Offline Multi-task Transfer RL with Representational Penalization

Avinandan Bose
University of Washington
avibose@cs.washington.edu

Simon Shaolei Du
University of Washington
ssdu@cs.washington.edu

Maryam Fazel
University of Washington
mfazel@uw.edu

Abstract

We study the problem of representational transfer in offline Reinforcement Learning (RL), where a learner has access to episodic data from a number of source tasks collected a priori, and aims to learn a shared representation to be used in finding a good policy for a target task. Unlike in online RL where the agent interacts with the environment while learning a policy, in the *offline* setting there cannot be such interactions in either the source tasks or the target task; thus multi-task offline RL can suffer from incomplete coverage.

We propose an algorithm to compute pointwise uncertainty measures for the learnt representation in low-rank MDPs, and establish a data-dependent upper bound for the suboptimality of the learnt policy for the target task. Our algorithm leverages the collective exploration done by source tasks to mitigate poor coverage at some points by a few tasks, thus overcoming the limitation of needing uniformly good coverage for a meaningful transfer by existing offline algorithms. We complement our theoretical results with empirical evaluation on a rich-observation MDP which requires many samples for complete coverage. Our findings illustrate the benefits of penalizing and quantifying the uncertainty in the learnt representation.

1 Introduction

The ability to leverage historical experiences from past tasks and transfer the shared skills to learn a new task with only a few interactions with the environment is a key aspect of machine intelligence. In this paper, we study this goal in the context of multi-task reinforcement learning (MTRL). Multi-task learning has been widely studied across different paradigms. [Caruana, 1997, Pan and Yang, 2009] study a transfer learning scenario where the learner is equipped with data from various source tasks during a pre-training phase. The objective is to learn features easily adaptable to a designated target task. Similar problems are also studied in meta-learning [Finn et al., 2017], lifelong learning [Parisi et al., 2019] and curriculum learning [Liu et al., 2021]. The effectiveness of representation transfer for RL has also been studied in Xu et al. [2020], Zhang et al. [2022], Mitchell et al. [2021], Kumar et al. [2022].

Notably, in all these applications, task datasets are available to the learner a priori. On the theoretical side, there has been a recent surge in emphasis on representation learning questions, driven by their practical significance in both supervised learning and reinforcement learning (RL). While the results in the supervised learning setup [Du et al., 2020, Tripuraneni et al., 2021, Sun et al., 2021] can work in the offline setting with the assumption that data was collected independently and identically from the underlying distributions, in RL data collection is tied to the deployed policy. The main focus has been on the online setting where the learner is able to interact with source tasks to construct datasets with good "coverage" by exploring extensively. Several recent papers study reward-free representation transfer learning [Jin et al., 2020a, Zhang et al., 2020b, Wang et al., 2020a, Misra et al., 2020, Agarwal et al., 2020, Modi et al., 2021, Agarwal et al., 2023]. These approaches are well-suited for scenarios with efficient data generators, such as game engines [Bellemare et al., 2013] and physics simulators [Todorov et al., 2012], serving as environments.

Online RL is harder in safety-critical domains, like precision medicine [Gottesman et al., 2019], autonomous driving [Shalev-Shwartz et al., 2016] and ride-sharing [Bose and Varakantham, 2021], where interactive data collection processes can be costly and risky. Offline

Proceedings of the 28[th] International Conference on Artificial Intelligence and Statistics (AISTATS) 2025, Mai Khao, Thailand. PMLR: Volume 258.

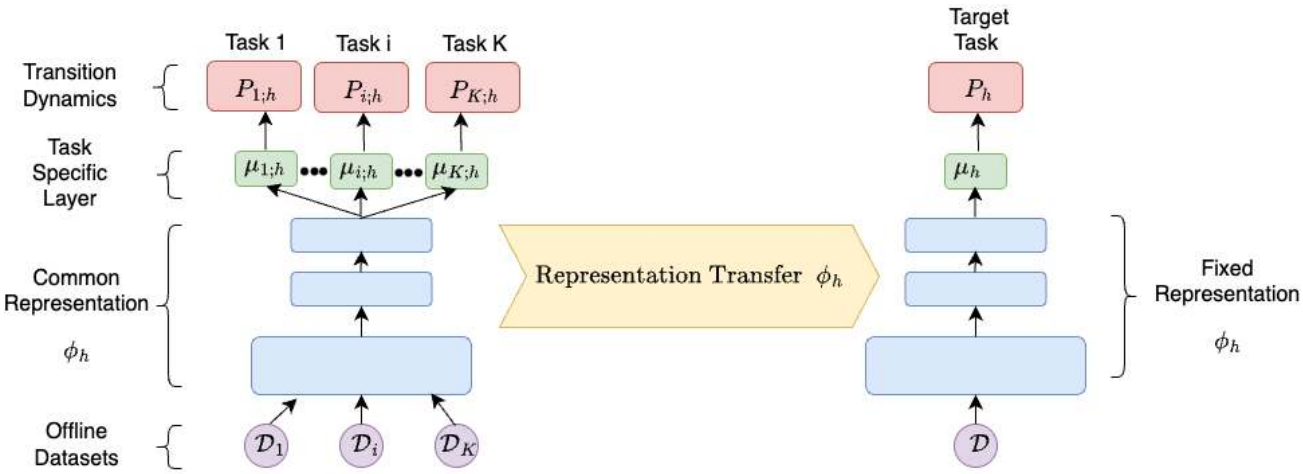


Figure 1: The learner has access to offline datasets from K source tasks and one target task all of which are modelled as Low-rank MDPs. First a common representation is learned across all source tasks, and keeping this representation fixed, the learner plans a near optimal policy using the target task's dataset.

datasets are often abundantly available, e.g., electronic health records for precision medicine [Chakraborty and Murphy, 2014], human driving trajectories for autonomous driving [Sun et al., 2020]. However, guarantees for current algorithms in offline RL exist under strong assumptions as discussed in [Levine et al., 2020, Lange et al., 2012, Wang et al., 2020b], which don't hold true on existing datasets.

In this paper we wish to ask the following question: *Can we design a provably sample efficient algorithm for offline MTRL under minimal assumptions on the datasets?*

We answer this question in the affirmative by introducing a novel algorithm with a theoretical analysis under suitable assumptions including a low-rank MDP model. We list our contributions below.

1. We address a bottleneck in offline MTRL for low-rank MDPs (Definition 2.1) by quantifying data-dependent pointwise uncertainty, while trying to model the target task transition dynamics with the representation learnt from source tasks (cf. Theorem 3.1). Quantifying pointwise uncertainty has not been addressed before even in single-task offline RL in low-rank MDPs due to non-linear function approximation [Uehara et al., 2021].
2. Inspired by ideas in non-parametric estimation, we introduce a quantity termed effective occupancy density (cf. Algorithm 1) which captures the coverage of a certain state-action pair across all source datasets. We show that representation transfer error scales inversely with the square root of the effective occupancy density (cf. Theorem 3.1). Our results show that extensively exploring *every state-action pair* for *every source task* is *not necessary* for uniformly low error for the representational transfer; failure to explore certain state-action pairs by some task can be balanced out by the exploration done by other tasks (cf. Corollary 3.1).
3. We derive a data-dependent bound on the suboptimality of the learnt policy for the target task (cf. Theorem 4.1) highlighting three key factors affecting the success of the process (i) source tasks' coverage of target task's optimal policy, (ii) source tasks' coverage of the offline samples from the target task, (iii) target task's coverage of its optimal policy.
4. We show that under mild assumptions on the policy collecting the data, the learner can achieve a near-optimal target policy by constructing source datasets of size polynomial in the covering number of the low-rank representation space, and target dataset only polynomial in the dimension of the representation. This allows leveraging typically available vast historical data from several source tasks and then performing few shot learning for the target task (cf. Corollary 4.1).
5. We empirically validate our algorithm on the benchmark in [Misra et al., 2020], and demonstrate that baselines without penalising the representation transfer end up with suboptimal cumulative rewards.

Related Work The main challenge in offline RL is insufficient dataset coverage, leading to distribution shift between trajectories in the dataset and those induced by the optimal policy [Wang et al., 2020b, Levine et al., 2020]. This issue is exacerbated by overparameterized function approximators, such as deep neural networks,

	Representation Learning	Multi-task	Purely offline (No access to MDPs)	Target Dataset Policy Independent
Agarwal et al. [2023]	✓	✓	✗	✗
Cheng et al. [2022]	✓	✓	✗	✗
Jin et al. [2021]	✗	✗	✓	✓
Our paper	✓	✓	✓	✓

Table 1: A comparison with lines of work closest to ours. [Jin et al., 2021]'s algorithm works for linear MDPs (assume a known representation) and is also for a single task. While both [Cheng et al., 2022, Agarwal et al., 2023] use maximum likelihood estimate on source datasets to compute a feature estimate (see Eq. (1)), both works assume access to underlying source task MDPs to construct well covered datasets and thus provide uniform representation transfer error guarantees. While [Agarwal et al., 2023] is purely online for target task, [Cheng et al., 2022] give results for downstream offline task with restrictive coverage assumptions on the policy collecting target trajectories. A detailed comparison is presented in Appendix A.

causing extrapolation errors on less covered states and actions [Fujimoto et al., 2019]. Theoretical study of offline RL typically requires one of these assumptions (i) the ratio between the visitation measure of the optimal policy and that of the data collecting policy to be upper bounded uniformly over the state-action space or (ii) the concentrability coefficient defined as the supremum of a similarly defined ratio over the state-action space needs to be upper bounded.

Recent algorithms proposed in [Yu et al., 2020, Kidambi et al., 2020, Kumar et al., 2020, Liu et al., 2020, Buckman et al., 2020, Jin et al., 2021] provably work without any coverage assumptions by penalizing the exploration in offline datasets. The work closest to ours is [Jin et al., 2021] who bound the suboptimality of the learnt policy in terms of an uncertainty quantifier for the limited exploration. For a special instance of low-rank MDPs where the representation is assumed to be known (linear MDP [Jin et al., 2020b]), [Jin et al., 2021] algorithmically construct an uncertainty quantifier. Our setup has the additional challenge of estimating the *unknown* representation, and bounding the suboptimality of the learnt policy in terms of insufficient coverage in the datasets used to learn the representation, as well as the dataset used to learn the policy. As discussed in [Uehara et al., 2021], the non-linear function approximation in Low-rank MDPs as opposed to linear MDPs makes the uncertainty quantification very challenging. Thus our techniques are of independent interest even for the single task offline RL in low-rank MDPs. More discussion on related work is deferred to Appendix A.

2 Preliminaries

In this paper, we study transfer learning in finite-horizon episodic Markov Decision Processes (MDPs), $\mathcal{M} = \langle H, \mathcal{S}, \mathcal{A}, \{P_h\}_{1:H}, \{r_h\}_{1:H}, d_1\rangle$, specified by the episode length or horizon H, state space $\mathcal{S}$, action space $\mathcal{A}$, *unknown* transition dynamics $P_h : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$, *known* reward function $r_h : \mathcal{S} \times \mathcal{A} \to [0, 1]$ and a *known* initial state distribution d_1. For any Markov policy $\pi : \mathcal{S} \to \mathcal{A}$, we use the shorthand notation $\mathbb{E}_{P,\pi}$ to denote the expectation under the distribution of the trajectory induced by executing the policy π in an MDP with transition dynamics $P = \{P_h\}_{1:H}$, i.e., start at an initial state $s_1 \sim d_1$, then for all $h \in [H]$, $a_h \sim \pi_h(s_h)$, $s_{h+1} \sim P_h(\cdot|s_h, a_h)$. The value function is the expected reward of a policy π starting at state s in step h, i.e., $V^{\pi}_{P,r;h}(s) = \mathbb{E}_{P,\pi}[\sum_{\tau=h}^{H} r_\tau(s_\tau, a_\tau)|s_h = s]$. The Q-function is $Q^{\pi}_{P,r;h}(s, a) = r_h(s, a) + \mathbb{E}_{s'\sim P_h(\cdot|s,a)} V^{\pi}_{P,r;h+1}(s')$. The expected total reward of a policy π is defined as $V^{\pi}_{P,r} = \mathbb{E}_{s_1\sim d_1} V^{\pi}_{P,r;1}(s_1)$ and the optimal policy π^* is denoted as the policy maximizing the expected total reward, i.e., $\pi^* = \text{argmax}_\pi V^{\pi}_{P,r}$. Our focus in this paper is on a special class of MDPs formalized below.

Definition 2.1. *(Low-Rank MDP [Jiang et al., 2017], [Agarwal et al., 2020]): A transition model $P_h : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ is low-rank with dimension d if there exist two unknown embedding functions $\phi_h : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$, $\mu_h : \mathcal{S} \to \mathbb{R}^d$ such that $\forall s, s' \in \mathcal{S}, a \in \mathcal{A}$: $P_h(s'|s, a) = \phi_h(s, a)^\top \mu_h(s')$, where $\|\phi_h(s, a)\|_2 \leq 1$ for all (s, a) and for any function $g : \mathcal{S} \to [0, 1]$, $\|\int g(s)\mu_h(s)\mathbf{d}(s)\|_2 \leq \sqrt{d}$. An MDP is low rank if P_h allows such a decomposition for all $h \in [H]$.*

Low-rank MDPs capture several classes of MDPs such as the latent variable model [Agarwal et al., 2020] where $\phi(s, a)$ is a distribution over a discrete latent state space $\mathcal{Z}$, and the block-MDP model [Du et al., 2019] where $\phi(s, a)$ is a one-hot encoding vector. Note that since ϕ can be a non-linear, flexible function class, the low-rank framework generalizes prior works with linear representations [Hu et al., 2021], [Yang et al., 2020a], [Yang et al., 2022].

The setup involves K source tasks and one target task all of which can be modeled as low rank MDPs. The

learning process can be classified into 2 steps: (i) Representation Learning: The learner learns a shared representation across all the source tasks, (ii) Planning: With the learnt representation, the learner plans a good policy for the target task. We first list a few common structural assumptions on the tasks which are needed for a meaningful representation transfer.

Assumption 2.1. *(Common Representation): All tasks share a common representation $\phi_h^\star(s,a)$.*

We denote the next state feature maps for the target task as μ_h^* and for the source tasks as $\{\mu_{1;h}^*, \ldots, \mu_{K;h}^*\}$.

Assumption 2.2. *(Pointwise Linear Span [Agarwal et al., 2023]) For any $h \in [H]$ and $s' \in \mathcal{S}$, there exists a vector $\alpha_h(s') \in \mathbb{R}^K$, such that $\mu_h^*(s') = \sum_{i\in[K]} \alpha_{i;h}(s')\mu_{i;h}^*(s')$, and $\alpha_{\max} = \max_{h;i;s'\in[H]\times[K]\times[S]} \alpha_{i;h}(s')$ is bounded.*

These assumptions capture a large class of MDPs. [Cheng et al., 2022] study unknown source models with the same ϕ^* thus satisfying Assumption 2.1. Assumption 2.2 is a strict generalization of mixture models where the target task transitions are a linear combination of the source tasks dynamics, studied by [Modi et al., 2020],[Ayoub et al., 2020], Block MDPs with shared latent dynamics [Du et al., 2019].

In this paper we consider tasks which can all be modeled as low rank MDPs satisfying Assumptions 2.1, 2.2. We study the offline setting where a learner has access to datasets from K source tasks, each containing N_S episodic trajectories. Let the dataset corresponding to task i be denoted as $\mathcal{D}_i = \{(s_h^{i;\tau}, a_h^{i;\tau})\}_{\tau,h=1}^{N_S,H}$. Since Assumption 2.1 states that all tasks share a common representation, our goal is to first learn a good estimate $\widehat{\phi_h}(s_h,a_h)$ of ϕ_h^* from the available offline data on source tasks and then do few shot offline training on a target task using this learned representation. The learner also has access to a dataset containing n (typically $n \ll N_S$) episodic trajectories from the target task, denoted by $\mathcal{D} = \{(s_h^\tau, a_h^\tau)\}_{\tau,h=1}^{n,H}$. Our main goal is to learn a good policy $\widehat{\pi}$ for the target task using the learnt representation $\widehat{\phi}_h$ that maximizes the expected total reward. The performance metric is defined below.

Definition 2.2. *(Suboptimality Gap): The suboptimality gap for any given policy π and initial state $s \in \mathcal{S}$ is the defined as $SubOpt(\pi, s) = V_1^{\pi^\star}(s) - V_1^{\pi}(s)$, where $\pi^\star$ is the optimal policy.*

In order to state the assumptions on the collected datasets, we begin with the following definition.

