumption 4.1 is satisfied, recall that comblock environment naturally satisfies Assumption 3.2, and identical emission implies that the conditional ratio of all observations between source and target is 1.

J.3. Construction of transfer setup in feature coverage setting

Now we introduce the construction of the Comblock with Partitioned Observation (Comblock-PO) environment, which we use in our second experiment. 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 $O = jOj$, the size of the observation for comblock-PO is KO. For the k-th source environment, where k 2 [K], the environment first generates the O-dimensional observation vector as in the original comblock, and then embed it to the $(k \quad 1)O$-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.

J.4. Implementation details

Our implementation builds on BRIEE (Zhang et al., 2022). [2]. 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 5.

To create the training dataset for Multi-task REPLEARN, for each $(i; j)$ environment pairs where i 6= j, we collect 500 samples for each timestep h. For each $(i; i)$ environment pairs, we collect 500 $(k \quad 1)$ k samples for each timestep h, 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 $(k \quad 1)$ k samples for each $(i; i)$ cross transition to ensure that the total number of samples is the same for G-REPTRANSFER and O-REPTRANSFER.

To sample the initial state action pair (i.e., $(s; a)$ pair as in (1)), for 90% of the samples, we follow the final policy from each source environment trained using BRIEE. For the remaining 10%, we

2. Code based on public repository: https://github.com/yudasong/briee.

follow the same policy to state s, and then take a uniform random policy to get a. With this sampling scheme we ensure that Assumption 3.2 is satisfied. In the setting of Section. **??**, we follow a more simple procedure to ensure that the samples are more balanced among the three states: we skip the first sampling step from environment i (i.e., sample s given (s, a)), and simply reset environment i 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-REPTRANSFER and O-REPTRANSFER for a fair comparison.

J.5. Hyperparameters

In this section, we record the hyperparameters we try and the final hyperparameter we use for each baselines. The hyperparameters for REPTRANSFER in the first experiment is in Table. 3. The hyperparameters for REPTRANSFER in the second experiment is in Table. 4. The hyperparameters for BRIEE is in Table. 5. We use the same set of hyperparameters for G-REPTRANSFER and O-REPTRANSFER.

Table 3: Hyperparameters for REPTRANSFER in 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 coefficient	{0.01}	0.01
Decoder softmax temperature	{1}	1
Decoder $_0$ softmax temperature	{0.1}	0.1
LSVI bonus coefficient	$\{1, \frac{H}{5}\}$	1
LSVI regularization coefficient	{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 f learning rate	{1e-2}	1e-2
Discriminator f hidden layer size	{256,512}	256
Discriminator f 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 f number of gradient steps	{64,80,128,256}	64
Decoder batch size	{256,512}	512
Discriminator f 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 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	{512}	512
Discriminator f 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

J.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. We provide an example in Fig. 3 on how to interpret the visualization: let the emission function of the target environment be o, and let the decoder that we are evaluating be , and to generate the blue block in Fig. 3, we sample 30 observations $\{s_n\}_{n=1}^{30}$ from the target environment at $z_{1;13}$, the latent state 1 (the title of the subplot) from timestep 13 (the x-axis). Concretely, $\{s_n\}_{n=1}^{30}$ $o(\ j\ z_{1;13})$. The blue block denotes the three-dimensional decoded latent states $\hat{z}$ from these 30 observations: $\hat{z} = \frac{1}{30}\sum_{n=1}^{30}$ (s_n).

In Figure. 2, we provide a runnning example that explains the results showed in Figure. 1 (b). We then follow the detailed visualizations in the following sections.

J.6.1. VISUALIZATIONS FROM THE OBSERVATIONAL COVERAGE EXPERIMENT

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

J.6.2. VISUALIZATIONS FROM THE FEATURE COVERAGE EXPERIMENT

We record the visualization of the 2 sources from Fig. 11 and Fig. 12; O-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.

