
Overcoming the Sim-to-Real Gap: Leveraging Simulation to Learn to Explore for Real-World RL

Andrew Wagenmaker*
University of California, Berkeley

Kevin Huang
University of Washington

Liyiming Ke
University of Washington

Kevin Jamieson
University of Washington

Abhishek Gupta
University of Washington

Abstract

In order to mitigate the sample complexity of real-world reinforcement learning, common practice is to first train a policy in a simulator where samples are cheap, and then deploy this policy in the real world, with the hope that it generalizes effectively. Such *direct sim2real* transfer is not guaranteed to succeed, however, and in cases where it fails, it is unclear how to best utilize the simulator. In this work, we show that in many regimes, while direct sim2real transfer may fail, we can utilize the simulator to learn a set of *exploratory* policies which enable efficient exploration in the real world. In particular, in the setting of low-rank MDPs, we show that coupling these exploratory policies with simple, practical approaches—least-squares regression oracles and naive randomized exploration—yields a polynomial sample complexity in the real world, an exponential improvement over direct sim2real transfer, or learning without access to a simulator. To the best of our knowledge, this is the first evidence that simulation transfer yields a provable gain in reinforcement learning in settings where direct sim2real transfer fails. We validate our theoretical results on several realistic robotic simulators and a real-world robotic sim2real task, demonstrating that transferring exploratory policies can yield substantial gains in practice as well.

1 Introduction

Over the last decade, reinforcement learning (RL) techniques have been deployed to solve a variety of real-world problems, with applications in robotics, the natural sciences, and beyond [27, 54, 52, 26, 46, 23]. While promising, the broad application of RL methods has been severely limited by its large sample complexity—the number of interactions with the environment required for the algorithm to learn to solve the desired task. In applications of interest, it is often the case that collecting samples is very costly, and the number of samples required by RL algorithms is prohibitively expensive.

In many domains, while collecting samples in the desired deployment environment may be very costly, we have access to a *simulator* where the cost of samples is virtually nonexistent. As a concrete example, in robotic applications where the goal is real-world deployment, directly training in the real world typically requires an infeasibly large number of samples. However, it is often possible to obtain a simulator—derived from first principles or knowledge of the robot’s actuation—which provides an approximate model of the real-world deployment environment. Given such a simulator, common practice is to first train a policy to accomplish the desired task in the simulator, and then deploy it in the real world, with the hope that the policy generalizes effectively from the simulator to the goal deployment environment. Indeed, such “sim2real” transfer has become a key piece in the

*Correspondance to: ajwagen@berkeley.edu

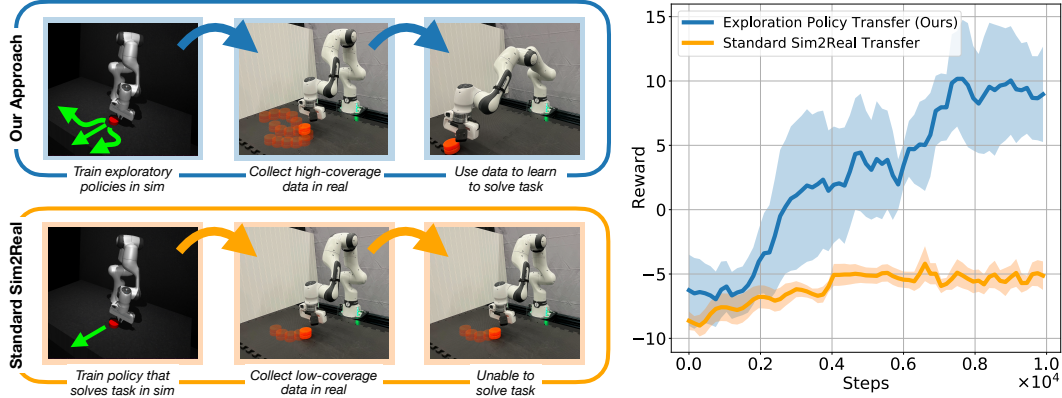


Figure 1: **Left:** Overview of our approach compared to standard sim2real transfer on puck pushing task. Standard sim2real transfer first trains a policy to solve the goal task in sim and then transfers this policy to real. This policy may fail to solve the task in real due to the sim2real gap, and furthermore may not provide sufficient data coverage to successfully learn a policy that does solve the goal task in real. In contrast, our approach trains a set of exploratory policies in sim which achieve high-coverage data when deployed in real, even if they are unable to solve the task 0-shot. This high-coverage data can then be used to successfully learn a policy that solves the goal task in real. **Right:** Quantitative results running our approach on the puck pushing task illustrated on left, compared to standard sim2real transfer. Over 6 real-world trials, our approach solves the task 6/6 times while standard sim2real transfer solves the task 0/6 times.

application of RL to robotic settings, as well as many other domains of interest such as the natural sciences [12, 15], and is a promising approach towards reducing the sample complexity of RL in real-world deployment [19, 4, 18].

Effective sim2real transfer can be challenging, however, as there is often a non-trivial mismatch between the simulated and real environments. The real world is difficult to model perfectly, and some discrepancy is inevitable. As such, directly transferring the policy trained in the simulator to the real world often fails, the mismatch between sim and real causing the policy—which may perfectly solve the task in sim—to never solve the task in real. While some attempts have been made to address this—for example, utilizing domain randomization to extend the space of environments covered by simulator [60, 49], or finetuning the policy learned in sim in the real world [50, 73]—these approaches are not guaranteed to succeed. In settings where such methods fail, can we still utilize a simulator to speed up real-world RL?

In this work we take steps towards developing principled approaches to sim2real transfer that addresses this question. Our key intuition is that it is often *easier to learn to explore than to learn to solve the goal task*. While solving the goal task may require very precise actions, collecting high-quality exploratory data can require significantly less precision. For example, successfully solving a complex robotic manipulation task requires a particular sequence of motions, but obtaining a policy that will interact with the object of interest in some way, providing useful exploratory data on its behavior, would require significantly less precision.

Formally, we show that, in the setting of low-rank MDPs where there is a mismatch in the dynamics between the “sim” and “real” environments, even when this mismatch is such that direct sim2real transfer fails, under certain conditions we can still effectively transfer a set of *exploratory* policies from sim to real. In particular, we demonstrate that access to such exploratory policies, coupled with random exploration and a least-squares regression oracle—which are insufficient for efficient learning on their own, but often still favored in practice due to their simplicity—enable provably efficient learning in real. Our results therefore demonstrate that simulators, when carefully applied, can yield a provable—exponential—gain over both naive sim2real transfer and learning without a simulator, and enable algorithms commonly used in practice to learn efficiently.

Furthermore, our results motivate a simple, easy-to-implement algorithmic principle: rather than training and transferring a policy that solves the task in the simulator, utilize the simulator to train a set of exploratory policies, and transfer these, coupled with random exploration, to generate high

quality exploratory data in real. We show experimentally—through a realistic robotic simulator and real-world sim2real transfer problem on the Franka robot platform—that this principle of transferring exploratory policies from sim to real yields a significant practical gain in sample efficiency, often enabling efficient learning in settings where naive transfer fails completely (see Figure 1).

2 Related Work

Provable Transfer in RL. Perhaps the first theoretical result on transfer in RL is the “simulation lemma”, which transforms a bound on the total-variation distance between the dynamics to a bound on policy value [24, 25, 6, 22]—we argue that we can do significantly better with exploration transfer. More recent work has considered transfer in the setting of block MDPs [34], but requires relatively strong assumptions on the similarity between source and target MDPs, or the meta-RL setting [69], but only consider tabular MDPs, and assume the target MDP is covered by the training distribution. Perhaps most relevant to this work is the work of [36], which presents several lower bounds showing that efficient transfer in RL is not feasible in general. In relation to this work, our work can be seen as providing a set of sufficient conditions that do enable efficient transfer; the lower bounds presented in [36] do not hold in the low-rank MDP setting we consider. Several other works exist, but either consider different types of transfer than what we consider (e.g., observation space mismatch), or only learn a policy that has suboptimality bounded by the sim2real mismatch [37, 56, 58]. Another somewhat tangential line of work considers representation transfer in RL, where it is assumed the source and target tasks share a common representation [35, 10, 2]. We remark as well that the formal sim2real setting we consider is a special case of the MF-MDP setting of [53].

Simulators and Low-Rank MDPs. Several existing works show that there are provable benefits to training a policy in “simulation” due to the ability to reset on command [67, 33, 5, 68, 70, 42]. These works do not consider the transfer problem, however. The setting of linear and low-rank MDPs which we consider has seen a significant amount of attention over the last several years, and many provably efficient algorithms exist [21, 1, 62, 63, 43, 41]. These works typically assume access to powerful oracles which enable efficient learning; we only consider access to a simple regression oracle. Beyond the theory literature, recent work has also shown that low-rank MDPs can effectively model a variety of standard RL settings in practice [72].

Sim2Real Transfer in Practice. The sim2real literature is vast and we only highlight particularly relevant works here; see [74] for a full survey. To mitigate the inconsistency between the simulator and real world’s physical parameters and modeling, domain randomization creates a variety of simulated environments with randomized properties to develop a robust policy [60, 49, 44, 8, 39]. Domain adaptation instead constructs encoding of deployment conditions (e.g., physical condition or past histories) and adapts to the deployment environment by matching the encoding [29, 9, 66, 55, 38, 40]. In contrast, our work assumes a fundamental sim2real mismatch where we do not expect the real system to match the simulator for any parameter settings. A related line of work shows that policies trained with robust exploration strategies generalize better to disturbed or unseen environments [13, 20]. Our work is complimentary to this work in that our goal is not to transfer a policy that solves the task in new environment, but rather explores the environment.

3 Preliminaries

We let $\Delta_{\mathcal{X}}$ denote the set of distributions over set \mathcal{X} , $[H] := \{1, 2, \dots, H\}$, and $\|P - Q\|_{\text{TV}}$ the total-variation distance between distributions P and Q .

Markov Decision Processes. We consider the setting of episodic Markov Decision Processes (MDPs). An MDP is denoted by a tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \{P_h\}_{h=1}^H, \{r_h\}_{h=1}^H, s_1, H)$, where \mathcal{S} denotes the set of states, \mathcal{A} the set of actions, $P_h : \mathcal{S} \times \mathcal{A} \rightarrow \Delta_{\mathcal{S}}$ the transition function, $r_h : \mathcal{S} \times \mathcal{A} \rightarrow [0, 1]$ the reward (which we assume is deterministic and known), s_1 the initial state, and H the horizon. We assume \mathcal{A} is finite and denote $A := |\mathcal{A}|$. Interaction with an MDP starts from state s_1 , the agent takes some action a_1 , transitions to state $s_2 \sim P_1(\cdot | s_1, a_1)$, and receives reward $r_1(s_1, a_1)$. This process continues for H steps at which points the episode terminates, and the process resets.