Definition 2.3. *(Compliance [Jin et al., 2021]) For a dataset $\mathcal{D} = \{(s_h^\tau, a_h^\tau)\}_{\tau,h=1}^{n,H}$, let $P_{\mathcal{D}}$ be the joint distribution of the data collecting process. We say $\mathcal{D}$ is compliant with the underlying MDP $\mathcal{M}$ if $P_{\mathcal{D}}(s_{h+1}^\tau = s' | \{(s_h^j, a_h^j)\}_{j=1}^\tau, \{s_{h+1}^j\}_{j=1}^{\tau-1}) = P_h(s_{h+1} = s' | s_h = s_h^\tau, a_h = a_h^\tau)$, for all $s' \in \mathcal{S}$ at each step $h \in [H]$ of each trajectory $\tau \in [n]$.*

Definition 2.3 implies that the data collecting process should satisfy the Markov property. At each step $h \in [H]$ of each trajectory $\tau \in [n]$, s_{h+1}^τ depends on $\{(s_h^j, a_h^j)\}_{j=1}^\tau \cup \{s_{h+1}^j\}_{j=1}^{\tau-1}$ only via (s_h^τ, a_h^τ) and the transition dynamics P_h of the underlying MDP $\mathcal{M}$. Thus the randomness in the $\{s_h^j, a_h^\tau, s_{h+1}^j\}_{j=1}^{\tau-1}$ is completely captured by (s_h^τ, a_h^τ) when we examine the randomness in s_{h+1}^τ.

Assumption 2.3. *(Data Collecting Process [Jin et al., 2021]) The offline source and target task datasets the learner has access to are compliant with their respective underlying MDPs.*

Assumption 2.3 is a weak assumption and captures several scenarios. (i) An experimenter collected the data according to a fixed policy, (ii) Experimenter sequentially improved the policy to collect data using any online RL algorithm, thus allowing the trajectories to be interdependent across each other, (iii) Experimenter collected the data by taking actions arbitrarily, say randomly or even any adaptive or adversarial manner and doesn't need to conform to any fixed policy. The important part is that Assumption 2.3 doesn't require the dataset to well explore the entire state-action space which is often the case with offline datasets such as electronic health records or human driving trajectories for autonomous driving.

3 Representation Learning

Recall from Definition 2.1, the transition dynamics of low rank MDPs can be expressed as a function of the representation. In our setting, all the MDPs have a shared representation (Assumption 2.1). Note that Assumption 2.2 implies that the transition dynamics of the target task lies in a linear span of the transition dynamics of the source tasks. Thus obtaining an estimate of the representation from the source tasks significantly reduces the sample complexity in the target task, since it allows the learner to model the transition dynamics of the target task in terms of this learnt representation. In this section we discuss the challenges of obtaining a good representation estimate without any coverage assumptions on the offline datasets and describe our methodology to overcome these challenges.

Learning a Joint Representation In order to learn a joint representation from the source tasks, for every $h \in [H]$ we perform a Maximum Likelihood Estimate (MLE) using the union of data across all source tasks

as follows:

$$\widehat{\mu}_{1:K;h}, \widehat{\phi}_h = \underset{\mu_{1:K}\in\Upsilon, \phi\in\Phi}{\operatorname{argmax}} \sum_{i=1}^{K}\sum_{\tau=1}^{N_S} \log \mu_i^\top(s_{h+1}^{i;\tau})\phi(s_h^{i;\tau}, a_h^{i;\tau}), \tag{1}$$

where Υ and Φ are finite hypothesis classes and we are working in the realizable setting, i.e. $\mu^*_{1:K;h} \in \Upsilon, \phi^*_h \in \Phi$. For special cases where the MDP is tabular or linear, the MLE objective is convex and the optimal solution has closed-form.

Pointwise Uncertainty in Learnt Representation Since we do not assume any coverage conditions on the collected datasets, the representation learnt by Equation (1) is likely to have estimation uncertainties. However, the magnitude of uncertainty for certain state-action pairs might be larger compared to others due to poor exploration. It is therefore desirable to quantify pointwise uncertainty in the estimation which is formally defined below.

Definition 3.1. *(Pointwise Uncertainty in Transition Dynamics) Given an arbitrary transition dynamics $\widehat{P} : \mathcal{S}\times\mathcal{A}\to\mathcal{S}$, its misspecification error at some state action pair $(s,a)\in\mathcal{S}\times\mathcal{A}$ w.r.t. the true transition dynamics P^* is defined as $\Delta_{\widehat{P}}(s,a) = \|\widehat{P}(\cdot|s,a) - P^*(\cdot|s,a)\|_{TV}^2$.*

In the context of low rank setting, the learner estimates the transition dynamics for task i as $\widehat{P}_{i;h}(\cdot|s,a) = \widehat{\mu}_{i;h}^\top(\cdot)\widehat{\phi}_h(s,a)$, where $\widehat{\mu}_{i;h}, \widehat{\phi}_h$ are obtained from Equation (1). As discussed in Uehara et al., 2021 the joint estimation of μ and ϕ in Equation (1) is an instance of non-linear function approximation. Therefore one cannot get pointwise uncertainty quantification via the typically used linear-regression based analysis. Due to this bottleneck, prior works extensively study this problem in the online setting to ensure good exploration and uniform coverage Agarwal et al., 2023 or in the offline scenario by imposing the strict assumption that all source datasets have uniformly explored all state action pairs. This allows for the construction of a uniform confidence bound, i.e. $\epsilon = \min_{(s,a)\in\mathcal{S}\times\mathcal{A}}\Delta_{\widehat{P}}(s,a)$ before transferring this representation for planning in the target task. The magnitude of ϵ impacts the suboptimality of the learnt policy for the target task. However, without uniform coverage assumptions this approach could be detrimental because even one failure mode, i.e. failing to explore some state action-pair even in one source task could lead to a large value of ϵ, rendering the suboptimality of the target task policy meaningless. This motivates us to develop an algorithm to quantify pointwise uncertainty in the transition dynamics estimation.

First we state a guarantee on the estimates in Equation (1). The following lemma states that the sum of the pointwise errors in the transition dynamics averaged over the points in the source datasets is upper bounded with high probability.

Lemma 3.1. *Let $\{\widehat{\mu}_{i;h}\}_{i\in[K]}, \widehat{\phi}_h$ be the learned MLE estimates from Equation (1). Then with probability at least $1-\delta$ we have the following bound:*

$$\underbrace{\sum_{i=1}^{K}\sum_{\tau=1}^{N_S}\frac{\|\widehat{\mu}_{i;h}(\cdot)^\top\widehat{\phi_h}(s_h^{i;\tau},a_h^{i;\tau}) - \mu^*_{i;h}(\cdot)^\top\phi^*_h(s_h^{i;\tau},a_h^{i;\tau})\|_{TV}^2}{N_S}}_{\text{average error on source } i\text{'s dataset}}$$
$$\leq \frac{2(\log(|\Phi|/\delta) + K\log(|\Upsilon|))}{N_S}. \tag{2}$$

It would useful to use the average sense guarantee in Lemma 3.1 to derive pointwise guarantees. To work our way towards this goal, we introduce the following concept.

Algorithm 1 Effective Occupancy Density

1: **Input:** Source Datasets : $\mathcal{D}_{i;h} = \{(s_h^{i;\tau}, a_h^{i;\tau})\}_{\tau=1}^{N_S}$ for all $i\in[K], h\in[H]$, $C = \frac{\log(|\Phi|/\delta)+K\log|\Upsilon|}{N_S d}$.

2: Define ν_i-neighborhood occupancy density

$$D^{\nu_i}_{i;h}(s,a) = \frac{1}{N_S}\inf_{\phi\in\Phi}\max_{\mathcal{C}\subseteq\mathcal{D}_{i;h}}|\mathcal{C}| \text{ such that } \|\phi(s,a)-\phi(s',a')\|_1 \leq \nu_i,\ \forall(s',a')\in\mathcal{C}. \tag{3}$$

3: Solve $\{\nu_1,\dots,\nu_K\}\subseteq\mathbb{R}_+^K$ such that:

$$\min_{i\in[K]} D^{\nu_i}_{i;h}(s,a)\cdot\textstyle\sum_{i\in[K]}\nu_i = C. \tag{4}$$

4: Define effective occupancy density:

$$D_h(s,a) = \frac{C}{\sum_{i\in[K]}\nu_i}. \tag{5}$$

Neighborhood Density: We borrow ideas from non-parametric estimation literature Epanechnikov, 1969, Kaplan and Meier, 1958 where the probability density at some point is estimated based on the observed data in its neighborhood (for, e.g., kernel density estimation Chen, 2017). Since (1) uses non-linear function estimation, we first need to formalize the concept of neighborhood in our setting. The ν_i-neighborhood occupancy density at some (s,a) in the dataset for source task i denoted by $D^{\nu_i}_{i;h}(s,a)$ is the fraction of points in the dataset within a distance of ν_i of (s,a) in the representation space $\mathbb{R}^d$ and is defined in Equation (3). $D^{\nu_i}_{i;h}(s,a)$ essentially *quantifies how well a dataset explores regions around (s,a) in the reprsentation space*. In the following lemma we focus our

attention on quantifying the pointwise uncertainty for source task i, where the transition dynamics is estimated as $\widehat{P}_{i;h}(\cdot|s,a) = \widehat{\mu}_{i;h}(\cdot)^\top \widehat{\phi_h}(s,a)$.

Lemma 3.2. *Let* $\Delta^i_{\widehat{\phi}_h,\widehat{\mu}_{i;h}}(s,a) = \|\widehat{\mu}_{i;h}(\cdot)^\top \widehat{\phi_h}(s,a) - \mu^*_{i;h}(\cdot)^\top \phi^*_h(s,a)\|^2_{TV}$ *denote the transition dynamics misspecification for source task i at time h for any $(s,a) \in \mathcal{S}\times\mathcal{A}$ specified by representations $\widehat{\phi}_h, \widehat{\mu}_{i;h}$ learnt from Equation (1). For some $\nu_i \geq 0$, $\Delta^i_{\widehat{\phi}_h,\widehat{\mu}_{i;h}}(s,a)$ can be upper bounded as follows:*

$$\Delta^i_{\widehat{\phi}_h,\widehat{\mu}_{i;h}}(s,a) \leq \underbrace{2d\cdot\nu_i}_{\text{bias}} + \underbrace{\sum_{(s',a',\cdot)\in\mathcal{D}_{i;h}} \frac{\|\widehat{\mu}_{i;h}(\cdot)^\top \widehat{\phi_h}(s',a') - \mu^*_{i;h}(\cdot)^\top \phi^*_h(s',a')\|^2_{TV}}{N_S D^{\nu_i}_{i;h}(s,a)}}_{\text{variance}}.$$

Note that the variance term is the average error on source i's dataset (Lemma 3.1) divided by the ν_i-neighborhood occupancy density $D^{\nu_i}_{i;h}(s,a)$. Since, $D^{\nu_i}_{i;h}(s,a)$ is a non-decreasing function of ν_i, the variance term is non-increasing in ν_i, whereas the bias term is increasing in ν_i. Thus there is a bias-variance tradeoff in choosing ν_i. We utilize this idea in Algorithm 1, which solves an optimization problem Equation (4) to optimally balance out the total variance and bias across all source tasks to return the effective occupancy density $D_h(s,a)$, as defined in Equation (5). Now, we are ready to state our main result and provide a proof sketch to highlight the main ideas.

Algorithm 2 Pessimisitc RepTransfer (PRT)

Input: Target Dataset $\mathcal{D} = \{(s^\tau_h, a^\tau_h)\}^{n,H}_{\tau,h=1}$, Learnt Representation $\widehat{\phi}_h(\cdot,\cdot)$, RepTransfer bound $\epsilon(\cdot,\cdot)$ for all points in $\mathcal{D}$, β, λ.
Initialization: Set $\widehat{V}_{H+1}(\cdot) \leftarrow 0$.
for $h = H, H-1, \ldots, 1$ **do**
  Set $\Lambda_h \leftarrow \frac{1}{n}\left(\sum_{\tau=1}^n \widehat{\phi}_h(s^\tau_h, a^\tau_h)\widehat{\phi}_h(s^\tau_h, a^\tau_h)^\top + \lambda\mathbb{I}\right)$.
  Set $\widehat{w}_h \leftarrow \Lambda_h^{-1}\left(\frac{1}{n}\sum_{\tau=1}^n \widehat{\phi}_h(s^\tau_h, a^\tau_h)\cdot \widehat{V}_{h+1}(s^\tau_{h+1})\right)$.
  Set $\epsilon_h \leftarrow \sqrt{\frac{1}{n}\sum_{\tau=1}^n \epsilon(s^\tau_h, a^\tau_h)^2}$
  Set $\Gamma_h(\cdot,\cdot) \leftarrow H(\beta+\epsilon_h)\cdot\|\widehat{\phi}_h(\cdot,\cdot)\|_{\Lambda_h} + H\epsilon(\cdot,\cdot)$.
  Set $\overline{Q}_h(\cdot,\cdot) \leftarrow r_h(\cdot,\cdot) + \widehat{\phi}_h(\cdot,\cdot)^\top \widehat{w}_h - \Gamma_h(\cdot,\cdot)$.
  Set $\widehat{Q}_h(\cdot,\cdot) \leftarrow \min\{\overline{Q}_h(\cdot,\cdot), H-h+1\}^+$.
  Set $\widehat{\pi}_h(\cdot|\cdot) \leftarrow \arg\max_{\pi_h} \widehat{Q}_h(\cdot,\cdot)^\top \pi_h$.
  Set $\widehat{V}_h(\cdot) \leftarrow \mathbb{E}_{a\sim\widehat{\pi}_h(\cdot|\cdot)}\widehat{Q}_h(\cdot,a)$.
end for
Return $\widehat{\pi} = \{\widehat{\pi}_h(\cdot|\cdot)\}^H_{h=1}$.

Theorem 3.1. *(Representation Transfer Error): Let $P^*_h(\cdot|s_h,a_h)$ denote the true transition dynamics of the target task, and $\widehat{\phi_h}(s,a)$ be the learnt representation from Equation (1). For all $h \in [H], (s,a) \in \mathcal{S}\times\mathcal{A}$, there exists $\mu'_h : \mathcal{S} \to \mathbb{R}^d$ such that the following bound holds with probability at least $1-\delta$*

$$\|\mu'_h(\cdot)^\top \widehat{\phi_h}(s,a) - P^*_h(\cdot|s,a)\|_{TV} \leq 2\alpha_{\max}\sqrt{\frac{K}{N_S}\frac{\log(|\Phi|/\delta) + K\log|\Upsilon|}{D_h(s,a)}},$$

where $D_h(s,a)$ is the effective occupancy density as computed in Algorithm 1.

Proof Sketch: We show in Lemma F.2 that there exists a transition model linear in the learnt representation $\widehat{\phi}_h$ such that the model misspecification error for the target task can be upper bounded in terms of the model misspecification errors of the individual source tasks, i.e. $\sum_{i\in[K]} \Delta^i_{\widehat{\phi}_h,\widehat{\mu}_{i;h}}(s,a)$. The sum of the variance terms can be upper bounded with high probability by utilizing the MLE guarantee in Lemma 3.1 with an additional multiplicative factor of the Importance sampling ratio $\frac{1}{\min_{i\in[K]} D^{\nu_i}_{i;h}(s,a)}$. The solution of the optimization problem in Equation (4) in Algorithm 1 optimally balances out the overall variance with the sum of bias terms.

One of the main implications of Theorem 3.1 is that the learner doesn't need to impose the strict assumption that every source task has extensively explored every state action pair in order to have a uniformly low representation transfer error. In fact in the following corollary we present a much more relaxed yet sufficient condition to ensure uniformly low estimation error. If for some $\nu' \in (0,1]$ and every $(s,a) \in \mathcal{S}\times\mathcal{A}$ the following optimization problem admits a feasible solution:

$$\left(\frac{1}{K}\sum_{i\in[K]}\frac{1}{D^{\nu_i}_{i;h}(s,a)}\right)^{-1} \geq \frac{\log(|\Phi|/\delta) + K\log|\Upsilon|}{N_S\cdot d\cdot\nu'} \quad \text{such that} \quad \frac{1}{K}\sum_{i\in[K]}\nu_i \leq \nu', \tag{6}$$

then the representation error is uniformly upper bounded and scales as $\sqrt{\nu'}$ as formalized below.

Corollary 3.1. *Let $P^*_h(\cdot|s_h,a_h)$ denote the true transition dynamics of the target task, and $\widehat{\phi_h}(s,a)$ be the learnt representation from Equation (1). For all $h \in [H], (s,a) \in \mathcal{S}\times\mathcal{A}$, there exists $\mu'_h : \mathcal{S} \to \mathbb{R}^d$ such that the following bound holds with probability at least $1-\delta$*

$$\|\mu'_h(\cdot)^\top \widehat{\phi_h}(s,a) - P^*_h(\cdot|s,a)\|_{TV} \leq 2\alpha_{\max}K\sqrt{d\cdot\nu'},$$

under the condition in Equation (6).

Thus we only need the harmonic means of the neighborhood densities to be lower bounded under an upper bounded average neighborhood size in order to get a uniformly low representation transfer error.