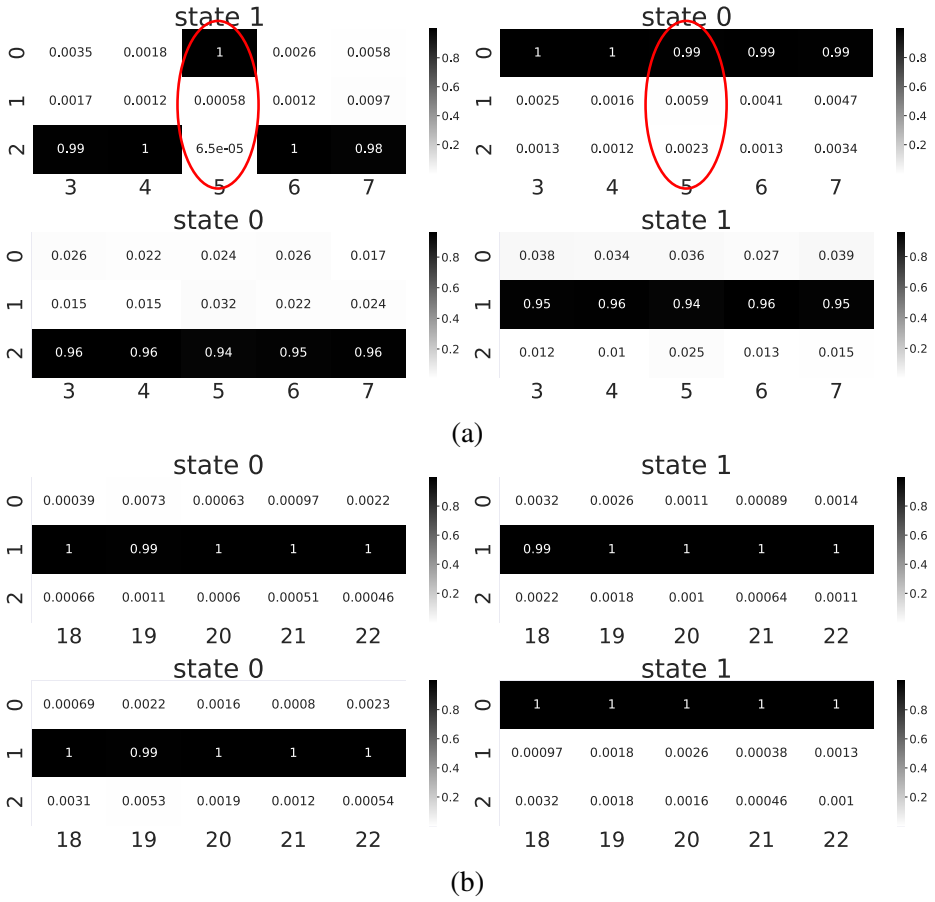


Figure 2: **(a):** Visualization of the decoder source (top) and G-REPTRANSFER (bottom). **(b):** Visualization of the decoder O-REPTRANSFER (top) and G-REPTRANSFER (bottom). For each baseline, The h-th column in the i-th image denotes the averaged decoded states from the 30 observations generated by latent state $z_{i;h}$, for $i \ 2 \ f0; 1; 2g$ and $h \ 2 \ [25]$, 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 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)). 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.