The goal of the learner is to find a policy $\pi = \{\pi_h\}_{h=1}^H$, $\pi_h : \mathcal{S} \rightarrow \Delta_{\mathcal{A}}$, that achieves maximum reward. We can quantify the reward received by some policy π in terms of the value and Q -value

functions. The Q -value function is defined as $Q_h^\pi(s, a) := \mathbb{E}^\pi[\sum_{h'=h}^H r_{h'}(s_{h'}, a_{h'}) \mid s_h = s, a_h = a]$, and value function is defined in terms of the Q -value function as $V_h^\pi(s) := \mathbb{E}_{a \sim \pi_h(\cdot \mid s)}[Q_h^\pi(s, a)]$. The value of policy π , its expected reward, is denoted by $V_0^\pi := V_1^\pi(s_1)$, and the value of the optimal policy, the maximum achievable reward, by $V_0^* := \sup_\pi V_0^\pi$.

In this work we are interested in the setting where we wish to solve some task in the “real” environment, represented as an MDP, and we have access to a simulator which approximates the real environment in some sense. We denote the real MDP as $\mathcal{M}^{\text{real}}$, and the simulator as \mathcal{M}^{sim} . We assume that $\mathcal{M}^{\text{real}}$ and \mathcal{M}^{sim} have the same state and actions spaces, reward function, and initial state, but different transition functions, P^{real} and P^{sim} . We denote value functions in $\mathcal{M}^{\text{real}}$ and \mathcal{M}^{sim} as $V_h^{\text{real}, \pi}(s)$ and $V_h^{\text{sim}, \pi}(s)$, respectively. We make the following assumption.

Assumption 1. For all $(s, a, h) \in \mathcal{S} \times \mathcal{A} \times [H]$ and some $\epsilon_{\text{sim}} > 0$, we have:

$$\|P_h^{\text{real}}(\cdot \mid s, a) - P_h^{\text{sim}}(\cdot \mid s, a)\|_{\text{TV}} \leq \epsilon_{\text{sim}}.$$

We do not assume that the value of ϵ_{sim} is known, simply that there exists some such ϵ_{sim} .

Function Approximation. In order to enable efficient learning, some structure on the MDPs of interest is required. We will assume that $\mathcal{M}^{\text{real}}$ and \mathcal{M}^{sim} are low-rank MDPs, as defined below.

Definition 3.1 (Low-Rank MDP). We say an MDP is a *low-rank* MDP with dimension d if there exists some featurization $\phi : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}^d$ and measure $\mu : [H] \times \mathcal{S} \rightarrow \mathbb{R}^d$ such that:

$$P_h(\cdot \mid s, a) = \langle \phi(s, a), \mu_h(\cdot) \rangle, \quad \forall s, a, h.$$

We assume that $\|\phi(s, a)\|_2 \leq 1$ for all (s, a) , and for all h , $\|\mu_h(\cdot)\|_2 = \|\int_{s \in \mathcal{S}} d\mu_h(s)\|_2 \leq \sqrt{d}$.

Formally, we make the following assumption on the structure of \mathcal{M}^{sim} and $\mathcal{M}^{\text{real}}$.

Assumption 2. Both \mathcal{M}^{sim} and $\mathcal{M}^{\text{real}}$ satisfy Definition 3.1 with feature maps and measures (ϕ^s, μ^s) and (ϕ^r, μ^r) , respectively. Furthermore, ϕ^s is known, but all of μ^s, ϕ^r , and μ^r are unknown.

In the literature, MDPs satisfying Definition 3.1 but where ϕ is known are typically referred to as “linear” MDPs, while MDPs satisfying Definition 3.1 but with ϕ unknown are typically referred to as “low-rank” MDPs. Given this terminology, we have that \mathcal{M}^{sim} is a linear MDP², while $\mathcal{M}^{\text{real}}$ is a low-rank MDP. We assume the following reachability condition on \mathcal{M}^{sim} .

Assumption 3. There $\exists \lambda_{\min}^* > 0$ with $\min_h \sup_\pi \lambda_{\min}(\mathbb{E}^{\mathcal{M}^{\text{sim}, \pi}}[\phi^s(s_h, a_h)\phi^s(s_h, a_h)^\top]) \geq \lambda_{\min}^*$.

Assumption 3 posits that each direction in the feature space in our simulator can be activated by some policy, and can be thought of as a measure of how easily each direction can be reached. Similar assumptions have appeared before in the literature on linear and low-rank MDPs [71, 3, 2]. Note that we only require this reachability assumption in \mathcal{M}^{sim} . We also assume we are given access to function classes $\mathcal{F}_h : \mathcal{S} \times \mathcal{A} \rightarrow [0, H]$ and let $\mathcal{F} := \mathcal{F}_1 \times \mathcal{F}_2 \times \dots \times \mathcal{F}_H$. Since no reward is collected in the $(H + 1)$ th step we take $f_{H+1} = 0$. For any $f : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$, we let $\pi_h^f(s) := \arg \max_{a \in \mathcal{A}} f_h(s, a)$. We define the *Bellman operator* on some function $f_{h+1} : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ as:

$$\mathcal{T}f_{h+1}(s, a) := r_h(s, a) + \mathbb{E}_{s' \sim P_h(\cdot \mid s, a)}[\max_{a'} f_{h+1}(s', a')].$$

We make the following standard assumption on \mathcal{F} .

Assumption 4 (Bellman Completeness). For all $f_{h+1} \in \mathcal{F}_{h+1}$, we have $\mathcal{T}^{\text{real}}f_{h+1}, \mathcal{T}^{\text{sim}}f_{h+1} \in \mathcal{F}_h$, where $\mathcal{T}^{\text{real}}$ and \mathcal{T}^{sim} denote the Bellman operators on $\mathcal{M}^{\text{real}}$ and \mathcal{M}^{sim} , respectively.

PAC Reinforcement Learning. Our goal is to find a policy $\hat{\pi}$ that achieves maximum reward in $\mathcal{M}^{\text{real}}$. Formally, we consider the PAC (Probably-Approximately-Correct) RL setting.

Definition 3.2 (PAC Reinforcement Learning). Given some $\epsilon > 0$ and $\delta > 0$, with probability at least $1 - \delta$ identify some policy $\hat{\pi}$ such that: $V_0^{\text{real}, \hat{\pi}} \geq V_0^{\text{real}, *} - \epsilon$.

We will be particularly interested in solving the PAC RL problem with the aid of a simulator, using the minimum number of samples from $\mathcal{M}^{\text{real}}$ possible, as we will formalize in the following. As we will see, while it is straightforward to achieve this objective using \mathcal{M}^{sim} if $\epsilon = \mathcal{O}(\epsilon_{\text{sim}})$, naive transfer methods can fail to achieve this completely if $\epsilon \ll \epsilon_{\text{sim}}$. As such, our primary focus will be on developing efficient sim2real methods in this regime.

²The assumption that ϕ^s is known is for simplicity only—similar results could be obtained were ϕ^s also unknown using more complex algorithmic tools in \mathcal{M}^{sim} .

4 Theoretical Results

In this section we provide our main theoretical results. We first present two negative results: in Section 4.1 showing that “naive exploration”—utilizing only a least-squares regression oracle and random exploration approaches such as ζ -greedy³—is provably inefficient, and in Section 4.2 showing that directly transferring the optimal policy from \mathcal{M}^{sim} to $\mathcal{M}^{\text{real}}$ is unable to efficiently obtain a policy with suboptimality better than $\mathcal{O}(\epsilon_{\text{sim}})$ in real. Then in Section 4.3 we present our main positive result, showing that by utilizing the same oracles as in Sections 4.1 and 4.2—a least-squares regression oracle, simulator access, and the ability to take actions randomly—we *can* efficiently learn an ϵ -optimal policy for $\epsilon \ll \epsilon_{\text{sim}}$ in $\mathcal{M}^{\text{real}}$ by carefully utilizing the simulator to learn exploration policies.

4.1 Naive Exploration is Provably Inefficient

While a variety of works have developed provably efficient methods for solving PAC RL in low-rank MDPs [1, 62, 43, 41], these works typically either rely on complex computation oracles or carefully directed exploration strategies which are rarely utilized in practice. In contrast, RL methods utilized in practice typically rely on “simple” computation oracles and exploration strategies. Before considering the sim2real setting, we first show that such “simple” strategies are insufficient for efficient PAC RL. To instantiate such strategies, we consider a least-squares regression oracle, often available in practice.

Oracle 4.1 (Least-Squares Regression Oracle). We assume access to a least-squares regression oracle such that, for any h and dataset $\mathcal{D} = \{(s^t, a^t, y^t)\}_{t=1}^T$, we can compute $\arg \min_{f \in \mathcal{F}_h} \sum_{t=1}^T (f(s^t, a^t) - y^t)^2$.

We couple this oracle with “naive exploration”, which here we use to refer to any method that explores by randomly perturbing the action recommended by the current estimate of the optimal policy. While a variety of instantiations of naive exploration exist (see e.g. [11]), we consider a particularly common formulation, ζ -greedy exploration.

Protocol 4.1 (ζ -Greedy Exploration). Given access to a regression oracle, any $\zeta \in [0, 1]$, and time horizon T , consider the following protocol:

1. Interact with $\mathcal{M}^{\text{real}}$ for T episodes. At every step of episode $t + 1$, play $\pi_h^{f_h^t}(s)$ with probability $1 - \zeta$, and $a \sim \text{unif}(\mathcal{A})$ otherwise, where:

$$f_h^t = \arg \min_{f \in \mathcal{F}_h} \sum_{t'=1}^t (f(s_h^{t'}, a_h^{t'}) - r_h^{t'} - \max_{a'} f_{h+1}^t(s_{h+1}^{t'}, a'))^2.$$

2. Using collected data in any way desired, propose a policy $\hat{\pi}$.

Protocol 4.1 forms the backbone of many algorithms used in practice. Despite its common application, as existing work [11] and the following result show, it is provably inefficient.

Proposition 1. *For any $H > 1$, $\zeta \in [0, 1]$, and $c \leq 1/6$, there exist some $\mathcal{M}^{\text{real},1}$ and $\mathcal{M}^{\text{real},2}$ such that both $\mathcal{M}^{\text{real},1}$ and $\mathcal{M}^{\text{real},2}$ satisfy Assumptions 2 and 4, and unless $T \geq \Omega(2^{H/2})$, when running Protocol 4.1 we have:*

$$\sup_{\mathcal{M}^{\text{real}} \in \{\mathcal{M}^{\text{real},1}, \mathcal{M}^{\text{real},2}\}} \mathbb{E}^{\mathcal{M}^{\text{real}}} [V_0^{\mathcal{M}^{\text{real}},*} - V_0^{\mathcal{M}^{\text{real}},\hat{\pi}}] \geq c/32.$$

Proposition 1 shows that, in a minimax sense, ζ -greedy exploration is insufficient for provably efficient reinforcement learning: on one of $\mathcal{M}^{\text{real},1}$ and $\mathcal{M}^{\text{real},2}$, ζ -greedy exploration will only be able to find a policy that is suboptimal by a constant factor, unless we take an exponentially large number of samples. While we focus on ζ -greedy exploration in Proposition 1, this result extends to other types of naive exploration, for example, those given in [11]. See Section 5.2 for further discussion of the construction for Proposition 1.