4 Representational Transfer in Target Task

In this section we present **P**essimistic**R**ep**T**ransfer (Algorithm 2) for policy planning in the target task using the learnt representation. Our algorithm is based on the idea of pessimism, i.e. penalizing the Q-function estimate for each (s,a) based on how uncertain the estimate is. The idea of pessimism in offline RL is very classical and has been explored in contextual bandits, tabular MDPs [Rashidinejad et al., 2021], and very recently for linear MDPs [Jin et al., 2021]. Ours is the first work for Low-rank MDPs. A detailed comparison to [Jin et al., 2021] is stated in Appendix A, and a description of Algorithm 2 can be found in Appendix B. In comparison with prior work which compute uncertainty estimates only for exploration in the downstream task, our setting has the additional challenge of bounding the uncertainty in learnt representation $\widehat{\phi}_h$ from upstream tasks, and we see how our novel techniques introduced in Section 3 allows us to state guarantees on the quality of the target task policy.

Theorem 4.1. *Let $\widehat{\pi}$ be the output of Algorithm 2. Then with probability at least $1-\delta$*

$$SubOpt(\widehat{\pi}, s) \le 2H \sum_{h=1}^{H} \mathbb{E}_{(s_h,a_h)\sim\pi^*,P_h^*}\Bigg[\underbrace{\epsilon(s_h,a_h)}_{\text{source coverage on } \pi^*} + (\beta + \underbrace{\epsilon_h}_{\text{source coverage on target}}) \cdot \underbrace{\|\widehat{\phi}_h(s_h,a_h)\|_{\Lambda_h}}_{\text{target coverage on } \pi^*} \mid s_1 = s \Bigg].$$

Here the expectation is taken with respect to the optimal policy $\pi^\star$ of the true underlying MDP of the target task. $\epsilon(s,a) = 2\alpha_{\max}\sqrt{K\frac{\log(2|\Phi|/\delta)+K\log|\Upsilon|}{N_S D_h(s,a)}}$, $\beta = c\cdot d\sqrt{\zeta}$, where $\zeta = \frac{\log(4dHn/\delta)}{n}$ and $c \ge 1$ is an absolute constant and $\epsilon_h = \sqrt{\frac{1}{n}\sum_{\tau=1}^{n}\epsilon(s_h^\tau,a_h^\tau)^2}$.

Below we discuss the factors affecting the suboptimality of the learnt target policy:

1. **Source Tasks' Coverage on Target Task's Optimal Policy π^*:** The source tasks' should have sufficient samples along the trajectory of the optimal policy of the target task: $\sum_{h\in[H]}\mathbb{E}_{(s_h,a_h)\sim\pi^*,P_h^*}\sqrt{1/D_h(s_h,a_h)}$.
2. **Source Tasks' Coverage on the offline samples from the Target Task :** Let $d_h(\cdot,\cdot)$ denote the target task's occupancy density based on the offline dataset $\mathcal{D}_h = \{s_h^\tau, a_h^\tau\}_{\tau=1}^n$. Evaluating the term ϵ_h we get: $\epsilon_h \propto \sqrt{\sum_{\tau\in[n]} 1/nD_h(s_h^\tau,a_h^\tau)} = \sqrt{\sum_{(s,a)\in\mathcal{S}\times\mathcal{A}} d_h(s,a)/D_h(s,a)}$. Note that ϵ_h doesn't depend on π^* or P_h^*. In order for the representational transfer to be effective, this term implies that the source tasks' must have sufficient coverage at all points covered in the target task.
3. **Target Task's Coverage on its Optimal Policy π^* :** $\Lambda_h = \frac{1}{n}\left(\sum_{\tau=1}^n \widehat{\phi}_h(s_h^\tau,a_h^\tau)\widehat{\phi}_h(s_h^\tau,a_h^\tau)^\top + \lambda\mathbb{I}\right)$ indicates the empirical covariance of the samples from the target task. For any arbitrary (s_h,a_h), the term $\|\widehat{\phi}_h(s_h,a_h)\|_{\Lambda_h} = \widehat{\phi}_h(s_h,a_h)^\top\Lambda_h^{-1}\widehat{\phi}_h(s_h,a_h)$ indicates how well (s_h,a_h) is covered by the offline samples from the target dataset. The suboptimality gap depends on how well the offline samples from the target task covers the trajectory of the taget task's optimal policy, i.e., $2H\sum_{h\in[H]}\mathbb{E}_{(s_h,a_h)\sim\pi^*,P_h^*}(\beta+\epsilon_h)\cdot\|\widehat{\phi}_h(s_h,a_h)\|_{\Lambda_h}$.

Remark 4.1. *The computational complexity of Algorithm 2 to obtain the guarantee in Theorem 4.1 is $\mathcal{O}(KN_Sn)$. The derivation and wall-clock run times can be found in Appendix C.*

Well Explored Source and Task Datasets We wish to study the suboptimality rates as a function of the number of source and target task samples. We examine this under the assumption that the data collecting process work with well exploratory policies, formally defined below.

Assumption 4.1. *(Bounded Density in Representation Space) Let $\pi_i = \{\pi_{i,1},\dots,\pi_{i,H}\}$ denote the policy that collects offline data for source task i. A feature map $\phi\in\Phi$ defines a distribution $d_{i,h}^{\pi_i,\phi}(\cdot)$ in the representation space $\mathbb{R}^d$. We assume that there exists policy $\bar{\pi}_i$ such that we can lower bound the density in the representation space, i.e.*

$$\inf_{\phi\in\Phi}\inf_{x\in\mathbb{R}^d} d_{i,h}^{\bar{\pi}_i,\phi}(x) \ge \psi \quad \text{and} \quad \sup_{\phi\in\Phi}\sup_{x\in\mathbb{R}^d} d_{i,h}^{\bar{\pi}_i,\phi}(x) \le 1,$$

for all $h\in[H]$.

Note that by Definition 2.1, every feature map $\phi\in\Phi$, satisfies $\|\phi(s,a)\|_2 \le 1\ \forall(s,a)\in\mathcal{S}\times\mathcal{A}$. Thus the representation space in $\mathbb{R}^d$ is the unit ℓ_2 norm ball in d dimensions, $\mathcal{B}_2^d$ which is a compact set. Assumption 4.1 thus states existence of policy with bounded density only on a compact set instead of the raw state action space which can be infinite.

Assumption 4.2. *[Jin et al., 2021] There exists a policy $\bar{\pi} = \{\bar{\pi}_1,\dots,\bar{\pi}_H\}$ for the target task such that*

$$\inf_{\phi\in\Phi}\lambda_{\min}(\Sigma_h^\phi) \ge c/d \text{ where } \Sigma_h^\phi = \mathbb{E}_{(s_h,a_h)\sim\bar{\pi},P^*}[\phi_h(s,a)\phi_h(s,a)^\top].$$

The following corollary gives a high probability bound on the target policy suboptimality as a function of the number of source and target task samples.

Corollary 4.1. *Let $\bar{\pi}$ be a policy satisfying Assumption 4.2 and $\{\bar{\pi}_1, \ldots, \bar{\pi}_K\}$ be policies satisfying Assumption 4.1. Suppose n i.i.d. trajectories are sampled from the target task by policy $\bar{\pi}$ and N_S i.i.d. trajectories are sampled from each task i by policy π_i. Then with probability at least $1-\delta$, the suboptimality gap is upper bounded as:*

$$SubOpt(\widehat{\pi}, s) \leq \tilde{\mathcal{O}}(\max({N_S}^{-\frac{1}{4d}}, n^{-\frac{1}{2}}) H^2 d^{\frac{3}{2}} K^{\frac{3}{4}} \sqrt{\log(1/\delta)}).$$

5 Experiments

In this section we empirically study[1] the benefits of penalizing the learnt representation in offline Multi-Task Transfer RL. We ask the following questions:

1. Does uncertainty quantification in the learnt representation reduce sample complexity of both source and task datasets?
2. Does running online algorithms such as UCB with inaccurate representation lead to convergence to suboptimal target policies?
3. Does our algorithm outperform baselines irrespective of the data collection policies for source and target tasks?

Our experiments suggest affirmative answer to all the questions above. We use the high dimesnional rich observation Combination Lock (comblock) benchmark (see Table 2).

Baselines: All baselines considered in our study leverage the representation learned from source tasks' offline datasets, obtained through Maximum Likelihood Estimation as described in Equation (1). The algorithm employed for the target task varies across these baselines and a complete description is provided in Appendix D. **RT-L** uses the LSVI (Least Squares Value Iteration) algorithm [Sutton and Barto, 2018], **RT-P** uses the Pessimistic Value Iteration (PEVI) algorithm [Jin et al., 2021], **PRT** denotes our Algorithm 2. These 3 are purely offline algorithms designed to work with the target task's offline dataset. **RT-LU** uses the learnt representation like the other baselines, but then can adaptively collect samples from the target task using the Upper Confidence Bound (UCB) algorithm [Sutton and Barto, 2018]. In Table 2, we vary the number of source trajectories N_S and target trajectories n, reporting the average reward (over 50 runs) for all baselines. For **RT-LU**, n is the number of trajectories for the algorithm to converge and is reported in parenthesis (we terminate when $n = 50000$ if it fails to converge).

Offline Dataset Construction: We run our experiments with 2 types of data collecting policies: **(i) Exploratory:** We use the Exploratory Policy Search (EPS) Algorithm proposed by [Agarwal et al., 2023] to identify exploratory policies for the source and target task. *Note that exploratory policies aim to cover as much of the feature space and are potentially very different from the optimal policy.* **(ii) Optimal / Expert Demonstrations:** The dataset comprises of trajectories optimally solving the task.

We independently and identically sampled N_S trajectories from source task and n trajectories from the target task to construct 3 types of offline datasets: **(a)** Exploratory Source and Optimal Target (Table 2), **(b)** Exploratory Source and Exploratory Target and **(c)** Optimal Source and Optimal Target (Table 3).

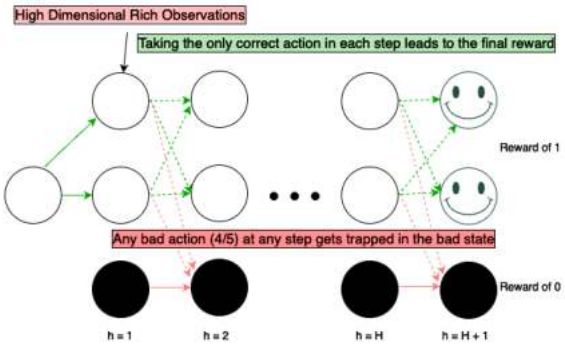
Results: In scenarios where the source tasks benefit from well explored datasets (i.e., large N_S and trajectories from Exploratory policies), the representation transfer error is uniformly low. All algorithms demonstrate strong performance under these well-covered conditions, as evident in Row 3 of Table 3 (left). Our focus, however, lies in situations where the source datasets are less explored, indicating a small N_S (rows 1-2 of Table 3(left)) or the source trajectories were from expert demonstrations (Table 3(right)). In such cases, our representation transfer penalty becomes crucial for selectively penalizing estimated representations for specific state-action pairs. We observe that both **RT-L** and **RT-P**, assuming the learned representation as ground truth, struggle to reach the optimal solution in these less-explored settings. While **RT-P** performs better than **RT-L** by penalizing points in the target dataset, it still falls short. On the other hand, **RT-LU** fails to recover the optimal policy even after 50000 episodes due to reliance on an inaccurate representation. Notably, our proposed algorithm, **PRT**, stands out as the only method capable of optimally solving the target task. Table 2 also demonstrates our algorithm's ability for few-shot learning when the source trajectories are sampled from exploratory policies and target trajectories are samples from optimal policies.

Wall Clock Times: We report the wall clock times for the MLE estimation step to learn a representation and the whole algorithm 2 under the various settings. We highlight 2 things (I) MLE is much more computationally expensive, (II) As the number of source trajectories N_S grows the additional time needed to construct uncertainty quantifiers is negligible to the gain in time for constructing uncertainty quantifiers. Thus implementation of the proposed method is feasible.

6 Conclusion

We address offline representation transfer in low-rank MDPs with a relaxed assumption that the trajectories

[1] All our code is available at `https://anonymous.4open.science/r/PessimisticRepTransfer-DBDE`



Source (N_S)	Target (n)	RT-L	RT-P	PRT (**Ours**)	RT-LU
250	50	0.21	0.30	**0.51**	0.04
	100	0.30	0.33	**0.55**	(50K)
500	50	0.25	0.35	**0.57**	0.05
	100	0.40	0.43	**0.65**	(50K)
1000	50	0.39	0.73	**0.96**	0.07
	100	0.41	0.81	**1.0**	(50K)

Table 2: **(Left)** A visualization of the rich observation CombLock environment. Our experiment uses $K = 5$ source tasks, $H = 5$ time steps and 5 actions in each step. See Appendix K for details. **(Right)** Average Rewards for CombLock across different algorithms and varying number of samples. We observe that our algorithm enables few shot learning in the target task by being able to recover a near optimal policy (last row) with very few target samples. Note that pre-training representation and fixing it, followed by online downstream learning on target (last column) fails to converge when the learned representation is inaccurate and uncertainty quantification isn't taken into account. This also underscores the poor performance of the purely offline baselines (first 2 columns).

Source (N_S)	Target (n)	RT-L	RT-P	PRT (**Ours**)	RT-LU
500	150	0.39	0.73	**1.0**	0.05 (50K)
	200	0.41	0.81	**1.0**	
	250	0.50	0.89	**1.0**	
1000	150	0.72	0.76	**1.0**	0.07 (50K)
	200	0.86	0.88	**1.0**	
	250	0.94	0.96	**1.0**	
1500	150	0.76	0.76	**1.0**	1.0 (572)
	200	0.88	0.89	**1.0**	
	250	0.96	0.98	**1.0**	

Source (N_S)	Target (n)	RT-L	RT-P	PRT (**Ours**)	RT-LU
500	150	0.16	0.32	**0.57**	0.04 (50K)
	200	0.21	0.41	**0.63**	
	250	0.40	0.43	**0.74**	
1000	150	0.44	0.54	**0.76**	0.05 (50K)
	200	0.60	0.78	**0.88**	
	250	0.72	0.86	**1.0**	
1500	150	0.55	0.76	**0.92**	0.76 (50K)
	200	0.65	0.89	**0.96**	
	250	0.77	0.96	**1.0**	

Table 3: Average Rewards for CombLock across different algorithms and varying number of samples **(Left)** Exploratory source and target policies, **(Right)** Optimal Source and Target policies.

N_S	n	MLE	PRT
500	150	332	2.85
500	200	332	3.78
500	250	332	4.72
1000	150	362	5.43
1000	200	362	7.42
1000	250	362	9.27
1500	150	450	8.26
1500	200	450	10.93
1500	250	450	12.77

Table 4: Wall clock time for MLE and PRT (Algorithm 2) as number of source and target samples is varied.

comply with the underlying MDPs, and contribute an algorithm for pointwise uncertainty quantification of the learned representation, demonstrating through theory and experiments that incorporating uncertainty improves the target policy, with future work focusing on source task selection and active error reduction in an online setting.

Future Work. Working completely in the offline setting means the learner incurs an irreducible sub-optimality from the error in the learnt representation. However, if the learner had online access to only the target task, then theoretical analysis of actively reducing the representation error is an interesting direction of future work. This is particularly useful in the RL finetuning of pre-trained language models for specific tasks [Bose et al., 2024b], [Bhatt et al., 2024]. Another promising direction of future work is the problem of source task selection. Typically domain experts are needed to select source tasks relevant for the corresponding target task. However, with the availability of offline datasets from a large number of source tasks available online necessitates principled approaches to select a small subset of tasks that are relevant to the target task [Chen et al., 2022, Bose et al., 2024a].

References

Alekh Agarwal, Sham Kakade, Akshay Krishnamurthy, and Wen Sun. Flambe: Structural complexity and representation learning of low rank mdps. *Advances in neural information processing systems*, 33: 20095–20107, 2020.

Alekh Agarwal, Yuda Song, Wen Sun, Kaiwen Wang, Mengdi Wang, and Xuezhou Zhang. Provable benefits of representational transfer in reinforcement learning. In *The Thirty Sixth Annual Conference on Learning Theory*, pages 2114–2187. PMLR, 2023.

András Antos, Csaba Szepesvári, and Rémi Munos. Fitted q-iteration in continuous action-space mdps. *Advances in neural information processing systems*, 20, 2007.

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.

Marc G Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47:253–279, 2013.

Gantavya Bhatt, Yifang Chen, Arnav M Das, Jifan Zhang, Sang T Truong, Stephen Mussmann, Yinglun Zhu, Jeffrey Bilmes, Simon S Du, Kevin Jamieson, et al. An experimental design framework for label-efficient supervised finetuning of large language models. *arXiv preprint arXiv:2401.06692*, 2024.

Avinandan Bose and Pradeep Varakantham. Conditional expectation based value decomposition for scalable on-demand ride pooling. *arXiv preprint arXiv:2112.00579*, 2021.