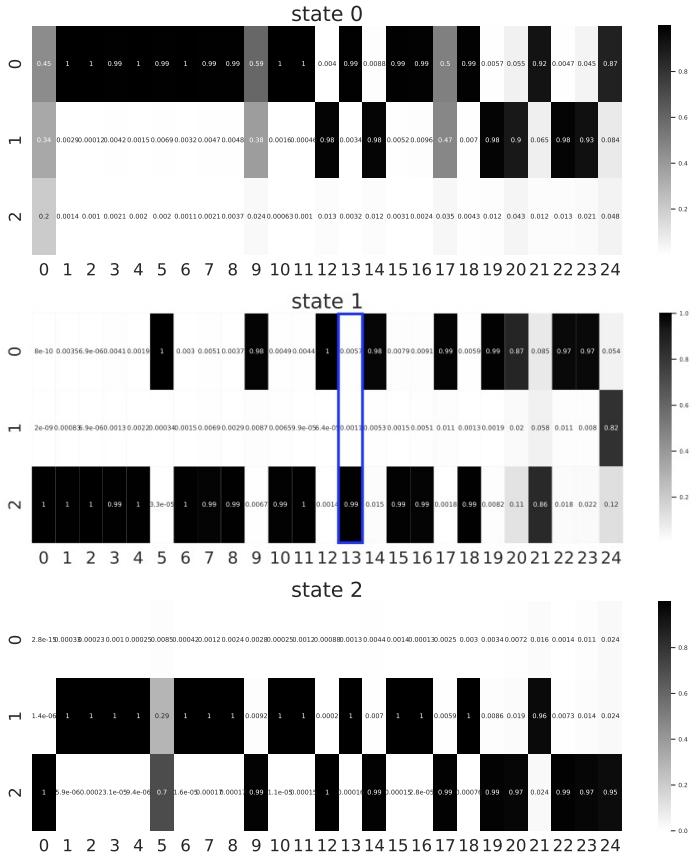


Figure 3: Visualization of decoders from source 1. Note the collapse happens at timestep 5, 9 and 17.

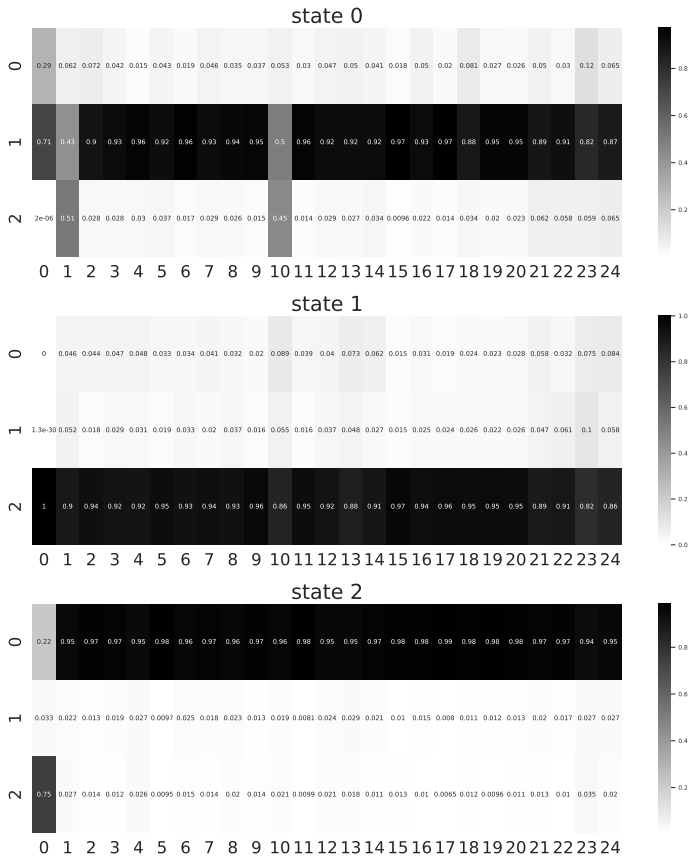


Figure 4: Visualization of decoders from source 2. Note the collapse happens at timestep 1 and 10.