4.2 Understanding the Limits of Direct sim2real Transfer

Proposition 1 shows that in general utilizing a least-squares regression oracle with ζ -greedy exploration is insufficient for provably efficient RL. Can this be made efficient with access to a simulator

³Throughout this paper, we use “ ζ -greedy” to refer to the method more commonly known as “ ϵ -greedy” in the literature, to avoid ambiguity between this ϵ and the ϵ in our definition of PAC RL, Definition 3.2.

\mathcal{M}^{sim} ? In practice, standard sim2real methodology typically trains a policy to accomplish the goal task in \mathcal{M}^{sim} , and then transfers this policy to $\mathcal{M}^{\text{real}}$. We refer to this methodology as *direct sim2real transfer*. The following canonical result, usually referred to as the “simulation lemma” [24, 25, 6, 22], provides a sufficient guarantee for direct sim2real transfer to succeed under Assumption 1.

Proposition 2 (Simulation Lemma). *Let $\pi^{\text{sim},*}$ denote an optimal policy in \mathcal{M}^{sim} . Then under Assumption 1 we have $V_0^{\text{real},\pi^{\text{sim},*}} \geq V_0^{\text{real},*} - 2H^2\epsilon_{\text{sim}}$.*

Proposition 2 shows that, as long as $\epsilon \geq 2H^2\epsilon_{\text{sim}}$, direct sim2real transfer succeeds in obtaining an ϵ -optimal policy in $\mathcal{M}^{\text{real}}$. While this justifies direct sim2real transfer in settings where \mathcal{M}^{sim} and $\mathcal{M}^{\text{real}}$ are sufficiently close, we next show that given access only to $\pi^{\text{sim},*}$ and a least-squares regression oracle—even when coupled with random exploration—we cannot hope to efficiently obtain a policy with suboptimality less than $\mathcal{O}(\epsilon_{\text{sim}})$ on $\mathcal{M}^{\text{real}}$ using naive exploration. To formalize this, we consider the following interaction protocol.

Protocol 4.2 (Direct sim2real Transfer with Naive Exploration). Given access to $\pi^{\text{sim},*}$, an optimal policy in \mathcal{M}^{sim} , any $\zeta \in [0, 1]$, and time horizon T , consider the following protocol:

1. Interact with $\mathcal{M}^{\text{real}}$ for T episodes, and at each step h and state s play $\pi_h^{\text{sim},*}(\cdot | s)$ with probability $1 - \zeta$, and $a \sim \text{unif}(\mathcal{A})$ with probability ζ .
2. Using collected data in any way desired, propose a policy $\hat{\pi}$.

Protocol 4.2 is a standard instantiation of direct sim2real transfer commonly found in the literature, and couples playing the optimal policy from \mathcal{M}^{sim} with naive exploration. We have the following.

Proposition 3. *With the same choice of $\mathcal{M}^{\text{real},1}$ and $\mathcal{M}^{\text{real},2}$ as in Proposition 1, there exists some \mathcal{M}^{sim} such that both $\mathcal{M}^{\text{real},1}$ and $\mathcal{M}^{\text{real},2}$ satisfy Assumption 1 with \mathcal{M}^{sim} for $\epsilon_{\text{sim}} \leftarrow c$, Assumptions 2 to 4 hold, and unless $T \geq \Omega(2^H)$ when running Protocol 4.2, we have:*

$$\sup_{\mathcal{M}^{\text{real}} \in \{\mathcal{M}^{\text{real},1}, \mathcal{M}^{\text{real},2}\}} \mathbb{E}^{\mathcal{M}^{\text{real}}} [V_0^{\mathcal{M}^{\text{real}},*} - V_0^{\mathcal{M}^{\text{real}},\hat{\pi}}] \geq \epsilon_{\text{sim}}/32.$$

Proposition 3 shows that there exists a setting where there are two possible $\mathcal{M}^{\text{real}}$ satisfying Assumption 1 with \mathcal{M}^{sim} , and where, using direct policy transfer, unless we interact with $\mathcal{M}^{\text{real}}$ for exponentially many episodes (in H), we cannot determine a better than $\Omega(\epsilon_{\text{sim}})$ -optimal policy for the worst-case $\mathcal{M}^{\text{real}}$. Together, Propositions 2 and 3 show that, while we can utilize direct sim2real transfer to learn a policy that is $\mathcal{O}(\epsilon_{\text{sim}})$ -optimal in $\mathcal{M}^{\text{real}}$, if our goal is to learn an ϵ -optimal policy for $\epsilon \ll \epsilon_{\text{sim}}$, direct sim2real transfer is unable to efficiently achieve this.

4.3 Efficient sim2real Transfer via Exploration Policy Transfer

Does there exist *some* way to utilize \mathcal{M}^{sim} and a least-squares regression oracle to enable efficient learning in $\mathcal{M}^{\text{real}}$, even when $\epsilon \ll \epsilon_{\text{sim}}$? Our key insight is that, rather than transferring the policy that optimally solves the task in \mathcal{M}^{sim} , we should instead transfer policies that *explore* effectively in \mathcal{M}^{sim} . While learning to solve a task may require very precise actions, we can often obtain sufficiently rich data with relatively imprecise actions—it is easier to learn to explore than learn to solve a task. In such settings, directly transferring a policy to solve the task will likely fail due to imprecision in the simulator, but it may be possible to still transfer a policy that generates exploratory data. To formalize this, we consider the following access model to \mathcal{M}^{sim} .

Oracle 4.2 (\mathcal{M}^{sim} Access). We may interact with \mathcal{M}^{sim} by either:

1. **(Trajectory Sampling)** For any policy π , sampling a trajectory $\{(s_h, a_h, r_h, s_{h+1})\}_{h=1}^H$ generated by playing π on \mathcal{M}^{sim} .
2. **(Policy Optimization)** For any reward \tilde{r} , computing a policy $\pi^{\text{sim}}(\tilde{r})$ maximizing \tilde{r} on \mathcal{M}^{sim} .

While access to such a policy optimization oracle is unrealistic in $\mathcal{M}^{\text{real}}$, where we want to minimize the number of samples collected, given cheap access to samples in \mathcal{M}^{sim} , such an oracle can often be (approximately) implemented in practice⁴. Note that under Oracle 4.2 we only assume *black-box*

⁴While for simplicity we assume that the truly optimal policy can be computed, our results easily extend to settings where we only have access to an oracle which can compute an approximately optimal policy.

access to our simulator—rather than allowing the behavior of the simulator to be queried at arbitrary states, we are simply allowed to roll out policies on \mathcal{M}^{sim} , and compute optimal policies. Given Oracle 4.2, as well as our least-squares regression oracle, Oracle 4.1, we propose the following algorithm.

Algorithm 1 sim2real Exploration Policy Transfer

```

1: input: budget  $T$ , confidence  $\delta$ , simulator  $\mathcal{M}^{\text{sim}}$ 
   // Learn policies  $\Pi_{\text{exp}}^h$  which cover feature space in  $\mathcal{M}^{\text{sim}}$ 
2:  $\Pi_{\text{exp}}^h \leftarrow \text{LEARNEXPOLICIES}(\mathcal{M}^{\text{sim}}, \delta, \frac{4A^3\epsilon}{H}, h)$  (Algorithm 5) for all  $h \in [H]$ 
3:  $\tilde{\Pi}_{\text{exp}}^h \leftarrow \{\tilde{\pi}_{\text{exp}} : \tilde{\pi}_{\text{exp}} \text{ plays } \pi_{\text{exp}} \text{ up to step } h, \text{ then plays actions randomly, } \forall \pi_{\text{exp}} \in \Pi_{\text{exp}}^h\}$ 
   // Explore in  $\mathcal{M}^{\text{real}}$  via  $\tilde{\Pi}_{\text{exp}}$ 
4: Play  $\pi_{\text{exp}} \sim \text{unif}(\{\text{unif}(\tilde{\Pi}_{\text{exp}}^h)\}_{h=1}^H)$  for  $T/2$  episodes in  $\mathcal{M}^{\text{real}}$ , add data to  $\mathcal{D}$ 
   // Estimate optimal policy on collected data
5: for  $h = H, H-1, \dots, 1$  do
6:    $\hat{f}_h \leftarrow \arg \min_{f \in \mathcal{F}} \sum_{(s,a,r,s') \in \mathcal{D}} (f_h(s,a) - r - \max_{a'} \hat{f}_{h+1}(s',a'))^2$ 
7:   Compute  $\pi^{\text{sim},*}$  via Oracle 4.2
8:   Play  $\pi^{\text{sim},*}$  for  $T/4$  episodes in real, compute average return  $\hat{V}_0^{\text{real}, \pi^{\text{sim},*}}$ 
9:   Play  $\pi^{\hat{f}}$  for  $T/4$  episodes in real, compute average return  $\hat{V}_0^{\text{real}, \pi^{\hat{f}}}$ 
10: return  $\hat{\pi} \leftarrow \arg \max_{\pi \in \{\pi^{\hat{f}}, \pi^{\text{sim},*}\}} \hat{V}_0^{\text{real}, \pi}$ 

```

Algorithm 1 first calls a subroutine LEARNEXPOLICIES, which learns a set of policies that provide rich data coverage on \mathcal{M}^{sim} —precisely, LEARNEXPOLICIES returns policies $\{\Pi_{\text{exp}}^h\}_{h \in [H]}$ which induce covariates with lower-bounded minimum eigenvalue on \mathcal{M}^{sim} and relies only on Oracle 4.2 (as well as knowledge of the featurization of \mathcal{M}^{sim} , ϕ^s) to find such policies. Algorithm 1 then plays these exploration policies in $\mathcal{M}^{\text{real}}$, coupled with random exploration, and applies the regression oracle to the data they collect. Finally, it estimates the value of the policy learned by the regression oracle and $\pi^{\text{sim},*}$, and returns whichever is best. We have the following.

Theorem 1. *If Assumptions 1 to 4 hold and*

$$\epsilon_{\text{sim}} \leq \frac{\lambda_{\min}^*}{64dHA^3}, \quad (4.1)$$

then as long as

$$T \geq c \cdot \frac{d^2 H^{16}}{\epsilon^8} \cdot \log \frac{H|\mathcal{F}|}{\delta},$$

with probability at least $1 - \delta$, Algorithm 1 returns a policy $\hat{\pi}$ such that $V_0^{\text{real},} - V_0^{\text{real},\hat{\pi}} \leq \epsilon$, and Oracles 4.1 and 4.2 are invoked at most $\text{poly}(d, H, \epsilon^{-1}, \log \frac{1}{\delta})$ times.*

Theorem 1 shows that, as long as ϵ_{sim} satisfies (4.1), utilizing a simulator and least-squares regression oracle, Oracles 4.1 and 4.2, allows for efficient learning in $\mathcal{M}^{\text{real}}$, achieving a complexity scaling polynomially in problem parameters. This yields an *exponential improvement* over learning without a simulator using naive exploration or direct sim2real transfer—which Propositions 1 and 3 show have complexity scaling exponentially in the horizon—despite utilizing the same practical computation oracles. To the best of our knowledge, this result provides the first theoretical evidence that sim2real transfer can yield provable gains in RL beyond trivial settings where direct transfer succeeds.