Avinandan Bose, Mihaela Curmei, Daniel Jiang, Jamie H Morgenstern, Sarah Dean, Lillian Ratliff, and Maryam Fazel. Initializing services in interactive ml systems for diverse users. *Advances in Neural Information Processing Systems*, 37:57701–57732, 2024a.

Avinandan Bose, Zhihan Xiong, Aadirupa Saha, Simon Shaolei Du, and Maryam Fazel. Hybrid preference optimization for alignment: Provably faster convergence rates by combining offline preferences with online exploration. *arXiv preprint arXiv:2412.10616*, 2024b.

Jacob Buckman, Carles Gelada, and Marc G Bellemare. The importance of pessimism in fixed-dataset policy optimization. *arXiv preprint arXiv:2009.06799*, 2020.

Rich Caruana. Multitask learning. *Machine learning*, 28:41–75, 1997.

Bibhas Chakraborty and Susan A Murphy. Dynamic treatment regimes. *Annual review of statistics and its application*, 1:447–464, 2014.

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

Yen-Chi Chen. A tutorial on kernel density estimation and recent advances. *Biostatistics & Epidemiology*, 1 (1):161–187, 2017.

Yifang Chen, Kevin Jamieson, and Simon Du. Active multi-task representation learning. In *International Conference on Machine Learning*, pages 3271–3298. PMLR, 2022.

Yuan Cheng, Songtao Feng, Jing Yang, Hong Zhang, and Yingbin Liang. Provable benefit of multitask representation learning in reinforcement learning. *Advances in Neural Information Processing Systems*, 35: 31741–31754, 2022.

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.

Vassiliy A Epanechnikov. Non-parametric estimation of a multivariate probability density. *Theory of Probability & Its Applications*, 14(1):153–158, 1969.

Jianqing Fan, Zhaoran Wang, Yuchen Xie, and Zhuoran Yang. A theoretical analysis of deep q-learning. In *Learning for dynamics and control*, pages 486–489. PMLR, 2020.

Mehrdad Farajtabar, Yinlam Chow, and Mohammad Ghavamzadeh. More robust doubly robust off-policy evaluation. In *International Conference on Machine Learning*, pages 1447–1456. PMLR, 2018.

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.

Scott Fujimoto, David Meger, and Doina Precup. Off-policy deep reinforcement learning without exploration. In *International conference on machine learning*, pages 2052–2062. PMLR, 2019.

Omer Gottesman, Fredrik Johansson, Matthieu Komorowski, Aldo Faisal, David Sontag, Finale Doshi-Velez, and Leo Anthony Celi. Guidelines for reinforcement learning in healthcare. *Nature medicine*, 25(1): 16–18, 2019.

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.

Haque Ishfaq, Thanh Nguyen-Tang, Songtao Feng, Raman Arora, Mengdi Wang, Ming Yin, and Doina Precup. Offline multitask representation learning for reinforcement learning. *Advances in Neural Information Processing Systems*, 37:70557–70616, 2024.

Nan Jiang and Jiawei Huang. Minimax value interval for off-policy evaluation and policy optimization. *Advances in Neural Information Processing Systems*, 33: 2747–2758, 2020.

Nan Jiang and Lihong Li. Doubly robust off-policy value evaluation for reinforcement learning. In *International Conference on Machine Learning*, pages 652–661. PMLR, 2016.

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.

Ying Jin, Zhuoran Yang, and Zhaoran Wang. Is pessimism provably efficient for offline rl? In *International Conference on Machine Learning*, pages 5084–5096. PMLR, 2021.

Nathan Kallus and Masatoshi Uehara. Efficiently breaking the curse of horizon in off-policy evaluation with double reinforcement learning. *Operations Research*, 70(6):3282–3302, 2022.

Edward L Kaplan and Paul Meier. Nonparametric estimation from incomplete observations. *Journal of the American statistical association*, 53(282):457–481, 1958.

Rahul Kidambi, Aravind Rajeswaran, Praneeth Netrapalli, and Thorsten Joachims. Morel: Model-based offline reinforcement learning. *Advances in neural information processing systems*, 33:21810–21823, 2020.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 33:1179–1191, 2020.

Aviral Kumar, Anikait Singh, Frederik Ebert, Mitsuhiko Nakamoto, Yanlai Yang, Chelsea Finn, and Sergey Levine. Pre-training for robots: Offline rl enables learning new tasks from a handful of trials. *arXiv preprint arXiv:2210.05178*, 2022.

Sascha Lange, Thomas Gabel, and Martin Riedmiller. Batch reinforcement learning. In *Reinforcement learning: State-of-the-art*, pages 45–73. Springer, 2012.

Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

Peng Liao, Zhengling Qi, Runzhe Wan, Predrag Klasnja, and Susan A Murphy. Batch policy learning in average reward markov decision processes. *Annals of statistics*, 50(6):3364, 2022.

Boyi Liu, Qi Cai, Zhuoran Yang, and Zhaoran Wang. Neural trust region/proximal policy optimization attains globally optimal policy. *Advances in neural information processing systems*, 32, 2019.

Minghuan Liu, Hanye Zhao, Zhengyu Yang, Jian Shen, Weinan Zhang, Li Zhao, and Tie-Yan Liu. Curriculum offline imitating learning. *Advances in Neural Information Processing Systems*, 34:6266–6277, 2021.

Qiang Liu, Lihong Li, Ziyang Tang, and Dengyong Zhou. Breaking the curse of horizon: Infinite-horizon off-policy estimation. *Advances in neural information processing systems*, 31, 2018.

Yao Liu, Adith Swaminathan, Alekh Agarwal, and Emma Brunskill. Provably good batch off-policy reinforcement learning without great exploration. *Advances in neural information processing systems*, 33: 1264–1274, 2020.

Rui Lu, Andrew Zhao, Simon S Du, and Gao Huang. Provable general function class representation learning in multitask bandits and mdp. *Advances in Neural Information Processing Systems*, 35:11507–11519, 2022.

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.

Eric Mitchell, Rafael Rafailov, Xue Bin Peng, Sergey Levine, and Chelsea Finn. Offline meta-reinforcement learning with advantage weighting. In *International Conference on Machine Learning*, pages 7780–7791. PMLR, 2021.

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.

Rémi Munos and Csaba Szepesvári. Finite-time bounds for fitted value iteration. *Journal of Machine Learning Research*, 9(5), 2008.

Ofir Nachum and Bo Dai. Reinforcement learning via fenchel-rockafellar duality. *arXiv preprint arXiv:2001.01866*, 2020.

Ofir Nachum, Yinlam Chow, Bo Dai, and Lihong Li. Dualdice: Behavior-agnostic estimation of discounted stationary distribution corrections. *Advances in neural information processing systems*, 32, 2019.

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.

Paria Rashidinejad, Banghua Zhu, Cong Ma, Jiantao Jiao, and Stuart Russell. Bridging offline reinforcement learning and imitation learning: A tale of pessimism. *Advances in Neural Information Processing Systems*, 34:11702–11716, 2021.

Bruno Scherrer, Mohammad Ghavamzadeh, Victor Gabillon, Boris Lesner, and Matthieu Geist. Approximate modified policy iteration and its application to the game of tetris. *J. Mach. Learn. Res.*, 16(49): 1629–1676, 2015.

Shai Shalev-Shwartz, Shaked Shammah, and Amnon Shashua. Safe, multi-agent, reinforcement learning for autonomous driving. *arXiv preprint arXiv:1610.03295*, 2016.

Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2446–2454, 2020.

Yue Sun, Adhyyan Narang, Ibrahim Gulluck, Samet Oymak, and Maryam Fazel. Towards sample-efficient overparameterized meta-learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

Ziyang Tang, Yihao Feng, Lihong Li, Dengyong Zhou, and Qiang Liu. Doubly robust bias reduction in infinite horizon off-policy estimation. *arXiv preprint arXiv:1910.07186*, 2019.

Philip Thomas and Emma Brunskill. Data-efficient off-policy policy evaluation for reinforcement learning. In *International Conference on Machine Learning*, pages 2139–2148. PMLR, 2016.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pages 5026–5033. IEEE, 2012.

Nilesh Tripuraneni, Chi Jin, and Michael Jordan. Provable meta-learning of linear representations. In *International Conference on Machine Learning*, pages 10434–10443. PMLR, 2021.

Masatoshi Uehara, Xuezhou Zhang, and Wen Sun. Representation learning for online and offline rl in low-rank mdps. *arXiv preprint arXiv:2110.04652*, 2021.

Lingxiao Wang, Qi Cai, Zhuoran Yang, and Zhaoran Wang. Neural policy gradient methods: Global optimality and rates of convergence. *arXiv preprint arXiv:1909.01150*, 2019.

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, 2020a.

Ruosong Wang, Dean P Foster, and Sham M Kakade. What are the statistical limits of offline rl with linear function approximation? *arXiv preprint arXiv:2010.11895*, 2020b.

Tengyang Xie and Nan Jiang. Q* approximation schemes for batch reinforcement learning: A theoretical comparison. In *Conference on Uncertainty in Artificial Intelligence*, pages 550–559. PMLR, 2020.

Tengyang Xie, Yifei Ma, and Yu-Xiang Wang. Towards optimal off-policy evaluation for reinforcement

learning with marginalized importance sampling. *Advances in Neural Information Processing Systems*, 32, 2019.

Zhiyuan Xu, Kun Wu, Zhengping Che, Jian Tang, and Jieping Ye. Knowledge transfer in multi-task deep reinforcement learning for continuous control. *Advances in Neural Information Processing Systems*, 33:15146–15155, 2020.

Jiaqi Yang, Wei Hu, Jason D Lee, and Simon S Du. Impact of representation learning in linear bandits. *arXiv preprint arXiv:2010.06531*, 2020a.

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.

Mengjiao Yang, Ofir Nachum, Bo Dai, Lihong Li, and Dale Schuurmans. Off-policy evaluation via the regularized lagrangian. *Advances in Neural Information Processing Systems*, 33:6551–6561, 2020b.

Ming Yin and Yu-Xiang Wang. Asymptotically efficient off-policy evaluation for tabular reinforcement learning. In *International Conference on Artificial Intelligence and Statistics*, pages 3948–3958. PMLR, 2020.

Ming Yin, Yu Bai, and Yu-Xiang Wang. Near-optimal provable uniform convergence in offline policy evaluation for reinforcement learning. In *International Conference on Artificial Intelligence and Statistics*, pages 1567–1575. PMLR, 2021.

Tianhe Yu, Garrett Thomas, Lantao Yu, Stefano Ermon, James Y Zou, Sergey Levine, Chelsea Finn, and Tengyu Ma. Mopo: Model-based offline policy optimization. *Advances in Neural Information Processing Systems*, 33:14129–14142, 2020.

Fuxiang Zhang, Chengxing Jia, Yi-Chen Li, Lei Yuan, Yang Yu, and Zongzhang Zhang. Discovering generalizable multi-agent coordination skills from multi-task offline data. In *The Eleventh International Conference on Learning Representations*, 2022.

Ruiyi Zhang, Bo Dai, Lihong Li, and Dale Schuurmans. Gendice: Generalized offline estimation of stationary values. *arXiv preprint arXiv:2002.09072*, 2020a.

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

Ding-Xuan Zhou. The covering number in learning theory. *Journal of Complexity*, 18(3):739–767, 2002.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes]
   (b) Complete proofs of all theoretical results. [Yes]
   (c) Clear explanations of any assumptions. [Yes]

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Yes]
   (b) The license information of the assets, if applicable. [Not Applicable]
   (c) New assets either in the supplemental material or as a URL, if applicable. [Not Applicable]
   (d) Information about consent from data providers/curators. [Not Applicable]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:

(a) The full text of instructions given to participants and screenshots. [Not Applicable]

(b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]

(c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

A More Discussion on Related Work

A.1 Online Multi-Task Transfer Learning in Low-rank MDPs

Our setup is similar to that studied in [Cheng et al., 2022, Lu et al., 2022, Agarwal et al., 2023] which learns a representation from the source tasks and then uses the learnt representation to learn a good policy in the target task, where all tasks are modeled as low-rank MDPs. However, all consider the online setting where the learner uses reward-free exploration in the source tasks to construct datasets with good coverage. As mentioned earlier, this can be costly or risky in applications such as precision medicine or autonomous driving, which preferably rely on offline data.

A.2 Offline RL

Theoretical study of offline RL typically requires one of these assumptions (i) the ratio between the visitation measure of the optimal policy and that of the data collecting policy to be upper bounded uniformly over the state-action space [Jiang and Li, 2016, Thomas and Brunskill, 2016, Farajtabar et al., 2018, Liu et al., 2018, Xie et al., 2019, Nachum et al., 2019, Nachum and Dai, 2020, Tang et al., 2019, Kallus and Uehara, 2022, Jiang and Huang, 2020, Uehara et al., 2021, Du et al., 2019, Yin and Wang, 2020, Yin et al., 2021, Yang et al., 2020b, Zhang et al., 2020a] or (ii) the concentrability coefficient defined as the supremum of a similarly defined ratio over the state-action space needs to be upper bounded [Antos et al., 2007, Munos and Szepesvári, 2008, Scherrer et al., 2015, Chen and Jiang, 2019, Liu et al., 2019, Wang et al., 2019, Fan et al., 2020, Xie and Jiang, 2020, Liao et al., 2022, Zhang et al., 2020a].

A.3 Comparison to Cheng et al. [2022]

We list the key differences to Cheng et al. [2022]:

1. Algorithm 1 ([Cheng et al., 2022]) needs access to the underlying MDPs, while in our setting we just have access to the trajectories and not to the underlying MDPs. Hence our setting does not have the ability to construct datasets, but rather compute policies on whatever dataset is available.

2. Theorem 4.1 ([Cheng et al., 2022]) is on the offline dataset curated by policies improved gradually over time by Algorithm 1 ([Cheng et al., 2022]), line 15. New trajectories are added to the dataset with the updated policies (lines 6,7). Thus although the representation is learnt via offline MLE (line 9) , the dataset itself is controlled by the online policy in Algorithm 1 ([Cheng et al., 2022])to have desirable properties.

3. Lemma 1 ([Cheng et al., 2022]) is derived under a very restrictive Assumption 2 ([Cheng et al., 2022]) which requires the dataset to be collected via an exploratory policy. This enables them to utilize concentration inequalities to uniformly bound the representation error in Lemma 1 ([Cheng et al., 2022]). In contrast, our work makes no assumptions on the policy collecting the dataset. Our proof direction is via our novel notion of neighborhood density and is thus very different from the usual route of concentration inequalities typically followed in offline RL papers.

4. We do not need Assumption 3 ([Cheng et al., 2022]) for the state space to be compact.

5. Assumption 3 ([Cheng et al., 2022]) is essentially an assumption imposed within the class of Low Rank MDPs. Hence their result doesn't hold for all LowRank MDPs. Our method doesn't need such an assumption and is applicable to any Low Rank MDP.

6. Assumption 5 ([Chen et al., 2022]) states that the transition dynamics of the source tasks can be linearly combined to approximate the target task transition dynamics up to an error of ξ. This ξ error is irreducible as noted in the definition of ξ_{down} before Lemma 1 ([Chen et al., 2022]). This notion of approximate linear can be trivially extended to Assumption 2, and our results will have an additional ξ error term which stems from triangle inequality. Thus we lose nothing in terms of the approximate linear span assumptions and it is only a matter of writing. The more important distinction is Assumption 2.2 in our paper allows for different weights for different states s' unlike fixed weights.

A.4 Comparison to Jin et al. [2021]

We follow the similar set of assumption on the trajectories in the dataset satisfying the Markov property of the underlying MDPs (Assumption 2.3).

We list down the key differences to Jin et al. [2021]:

1. Definition 4.1 and Theorem 4.2 in [Jin et al., 2021] are non-constructive for general MDPs, and they derive an uncertainty quantifier for the special case of linear MDPs. Our Lemma B.1 is the first work on uncertainty quantifier for low-rank MDPs in the offline setting which leads to the main result of our paper in Theorem 4.1.
2. Our work focuses on the Multi-task setting, where first one needs to learn a representation from source datasets. Our contributions over [Jin et al., 2021] are two fold: (i) We construct uncertainty estimators for the representation learning stage, (ii) Combine both the represntation learning errors and penalize the poor exploration in target task to bound suboptimality on the output policy.

A.5 Comparison to Agarwal et al. [2023]

We follow the similar setup with the source task relatedness to the target task as stated in Assumption 2.2. However, Agarwal et al. [2023] focus on the purely online setting, with access to both source and target MDPs. Hence our work diverges from their direction.

A.6 Comparison to concurrent work [Ishfaq et al., 2024]

In a concurrent work [Ishfaq et al., 2024] study the same problem as us. Unlike our Theorem 3.1, which leverages our algorithmic contribution in terms of the neighborhood density function to get a pointwise uncertainty bound, [Ishfaq et al., 2024] rely on prior techniques on condition number to provide an uniform upper bound on the estimated transition kernels. For the downstream target task, while we only consider the offline setting, Ishfaq et al. [2024] primarily focus and on the study of reward-free setting.