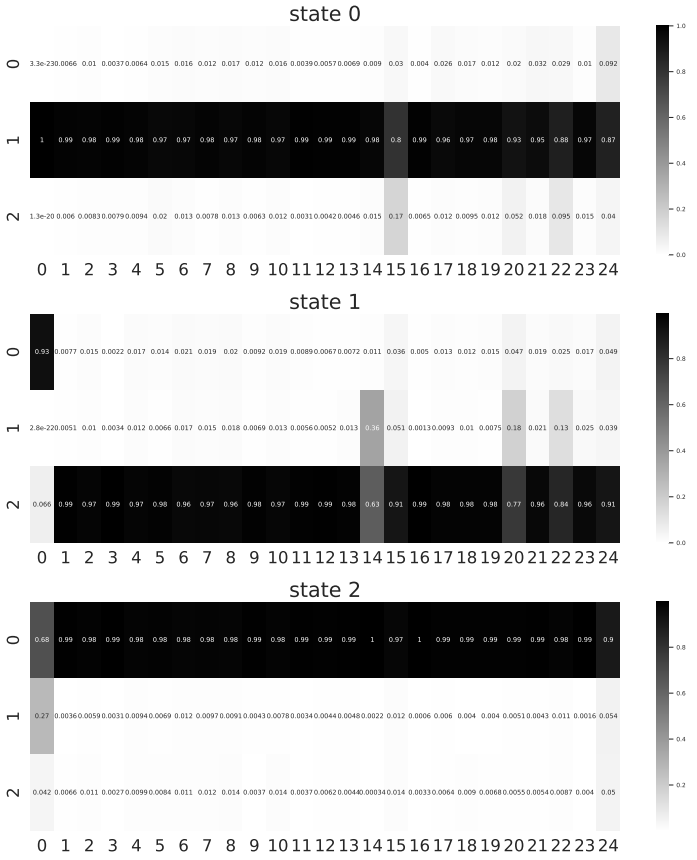


Figure 5: Visualization of decoders from source 3. Note the collapse happens at timestep 14 and 15.

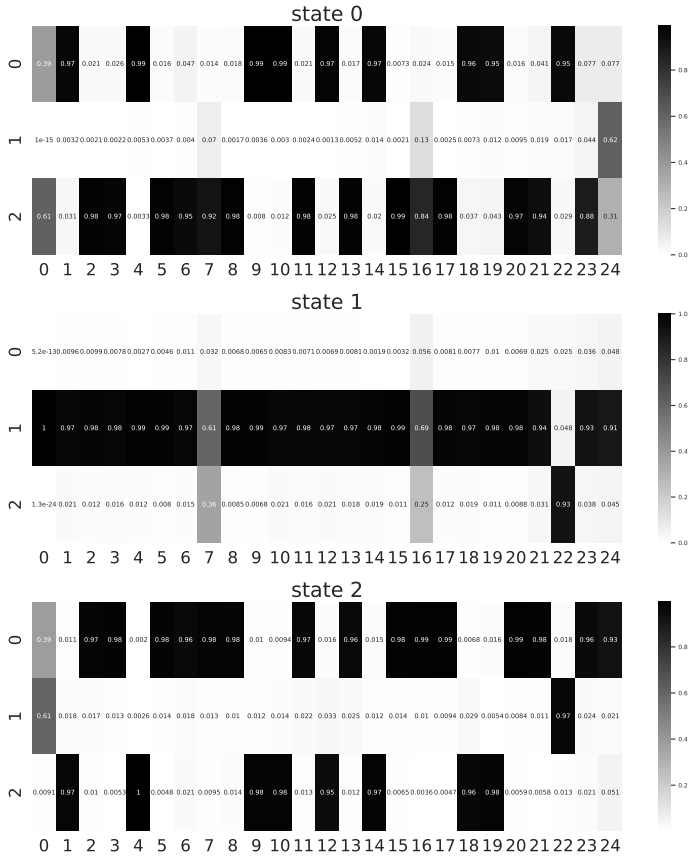


Figure 6: Visualization of decoders from source 4. Note the collapse happens at timestep 7, 16, 24.