Note that the condition in (4.1) is independent of ϵ —unlike direct sim2real transfer, which requires $\epsilon = \mathcal{O}(\epsilon_{\text{sim}})$, we simply must assume ϵ_{sim} is small enough that (4.1) holds, and Theorem 1 shows that we can efficiently learn an ϵ -optimal policy in $\mathcal{M}^{\text{real}}$ for any $\epsilon > 0$. In Appendix B.4, we also present an extended version of Theorem 1, Theorem 3, which utilizes data from \mathcal{M}^{sim} to reduce the dependence on $\log |\mathcal{F}|$. In particular, instead of scaling with $\log |\mathcal{F}|$, it only scales with the log-cardinality of functions that are (approximately) Bellman-consistent on \mathcal{M}^{sim} . To illustrate the effectiveness of Theorem 1, we return to the instance of Propositions 1 and 3, where naive exploration and direct sim2real transfer fails. We have the following.

Proposition 4. *In the setting of Propositions 1 and 3 and assuming that $\epsilon_{\text{sim}} \leq \frac{1}{8192} \cdot \frac{1}{H}$, running Algorithm 1 will require $\text{poly}(H, \epsilon^{-1}) \cdot \log \frac{1}{\delta}$ samples from $\mathcal{M}^{\text{real}}$ in order to identify an ϵ -optimal policy in $\mathcal{M}^{\text{real}}$ with probability at least $1 - \delta$, for any $\epsilon > 0$.*

Note that the condition required by Proposition 4 is simply that $\epsilon_{\text{sim}} \lesssim 1/H$ —as long as our simulator satisfies this condition, we can efficiently transfer exploration policies to learn an ϵ -optimal policy, for any $\epsilon > 0$, while naive methods would be limited to only obtaining an $\Omega(1/H)$ -optimal policy.

Remark 4.1 (Necessity of Random Exploration). Algorithm 1 achieves efficient exploration in $\mathcal{M}^{\text{real}}$ by learning policies Π_{exp}^h in \mathcal{M}^{sim} that span the feature space of \mathcal{M}^{sim} (Line 2), and then playing these policies in $\mathcal{M}^{\text{real}}$, coupled with random exploration (Line 4). This use of random exploration is critical to obtaining Theorem 1. As we show in Proposition 5, if we omit the random exploration, Assumption 1 is not sufficient to guarantee Π_{exp}^h explores effectively in $\mathcal{M}^{\text{real}}$, even when (4.1) holds.

Remark 4.2 (Computational Efficiency). Algorithm 1, as well as its main subroutine LEARNEXPOLICIES, relies only on calls to Oracle 4.1 and Oracle 4.2. Thus, assuming we can efficiently implement these oracles, which is often the case in problem settings of interest, Algorithm 1 can be run in a computationally efficient manner.

5 Practical Algorithm and Experiments

We next validate the effectiveness of our proposal in practice: can a set of diverse exploration policies obtained from simulation improve the efficiency of real-world reinforcement learning? We start by showing that this holds for a simple, didactic, tabular environment in Section 5.2. From here, we consider several more realistic task domains: simulators inspired by real-world robotic manipulation tasks (sim2sim transfer, Section 5.3); and an actual real-world sim2real experiment on a Franka robotic platform (sim2real transfer, Section 5.4). Further details on all experiments, including additional baselines, can be found in Appendix E. Before stating our experimental results, we first provide a practical instantiation of Algorithm 1 that we can apply with real robotic systems and neural network function approximators.

5.1 Practical Instantiation of Exploration Policy Transfer

The key idea behind Algorithm 1 is quite simple: learn a set of exploratory policies in \mathcal{M}^{sim} —policies which provide rich data coverage in \mathcal{M}^{sim} —and transfer these policies to $\mathcal{M}^{\text{real}}$, coupled with random exploration, using the collected data to determine a near-optimal policy for $\mathcal{M}^{\text{real}}$. Algorithm 1 provides a particular instantiation of this principle, learning exploratory policies in \mathcal{M}^{sim} via the LEARNEXPOLICIES subroutine, which aims to cover the feature space of \mathcal{M}^{sim} , and utilizing a least-squares regression oracle to compute an optimal policy given the data collected in $\mathcal{M}^{\text{real}}$. In practice, however, other instantiations of this principle are possible by replacing LEARNEXPOLICIES with any procedure which generates exploratory policies in \mathcal{M}^{sim} , and replacing the regression oracle with any RL algorithm able to learn from off-policy data. We consider a general meta-algorithm instantiating this in Algorithm 2.

Algorithm 2 Practical sim2real Exploration Policy Transfer Meta-Algorithm

- 1: **Input:** Simulator \mathcal{M}^{sim} , real environment $\mathcal{M}^{\text{real}}$, simulator budget T_{sim} , real budget T , algorithm to generate exploratory policies in \mathcal{M}^{sim} $\mathfrak{A}_{\text{exp}}$, algorithm to solve policy optimization in real \mathfrak{A}_{po}
// Learn exploratory policies in \mathcal{M}^{sim}
 - 2: Run $\mathfrak{A}_{\text{exp}}$ for T_{sim} steps in \mathcal{M}^{sim} to generate set of exploratory policies Π_{exp}
// Deploy exploratory policies in $\mathcal{M}^{\text{real}}$
 - 3: **for** $t = 1, 2, \dots, T/2$ **do**
 - 4: Draw $\pi_{\text{exp}} \sim \text{unif}(\Pi_{\text{exp}})$, play in $\mathcal{M}^{\text{real}}$ for one episode, add data to replay buffer of \mathfrak{A}_{po}
 - 5: Run \mathfrak{A}_{po} for one episode // optional if \mathfrak{A}_{po} learns fully offline
-

In practice, $\mathfrak{A}_{\text{exp}}$ and \mathfrak{A}_{po} can be instantiated with a variety of algorithms. For example, we might take $\mathfrak{A}_{\text{exp}}$ to be an RND [7] or bootstrapped Q-learning-style [45, 31] algorithm, or any unsupervised RL procedure [48, 14, 32, 47], and \mathfrak{A}_{po} to be an off-policy policy optimization algorithm such as soft actor-critic (SAC) [16] or implicit Q-learning (IQL) [28].

For the following experiments, we instantiate Algorithm 2 by setting $\mathfrak{A}_{\text{exp}}$ to an algorithm inspired by recent work on inducing diverse behaviors in RL [14, 30], and \mathfrak{A}_{po} to SAC. In particular, $\mathfrak{A}_{\text{exp}}$ simultaneously trains an ensemble of policies $\Pi_{\text{exp}} = \{\pi_{\text{exp}}^i\}_{i=1}^n$ and a discriminator $d_\theta : \mathcal{S} \times [n] \rightarrow \mathbb{R}$, where d_θ is trained to discriminate between the behaviors of each policy π_{exp}^i , and π_{exp}^i

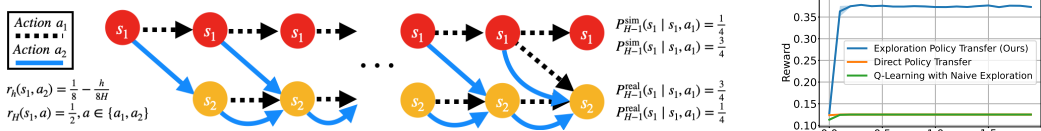


Figure 2: **Left:** Illustration of Combination Lock Example. **Right:** Results on Combination Lock.

is optimized on a weighting of the true task reward and the exploration reward induced by the discriminator, $r_e(s, i) := \log \frac{\exp(d_\theta(s, i))}{\sum_{j \in [n]} \exp(d_\theta(s, j))}$. As shown in existing work [14, 30], this simple training objective effectively induces diverse behavior with temporally correlated exploration while remaining within the vicinity of the optimal policy, using standard optimization techniques. Note that the particular choice of algorithm is less critical here than abiding by the recipes laid out in the meta-algorithm (Algorithm 2). The particular instantiation that we run for our experiments is detailed in Algorithm 6, along with further details in Appendix E.2.

5.2 Didactic Combination Lock Experiment

We first consider a variant of the construction used to prove Propositions 1 and 3, itself a variant of the classic combination lock instance. We illustrate this instance in Figure 2. Unless noted, all transitions occur with probability 1, and rewards are 0. Here, in \mathcal{M}^{sim} the optimal policy, $\pi^{\text{sim},*}$, plays action a_2 for all steps $h < H - 1$, while in $\mathcal{M}^{\text{real}}$, the optimal policy plays action a_1 at every step. Which policy is optimal is determined by the outgoing transition from s_1 at the $(H - 1)$ th step and, as such, to identify the optimal policy, any algorithm must reach s_1 at the $(H - 1)$ th step. As s_1 will only be reached at step $H - 1$ by playing a_1 for $H - 1$ consecutive times, any algorithm relying on naive exploration will take exponentially long to identify the optimal policy. Furthermore, playing $\pi^{\text{sim},*}$ coupled with random exploration will similarly take an exponential number of episodes, since $\pi^{\text{sim},*}$ always plays a_2 . As such, both direct sim2real policy transfer as well as Q -learning with naive exploration (Protocol 4.1) will fail to find the optimal policy in $\mathcal{M}^{\text{real}}$. However, if we transfer exploratory policies from \mathcal{M}^{sim} , since \mathcal{M}^{sim} and $\mathcal{M}^{\text{real}}$ behave identically up to step $H - 1$, these policies can efficiently traverse $\mathcal{M}^{\text{real}}$, reach s_1 at step $H - 1$, and identify the optimal policy. We compare our approach of exploration policy transfer to these baseline methods and illustrate the performance of each in Figure 2. As this is a simple tabular instance, we implement Algorithm 1 directly here. As Figure 2 shows, the intuition described above leads to real gains in practice—exploration policy transfer quickly identifies the optimal policy, while more naive approaches fail completely over the time horizon we considered.

5.3 Realistic Robotics sim2sim Experiment

To test the ability of our proposed method to scale to more complex problems, we next experiment on a sim2sim transfer setting with a realistic robotic simulator. We consider TychoEnv, a simulator of the 7DOF Tycho robotics platform introduced by [73], and shown in Figure 3. We test sim2sim transfer on a reaching task where the goal is to touch a small ball hanging in the air with the tip of the chopstick end effector. The agent perceives the ball and its own end effector pose and outputs a delta in its desired end effector pose as a command. We set \mathcal{M}^{sim} and $\mathcal{M}^{\text{real}}$ to be two instances of TychoEnv with slightly different parameters to model real-world sim2real transfer. Precisely, we change the action bounds and control frequency from \mathcal{M}^{sim} to $\mathcal{M}^{\text{real}}$.