B PRT Description

First we present a brief overview of the standard Value Iteration algorithm [Sutton and Barto, 2018], which under the assumption of known transition dynamics $P_h^*(\cdot|s,a)$ returns the optimal policy. Recall the definition of the Q-function : $Q_h(s,a) = r_h(s,a) + \mathbb{E}_{s'\sim P_h^*(\cdot|s,a)} V_{h+1}(s')$. The Value Iteration Algorithm initializes $V_{H+1} = 0$ and goes backwards by setting the policy $\pi_H(s) = \text{argmax}_{a\in\mathcal{A}} Q_H(s,a)$, and the corresponding value for this policy $V_H(s) = \max_{a\in\mathcal{A}} Q_H(s,a)$. Doing this iteratively for all $h = H-1,\dots,1$, the learner is able to obtain the optimal policy $\pi_1^*,\dots,\pi_H^*$.

However, since $P_h^*(\cdot|s,a)$ is unknown in our setting, the learner is unable to accurately compute $\mathbb{E}_{s'\sim P_h^*(\cdot|s,a)} V_{h+1}(s')$ at any arbitrary step h. However based on the available offline data and using the low-rank structure, the learner can form an estimate $\mathbb{E}_{s'\sim \widehat{P}_h(\cdot|s,a)} \widehat{V}_{h+1}(s') = \widehat{\phi}_h(s,a)^\top \widehat{w}_h$, using Least Squares regression (see Lines 4-5 in Algorithm 2). Since this estimate is likely to have uncertainties, before constructing the Q-function it is necessary to penalise every (s,a) based on how uncertain the estimation is. The following lemma introduces such an uncertainty quantifier $\Gamma_h(s,a)$ with high probability.

Lemma B.1. *In Algorithm 2, setting* $\lambda = 1, \beta = c\cdot d\sqrt{\zeta}$ *and* $\epsilon(s,a) = 2\alpha_{\max}\sqrt{K\frac{\log(2|\Phi|/\delta)+K\log|\Upsilon|}{N_S D_h(s,a)}}$ *where* $\zeta = \frac{\log(4dHn/\delta)}{n}$. *Here* $c \geq 1$ *is an absolute constant and* $\delta \in (0,1)$ *is the confidence parameter and* $\epsilon_h = \sqrt{\frac{1}{n}\sum_{\tau=1}^n \epsilon(s_h^\tau, a_h^\tau)^2}$. *Define the event* $\mathcal{E}$:

$$\{\Gamma_h(s,a) = |\mathbb{E}_{s'\sim P_h^*(\cdot|s,a)} \widehat{V}_{h+1}(s') - \mathbb{E}_{s'\sim \widehat{P}_h(\cdot|s,a)} \widehat{V}_{h+1}(s')| \leq H(\beta+\epsilon_h)\cdot\|\widehat{\phi}_h(s,a)\|_{\Lambda_h} + H\epsilon(s,a)\}.$$

Then $\mathcal{E}$ *satisfies* $P_{\mathcal{D}}(\mathcal{E}) \geq 1-\delta$, *where* $P_{\mathcal{D}}$ *is the data generating process for the tasks satisfying Assumption 2.3.*

By penalizing the Q-functions by the uncertainty quantifier (lines 8-9 Algorithm 2), the learner chooses the policy as the action maximizing the $Q-$function for each corresponding state (line 10 Algorithm 2). Doing it for all steps $h \in [H]$ as described in Algorithm 2 gives the target policy.

C Computational Complexity

Remark C.1. ***(Computation of MLE)*** *The MLE estimation is in general a non-convex optimization problem when ϕ and μ are general nonlinear function approximators. However, this is treated as a standard supervised learning ERM oracle in the literature [Uehara et al., 2021, Agarwal et al., 2020, 2023].*

Remark 4.1. *The computational complexity of Algorithm 2 to obtain the guarantee in Theorem 4.1 is $\mathcal{O}(KN_Sn)$. The derivation and wall-clock run times can be found in Appendix C.*

First, we provide the description of an efficient algorithm to compute Eq. (3) and Eq. (4) in Algorithm 1. For any given state action pair (s, a), and any chosen $\phi \in \Phi$, for all N_S points in the source task i, $\mathcal{D}_{i;h}$ pre-compute the distances $|\phi(s, a) - \phi(s', a')| : (s', a') \in \mathcal{D}_{i;h}$. One can do this for all $\phi \in \Phi$ in $\mathcal{O}(N_S|\Phi|)$ time. Notice that the neighborhood density function is a piecewise constant function with a maximum of n jumps, and can be computed in $\mathcal{O}(N_S|\Phi|)$ time. We solve Eq. (4) by first initializing all $\nu_1, \ldots, \nu_K$ to 0 and then incrementing them at the pre-assigned points $KN_S|\Phi|$ points of discontinuity. Thus the overall theoretical computational cost of the algorithm to compute the neighborhood density for a given (s, a) pair is $\mathcal{O}(KN_S|\Phi|)$. For our experiments, where our representation class is a parametrized neural network we use the MLE estimate $\widehat{\phi}$ to compute the distances in Eq (3), thus the practical computational cost for any (s, a) is $\mathcal{O}(KN_S)$. Note that the neighborhood density only needs to be computed for points in the target dataset in Algorithm 2, so that we can penalize the representation error. we state that these need to be done for only the points in the target task since uncertainty needs to be computed only on the seen target samples, thus the overall cost of the uncertainty quantifier along with Algorithm 2 is $\mathcal{O}(KN_Sn)$.

D Missing Algorithms

In this section we present the algorithms we used for our baselines. All of these use the learnt representation $\widehat{\phi}_h(\cdot, \cdot)$ from Equation (1).

Algorithm 3 (RepTransfer Least Squares Value Iteration) RT-L

Input: Target Dataset $\mathcal{D} = \{(s_h^\tau, a_h^\tau)\}_{\tau,h=1}^{n,H}$, Learnt Representation $\widehat{\phi}_h(\cdot, \cdot)$.
Initialization: Set $\widehat{V}_{H+1}(\cdot) \leftarrow 0$.
for $h = H, H-1, \ldots, 1$ **do**
  Set $\Lambda_h \leftarrow \frac{1}{n}\left(\sum_{\tau=1}^{n} \phi_h(s_h^\tau, a_h^\tau)\phi_h(s_h^\tau, a_h^\tau)^\top + \lambda \mathbb{I}\right)$.
  Set $\widehat{w}_h \leftarrow \Lambda_h^{-1}\left(\frac{1}{n}\sum_{\tau=1}^{n} \phi_h(s_h^\tau, a_h^\tau) \cdot \widehat{V}_{h+1}(s_{h+1}^\tau)\right)$.
  Set $\overline{Q}_h(\cdot, \cdot) \leftarrow r_h(\cdot, \cdot) + \phi_h(\cdot, \cdot)^\top \widehat{w}_h$.
  Set $\widehat{Q}_h(\cdot, \cdot) \leftarrow \min\{\overline{Q}_h(\cdot, \cdot), H - h + 1\}^+$.
  Set $\widehat{\pi}_h(\cdot|\cdot) \leftarrow \arg\max_{\pi_h} \widehat{Q}_h(\cdot, \cdot)^\top \pi_h$.
  Set $\widehat{V}_h(\cdot) \leftarrow \mathbb{E}_{a\sim\widehat{\pi}_h(\cdot|\cdot)}\widehat{Q}_h(\cdot, a)$.
end for
Return $\widehat{\pi} = \{\widehat{\pi}_h(\cdot|\cdot)\}_{h=1}^{H}$.

Algorithm 4 RepTransfer Pessimistic Value Iteration (RT-P)

Input: Target Dataset $\mathcal{D} = \{(s_h^\tau, a_h^\tau)\}_{\tau,h=1}^{n,H}$, Learnt Representation $\widehat{\phi}_h(\cdot,\cdot)$.
Initialization: Set $\widehat{V}_{H+1}(\cdot) \leftarrow 0$.
for $h = H, H-1, \ldots, 1$ **do**
  Set $\Lambda_h \leftarrow \frac{1}{n}\left(\sum_{\tau=1}^n \phi_h(s_h^\tau, a_h^\tau)\phi_h(s_h^\tau, a_h^\tau)^\top + \lambda \mathbb{I}\right)$.
  Set $\widehat{w}_h \leftarrow \Lambda_h^{-1}\left(\frac{1}{n}\sum_{\tau=1}^n \phi_h(s_h^\tau, a_h^\tau)\cdot \widehat{V}_{h+1}(s_{h+1}^\tau)\right)$.
  Set $\Gamma_h(\cdot,\cdot) \leftarrow \beta \cdot \|\phi_h(\cdot,\cdot)\|_{\Lambda_h}$.
  Set $\overline{Q}_h(\cdot,\cdot) \leftarrow r_h(\cdot,\cdot) + \phi_h(\cdot,\cdot)^\top \widehat{w}_h - \Gamma_h(\cdot,\cdot)$.
  Set $\widehat{Q}_h(\cdot,\cdot) \leftarrow \min\{\overline{Q}_h(\cdot,\cdot), H-h+1\}^+$.
  Set $\widehat{\pi}_h(\cdot|\cdot) \leftarrow \arg\max_{\pi_h} \widehat{Q}_h(\cdot,\cdot)^\top \pi_h$.
end for
Return $\widehat{\pi} = \{\widehat{\pi}_h(\cdot|\cdot)\}_{h=1}^H$.
Set $\widehat{V}_h(\cdot) \leftarrow \mathbb{E}_{a\sim\widehat{\pi}_h(\cdot|\cdot)}\widehat{Q}_h(\cdot, a)$.

Algorithm 5 RepTransfer Least Squares Value Iteration Upper Confidence Bound (RT-LU)

Input: Learnt Representation $\widehat{\phi}_h(\cdot,\cdot)$, Access to draw samples from Target MDP.
Initialization: Set $\widehat{V}_{H+1}(\cdot) \leftarrow 0$, Randomly initialize $\{\widehat{\pi}_h\}_{h\in[H]}$.
for $l = 1, \ldots, n$ **do**
  Sample $s_1^l \sim d_1$.
  for $h = 1, \ldots, H$ **do**
    Perform $a_h^l \sim \widehat{\pi}_h(\cdot|s_h^l)$.
    Collect $s_{h+1}^l \sim P_h(\cdot|s_h^l, a_h^l)$.
  end for
  for $h = H, H-1, \ldots, 1$ **do**
    Set $\Lambda_h \leftarrow \frac{1}{l}\left(\sum_{\tau=1}^l \phi_h(s_h^\tau, a_h^\tau)\phi_h(s_h^\tau, a_h^\tau)^\top + \lambda \mathbb{I}\right)$.
    Set $\widehat{w}_h \leftarrow \Lambda_h^{-1}\left(\frac{1}{l}\sum_{\tau=1}^l \phi_h(s_h^\tau, a_h^\tau)\cdot \widehat{V}_{h+1}(s_{h+1}^\tau)\right)$.
    Set $\Gamma_h(\cdot,\cdot) \leftarrow \beta \cdot \|\phi_h(\cdot,\cdot)\|_{\Lambda_h}$.
    Set $\overline{Q}_h(\cdot,\cdot) \leftarrow r_h(\cdot,\cdot) + \phi_h(\cdot,\cdot)^\top \widehat{w}_h + \Gamma_h(\cdot,\cdot)$.
    Set $\widehat{Q}_h(\cdot,\cdot) \leftarrow \min\{\overline{Q}_h(\cdot,\cdot), H-h+1\}^+$.
    Set $\widehat{\pi}_h(\cdot|\cdot) \leftarrow \arg\max_{\pi_h} \widehat{Q}_h(\cdot,\cdot)^\top \pi_h$.
    Set $\widehat{V}_h(\cdot) \leftarrow \mathbb{E}_{a\sim\widehat{\pi}_h(\cdot|\cdot)}\widehat{Q}_h(\cdot, a)$.
  end for
end for
Return $\widehat{\pi} = \{\widehat{\pi}_h(\cdot|\cdot)\}_{h=1}^H$.

E Proof of MLE Guarantee

We first state an auxiliary lemma which allows us to work our way to the MLE guarantee for Equation 1.

Lemma E.1. *Consider a class of conditional probability distribution functions $\mathcal{F} : \{f | f(y|x) \to [0,1]\}$. Suppose we have data samples $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n \subset \mathcal{X} \times \mathcal{Y}$ where $y|x \sim f^*(\cdot|x)$ $(f^* \in \mathcal{F})$. We find an MLE estimate:*

$$\widehat{f} = \underset{f \in \mathcal{F}}{\operatorname{argmax}} \sum_{i=1}^{n} \log f(y_i|x_i).$$

Then the following bound holds with probability at least $1 - \delta$:

$$\frac{1}{n} \sum_{i \in [n]} \|\widehat{f}(\cdot|x_i) - f^*(\cdot|x_i)\|_{TV}^2 \leq \frac{2 \log(|\mathcal{F}|/\delta)}{n}.$$

Proof. Given an set of points $\{x_1, \ldots, x_n\}$, we observe samples $\{y_1, \ldots, y_n\}$ from f^*. We wish to understand how well the estimate $\widehat{f}$ captures the randomness in the $\mathcal{Y}$ space on the empirical distribution over $\{x_1, \ldots, x_n\}$. $\frac{1}{n} \sum_{i \in [n]} \|\widehat{f}(\cdot|x_i) - f^*(\cdot|x_i)\|_{TV}^2$ is a measure of the quality of this estimate.

We invoke Theorem 18 from [Agarwal et al., 2020], with a slight variation. Given offline source data $\mathcal{D} = \{(x_i, y_i)\}_{i \in [n]}$, we create a tangent sequence $\mathcal{D}' = \{(x_i', y_i')\}_{i \in [n]}$, where $x_i' = x_i$ and $y_i' \sim f^*(\cdot|x_i')$. Rest of the proof follows after making this choice of $\mathcal{D}'$. In [Agarwal et al., 2020], they consider the randomness in the $\mathcal{X}$ space as well, but since we are working with a offline dataset, we don't need to take that into account. □

Lemma 3.1. *Let $\{\widehat{\mu}_{i;h}\}_{i \in [K]}, \widehat{\phi}_h$ be the learned MLE estimates from Equation (1). Then with probability at least $1 - \delta$ we have the following bound:*

$$\sum_{i=1}^{K} \underbrace{\sum_{\tau=1}^{N_S} \frac{\|\widehat{\mu}_{i;h}(\cdot)^\top \widehat{\phi_h}(s_h^{i;\tau}, a_h^{i;\tau}) - \mu_{i;h}^*(\cdot)^\top \phi_h^*(s_h^{i;\tau}, a_h^{i;\tau})\|_{TV}^2}{N_S}}_{\text{average error on source } i\text{'s dataset}}$$
$$\leq \frac{2(\log(|\Phi|/\delta) + K \log(|\Upsilon|))}{N_S}. \tag{2}$$

Proof. This follows from Lemma E.1, where the function class is expressed as $\mathcal{F} = \Phi \times \Upsilon^K$ and the number of samples is N_S. □

F Proof of Theorem 3.1

F.1 Error Bounds for one Source Task

We first derive an upper bound on the pointwise uncertainty error for any low-rank MDP in the following lemma.

Lemma F.1. *For all $\phi \in \Phi, \mu \in \Upsilon$, the pointwise model misspecification $\Delta_{\phi,\mu}(\cdot,\cdot)$ can be bounded as*

$$\Delta_{\phi,\mu}(s,a) \leq \frac{1}{|\mathcal{D}|} \left(\sum_{(s',a') \in \mathcal{D}} \Delta_{\phi,\mu}(s',a') + 2d \sup_{\phi \in \Phi} \sum_{(s',a') \in \mathcal{D}} \|\phi(s,a) - \phi(s',a')\|_1 \right),$$

for all $(s,a) \in \mathcal{S} \times \mathcal{A}$ and all $\mathcal{D} \subseteq \mathcal{S} \times \mathcal{A}$.