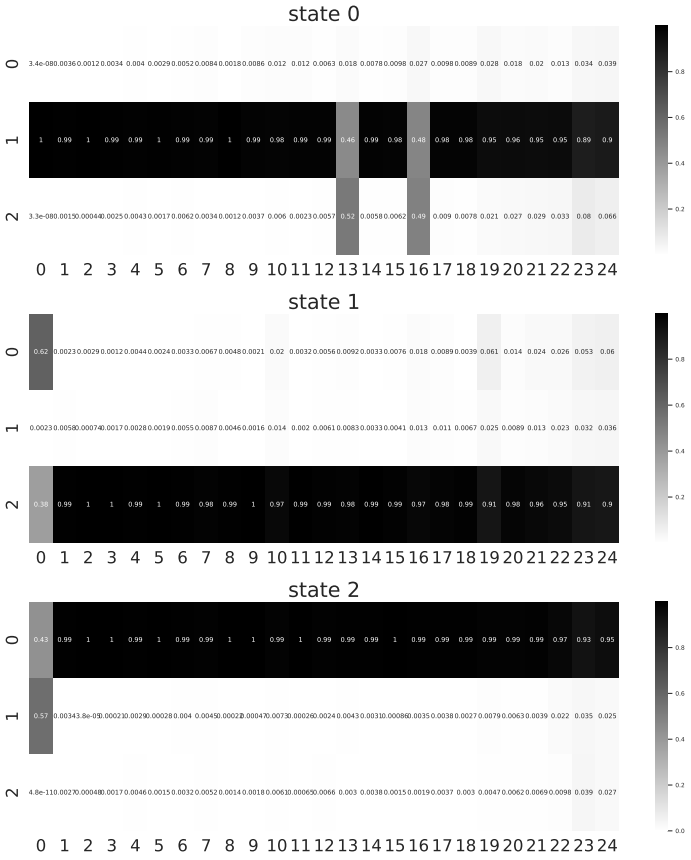


Figure 7: Visualization of decoders from source 3. Note the collapse happens at timestep 13 and 16.