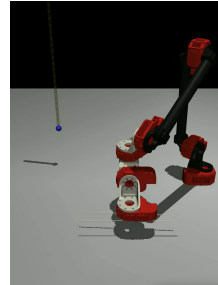


Figure 3: TychoEnv Reach Task Setup

We aim to compare our approach of exploration policy transfer with direct sim2real policy transfer. To this end, we first train a policy in \mathcal{M}^{sim} that solves the task in \mathcal{M}^{sim} , $\pi^{\text{sim},*}$, and then utilize this policy in place of Π_{exp} in Algorithm 2. We instantiate our approach of exploration policy transfer as outlined above. Our aim in this experiment is to illustrate how the quality of the data provided by direct policy transfer vs. exploration policy transfer affects learning. As such, for both approaches we simply initialize our SAC agent in $\mathcal{M}^{\text{real}}$, \mathfrak{A}_{po} , from scratch, and set the reward in $\mathcal{M}^{\text{real}}$ to be sparse: the agent only receives a non-zero reward if it successfully touches the ball. For each approach, we repeat the process of training in \mathcal{M}^{sim} four times, and for each of these run them for two trials in $\mathcal{M}^{\text{real}}$.

We illustrate our results in Figure 4. As this figure illustrates, direct policy transfer fails to learn completely, while exploration policy transfer successfully solves the task. Investigating the behavior of each method, we find that the policies transferred via exploration policy transfer, while failing to solve the task with perfect accuracy, when coupled with naive exploration are able to successfully make contact with the ball on occasion. This provides sufficiently rich data for SAC to ultimately learn to solve the task. In contrast, direct policy transfer fails to collect any reward when run in $\mathcal{M}^{\text{real}}$, and, given the sparse reward nature of the task, SAC is unable to locate any reward and learn. We include an additional sim2sim experiment on the Franka Emika Panda Robot Arm in Appendix E.4.

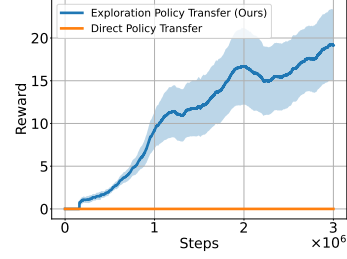


Figure 4: Results on sim2sim Transfer in TychoEnv Simulator

5.4 Real-World Robotic sim2real Experiment

Finally, we demonstrate our algorithm for actual sim2real policy transfer for a manipulation task on a real-world Franka Emika Panda robot arm [17] with a parallel gripper. Our task is to push a 75mm diameter cylindrical “puck” from the center to the edge of the surface, as shown in Figure 1, with the arm initialized at random locations. The observed state $s = [\mathbf{p}_{\text{ee}}, \mathbf{p}_{\text{obj}}] \in \mathbb{R}^4$ consists of the planar Cartesian coordinate of the end effector \mathbf{p}_{ee} along with the center of mass of the puck \mathbf{p}_{obj} . Our policy outputs planar end effector position deltas $a = \Delta \mathbf{p}_{\text{ee}} \in \mathbb{R}^2$, evaluated at 8 Hz, which are passed into a lower-level joint position PID controller. We use an Intel Realsense D435 depth camera to track the location of the puck. Our reward function is a sum of a success indicator (indicating when the puck has been pushed to the edge of the surface) and terms which give negative reward if the distance from the end effector to the puck, or puck to the goal, are too large (see (E.1)); in particular, a reward greater than 0 indicates success.

We run the instantiation of Algorithm 2 outlined above. In particular, we train an ensemble of $n = 15$ exploration policies, training for 20 million steps in \mathcal{M}^{sim} . In addition, we train a policy that solves the task in \mathcal{M}^{sim} , $\pi^{\text{sim},*}$. We use a custom simulator of the arm, where during training the friction of the table is randomized and noise is added to the observations.

We observe a substantial sim2real gap between our simulator and the real robot, with policies trained in simulation failing to complete the pushing task zero shot in real, even when trained with domain randomization. We compare direct sim2real policy transfer against our method of transferring exploration policies. For direct policy transfer, we simply run SAC to finetune $\pi^{\text{sim},*}$ in the real world, using the current policy to collect data. For exploration policy transfer, we instead utilize Π_{exp} , our ensemble of exploration policies, to collect data in the real world. We run this in tandem with an SAC agent, feeding the data from the exploration policies into the SAC agent’s replay buffer. Unlike in Section 5.3, rather than initializing the SAC policy from scratch, we set the initial policy as $\pi^{\text{sim},*}$, and fine-tune from this on the data collected from playing Π_{exp} . See Appendix E.5 for additional details.

Our results are shown on the right side of Figure 1. Statistics are computed over 6 runs for each method. Direct policy transfer with finetuning is unable to solve the task in real in each of the 6 runs, and converges to a suboptimal solution. However, our method is able to solve the task successfully each time and achieve a substantially higher reward.

6 Discussion

In this work, we have demonstrated that simulators can make naive exploration efficient even in settings where direct sim2real transfer fails, if they are used to train a set of exploration policies. We highlight several limitations of this work, which we believe are interesting future research questions:

- Our focus is purely on dynamics shift—where the dynamics of sim and real differ, but the environments are otherwise the same. While dynamics shift is common in many scenarios, other types of shift can exist as well, for example perceptual shift. How can we best handle these types of shift?
- How can we utilize a simulator in sim2real transfer if we can reset it arbitrarily, rather than just allowing for black-box access? Does the ability to reset allow us to improve sample efficiency further?
- Is the reachability condition, Assumption 3, necessary for successful exploration transfer?

Acknowledgements

The work of AW and KJ was partially supported by the NSF through the University of Washington Materials Research Science and Engineering Center, DMR-2308979, and awards CCF 2007036 and CAREER 2141511. The work of LK was partially supported by Toyota Research Institute URP.

References

- [1] 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.
- [2] 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.
- [3] Naman Agarwal, Syomantak Chaudhuri, Prateek Jain, Dheeraj Nagaraj, and Praneeth Netrapalli. Online target q-learning with reverse experience replay: Efficiently finding the optimal policy for linear mdps. *arXiv preprint arXiv:2110.08440*, 2021.
- [4] Ilge Akkaya, Marcin Andrychowicz, Maciek Chociej, Mateusz Litwin, Bob McGrew, Arthur Petron, Alex Paino, Matthias Plappert, Glenn Powell, Raphael Ribas, et al. Solving rubik’s cube with a robot hand. *arXiv preprint arXiv:1910.07113*, 2019.
- [5] Philip Amortila, Nan Jiang, Dhruv Madeka, and Dean P Foster. A few expert queries suffices for sample-efficient rl with resets and linear value approximation. *Advances in Neural Information Processing Systems*, 35:29637–29648, 2022.
- [6] Ronen I Brafman and Moshe Tennenholtz. R-max-a general polynomial time algorithm for near-optimal reinforcement learning. *Journal of Machine Learning Research*, 3(Oct):213–231, 2002.
- [7] Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. Exploration by random network distillation. *arXiv preprint arXiv:1810.12894*, 2018.
- [8] Yevgen Chebotar, Ankur Handa, Viktor Makoviychuk, Miles Macklin, Jan Issac, Nathan Ratliff, and Dieter Fox. Closing the sim-to-real loop: Adapting simulation randomization with real world experience. In *ICRA*, 2019.
- [9] Tao Chen, Megha Tippur, Siyang Wu, Vikash Kumar, Edward Adelson, and Pulkit Agrawal. Visual dexterity: In-hand reorientation of novel and complex object shapes. *Science Robotics*, 8(84):eadc9244, 2023.
- [10] 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.
- [11] Chris Dann, Yishay Mansour, Mehryar Mohri, Ayush Sekhari, and Karthik Sridharan. Guarantees for epsilon-greedy reinforcement learning with function approximation. In *International conference on machine learning*, pages 4666–4689. PMLR, 2022.
- [12] Jonas Degraeve, Federico Felici, Jonas Buchli, Michael Neunert, Brendan Tracey, Francesco Carpanese, Timo Ewalds, Roland Hafner, Abbas Abdolmaleki, Diego de Las Casas, et al. Magnetic control of tokamak plasmas through deep reinforcement learning. *Nature*, 602(7897):414–419, 2022.
- [13] Benjamin Eysenbach and Sergey Levine. Maximum entropy rl (provably) solves some robust rl problems. *arXiv preprint arXiv:2103.06257*, 2021.
- [14] Benjamin Eysenbach, Abhishek Gupta, Julian Ibarz, and Sergey Levine. Diversity is all you need: Learning skills without a reward function. *arXiv preprint arXiv:1802.06070*, 2018.
- [15] Raj Ghugare, Santiago Miret, Adriana Hugessen, Mariano Phielipp, and Glen Berseth. Searching for high-value molecules using reinforcement learning and transformers. *arXiv preprint arXiv:2310.02902*, 2023.
- [16] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pages 1861–1870. PMLR, 2018.