Proof. We use $\|\mu(\cdot)\| = \|\int_{s \in \mathcal{S}} \mathbf{d}\mu(s)\|$. By Definition 2.1, choosing $g(s) = 1 \; \forall s \in \mathcal{S}$, we have $\|\mu(\cdot)\|_2 \leq \sqrt{d}$. By Cauchy Schwartz inequality $\|\mu(\cdot)\|_1 \leq d$. Noting that that the total variation distance between two distributions

is the ℓ_1 norm of their difference, we can write:

$$\begin{aligned}
&\left|\Delta_{\phi,\mu}(s',a') - \Delta_{\phi,\mu}(s,a)\right| \\
&= \left|\frac{1}{4}\|\mu^\top(\cdot)\phi(s',a') - {\mu^*}^\top(\cdot)\phi^*(s',a')\|_1^2 - \frac{1}{4}\|\mu^\top(\cdot)\phi(s,a) - {\mu^*}^\top(\cdot)\phi^*(s,a)\|_1^2\right| \\
&= \frac{1}{4}\left|\|\mu^\top(\cdot)\phi(s',a') - {\mu^*}^\top(\cdot)\phi^*(s',a')\|_1 - \|\mu^\top(\cdot)\phi(s,a) - {\mu^*}^\top(\cdot)\phi^*(s,a)\|_1\right| \\
&\cdot \left(\|\mu^\top(\cdot)\phi(s',a') - {\mu^*}^\top(\cdot)\phi^*(s',a')\|_1 + \|\mu^\top(\cdot)\phi(s,a) - {\mu^*}^\top(\cdot)\phi^*(s,a)\|_1\right) \\
&\leq \left|\|\mu^\top(\cdot)\phi(s',a') - {\mu^*}^\top(\cdot)\phi^*(s',a')\|_1 - \|\mu^\top(\cdot)\phi(s,a) - {\mu^*}^\top(\cdot)\phi^*(s,a)\|_1\right| \\
&\text{(Since } \|\mu^\top(\cdot)\phi(s,a)\|_1 = 1 \ \forall (s,a) \in \mathcal{S}\times\mathcal{A} \text{ and } \forall \phi \in \Phi, \mu \in \Upsilon\text{; it is a probability distribution)} \\
&\leq \|\mu^\top(\cdot)\phi(s',a') - \mu^\top(\cdot)\phi(s,a)\|_1 + \|{\mu^*}^\top(\cdot)\phi^*(s',a') - {\mu^*}^\top(\cdot)\phi^*(s,a)\|_1 \quad \text{(Triangle Inequality)} \\
&\leq d\left(\|\phi(s',a') - \phi(s,a)\|_1 + \|\phi^*(s',a') - \phi^*(s,a)\|_1\right) \quad \text{(Since } \|\mu(\cdot)\|_1 \leq d \quad \forall \mu \in \Upsilon\text{)}.
\end{aligned}$$

Now given a set of state-action pairs $\mathcal{D} \subseteq \mathcal{S}\times\mathcal{A}$, we can write:

$$\begin{aligned}
&\left|\frac{1}{|\mathcal{D}|}\sum_{(s',a')\in\mathcal{D}} \left(\Delta_{\phi,\mu}(s,a) - \Delta_{\phi,\mu}(s',a')\right)\right| \\
&\leq \frac{1}{|\mathcal{D}|}\sum_{(s',a')\in\mathcal{D}} \left|\left(\Delta_{\phi,\mu}(s,a) - \Delta_{\phi,\mu}(s',a')\right)\right| \\
&\leq \frac{1}{|\mathcal{D}|}\sum_{(s',a')\in\mathcal{D}} d\left(\|\phi(s',a') - \phi(s,a)\|_1 + \|\phi^*(s',a') - \phi^*(s,a)\|_1\right) \\
&\leq 2d \sup_{\phi\in\Phi} \sum_{(s',a')\in\mathcal{D}} \|\phi(s,a) - \phi(s',a')\|_1.
\end{aligned}$$

This completes the proof. □

Note that the lemma above allows us to write the uncertainty at some point (s,a) in terms of the distances in the representation space for any arbitrary $\mathcal{D} \subseteq \mathcal{S}\times\mathcal{A}$. In the following lemma we are going to restrict $\mathcal{D}$ to be a subset of the offline dataset and use the MLE guarantee (Lemma 3.1).

Lemma 3.2. *Let* $\Delta^i_{\widehat{\phi}_h,\widehat{\mu}_{i;h}}(s,a) = \|\widehat{\mu}_{i;h}(\cdot)^\top\widehat{\phi_h}(s,a) - \mu^*_{i;h}(\cdot)^\top\phi^*_h(s,a)\|^2_{TV}$ *denote the transition dynamics misspecification for source task* i *at time* h *for any* $(s,a)\in\mathcal{S}\times\mathcal{A}$ *specified by representations* $\widehat{\phi}_h, \widehat{\mu}_{i;h}$ *learnt from Equation (1). For some* $\nu_i \geq 0$, $\Delta^i_{\widehat{\phi}_h,\widehat{\mu}_{i;h}}(s,a)$ *can be upper bounded as follows:*

$$\Delta^i_{\widehat{\phi}_h,\widehat{\mu}_{i;h}}(s,a) \leq \underbrace{2d\cdot\nu_i}_{\text{bias}} + \underbrace{\sum_{(s',a',\cdot)\in\mathcal{D}_{i;h}} \frac{\|\widehat{\mu}_{i;h}(\cdot)^\top\widehat{\phi_h}(s',a') - \mu^*_{i;h}(\cdot)^\top\phi^*_h(s',a')\|^2_{TV}}{N_S D^{\nu_i}_{i;h}(s,a)}}_{\text{variance}}.$$

Proof. For a given (s,a), choose $\mathcal{D} = S_{i;h}(s,a,\nu_i)$ where

$$\begin{aligned}
&S_{i;h}(s,a,\nu_i) = \frac{1}{N_S}\inf_{\phi\in\Phi}\operatorname*{argmax}_{\mathcal{C}\subseteq\mathcal{D}_{i;h}} |\mathcal{C}| \\
&\text{such that } \|\phi(s,a) - \phi(s',a')\|_1 \leq \nu_i, \ \forall (s',a') \in \mathcal{C},
\end{aligned}$$

as the subset of datapoints optimizing Equation 3 in Algorithm 1. Plugging this choice of $\mathcal{D}$ in Lemma F.1, we can write:

$$\Delta_{\phi,\mu}(s,a) \leq \frac{1}{|S_{i;h}(s,a,\nu_i)|}\left(\sum_{(s',a')\in S_{i;h}(s,a,\nu_i)} \Delta_{\phi,\mu}(s',a') + 2d\sup_{\phi\in\Phi}\sum_{(s',a')\in S_{i;h}(s,a,\nu_i)} \|\phi(s,a)-\phi(s',a')\|_1\right).$$

The second term on the right hand side is $\leq \nu_i$ by the condition of the optimization problem. Now we will use importance sampling (IS) to bound the first term on the right hand side by the average error on dataset (Lemma 3.1). Consider a support as the collection of state action pairs in $\mathcal{D}_{i;h}$. For the expression above, the probability density is:

$$q_i(s',a') = \begin{cases} \frac{1}{|S_{i;h}(s,a,\nu_i)|} & \text{if}(s',a') \in S_{i;h}(s,a,\nu_i) \\ 0 & \text{otherwise} \end{cases}$$

The probability distribution for the average error on dataset is uniform $\frac{1}{N_S}$ on the support. Therefore, the IS ratio $\max_{(s',a',\cdot)\in\mathcal{D}_{i;h}} \frac{q_i(s',a')}{p_i(s',a')} = \frac{N_S}{|S_{i;h}(s,a,\nu_i)|} = \frac{1}{D^{\nu_i}_{i;h}(s,a)}$. Hence, we can write:

$$\Delta^i_{\widehat{\phi},\widehat{\mu}_{i;h}}(s,a) \leq \frac{1}{D^{\nu_i}_{i;h}(s,a)}\sum_{i\in[K]}\frac{1}{N_S}\sum_{(s',a')\in\mathcal{D}_{i;h}} \|\widehat{\mu}_{i;h}(\cdot)^\top\widehat{\phi_h}(s',a') - \mu^*_{i;h}(\cdot)^\top\phi^*_h(s',a')\|^2_{TV} + 2d\cdot\nu_i.$$

This completes the proof. □

F.2 Error Bounds for Target Task

First we show the existence of a transition function linear in $\widehat{\phi}_h$, such that the pointwise error of this transition function with respect to the true transition dynamics of the target task can be decomposed into the sum of pointwise errors of the individual source tasks.

Lemma F.2. *Let $P^*_h(\cdot|s_h,a_h)$ denote the true transition dynamics of the target task, and $\widehat{\phi_h}(s,a)$ be the learnt representation from Equation (1). For all $h\in[H], (s,a)\in\mathcal{S}\times\mathcal{A}$, there exists $\mu'_h:\mathcal{S}\to\mathbb{R}^d$ such that:*

$$\|\mu'_h(\cdot)^\top\widehat{\phi_h}(s,a) - P^*_h(\cdot|s,a)\|^2_{TV} \leq \alpha^2_{\max}K\sum_{i\in[K]}\Delta^i_{\widehat{\phi}_h,\widehat{\mu}_{i;h}}(s,a).$$

Proof. Denote $\mu'_h(s') = \sum_{i\in[K]}\alpha_{i;h}(s')\widehat{\mu}_{i:h}(s')$ where $\alpha_{i;h}(s')$ is as defined in Assumption 2.2.

$$\begin{aligned}
\Delta_{\widehat{\phi}_h,\mu'_h}(s,a) &= \|\mu'_h(\cdot)^\top\widehat{\phi_h}(s,a) - P^*_h(\cdot|s,a)\|^2_{TV} \\
&= \|\mu'_h(\cdot)^\top\widehat{\phi_h}(s,a) - \mu^*_h(\cdot)^\top\phi^*_h(s,a)\|^2_{TV} \\
&= \|\sum_{i\in[K]}\alpha_{i;h}(\cdot)\left(\widehat{\mu}_{i;h}(\cdot)^\top\widehat{\phi_h}(s,a) - \mu^*_{i;h}(\cdot)^\top\phi^*_h(s,a)\right)\|^2_{TV} \quad \text{(By Assumption 2.2)} \\
&\leq \alpha^2_{\max}\|\sum_{i\in[K]}\widehat{\mu}_{i;h}(\cdot)^\top\widehat{\phi_h}(s,a) - \mu^*_{i;k}(\cdot)^\top\phi^*_h(s,a)\|^2_{TV} \\
&\leq \alpha^2_{\max}K\sum_{i\in[K]}\|\widehat{\mu}_{i;h}(\cdot)^\top\widehat{\phi_h}(s,a) - \mu^*_{i;h}(\cdot)^\top\phi^*_h(s,a)\|^2_{TV} \quad \text{(By Cauchy Schwartz)} \\
&= \alpha^2_{\max}K\sum_{i\in[K]}\Delta^i_{\widehat{\phi}_h,\widehat{\mu}_{i;h}}(s,a).
\end{aligned}$$

□

Now we show that the solution of Algorithm 1, allows to use the MLE guarantee (Lemma 3.1) to get a high probability pointwise uncertainty error bound for the target task.

Theorem 3.1. *(Representation Transfer Error): Let $P_h^*(\cdot|s_h, a_h)$ denote the true transition dynamics of the target task, and $\widehat{\phi_h}(s,a)$ be the learnt representation from Equation (1). For all $h \in [H], (s,a) \in \mathcal{S} \times \mathcal{A}$, there exists $\mu_h' : \mathcal{S} \to \mathbb{R}^d$ such that the following bound holds with probability at least $1-\delta$*

$$\|\mu_h'(\cdot)^\top \widehat{\phi_h}(s,a) - P_h^*(\cdot|s,a)\|_{TV} \leq 2\alpha_{\max}\sqrt{\frac{K}{N_S}\frac{\log(|\Phi|/\delta) + K\log|\Upsilon|}{D_h(s,a)}},$$

where $D_h(s,a)$ is the effective occupancy density as computed in Algorithm 1.

Proof. Let's use $\Delta_{\widehat{\phi}_h,\mu'}(s,a)$ to denote $\|\mu_h'(\cdot)^\top \widehat{\phi_h}(s,a) - P_h^*(\cdot|s,a)\|_{TV}^2$. By Lemma F.2, we can write:

$$\Delta_{\widehat{\phi}_h,\mu'}(s,a) \leq \alpha_{\max}^2 K \sum_{i\in[K]} \Delta^i_{\widehat{\phi}_h,\widehat{\mu}_{i;h}}(s,a).$$

For some choice of $\{\nu_1, \ldots, \nu_K\}$, using Lemma 3.2 for the right hand side, we can write:

$$\Delta_{\widehat{\phi},\mu'}(s,a) \leq \alpha_{\max}^2 K \left(\sum_{i\in[K]} \frac{1}{D^{\nu_i}_{i;h}(s,a)} \frac{1}{N_S} \sum_{(s',a')\in\mathcal{D}_{i;h}} \|\widehat{\mu}_{i;h}(\cdot)^\top \widehat{\phi_h}(s',a') - \mu^*_{i;h}(\cdot)^\top \phi^*_h(s',a')\|_{TV}^2 + 2dL\sum_{i\in K}\nu_i \right).$$

Invoking Lemma 3.1, with probability at least $1-\delta$, we have:

$$\Delta_{\widehat{\phi},\mu'}(s,a) \leq \alpha_{\max}^2 K \left(\max_{i\in[K]} \frac{2}{D^{\nu_i}_{i;h}(s,a)} \frac{\log(|\Phi|/\delta) + K\log|\Upsilon|}{N_S} + 2dL\sum_{i\in K}\nu_i \right).$$

Choosing $\{\nu_1, \ldots, \nu_K\}$ by Algorithm 1 we can write:

$$\Delta_{\widehat{\phi},\mu'}(s,a) \leq 4\alpha_{\max}^2 K \frac{\log(|\Phi|/\delta) + K\log|\Upsilon|}{N_S D_h(s,a)}.$$

This completes the proof. □

G Proof of Corollary 3.1

Corollary 3.1. *Let $P_h^*(\cdot|s_h, a_h)$ denote the true transition dynamics of the target task, and $\widehat{\phi_h}(s,a)$ be the learnt representation from Equation (1). For all $h \in [H], (s,a) \in \mathcal{S} \times \mathcal{A}$, there exists $\mu_h' : \mathcal{S} \to \mathbb{R}^d$ such that the following bound holds with probability at least $1-\delta$*

$$\|\mu_h'(\cdot)^\top \widehat{\phi_h}(s,a) - P_h^*(\cdot|s,a)\|_{TV} \leq 2\alpha_{\max} K\sqrt{d\cdot\nu'},$$

under the condition in Equation (6).

Proof. For a given $(s,a) \in \mathcal{S}\times\mathcal{A}$, let $\nu_1, \ldots, \nu_K$ be such that the following are satisfied:

$$\left(\frac{1}{K}\sum_{i\in[K]} \frac{1}{D^{\nu_i}_{i;h}(s,a)}\right)^{-1} \geq \frac{\log(|\Phi|/\delta) + K\log|\Upsilon|}{N_S\cdot d\cdot\nu'} \quad \text{and} \quad \frac{1}{K}\sum_{i\in[K]}\nu_i \leq \nu'.$$

Given any arbitrary set of positive numbers $\{a_1, \ldots, a_K\}$, using the properties that $\mathbf{HM}(a_1, \ldots, a_K) \leq K\min\{a_1, \ldots, a_K\}$, we get: $\min_{i\in[K]} D^{\nu_i}_{i;h}(s,a) \geq \frac{\log(|\Phi|/\delta)+K\log|\Upsilon|}{K\cdot N_S\cdot d\cdot\nu'}$.

Since $\min_{i\in[K]} D^{\nu_i}_{i;h}(s,a)\cdot\sum_{i\in[K]}\nu_i$ is an increasing function in $\{\nu_1, \ldots, \nu_K\}$, thus there exists $\{\nu_1', \ldots, \nu_K'\}$ such that $\nu_i' \geq \nu_i\ \forall i \in [K]$ and $\sum_{i\in[K]}\nu_i' = K\nu'$ satisfies : $\min_{i\in[K]} D^{\nu_i'}_{i;h}(s,a)\cdot\sum_{i\in[K]}\nu_i' \geq \frac{\log(|\Phi|/\delta)+K\log|\Upsilon|}{N_S\cdot d}$.

Therefore there exists $\nu_1^*, \ldots, \nu_K^*$ such that $\nu_i^* \leq \nu_i'\ \forall i \in [K]$ and $\sum_{i\in[K]}\nu_i^* \leq K\nu'$ which is the solution of Equation (4). Therefore by Equation (5): $D_h(s,a) = \frac{\log(|\Phi|/\delta)+K\log|\Upsilon|}{N_S\cdot d\cdot\sum_{i\in[K]}\nu_i^*} \geq \frac{\log(|\Phi|/\delta)+K\log|\Upsilon|}{K\cdot N_S\cdot d\cdot\nu'}$. Plugging this back in Theorem 3.1 gives us the desired result. □

H Proof of Theorem 4.1

We introduce the following standard definition to ease the presentation of the results in this section.

Definition H.1. *(Transition Operator): Given any function $f : \mathcal{S} \to \mathbb{R}$, the Transition operator P_h at step $h \in [H]$ is defined as:*

$$(P_h f)(s,a) = \mathbb{E}_{s' \sim P_h(\cdot|s,a)} f(s').$$

The following lemma states that for a low-rank MDP the Transition Operator P_h can be written as a linear function of the representation $\phi_h(\cdot,\cdot)$.