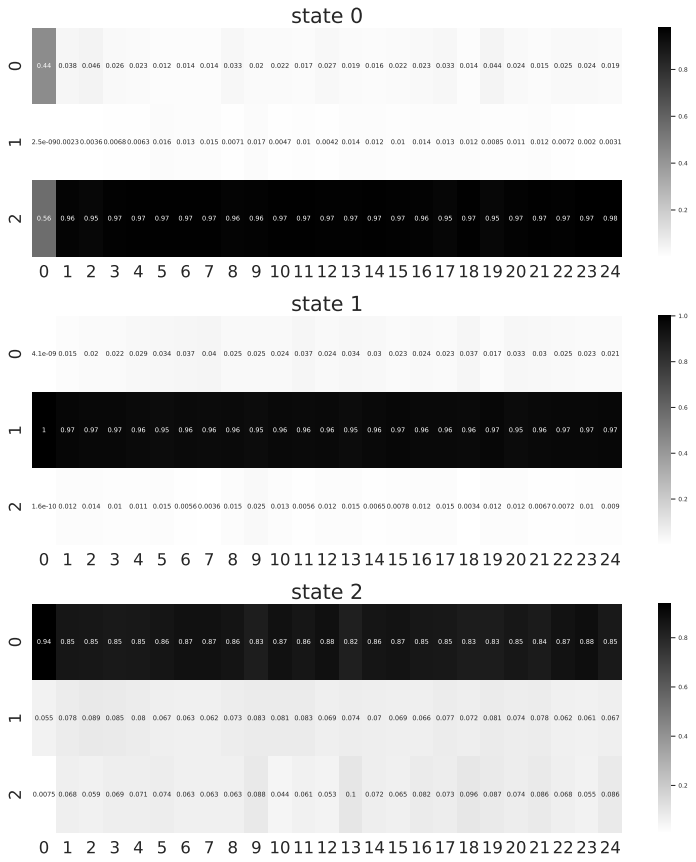


Figure 8: Visualization of decoders from O-REPTRANSFER

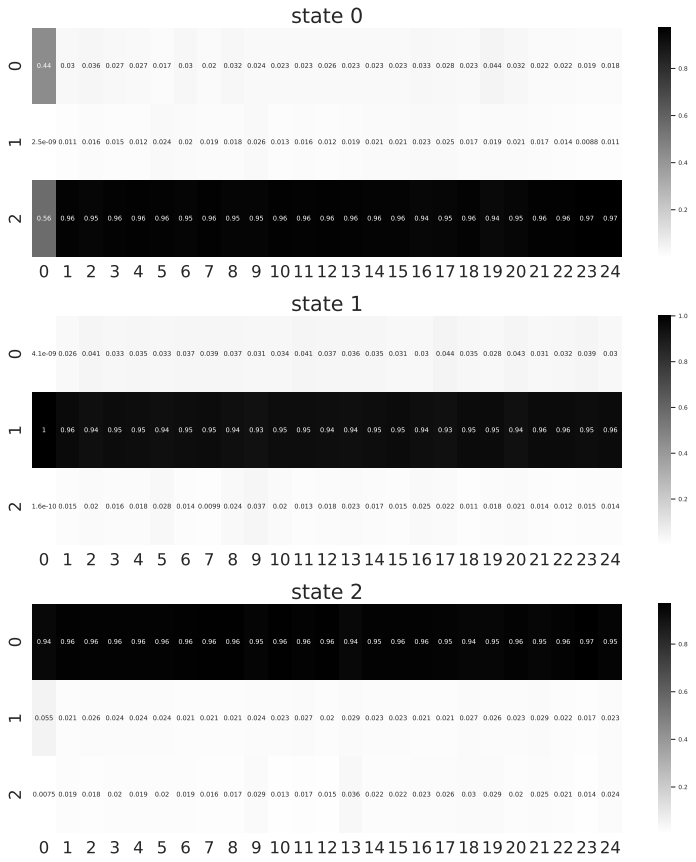


Figure 9: Visualization of decoders from G-REPTRANSFER

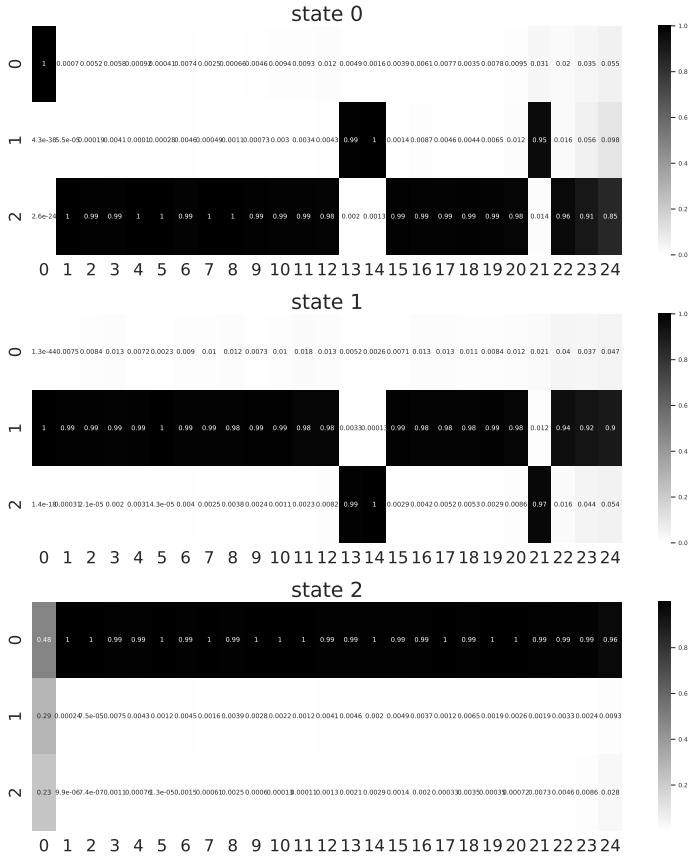


Figure 10: Visualization of decoders from running BRIEE on target.

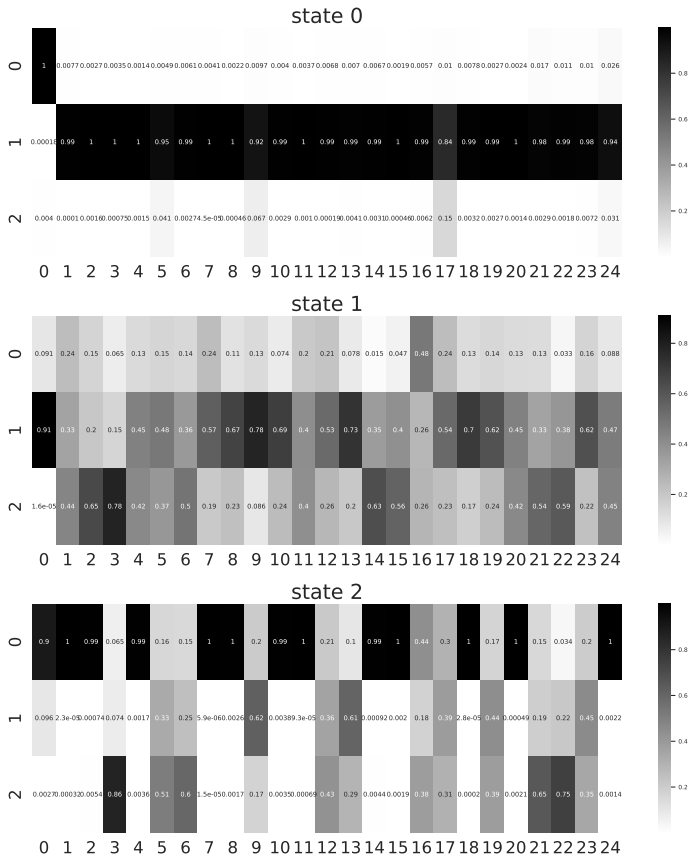


Figure 11: Visualization of decoders from source environment 1.