- [17] Sami Haddadin, Sven Parusel, Lars Johannsmeier, Saskia Golz, Simon Gabl, Florian Walch, Mohamadreza Sabaghian, Christoph Jähne, Lukas Hausperger, and Simon Haddadin. The franka emika robot: A reference platform for robotics research and education. *IEEE Robotics & Automation Magazine*, 29(2):46–64, 2022. doi: 10.1109/MRA.2021.3138382.
- [18] Sebastian Höfer, Kostas Bekris, Ankur Handa, Juan Camilo Gamboa, Melissa Mozifian, Florian Golemo, Chris Atkeson, Dieter Fox, Ken Goldberg, John Leonard, et al. Sim2real in robotics and automation: Applications and challenges. *IEEE transactions on automation science and engineering*, 18(2):398–400, 2021.
- [19] Stephen James, Paul Wohlhart, Mrinal Kalakrishnan, Dmitry Kalashnikov, Alex Irpan, Julian Ibarz, Sergey Levine, Raia Hadsell, and Konstantinos Bousmalis. Sim-to-real via sim-to-sim: Data-efficient robotic grasping via randomized-to-canonical adaptation networks. arxiv e-prints, page. *arXiv preprint arXiv:1812.07252*, 2018.
- [20] Yiding Jiang, J Zico Kolter, and Roberta Raileanu. On the importance of exploration for generalization in reinforcement learning. *arXiv preprint arXiv:2306.05483*, 2023.
- [21] Chi Jin, Zhuoran Yang, Zhaoran Wang, and Michael I Jordan. Provably efficient reinforcement learning with linear function approximation. In *Conference on Learning Theory*, pages 2137–2143. PMLR, 2020.
- [22] Sham Kakade, Michael J Kearns, and John Langford. Exploration in metric state spaces. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pages 306–312, 2003.
- [23] Elia Kaufmann, Leonard Bauersfeld, Antonio Loquercio, Matthias Müller, Vladlen Koltun, and Davide Scaramuzza. Champion-level drone racing using deep reinforcement learning. *Nature*, 620(7976):982–987, 2023.
- [24] Michael Kearns and Daphne Koller. Efficient reinforcement learning in factored mdps. In *IJCAI*, volume 16, pages 740–747, 1999.
- [25] Michael Kearns and Satinder Singh. Near-optimal reinforcement learning in polynomial time. *Machine learning*, 49:209–232, 2002.
- [26] B Ravi Kiran, Ibrahim Sobh, Victor Talpaert, Patrick Mannion, Ahmad A Al Sallab, Senthil Yogamani, and Patrick Pérez. Deep reinforcement learning for autonomous driving: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 23(6):4909–4926, 2021.
- [27] Jens Kober, J Andrew Bagnell, and Jan Peters. Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research*, 32(11):1238–1274, 2013.
- [28] Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit q-learning. *arXiv preprint arXiv:2110.06169*, 2021.
- [29] Ashish Kumar, Zipeng Fu, Deepak Pathak, and Jitendra Malik. Rma: Rapid motor adaptation for legged robots. *arXiv preprint arXiv:2107.04034*, 2021.
- [30] Saurabh Kumar, Aviral Kumar, Sergey Levine, and Chelsea Finn. One solution is not all you need: Few-shot extrapolation via structured maxent rl. *Advances in Neural Information Processing Systems*, 33:8198–8210, 2020.
- [31] Kimin Lee, Michael Laskin, Aravind Srinivas, and Pieter Abbeel. Sunrise: A simple unified framework for ensemble learning in deep reinforcement learning. In *International Conference on Machine Learning*, pages 6131–6141. PMLR, 2021.
- [32] Lisa Lee, Benjamin Eysenbach, Emilio Parisotto, Eric Xing, Sergey Levine, and Ruslan Salakhutdinov. Efficient exploration via state marginal matching. *arXiv preprint arXiv:1906.05274*, 2019.
- [33] Gen Li, Yuxin Chen, Yuejie Chi, Yuantao Gu, and Yuting Wei. Sample-efficient reinforcement learning is feasible for linearly realizable mdps with limited revisiting. *Advances in Neural Information Processing Systems*, 34:16671–16685, 2021.
- [34] Yao Liu, Dipendra Misra, Miro Dudík, and Robert E Schapire. Provably sample-efficient rl with side information about latent dynamics. *Advances in Neural Information Processing Systems*, 35:33482–33493, 2022.
- [35] Rui Lu, Gao Huang, and Simon S Du. On the power of multitask representation learning in linear mdp. *arXiv preprint arXiv:2106.08053*, 2021.

- [36] Dhruv Malik, Yuanzhi Li, and Pradeep Ravikumar. When is generalizable reinforcement learning tractable? *Advances in Neural Information Processing Systems*, 34:8032–8045, 2021.
- [37] Timothy A Mann and Yoonsuck Choe. Directed exploration in reinforcement learning with transferred knowledge. In *European Workshop on Reinforcement Learning*, pages 59–76. PMLR, 2013.
- [38] Gabriel B Margolis, Xiang Fu, Yandong Ji, and Pulkit Agrawal. Learning physically grounded robot vision with active sensing motor policies. In *CoRL*, 2023.
- [39] Bhairav Mehta, Manfred Diaz, Florian Golemo, Christopher J Pal, and Liam Paull. Active domain randomization. In *CoRL*, 2020.
- [40] Marius Memmel, Andrew Wagenmaker, Chuning Zhu, Patrick Yin, Dieter Fox, and Abhishek Gupta. Asid: Active exploration for system identification in robotic manipulation. *arXiv preprint arXiv:2404.12308*, 2024.
- [41] Zak Mhammedi, Adam Block, Dylan J Foster, and Alexander Rakhlin. Efficient model-free exploration in low-rank mdps. *Advances in Neural Information Processing Systems*, 36, 2024.
- [42] Zakaria Mhammedi, Dylan J Foster, and Alexander Rakhlin. The power of resets in online reinforcement learning. *arXiv preprint arXiv:2404.15417*, 2024.
- [43] Aditya Modi, Jinglin Chen, Akshay Krishnamurthy, Nan Jiang, and Alekh Agarwal. Model-free representation learning and exploration in low-rank mdps. *Journal of Machine Learning Research*, 25(6):1–76, 2024.
- [44] Fabio Muratore, Michael Gienger, and Jan Peters. Assessing transferability from simulation to reality for reinforcement learning. *IEEE transactions on pattern analysis and machine intelligence*, 2019.
- [45] Ian Osband, Charles Blundell, Alexander Pritzel, and Benjamin Van Roy. Deep exploration via bootstrapped dqn. *Advances in neural information processing systems*, 29, 2016.
- [46] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.
- [47] Seohong Park, Oleh Rybkin, and Sergey Levine. Metra: Scalable unsupervised rl with metric-aware abstraction. *arXiv preprint arXiv:2310.08887*, 2023.
- [48] Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *International conference on machine learning*, pages 2778–2787. PMLR, 2017.
- [49] Xue Bin Peng, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. Sim-to-real transfer of robotic control with dynamics randomization. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 3803–3810. IEEE, 2018.
- [50] Xue Bin Peng, Erwin Coumans, Tingnan Zhang, Tsang-Wei Lee, Jie Tan, and Sergey Levine. Learning agile robotic locomotion skills by imitating animals. *arXiv preprint arXiv:2004.00784*, 2020.
- [51] Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22(268):1–8, 2021.
- [52] Aravind Rajeswaran, Vikash Kumar, Abhishek Gupta, Giulia Vezzani, John Schulman, Emanuel Todorov, and Sergey Levine. Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. *arXiv preprint arXiv:1709.10087*, 2017.
- [53] FL Silva, Jiachen Yang, Mikel Landajuela, Andre Goncalves, Alexander Ladd, Daniel Faissol, and Brenden Petersen. Toward multi-fidelity reinforcement learning for symbolic optimization. Technical report, Lawrence Livermore National Laboratory (LLNL), Livermore, CA (United States), 2023.
- [54] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

- [55] Rohan Sinha, James Harrison, Spencer M Richards, and Marco Pavone. Adaptive robust model predictive control with matched and unmatched uncertainty. In *2022 American Control Conference (ACC)*, 2022.
- [56] Yuda Song, Aditi Mavalankar, Wen Sun, and Sicun Gao. Provably efficient model-based policy adaptation. *arXiv preprint arXiv:2006.08051*, 2020.
- [57] Yuda Song, Yifei Zhou, Ayush Sekhari, J Andrew Bagnell, Akshay Krishnamurthy, and Wen Sun. Hybrid rl: Using both offline and online data can make rl efficient. *arXiv preprint arXiv:2210.06718*, 2022.
- [58] Yanchao Sun, Ruijie Zheng, Xiyao Wang, Andrew Cohen, and Furong Huang. Transfer rl across observation feature spaces via model-based regularization. *arXiv preprint arXiv:2201.00248*, 2022.
- [59] Andrea Tirinzoni, Matteo Pirota, and Alessandro Lazaric. A fully problem-dependent regret lower bound for finite-horizon mdps. *arXiv preprint arXiv:2106.13013*, 2021.
- [60] Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 23–30. IEEE, 2017.
- [61] 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, 2012. doi: 10.1109/IROS.2012.6386109.
- [62] Masatoshi Uehara, Xuezhou Zhang, and Wen Sun. Representation learning for online and offline rl in low-rank mdps. *arXiv preprint arXiv:2110.04652*, 2021.
- [63] Andrew Wagenmaker and Kevin G Jamieson. Instance-dependent near-optimal policy identification in linear mdps via online experiment design. *Advances in Neural Information Processing Systems*, 35:5968–5981, 2022.
- [64] Andrew Wagenmaker, Guanya Shi, and Kevin Jamieson. Optimal exploration for model-based rl in nonlinear systems. *arXiv preprint arXiv:2306.09210*, 2023.
- [65] Andrew J Wagenmaker, Yifang Chen, Max Simchowitz, Simon Du, and Kevin Jamieson. Reward-free rl is no harder than reward-aware rl in linear markov decision processes. In *International Conference on Machine Learning*, pages 22430–22456. PMLR, 2022.
- [66] Jane X Wang, Zeb Kurth-Nelson, Dhruva Tirumala, Hubert Soyer, Joel Z Leibo, Remi Munos, Charles Blundell, Dharshan Kumaran, and Matt Botvinick. Learning to reinforcement learn. *arXiv preprint arXiv:1611.05763*, 2016.
- [67] Gellert Weisz, Philip Amortila, Barnabás Janzer, Yasin Abbasi-Yadkori, Nan Jiang, and Csaba Szepesvári. On query-efficient planning in mdps under linear realizability of the optimal state-value function. In *Conference on Learning Theory*, pages 4355–4385. PMLR, 2021.
- [68] Gellert Weisz, András György, Tadashi Kozuno, and Csaba Szepesvári. Confident approximate policy iteration for efficient local planning in q^π -realizable mdps. *Advances in Neural Information Processing Systems*, 35:25547–25559, 2022.
- [69] Haotian Ye, Xiaoyu Chen, Liwei Wang, and Simon Shaolei Du. On the power of pre-training for generalization in rl: provable benefits and hardness. In *International Conference on Machine Learning*, pages 39770–39800. PMLR, 2023.
- [70] Dong Yin, Botao Hao, Yasin Abbasi-Yadkori, Nevena Lazić, and Csaba Szepesvári. Efficient local planning with linear function approximation. In *International Conference on Algorithmic Learning Theory*, pages 1165–1192. PMLR, 2022.
- [71] Andrea Zanette, Alessandro Lazaric, Mykel J Kochenderfer, and Emma Brunskill. Provably efficient reward-agnostic navigation with linear value iteration. *Advances in Neural Information Processing Systems*, 33:11756–11766, 2020.
- [72] Tianjun Zhang, Tongzheng Ren, Mengjiao Yang, Joseph Gonzalez, Dale Schuurmans, and Bo Dai. Making linear mdps practical via contrastive representation learning. In *International Conference on Machine Learning*, pages 26447–26466. PMLR, 2022.
- [73] Yunchu Zhang, Liyiming Ke, Abhay Deshpande, Abhishek Gupta, and Siddhartha Srinivasa. Cherry-picking with reinforcement learning. *arXiv preprint arXiv:2303.05508*, 15, 2023.

- [74] Wenshuai Zhao, Jorge Peña Queralta, and Tomi Westerlund. Sim-to-real transfer in deep reinforcement learning for robotics: a survey. In *2020 IEEE symposium series on computational intelligence (SSCI)*, pages 737–744. IEEE, 2020.

A Technical Results

We denote the state-visitations for some policy π as $w_h^\pi(s, a) := \mathbb{P}^\pi[(s_h, a_h) = (s, a)]$, $w_h^\pi(\mathcal{Z}) := \mathbb{P}^\pi[(s_h, a_h) \in \mathcal{Z}]$, for $\mathcal{Z} \subseteq \mathcal{S} \times \mathcal{A}$. For $\mathcal{X} \subseteq \mathbb{R}^d$, we denote $w_h^\pi(\mathcal{X}) := \mathbb{P}^\pi[\phi(s_h, a_h) \in \mathcal{X}]$, for ϕ the featurization of the environment.