Lemma H.1. *For a low rank MDP, given any function $f : \mathcal{S} \to \mathbb{R}$ there exists an unknown $w_h \in \mathbb{R}^d$ such that*

$$(P_h f)(s,a) = \phi_h(s,a)^\top w_h.$$

Proof. By Definition 2.1 and H.1 we have:

$$\begin{aligned}(P_h f)(s,a) &= \int_{\mathcal{S}} \phi_h(s,a)^\top \mu_h(s') f(s') \mathbf{d}s' \\ &= \phi_h(s,a)^\top \int_{\mathcal{S}} \mu_h(s') f(s') \mathbf{d}s'.\end{aligned}$$

Thus $w_h = \int_{\mathcal{S}} \mu_h(s') f(s') \mathbf{d}s'$. □

Since the true representation $\phi^*_h(\cdot,\cdot)$ is not known, a key step is proving under Theorem 3.1, the existence of a transition operator P'_h with high probability which is linear in the learnt representation $\widehat{\phi}_h(\cdot,\cdot)$ that is close to the true transition operator P^*_h.

Lemma H.2. *Let $\widehat{\phi}_h(\cdot,\cdot)$ be a representation from Equation (1). Given any function $f : \mathcal{S} \to \mathbb{R}$, there exists an unknown $w'_h \in \mathbb{R}^d$, such that $(P'_h f)(s,a) = \widehat{\phi}_h(s,a)^\top w'_h$ satisfies the following bound with probability at least $1 - \delta/2$:*

$$|(P^*_h f)(s,a) - (P'_h f)(s,a)| \le \max_{\mathcal{S}} |f| \cdot \epsilon(s,a),$$

where $\epsilon(s,a) = 2\alpha_{\max}\sqrt{K \frac{\log(2|\Phi|/\delta) + K \log|\Upsilon|}{n_{\nu;h}(s,a)}}$.

Proof. Let $w'_h = \int_{\mathcal{S}} f(s') \mu'_h(s') \mathbf{d}s'$, where $\mu'_h(\cdot)$ is as defined in Theorem 3.1. Then we have:

$$\begin{aligned}&|(P^*_h f)(s,a) - (P'_h f)(s,a)| \\ =&|\int_{\mathcal{S}} P^*_h(s'|s,a) f(s') \mathbf{d}s' - \int_{\mathcal{S}} \widehat{\phi}_h(s,a)^\top \mu'_h(s') f(s') \mathbf{d}s'| \\ \le& \max_{\mathcal{S}} |f| \cdot |\int_{\mathcal{S}} P^*_h(s'|s,a) \mathbf{d}s' - \int_{\mathcal{S}} \widehat{\phi}_h(s,a)^\top \mu'_h(s') \mathbf{d}s'|.\end{aligned}$$

Now use Theorem 3.1 with tolerance $\delta/2$ to bound the second term with probability at least $1 - \delta/2$ completes the proof. □

Lemma B.1. *In Algorithm 2, setting $\lambda = 1, \beta = c \cdot d\sqrt{\zeta}$ and $\epsilon(s,a) = 2\alpha_{\max}\sqrt{K \frac{\log(2|\Phi|/\delta) + K \log|\Upsilon|}{N_S D_h(s,a)}}$ where $\zeta = \frac{\log(4dHn/\delta)}{n}$. Here $c \ge 1$ is an absolute constant and $\delta \in (0,1)$ is the confidence parameter and $\epsilon_h = \sqrt{\frac{1}{n}\sum_{\tau=1}^n \epsilon(s^\tau_h, a^\tau_h)^2}$. Define the event $\mathcal{E}$:*

$$\{\Gamma_h(s,a) = |\mathbb{E}_{s' \sim P^*_h(\cdot|s,a)} \widehat{V}_{h+1}(s') - \mathbb{E}_{s' \sim \widehat{P}_h(\cdot|s,a)} \widehat{V}_{h+1}(s')| \le H(\beta + \epsilon_h) \cdot \|\widehat{\phi}_h(s,a)\|_{\Lambda_h} + H\epsilon(s,a)\}.$$

Then $\mathcal{E}$ satisfies $P_{\mathcal{D}}(\mathcal{E}) \ge 1 - \delta$, where $P_{\mathcal{D}}$ is the data generating process for the tasks satisfying Assumption 2.3.

Proof. Note that $\Gamma_h(s,a) = |(P_h^* \widehat{V}_{h+1})(s,a) - (\widehat{P}_h \widehat{V}_{h+1})(s,a)|$. We now use triangle inequality to upper bound $\Gamma_h(s,a)$.

$$\begin{aligned}
&|(P_h^* \widehat{V}_{h+1})(s,a) - (\widehat{P}_h \widehat{V}_{h+1})(s,a)| \\
\leq &\underbrace{|(P_h^* \widehat{V}_{h+1})(s,a) - (P'_h \widehat{V}_{h+1})(s,a)|}_{(i)} + \underbrace{|(P'_h \widehat{V}_{h+1})(s,a) - (\widehat{P}_h \widehat{V}_{h+1})(s,a)|}_{(ii)}
\end{aligned}$$

(i) can be bounded by Lemma H.2, with $f = \widehat{V}_{h+1}$ and $\max_{\mathcal{S}} |\widehat{V}_{h+1}| \leq H$. Thus $(i) \leq H\epsilon(s,a)$ with probability at least $1 - \delta/2$.

Let us define

$$w'_h = \int_{\mathcal{S}} \widehat{V}_{h+1}(s') \mu'_h(s') \mathbf{d}s'.$$

Thus we can write

$$(P'_h \widehat{V}_{h+1})(s,a) = \widehat{\phi}_h(s,a)^\top w'_h.$$

Now we analyse (ii).

$$\begin{aligned}
(ii) &= \widehat{\phi}_h(s,a)^\top (w'_h - \widehat{w}_h) \\
&= \widehat{\phi}_h(s,a)^\top \left(w'_h - \Lambda_h^{-1} (\frac{1}{n} \sum_{\tau=1}^n \widehat{\phi}_h(s_h^\tau, a_h^\tau) \cdot \widehat{V}_{h+1}(s_{h+1}^\tau)) \right) \\
&= \underbrace{\widehat{\phi}_h(s,a)^\top \left(w'_h - \Lambda_h^{-1} \left(\frac{1}{n} \sum_{\tau=1}^n \widehat{\phi}(s_h^\tau, a_h^\tau) \cdot (P'_h \widehat{V}_{h+1})(s_h^\tau, a_h^\tau) \right) \right)}_{(iii)} \\
&\quad \underbrace{- \widehat{\phi}_h(s,a)^\top \Lambda_h^{-1} \left(\frac{1}{n} \sum_{\tau=1}^n \widehat{\phi}(s_h^\tau, a_h^\tau) \cdot (\widehat{V}_{h+1}(s_{h+1}^\tau) - (P_h \widehat{V}_{h+1})(s_h^\tau, a_h^\tau) \right)}_{(iv)} \\
&\quad \underbrace{+ \widehat{\phi}_h(s,a)^\top \Lambda_h^{-1} \left(\frac{1}{n} \sum_{\tau=1}^n \widehat{\phi}(s_h^\tau, a_h^\tau) \cdot (P'_h \widehat{V}_{h+1})(s_h^\tau, a_h^\tau) - (P_h \widehat{V}_{h+1})(s_h^\tau, a_h^\tau)) \right)}_{(v)}.
\end{aligned}$$

Using triangle inequality we get $|(ii)| \leq |(iii)| + |(iv)| + |(v)|$. The analysis for $|(iii)|, |(iv)|$ follows similarly to the proof of Lemma 5.2 in Jin et al. [2021]. We state the bounds:

$$|(iii)| \leq H \sqrt{\frac{d\lambda}{n}} \sqrt{\left(\widehat{\phi}_h(s,a)^\top \Lambda_h^{-1} \widehat{\phi}_h(s,a) \right)} \leq H \frac{\beta}{2} \sqrt{\left(\widehat{\phi}_h(s,a)^\top \Lambda_h^{-1} \widehat{\phi}_h(s,a) \right)}.$$

$$P_{\mathcal{D}} \left(|(iv)| \leq H \frac{\beta}{2} \sqrt{\left(\widehat{\phi}_h(s,a)^\top \Lambda_h^{-1} \widehat{\phi}_h(s,a) \right)} \right) \geq 1 - \delta/2,$$

where $P_{\mathcal{D}}$ is the data generating distribution.

Let us now bound $|(v)|$.

$$\begin{aligned}
|(v)| &= |\widehat{\phi}_h(s,a)^\top \Lambda_h^{-1} \left(\frac{1}{n} \sum_{\tau=1}^n \widehat{\phi}(s_h^\tau, a_h^\tau) \cdot (P'_h \widehat{V}_{h+1})(s_h^\tau, a_h^\tau) - (P_h \widehat{V}_{h+1})(s_h^\tau, a_h^\tau)) \right)| \\
&\leq \sqrt{\frac{1}{n} \left(\sum_{\tau=1}^n \left(\widehat{\phi}_h(s,a)^\top \Lambda_h^{-1} \widehat{\phi}(s_h^\tau, a_h^\tau) \right)^2 \right) \cdot \left(\frac{1}{n} \sum_{\tau=1}^n \left((P'_h \widehat{V}_{h+1})(s_h^\tau, a_h^\tau) - (P_h \widehat{V}_{h+1})(s_h^\tau, a_h^\tau) \right)^2 \right)}
\end{aligned}$$

This follows from Cauchy Schwartz inequality. From Theorem 3.1 we can bound the second term with probability at least $1-\delta/2$, where $\epsilon(s,a) = 2\alpha_{\max}\sqrt{K\frac{\log(2|\Phi|/\delta)+K\log|\Upsilon|}{n_{\nu;h}(s,a)}}$ and we get:

$$
\begin{aligned}
|(v)| &\leq H \cdot \sqrt{\left(\widehat{\phi}_h(s,a)^\top \Lambda_h^{-1}\left(\frac{1}{n}\sum_{\tau=1}^{n}\widehat{\phi}(s_h^\tau,a_h^\tau)\widehat{\phi}(s_h^\tau,a_h^\tau)^\top\right)\Lambda_h^{-1}\widehat{\phi}_h(s,a)\right)}\sqrt{\frac{1}{n}\sum_{\tau=1}^{n}\epsilon(s_h^\tau,a_h^\tau)^2} \\
&= H \cdot \sqrt{\left(\widehat{\phi}_h(s,a)^\top \Lambda_h^{-1}\left(\Lambda_h - \lambda\mathbb{I}\right)\Lambda_h^{-1}\widehat{\phi}_h(s,a)\right)}\sqrt{\frac{1}{n}\sum_{\tau=1}^{n}\epsilon(s_h^\tau,a_h^\tau)^2} \\
&\leq H \cdot \sqrt{\left(\widehat{\phi}_h(s,a)^\top \Lambda_h^{-1}\widehat{\phi}_h(s,a)\right)}\sqrt{\frac{1}{n}\sum_{\tau=1}^{n}\epsilon(s_h^\tau,a_h^\tau)^2}
\end{aligned}
$$

The first inequality follows from Lemma H.2 by noting $\max_{\mathcal{S}}\widehat{V}_{h+1} \leq H$. The second equation follows from definition of Λ_h.

Combining the bounds by taking union bound concludes the proof of the Lemma. □

Theorem 4.1. *Let $\widehat{\pi}$ be the output of Algorithm 2. Then with probability at least $1-\delta$*

$$
\begin{aligned}
\mathit{SubOpt}(\widehat{\pi}, s) \leq 2H\sum_{h=1}^{H}\mathbb{E}_{(s_h,a_h)\sim\pi^*,P_h^*}\Bigg[&\underbrace{\epsilon(s_h,a_h)}_{\text{source coverage on } \pi^*} + \\
(\beta + &\underbrace{\epsilon_h}_{\text{source coverage on target}}) \cdot \underbrace{\|\widehat{\phi}_h(s_h,a_h)\|_{\Lambda_h}}_{\text{target coverage on } \pi^*} \; | s_1 = s\Bigg].
\end{aligned}
$$

Here the expectation is taken with respect to the optimal policy $\pi^\star$ of the true underlying MDP of the target task. $\epsilon(s,a) = 2\alpha_{\max}\sqrt{K\frac{\log(2|\Phi|/\delta)+K\log|\Upsilon|}{N_S D_h(s,a)}}$, $\beta = c\cdot d\sqrt{\zeta}$, where $\zeta = \frac{\log(4dHn/\delta)}{n}$ and $c \geq 1$ is an absolute constant and $\epsilon_h = \sqrt{\frac{1}{n}\sum_{\tau=1}^{n}\epsilon(s_h^\tau,a_h^\tau)^2}$.

Proof. Follows from Theorem 4.2 in Jin et al. [2021] by plugging in uncertainty quantifiers $\Gamma_h(\cdot,\cdot)$ satisfying guarantees in Lemma B.1. □

I Proof for Uniform Cover in Source Tasks (Corolloary 4.1)

I.1 Recap on Covering and Packing Numbers

Definition I.1. *(ν-Covering) Let $(\mathcal{V}, \|\cdot\|)$ be a normed space and $\mathcal{X} \subseteq \mathcal{V}$. A set $\mathcal{A}$ is called a ν-covering of $\mathcal{X}$, if for all $x \in \mathcal{X}$ $\exists x' \in \mathcal{A}$ such that $\|x - x'\| \leq \nu$. The collection of such sets is denoted by denoted by $\mathcal{N}(\mathcal{X}, \|\cdot\|, \nu)$.*

Definition I.2. *(ν-Covering Number) The size of the minimal set which is ν-covering is defined as the ν-covering number, that is*

$$N(\mathcal{X}, \|\cdot\|, \nu) = \min_{\mathcal{A}\in\mathcal{N}(\mathcal{X},\|\cdot\|,\nu)} |A|.$$

Definition I.3. *(ν-Packing) Let $(\mathcal{V}, \|\cdot\|)$ be a normed space and $\mathcal{X} \subseteq \mathcal{V}$. A set $\mathcal{A}$ is called a ν-packing of $\mathcal{X}$, if for all $x, x' \in \mathcal{A}$ $\|x - x'\| \geq \nu$. The collection of such sets is denoted by denoted by $\mathcal{M}(\mathcal{X}, \|\cdot\|, \nu)$.*

Definition I.4. *(ν-Packing Number) The size of the maximal set which is ν-packing is defined as the ν-packing number, that is*

$$M(\mathcal{X}, \|\cdot\|, \nu) = \max_{\mathcal{A}\in\mathcal{M}(\mathcal{X},\|\cdot\|,\nu)} |A|.$$

We introduce some notation and state some bounds on covering and packing numbers. The unit ℓ_p norm ball in $\mathbb{R}^d$ is defined as :

$$\mathcal{B}_p^d = \{x | x \in \mathbb{R}^d \, ; \|x\|_p \leq 1\}.$$

The following lemmas are borrowed from [Zhou, 2002].

Lemma I.1. *The ν-covering number of $\mathcal{B}_2^d$ satisfies:*

$$N(\mathcal{B}_2^d, \|\cdot\|_1, \nu) \leq \left(\frac{3\sqrt{d}}{\nu}\right)^d.$$

Lemma I.2. *The ν-packing number of $\mathcal{B}_1^d$ satisfies:*

$$M(\mathcal{B}_1^d, \|\cdot\|_1, \nu) \geq \left(\frac{1}{\nu}\right)^d$$

I.2 Some Results using Covering and Packing Numbers

Lemma I.3. *A set $\mathcal{D} \subseteq \mathcal{X}$ is a ν-covering of $\mathcal{X}$, i.e. $\mathcal{D} \in \mathcal{N}(\mathcal{X}, \|\cdot\|, \nu)$ if it is a $\nu/2$-covering for some $\mathcal{A} \in \mathcal{N}(\mathcal{X}, \|\cdot\|, \nu/2)$.*

Proof. Let us consider a set $\mathcal{D} \subseteq \mathcal{X}$ that is a $\nu/2$-covering for some $\mathcal{A} \in \mathcal{N}(\mathcal{X}, \|\cdot\|, \nu/2)$. Therefore for all $x' \in \mathcal{A}$ there exists $x'' \in \mathcal{D}$ such that $\|x'' - x'\| \leq \nu/2$.

Now by definition of $\mathcal{A}$, for every $x \in \mathcal{X}$ there exists $x' \in \mathcal{A}$ such that $\|x' - x\| \leq \nu/2$.

For any $x \in \mathcal{X}$, by the existence results above there exists $x' \in \mathcal{A}, x'' \in \mathcal{D}$ such that $\|x'' - x'\| \leq \nu/2$ and $\|x' - x\| \leq \nu/2$.