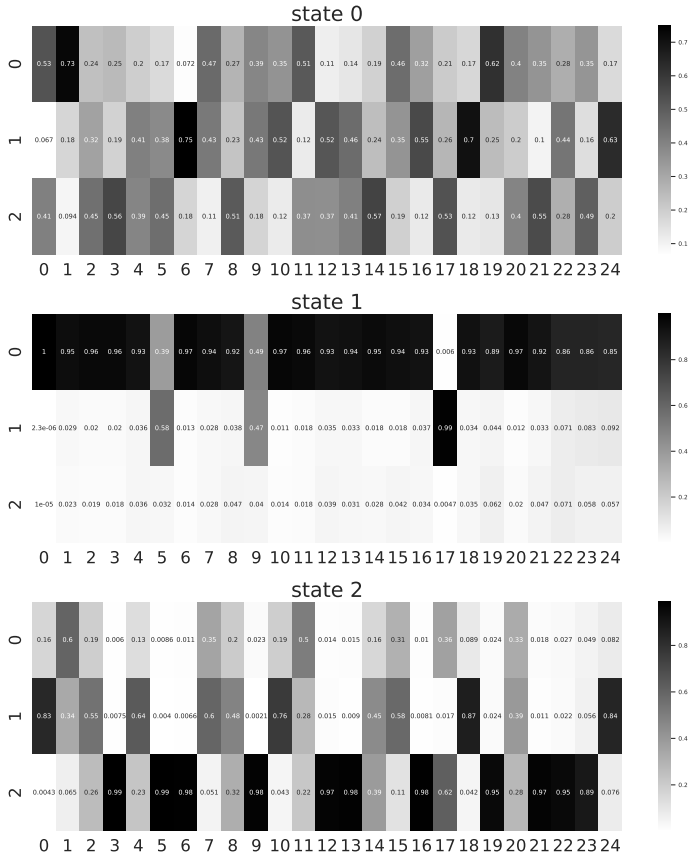


Figure 12: Visualization of decoders from source environment 2.

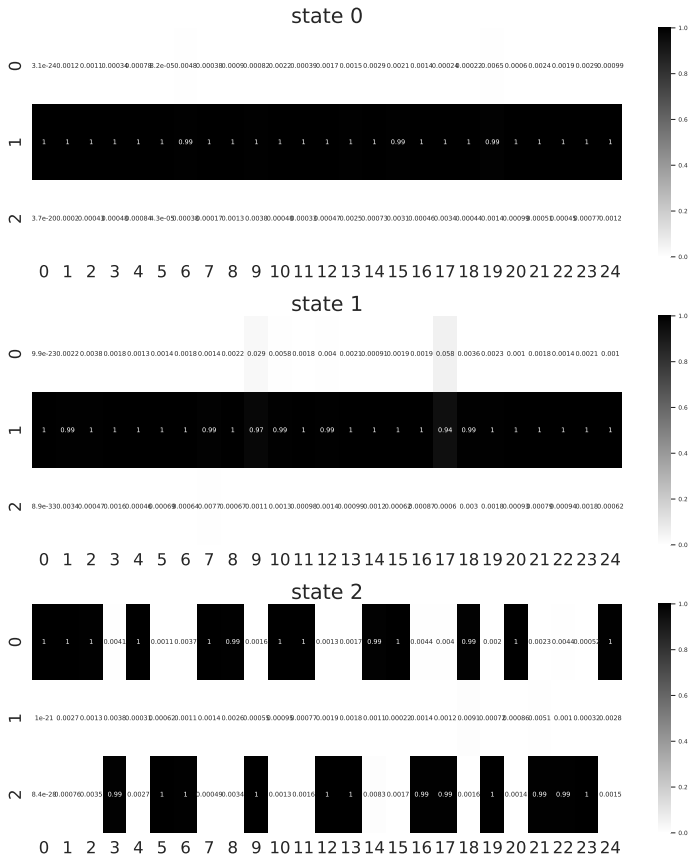


Figure 13: Visualization of decoders from O-REPTRANSFER.