Lemma A.1. *Consider MDPs M and \widetilde{M} with transition kernels P and \widetilde{P} . Assume that both M and \widetilde{M} start in the same state s_0 and that, for each (s, a, h) :*

$$\|P_h(\cdot | s, a) - \widetilde{P}_h(\cdot | s, a)\|_{\text{TV}} \leq \epsilon_{\text{sim}}. \quad (\text{A.1})$$

Consider some reward function r such that $\sum_{h=1}^H r_h(s_h, a_h) \leq R$ for all possible sequences $\{(s_h, a_h)\}_{h=1}^H$. Then it follows that, for any π and (s, a, h) ,

$$|Q_h^{M,\pi}(s, a) - Q_h^{\widetilde{M},\pi}(s, a)| \leq HR \cdot \epsilon_{\text{sim}}.$$

Proof. We prove this by induction. First, assume that for some h and all s, a , we have $|Q_{h+1}^{M,\pi}(s, a) - Q_{h+1}^{\widetilde{M},\pi}(s, a)| \leq \epsilon_{h+1}$. By definition we have

$$Q_h^{M,\pi}(s, a) = r_h(s, a) + \mathbb{E}^{M,\pi}[Q_{h+1}^{M,\pi}(s_{h+1}, a_{h+1}) | s_h = s, a_h = a]$$

and similarly for $Q_{h+1}^{\widetilde{M},\pi}(s, a)$. Thus:

$$\begin{aligned} |Q_h^{M,\pi}(s, a) - Q_h^{\widetilde{M},\pi}(s, a)| &\stackrel{(a)}{\leq} |\mathbb{E}^{M,\pi}[Q_{h+1}^{M,\pi}(s_{h+1}, a_{h+1}) | s_h = s, a_h = a] - \mathbb{E}^{\widetilde{M},\pi}[Q_{h+1}^{M,\pi}(s_{h+1}, a_{h+1}) | s_h = s, a_h = a]| \\ &\quad + |\mathbb{E}^{\widetilde{M},\pi}[Q_{h+1}^{M,\pi}(s_{h+1}, a_{h+1}) - Q_{h+1}^{\widetilde{M},\pi}(s_{h+1}, a_{h+1}) | s_h = s, a_h = a]| \\ &\stackrel{(b)}{\leq} |\mathbb{E}^{M,\pi}[Q_{h+1}^{M,\pi}(s_{h+1}, a_{h+1}) | s_h = s, a_h = a] - \mathbb{E}^{\widetilde{M},\pi}[Q_{h+1}^{M,\pi}(s_{h+1}, a_{h+1}) | s_h = s, a_h = a]| + \epsilon_{h+1} \end{aligned}$$

where (a) follows from the triangle inequality and (b) follows from the inductive hypothesis. Under (A.1), we can bound

$$|\mathbb{E}^{M,\pi}[Q_{h+1}^{M,\pi}(s_{h+1}, a_{h+1}) | s_h = s, a_h = a] - \mathbb{E}^{\widetilde{M},\pi}[Q_{h+1}^{M,\pi}(s_{h+1}, a_{h+1}) | s_h = s, a_h = a]| \leq \epsilon_{\text{sim}} \cdot R.$$

It follows that for any (s, a) , $|Q_h^{M,\pi}(s, a) - Q_h^{\widetilde{M},\pi}(s, a)| \leq \epsilon_h =: \epsilon_{\text{sim}} R + \epsilon_{h+1}$.

The base case follows trivially with $\epsilon_H = 0$ since for any MDP we have that $Q_H^{M,\pi}(s, a) = r_H(s, a) = Q_H^{\widetilde{M},\pi}(s, a)$. \square

Lemma A.2. *Under the same setting as Lemma A.1 and for any h, π , and $\mathcal{Z} \subseteq \mathcal{S} \times \mathcal{A}$, we have*

$$|w_h^{M,\pi}(\mathcal{Z}) - w_h^{\widetilde{M},\pi}(\mathcal{Z})| \leq H \epsilon_{\text{sim}}.$$

Proof. This is an immediate consequence of Lemma A.1 since, setting the reward $r_{h'}(s, a) = \mathbb{I}\{(s, a) \in \mathcal{Z}, h' = h\}$, we can set $R = 1$ and have $V_0^{M,\pi} = w_h^{M,\pi}(\mathcal{Z})$. \square

Lemma A.3 (Proposition 2). *Under Assumption 1, we have that*

$$V_0^{\text{real},*} - V_0^{\text{real},\pi^{\text{sim},*}} \leq 2H^2 \epsilon_{\text{sim}} \quad \text{and} \quad V_0^{\text{sim},*} - V_0^{\text{sim},\pi^{\text{real},*}} \leq 2H^2 \epsilon_{\text{sim}}.$$

Proof. We prove the result for real—the result for sim follows analogously. We have

$$\begin{aligned} V_0^{\text{real},*} - V_0^{\text{real},\pi^{\text{sim},*}} &= V_0^{\text{real},\pi^{\text{real},*}} - V_0^{\text{sim},\pi^{\text{real},*}} + \underbrace{V_0^{\text{sim},\pi^{\text{real},*}} - V_0^{\text{sim},\pi^{\text{sim},*}}}_{\leq 0} + V_0^{\text{sim},\pi^{\text{sim},*}} - V_0^{\text{real},\pi^{\text{sim},*}} \\ &\leq |V_0^{\text{real},\pi^{\text{real},*}} - V_0^{\text{sim},\pi^{\text{real},*}}| + |V_0^{\text{sim},\pi^{\text{sim},*}} - V_0^{\text{real},\pi^{\text{sim},*}}|. \end{aligned}$$

The result then follows by applying Lemma A.1 to bound each of these terms by $H^2 \epsilon_{\text{sim}}$. \square

Lemma A.4. For any $f \in \mathcal{F}$,

$$V_0^* - V_0^{\pi^f} \leq \max_{\pi \in \{\pi^f, \pi^*\}} \sum_{h=0}^{H-1} 2 |\mathbb{E}^\pi[f_h(s_h, a_h) - \mathcal{T}f_{h+1}(s_h, a_h)]|.$$

Proof. We write

$$V_0^* - V_0^{\pi^f} = \underbrace{V_0^* - \max_a f_0(s_0, a)}_{(a)} + \underbrace{\max_a f_0(s_0, a) - V_0^{\pi^f}}_{(b)}$$

and then bound each of these terms separately. By Lemma 5 of [57] we have

$$\begin{aligned} (a) &\leq \sum_{h=0}^H \left| \mathbb{E}^{\pi^*}[f_h(s_h, a_h) - r_h - \max_{a'} f_{h+1}(s_{h+1}, a')] \right| \\ &= \sum_{h=0}^{H-1} \left| \mathbb{E}^{\pi^*}[f_h(s_h, a_h) - \mathbb{E}[r_h + \max_{a'} f_{h+1}(s_{h+1}, a') \mid s_h, a_h]] \right|. \end{aligned}$$

Similarly, by Lemma 4 of [57] we have

$$\begin{aligned} (b) &\leq \sum_{h=0}^{H-1} \left| \mathbb{E}^{\pi^f}[f_h(s_h, a_h) - r_h - \max_{a'} f_{h+1}(s_{h+1}, a')] \right| \\ &= \sum_{h=0}^{H-1} \left| \mathbb{E}^{\pi^f}[f_h(s_h, a_h) - \mathbb{E}[r_h + \max_{a'} f_{h+1}(s_{h+1}, a') \mid s_h, a_h]] \right|. \end{aligned}$$

□

B Proof of Main Results

In Appendix B.1 we first provide a general result on learning in real when collecting data via a fixed set of exploration policies, given a particular coverage assumption. Then in Appendix B.2, we show that by playing a set of policies which induce full-rank covariates in sim, these policies provide sufficient coverage for learning in real. Finally in Appendices B.3 and B.4, we use these results to prove Theorems 1 and 3. Throughout the appendix we develop the supporting lemmas for our more general result, Theorem 3, which utilizes the simulator to restrict the version space (i.e. the dependence on $|\mathcal{F}|$) in addition to utilizing the simulator to aid in exploration.

Throughout this and the following section we assume that Assumption 4 holds. We also assume that $f_h \in [0, V_{\max}]$ instead of $f_h \in [0, H]$, for some $V_{\max} > 0$. For any $f \in \mathcal{F}$, we denote the Bellman residual as

$$\mathcal{E}_h(f)(s, a) := \mathcal{T}f_{h+1}(s, a) - f_h(s, a).$$

Note that by assumption on \mathcal{F} , we have $\mathcal{E}_h(f)(s, a) \in [-V_{\max}, V_{\max}]$.

For any policy π , we denote $\Lambda_{\pi, h}^s := \mathbb{E}^{\text{sim}, \pi}[\phi^s(s_h, a_h)\phi^s(s_h, a_h)^\top]$ and $\Lambda_{\pi, h}^r := \mathbb{E}^{\text{real}, \pi}[\phi^r(s_h, a_h)\phi^r(s_h, a_h)^\top]$.

Necessity of Random Exploration. Algorithm 1 achieves efficient exploration in $\mathcal{M}^{\text{real}}$ by first learning a set of policies Π_{exp}^h in \mathcal{M}^{sim} that span the feature space of \mathcal{M}^{sim} (Line 2), achieving

$$\lambda_{\min} \left(\frac{1}{|\Pi_{\text{exp}}^h|} \sum_{\pi \in \Pi_{\text{exp}}^h} \mathbb{E}^{\mathcal{M}^{\text{sim}}, \pi}[\phi^s(s_h, a_h)\phi^s(s_h, a_h)^\top] \right) \gtrsim \lambda_{\min}^*, \quad (\text{B.1})$$

and then playing these policies in $\mathcal{M}^{\text{real}}$, coupled with random exploration (Line 4). In particular, Algorithm 1 plays policies from $\tilde{\Pi}_{\text{exp}}^h$, where each $\tilde{\pi}_{\text{exp}} \in \tilde{\Pi}_{\text{exp}}^h$ is defined as the policy which plays some $\pi_{\text{exp}} \in \Pi_{\text{exp}}^h$ up to step h , and then for steps $h' = h + 1, \dots, H$ chooses actions uniformly at random. This use of random exploration is critical to obtaining Theorem 1. Indeed, under our

transfer model, condition (4.1) of Theorem 1 is not strong enough to ensure that policies satisfying (B.1) collect rich enough data in $\mathcal{M}^{\text{real}}$ to allow for learning a near-optimal policy. While (4.1) is sufficient to guarantee that playing Π_{exp}^h on $\mathcal{M}^{\text{real}}$ collects data which spans the feature space of \mathcal{M}^{sim} —that is, satisfying (B.1) but with the expectation over \mathcal{M}^{sim} replaced by an expectation of $\mathcal{M}^{\text{real}}$ —this is insufficient for learning, as the following result shows.