Using triangle inequality:

$$\begin{aligned}\|x - x''\| &= \|(x - x') + (x' - x'')\| \\ &\leq \|(x - x')\| + \|(x' - x'')\| \\ &\leq \nu.\end{aligned}$$

This proves that $\mathcal{D}$ is a ν-covering of $\mathcal{X}$. □

Lemma I.4. *If $\mathcal{D} \subseteq \mathcal{X}$ is a ν/l-covering of $\mathcal{X}$ then for every $x \in \mathcal{X}$ there exists at least $M(\mathcal{B}_1, \|\cdot\|_1, \frac{1}{l})$ number of points $x' \in \mathcal{D}$ such that $\|x' - x\|_1 \leq \nu$.*

Proof. Pick any $x \in \mathcal{X}$ and construct an ℓ_1 norm ball of radius ν centered at x, $x + \nu\mathcal{B}_1$. The maximum number $\frac{\nu}{l}\mathcal{B}_1$ balls we can pack in $\nu\mathcal{B}_1$ is given by $M(\mathcal{B}_1, \|\cdot\|_1, \frac{1}{l})$. Since, $\mathcal{D}$ covers $\mathcal{X}$ and consequently $x + \nu\mathcal{B}_1$, there are at least $M(\mathcal{B}_1, \|\cdot\|_1, \frac{1}{l})$ points in $\mathcal{D}$ that are contained in $x + \nu\mathcal{B}_1$. □

J Proof of Corollary 4.1

Lemma J.1. *Let π_i be a policy satisfying Assumption 4.1 used to collect n i.i.d. trajectories. Let $\mathcal{D}_n^h = \{(s_1, a_1), \ldots, (s_n, a_n)\}$ denote the n state action pairs in the offline dataset at time step h. Then with probability at least $1 - \delta$, for every $(s, a) \in \mathcal{S} \times \mathcal{A}$ and for all $h \in [H]$ there exists $(s', a') \in \mathcal{D}_n^h$ such that $\sup_{\phi \in \Phi} \|\phi(s, a) - \phi(s', a')\|_1 \leq \nu$ if*

$$n \geq C\nu^{-d},$$

where $C = c^{-d}\psi^{-d}(6\sqrt{d})^d \cdot \Big(d\log(6\sqrt{d}/\delta)\Big)$.

Proof. The condition for every $(s, a) \in \mathcal{S} \times \mathcal{A}$ and for all $h \in [H]$ there exists $(s', a') \in \mathcal{D}_n^h$ such that $\sup_{\phi \in \Phi} \|\phi(s, a) - \phi(s', a')\|_1 \leq \nu$ implies that we need the offline dataset to be ν-covering in the representation space. For a particular $\phi \in \Phi$, we use $\mathcal{D}_n^\phi = \{\phi(s, a) | (s, a) \in \mathcal{D}_n^h\}$ to denote the mappings of the state action pairs in $\mathcal{D}_n^h$ in the representation space.

By Lemma I.3 it is sufficient for $\mathcal{D}_n^\phi$ to be an $\nu/2$-covering of some $\mathcal{A} \in \mathcal{N}(\mathcal{B}_2^d, \|\cdot\|_1, \nu/2)$ to be an ν-covering of $\mathcal{B}_2^d$ (since the representation space is $\mathcal{B}_2^d$). We choose the minimal set $\mathcal{A}$ such that $|\mathcal{A}| = N(\mathcal{B}_2^d, \|\cdot\|_1, \nu/2)$. We need to show that the worst case (over $\phi \in \Phi$) $\mathcal{D}_n^\phi$ is ν-covering with high probability.

Lets construct bins $\mathcal{A}_y = \{y' | y' \in \mathcal{B}_2^d, \|y' - y\|_1 \leq \nu/2\}$ for all $y \in \mathcal{A}$. Note that $\cup_{y\in\mathcal{A}} \mathcal{A}_y = \mathcal{B}_2^d$. These sets are ℓ_1 norm balls of radius $\nu/2$, i.e. $\nu/2 \cdot \mathcal{B}_1^d$ if they lie in the complete interior of $\mathcal{B}_2^d$. For those sets on the boundary, their volume is at least some fraction c times the volume of $\nu/2 \cdot \mathcal{B}_1^d$ since their center is within $\mathcal{B}_2^d$, for some finite $c < 1$. Thus we can argue

$$N(\mathcal{B}_2^d, \|\cdot\|_1, \nu/2)\text{Vol}(\mathcal{A}_y) \geq c\text{Vol}(\mathcal{B}_2^d) \quad \forall y \in \mathcal{A}$$
$$\implies \frac{\psi \cdot \text{Vol}(\mathcal{A}_y)}{1 \cdot \text{Vol}(\mathcal{B}_2^d)} \geq \frac{c\psi}{N(\mathcal{B}_2^d, \|\cdot\|_1, \nu/2)}.$$

By Assumption 4.1, we have $\inf_{\phi\in\Phi} \inf_{x\in\mathbb{R}^d} d_{i,h}^{\pi_i,\phi}(x) \geq \psi$ and $\sup_{\phi\in\Phi} \sup_{x\in\mathbb{R}^d} d_{i,h}^{\pi_i,\phi}(x) \geq 1$. Thus we can write

$$\inf_{\phi\in\Phi} P(x \in \mathcal{A}_y | x \sim d_{i,h}^{\pi_i,\phi}) \geq \frac{\psi \cdot \text{Vol}(\mathcal{A}_y)}{1 \cdot \text{Vol}(\mathcal{B}_2^d)} \geq \frac{c\psi}{N(\mathcal{B}_2^d, \|\cdot\|_1, \nu/2)} = p.$$

This statement implies that the map of a randomly sampled state action pair via π_i in the representation space lies in the bin $\mathcal{A}_y$ with probability at least p.

We upper bound the probability that none of the n i.i.d. draws lies in $\mathcal{A}_y$ as follows:

$$\sup_{\phi\in\Phi} P(\nexists x \in \mathcal{D}_n^\phi \, ; x \in \mathcal{A}_y) \leq (1-p)^n \leq \exp(-np).$$

Now we wish to lower bound the probability that given n i.i.d. draws we sample at least 1 point from each of these bins. This is achieved as follows:

$$\begin{aligned}\inf_{\phi\in\Phi} P(\cap_{y\in\mathcal{A}} \exists x \in \mathcal{D}_n \, ; x \in \mathcal{A}_y) &= 1 - \sup_{\phi\in\Phi} P(\cup_{y\in\mathcal{A}} \nexists x \in \mathcal{D}_n \, ; x \in \mathcal{A}_y) \\ &\geq 1 - \sum_{y\in\mathcal{A}} \sup_{\phi\in\Phi} P(\nexists x \in \mathcal{D}_n \, ; x \in \mathcal{A}_y) \\ &\geq 1 - N(\mathcal{B}_2^d, \|\cdot\|_1, \nu/2) \exp(-np).\end{aligned}$$

For this probability to be greater than $1-\delta$, we need

$$n = \frac{N(\mathcal{B}_2^d, \|\cdot\|_1, \nu/2)}{c\psi} \log\left(\frac{N(\mathcal{B}_2^d, \|\cdot\|_1, \nu/2)}{\delta}\right)$$

samples. Plugging in Lemma I.1

$$n = c^{-d}\psi^{-d}(6\sqrt{d})^d \cdot \left(d\log(6\sqrt{d}/\delta)\right)\nu^{-d}.$$

samples are needed for this event to happen with probability at least $1-\delta$.

□

Lemma J.2. *Let π_i be a policy satisfying Assumption 4.1 used to collect n i.i.d. trajectories. Let $\mathcal{D}_n^h = \{(s_1, a_1), \ldots, (s_n, a_n)\}$ denote the n state action pairs in the offline dataset at time step h. Then with probability at least $1-\delta$, for every $(s,a) \in \mathcal{S}\times\mathcal{A}$ and for all $h \in [H]$ there exists $\gamma \in \mathbb{N}$ state action pairs $(s', a') \in \mathcal{D}_n^h$ such that $\sup_{\phi\in\Phi} \|\phi(s,a) - \phi(s',a')\|_1 \leq \nu$ if*

$$n \geq C_1 \nu^{-d},$$

where $C_1 = \gamma^d c^{-d}\psi^{-d}(6\sqrt{d})^d \cdot \left(d\log(6\sqrt{d}/\delta)\right)$.

Proof. Setting $\nu \to \nu/\gamma$ in Lemma J.1 and using Lemma I.4 gives us the result. □

Lemma J.3. *Let $\{\pi_1, \ldots, \pi_K\}$ be policies satisfying Assumption 4.1. Suppose N_S i.i.d. trajectories are sampled from each task i by policy π_i, then with probability at least $1-\delta$, for all $(s,a) \in \mathcal{S} \times \mathcal{A}$ we can upper bound the transition model estimation error as:*

$$\|\mu_h'(\cdot)^\top \widehat{\phi_h}(s,a) - P_h^*(\cdot|s,a)\|_{TV} \leq C_2 {N_S}^{-1/4d} \alpha_{\max} \left((\log(2|\Phi|/\delta) + K \log|\Upsilon|) \right)^{1/4} K^{3/4} d^{1/2} \psi^{-1/4},$$

where C_2 is a finite constant.

Proof. Let us use ϵ to denote the desired error tolerance in the transition model. We choose $\gamma = 4\epsilon^{-2}\alpha_{\max}^2(K\log(2|\Phi|/\delta) + K^2 \log|\Upsilon|)$. If we sample $N_S = \gamma^d c^{-d} \psi^{-d} (6\sqrt{d})^d \cdot \left(d \log(6\sqrt{d}/\delta)\right) \nu^{-d}$,

$$\min_{i\in[K]} D_{i;h}^{\nu}(s,a) N_S \geq \gamma = 4\epsilon^{-2}\alpha_{\max}^2(K\log(2|\Phi|/\delta) + K^2 \log|\Upsilon|)$$

$$\implies 2\alpha_{\max} \sqrt{\frac{K\log(2|\Phi|/\delta) + K^2\log|\Upsilon|}{\min_{i\in[K]} D_{i;h}^{\nu}(s,a) N_S}} \leq \epsilon$$

Writing the total variance as a function of ν (setting all ν_i identical to ν). With probability at least $1-\delta/2$ we can upper bound the total variance as

$$\max_{i\in[K]} \frac{2}{D_{i;h}^{\nu}(s,a)} \frac{\log(2|\Phi|/\delta) + K\log|\Upsilon|}{N_S}.$$

By our choice of γ, by the union bound with probability at least $1-\delta$, the total variance variance gets upper bounded by $\frac{\epsilon^2}{2K\alpha_{\max^2}}$.

Note that the total bias is $2\nu dK$. Since the sample complexity is decreasing in ν. We want to find largest ν such that variance is larger than bias. Thus we equate upper bound on variance to bias to compute ν.

This gives the optimal $\nu = \frac{\epsilon^2}{4dK^2\alpha_{\max}^2}$.

Plugging the values of ν and γ, we get

$$N_S = \epsilon^{-4d} \left(2\alpha_{\max}\right)^{4d} \left((\log(2|\Phi|/\delta) + K\log|\Upsilon|) \right)^d c^{-d}\psi^{-d}(6\sqrt{d})^d \cdot \left(d\log(6\sqrt{d}/\delta)\right) d^d K^{3d}.$$

We can write the error bound ϵ in terms of N_S, we get:

$$\epsilon = C_2 {N_S}^{-1/4d} \alpha_{\max} \left((\log(2|\Phi|/\delta) + K\log|\Upsilon|) \right)^{1/4} K^{3/4} d^{1/2} \psi^{-1/4},$$

where C_2 is a positive constant (upto factor in $\log(6\sqrt{d}/\delta)^{1/4d}$). □

Lemma J.4. *Suppose $\bar{\pi}$ is a policy satisfying Assumption 4.2 and n i.i.d. trajectories are sampled from the target task by policy $\bar{\pi}$. Then with probability at least $1-\delta/2$ we can show that:*

$$2H\beta \sum_{h=1}^{H} \mathbb{E}_{(s_h,a_h)\sim\pi^*,P_h^*}[\|\widehat{\phi}_h(s_h,a_h)\|_{\Lambda_h}|s_1 = s] \leq c_1 n^{-1/2} d^{3/2} H^2 \sqrt{\log(4dHn/\delta)},$$

for some finite constant c_1.

Proof. Proof directly follows from proof of Corollary 4.6 in [Jin et al., 2021]. □

Corollary 4.1. *Let $\bar{\pi}$ be a policy satisfying Assumption 4.2 and $\{\bar{\pi}_1, \ldots, \bar{\pi}_K\}$ be policies satisfying Assumption 4.1. Suppose n i.i.d. trajectories are sampled from the target task by policy $\bar{\pi}$ and N_S i.i.d. trajectories are sampled from each task i by policy π_i. Then with probability at least $1-\delta$, the suboptimality gap is upper bounded as:*

$$SubOpt(\widehat{\pi}, s) \leq \tilde{\mathcal{O}}(\max({N_S}^{-\frac{1}{4d}}, n^{-\frac{1}{2}}) H^2 d^{\frac{3}{2}} K^{\frac{3}{4}} \sqrt{\log(1/\delta)}).$$

Proof. Let consider 2 events as the ones stated in Lemma J.3 and Lemma J.4, each likely to happen with probability at least $1-\delta/2$. By using union bound it is easy to show that both these events hold simultaneously with probability at least $1-\delta$. Plugging these upper bounds in Theorem 4.1 we get with probability at least $1-\delta$:

$$\text{SubOpt}(\widehat{\pi}, s) \leq 2H \sum_{h=1}^{H} \mathbb{E}_{(s_h,a_h)\sim\pi^*,P_h^*}\Bigg[\underbrace{\epsilon(s_h,a_h)}_{\text{source coverage on } \pi^*} + (\beta + \underbrace{\epsilon_h}_{\text{source coverage on target}}) \cdot \underbrace{\|\widehat{\phi}_h(s_h,a_h)\|_{\Lambda_h}}_{\text{target coverage on } \pi^*} \mid s_1 = s \Bigg]$$

$$\leq C_1' H^2 {N_S}^{-1/4d} \alpha_{\max} \Big((\log(2|\Phi|/\delta) + K \log|\Upsilon|) \Big)^{1/4} K^{3/4} d^{1/2} \psi^{-1/4} k$$

$$+ C_2' n^{-1/2} d^{3/2} H^2 \sqrt{\log(4dHn/\delta)}$$

$$+ C_3' H^2 {N_S}^{-1/4d} \alpha_{\max} \Big((\log(2|\Phi|/\delta) + K \log|\Upsilon|) \Big)^{1/4} K^{3/4} d \psi^{-1/4} k$$

$$\leq \tilde{\mathcal{O}}({N_S}^{-\frac{1}{4d}} n^{-\frac{1}{2}} H^2 d^{\frac{3}{2}} K^{\frac{3}{4}} \sqrt{\log(1/\delta)}).$$

□

K Experiment Details

Environment Description: In this section, we introduce the Combination lock (Comblock) environment, a widely adopted benchmark for algorithms designed for Block Markov Decision Processes (MDPs). Figure 2 provides a visualization of the Comblock environment. Specifically, the environment encompasses a horizon denoted as H, and at each timestep h, it includes 3 latent states $z_{i;h}$, where $i \in \{0,1,2\}$, along with 5 possible actions. Within the three latent states, we designate z_0 and z_1 as the desirable states leading to the final reward, while z_2 represents undesirable states. At the onset of the task, the environment uniformly and independently samples one out of 5 possible actions for each good state $z_{0;h}$ and $z_{1;h}$ at each timestep h. These sampled actions, denoted as $a_{0;h}$ and $a_{1;h}$, respectively, are considered optimal actions corresponding to each latent state. These optimal actions, in conjunction with the task itself, dictate the dynamics of the environment. At each good latent state $s_{0;h}$ or $s_{1;h}$, taking the correct action results in a transition to either good state at the next timestep (i.e., $s_{0;h+1}$, $s_{1;h+1}$) with equal probability. Conversely, if the agent chooses any of the four bad actions, the environment deterministically transitions to the bad state $s_{2;h+1}$, and the bad states transition only to bad states at the subsequent timestep. The agent receives a reward in two scenarios: firstly, upon reaching the good states at the last timestep, the agent receives a reward of 1; secondly, upon the first transition into the bad state, the agent receives an "anti-shaped" reward of 0.1 with a probability of 0.5. This design renders greedy algorithms, lacking strategic exploration such as policy optimization methods, susceptible to failure. Regarding the initial state distribution, the environment begins in either $s_{0;0}$ or $s_{1;0}$ with equal probability. The dimension of the observation is $2^{\log(H+|S|+1)}$. For the emission distribution, given a latent state $s_{i;h}$, the observation is generated by concatenating the one-hot vectors of the state and the horizon. Additionally, i.i.d. $\mathcal{N}(0, 0.1)$ noise is added at each entry, and if necessary, a 0 is appended at the end. Finally, a linear transformation is applied to the observation using a Hadamard matrix. It's noteworthy that, without effective features or strategic exploration, it requires 5^H actions with random actions to reach the final goal.

Generating Source and Target Tasks: To create the source environment, we randomly generate five instances of the Comblock environment as described. It's important to note that this approach ensures a shared emission distribution across the sources, while the latent dynamics differ due to independently and randomly selected optimal actions. To construct the target environment, for each timestep h, we randomly select optimal actions at h from one of the sources and designate them as the optimal actions for the target environment at timestep h. This is contingent upon the condition that the selected optimal actions differ for the two good states. If the optimal actions are the same, we continue sampling until distinct actions are obtained. This procedure ensures variability in the optimal actions, introducing diversity in the latent dynamics of the target environment.