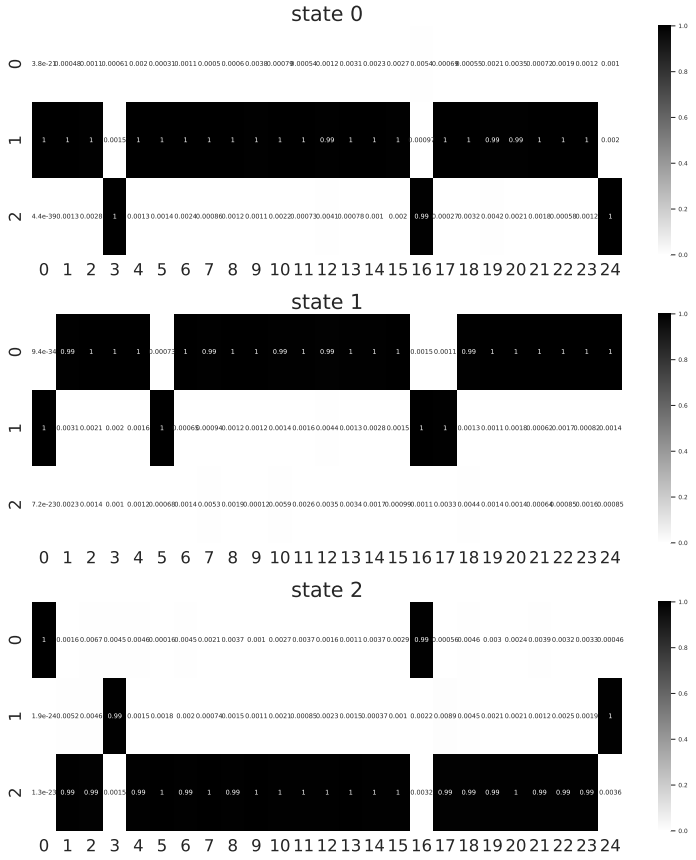


Figure 14: Visualization of decoders from G-REPTRANSFER.

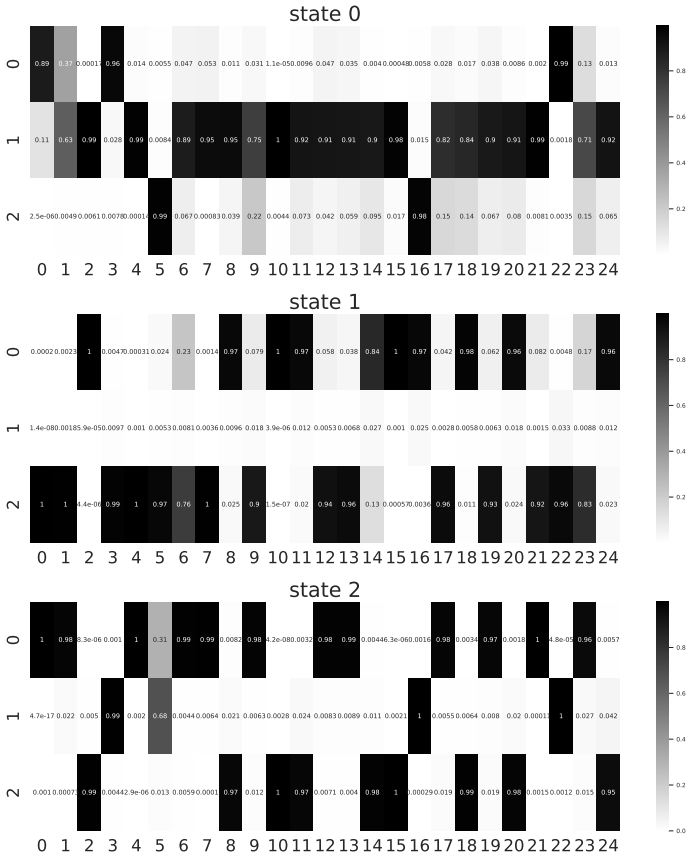


Figure 15: Visualization of decoders running BRIEE in the target.