Proposition 5. *For any $\epsilon_{\text{sim}} \leq 1/2$, there exist some \mathcal{M}^{sim} , $\mathcal{M}^{\text{real},1}$, and $\mathcal{M}^{\text{real},2}$ such that:*

1. *Both $\mathcal{M}^{\text{real},1}$ and $\mathcal{M}^{\text{real},2}$ satisfy Assumption 1 with \mathcal{M}^{sim} and Assumptions 2 to 4 hold.*
2. *There exists some policy π_{exp} such that $\lambda_{\min}(\mathbb{E}^{\mathcal{M}^{\text{sim}}, \pi_{\text{exp}}}[\phi^s(s_h, a_h)\phi^s(s_h, a_h)^\top]) = 1/2$, $\forall h \in [H]$, and for any $T \geq 0$, if we play π_{exp} on $\mathcal{M}^{\text{real}}$ for T steps, we have:*

$$\inf_{\hat{\pi}} \sup_{\mathcal{M}^{\text{real}} \in \{\mathcal{M}^{\text{real},1}, \mathcal{M}^{\text{real},2}\}} \mathbb{E}^{\mathcal{M}^{\text{real}}, \pi_{\text{exp}}} [V_0^{\mathcal{M}^{\text{real}}, \star} - V_0^{\mathcal{M}^{\text{real}}, \hat{\pi}}] \geq \epsilon_{\text{sim}}.$$

Proposition 5 holds because two MDPs may be “close” in the sense of Assumption 1 but admit very different feature representations. As a result, transferring a policy that covers the feature space of \mathcal{M}^{sim} is not necessarily sufficient for covering the feature space of $\mathcal{M}^{\text{real}}$, which ultimately means that data collected from π_{exp} is unable to identify the optimal policy in $\mathcal{M}^{\text{real}}$. Our key technical result, Lemma B.4, shows, however, that under Assumption 1 and (4.1), policies which achieve high coverage in \mathcal{M}^{sim} (i.e. satisfy (B.1)) are able to reach within a *logarithmic* number of steps of relevant states in $\mathcal{M}^{\text{real}}$. While the sample complexity of random exploration typically scales exponentially in the horizon, if the horizon over which we must explore is only logarithmic, the total complexity is then only polynomial. Theorem 1 critically relies on these facts—by playing policies in Π_{exp}^h up to step h and then exploring randomly, and repeating this for each $h \in [H]$, we show that sufficiently rich data is collected in $\mathcal{M}^{\text{real}}$ for learning an ϵ -optimal policy.

B.1 Learning in real with Fixed Exploration Policies

Algorithm 3 sim2real transfer with fixed exploration policies (EXPLOREREAL)

- 1: **input:** exploration policies $\{\pi_{\text{exp}}^h\}_{h=1}^H$, budget T , sim date \mathcal{D}_{sim} , sim regularization γ
- 2: Play $\pi_{\text{exp}} = \text{unif}(\{\pi_{\text{exp}}^h\}_{h=1}^H)$ for T episodes in real, add data to \mathcal{D}
- 3: **for** $h = H, H-1, \dots, 1$ **do**
- 4:

$$\begin{aligned} \tilde{f}_h &\leftarrow \arg \min_{f \in \mathcal{F}} \sum_{(s,a,r,s') \in \mathcal{D}_{\text{sim}}^h} (f_h(s,a) - r - \max_{a'} \hat{f}_{h+1}(s',a'))^2 \\ \hat{f}_h &\leftarrow \arg \min_{f \in \mathcal{F}} \sum_{(s,a,r,s') \in \mathcal{D}_h} (f_h(s,a) - r - \max_{a'} \hat{f}_{h+1}(s',a'))^2 \\ \text{s.t. } &\frac{1}{|\mathcal{D}_{\text{sim}}|} \sum_{(s,a) \in \mathcal{D}_{\text{sim}}^h} (f_h(s,a) - \tilde{f}_h(s,a))^2 \leq \gamma \end{aligned} \tag{B.2}$$

5: **return** $\pi^{\hat{f}}$

Lemma B.1. *Consider running Algorithm 3. Assume that \mathcal{D}_{sim} was generated as in Assumption 5, via the procedure of Lemma C.3 run with some parameter β , and γ satisfies*

$$2V_{\max}^2 \epsilon_{\text{sim}}^2 + \frac{43V_{\max}^2 \beta^2}{dH} \cdot \log \frac{8H|\mathcal{F}_h|}{\delta} + 6V_{\max}^2 \beta \sqrt{\frac{\log \frac{8H|\mathcal{F}_h|}{\delta}}{dH}} \leq \gamma.$$

Furthermore, assume that there exists some $\mathfrak{C}, \epsilon > 0$ such that, for any $\pi, h \in [H]$, and $\mathcal{Z}' \subseteq \mathcal{S} \times \mathcal{A}$, we have:

$$w_h^{\text{real}, \pi}(\mathcal{Z}') \leq \mathfrak{C} \cdot w_h^{\text{real}, \pi_{\text{exp}}}(\mathcal{Z}') + \epsilon. \tag{B.3}$$

Then with probability at least $1 - 2\delta$, the policy $\pi^{\hat{f}}$ generated by Algorithm 3 satisfies

$$V_0^* - V_0^{\pi^{\hat{f}}} \leq 4\mathfrak{C}H\sqrt{\frac{256V_{\max}^2 \log(4H|\tilde{\mathcal{F}}(\pi_{\exp}^{\text{sim}})|/\delta)}{T}} + 4HV_{\max}\epsilon$$

for

$$\tilde{\mathcal{F}}(\pi_{\exp}^{\text{sim}}) := \{f \in \mathcal{F} : \mathbb{E}^{\text{sim}, \pi_{\exp}^{\text{sim}}}[(f_h(s_h, a_h) - \mathcal{T}^{\text{sim}} f_{h+1}(s_h, a_h))^2] \leq 2\gamma, \forall h \in [H]\}.$$

Proof. Let \mathcal{E} denote the good event of Lemma B.2, which holds with probability at least $1 - 2\delta$. By Lemma A.4 we have

$$\begin{aligned} V_0^{\text{real}, \star} - V_0^{\text{real}, \pi^{\hat{f}}} &\leq \max_{\pi \in \{\pi^{\hat{f}}, \pi^{\text{real}, \star}\}} \sum_{h=0}^{H-1} 2 \left| \mathbb{E}^{\text{real}, \pi}[\hat{f}_h(s_h, a_h) - \mathcal{T}^{\text{real}} \hat{f}_{h+1}(s_h, a_h)] \right| \\ &\leq \max_{\pi} \sum_{h=0}^{H-1} 2 \mathbb{E}^{\text{real}, \pi}[\|\mathcal{E}_h^{\text{real}}(\hat{f})(s_h, a_h)\|]. \end{aligned}$$

Let

$$\mathcal{Z}_{h,i} := \{(s, a) : \|\mathcal{E}_h^{\text{real}}(\hat{f})(s, a)\| \in [V_{\max} \cdot 2^{-i}, V_{\max} \cdot 2^{-i+1}]\}.$$

Then we have, for any π ,

$$\begin{aligned} \mathbb{E}^{\text{real}, \pi}[\|\mathcal{E}_h^{\text{real}}(\hat{f})(s_h, a_h)\|] &\leq \sum_{i=1}^{\infty} w_h^{\text{real}, \pi}(\mathcal{Z}_{h,i}) \cdot V_{\max} 2^{-i+1} \\ &\leq \mathfrak{C} \cdot \sum_{i=1}^{\infty} w_h^{\text{real}, \pi_{\exp}}(\mathcal{Z}_{h,i}) \cdot V_{\max} 2^{-i+1} + 2V_{\max}\epsilon \\ &\leq 2\mathfrak{C} \cdot \mathbb{E}^{\text{real}, \pi_{\exp}}[\|\mathcal{E}_h^{\text{real}}(\hat{f})(s_h, a_h)\|] + 2V_{\max}\epsilon \end{aligned}$$

where the second inequality follows from (B.3). On \mathcal{E} , by Lemma B.2 and Jensen's inequality, we have

$$\mathbb{E}^{\text{real}, \pi_{\exp}}[\|\mathcal{E}_h^{\text{real}}(\hat{f})(s_h, a_h)\|] \leq \sqrt{\mathbb{E}^{\text{real}, \pi_{\exp}}[\mathcal{E}_h^{\text{real}}(\hat{f})(s_h, a_h)^2]} \leq \sqrt{\frac{1}{T} \cdot 256V_{\max}^2 \log \frac{2H|\tilde{\mathcal{F}}_h(\pi_{\exp}^{\text{sim}})|}{\delta}}.$$

As this holds for each h and π , we have therefore shown that

$$\begin{aligned} V_0^{\text{real}, \star} - V_0^{\text{real}, \pi^{\hat{f}}} &\leq 4\mathfrak{C} \cdot \sum_{h=0}^{H-1} \sqrt{\frac{1}{T} \cdot 256V_{\max}^2 \log \frac{2H|\tilde{\mathcal{F}}_h(\pi_{\exp}^{\text{sim}})|}{\delta}} + 4HV_{\max}\epsilon \\ &\leq 4\mathfrak{C}H\sqrt{\frac{1}{T} \cdot 256V_{\max}^2 \log \frac{2H|\tilde{\mathcal{F}}(\pi_{\exp}^{\text{sim}})|}{\delta}} + 4HV_{\max}\epsilon. \end{aligned}$$

This proves the result. \square

Lemma B.2. With probability at least $1 - 2\delta$, for each $h \in [H]$ simultaneously, as long as the conditions on γ given in Lemma B.3 hold, we have

$$\mathbb{E}^{\text{real}, \pi_{\exp}}[(\hat{f}_h(s_h, a_h) - \mathcal{T}^{\text{real}} \hat{f}_{h+1}(s_h, a_h))^2] \leq \frac{1}{T} \cdot 256V_{\max}^2 \log(2H|\tilde{\mathcal{F}}_h(\pi_{\exp}^{\text{sim}})|/\delta),$$

and $\hat{f}_h \in \tilde{\mathcal{F}}_h(\pi_{\exp}^{\text{sim}})$ for all $h \in [H]$, where

$$\tilde{\mathcal{F}}_h(\pi_{\exp}^{\text{sim}}) := \{f_h \in \mathcal{F}_h : \exists f_{h+1} \in \mathcal{F}_{h+1} \text{ s.t. } \mathbb{E}^{\text{sim}, \pi_{\exp}^{\text{sim}}}[(f_h(s_h, a_h) - \mathcal{T}^{\text{sim}} f_{h+1}(s_h, a_h))^2] \leq 2\gamma\}.$$

Proof. Let $\hat{\mathcal{F}}_h$ denote the feasible set of (B.2) at step h . By Lemma B.3, with probability at least $1 - \delta$, $\hat{\mathcal{F}}_h^t \subseteq \tilde{\mathcal{F}}_h$, and, furthermore, that $\mathcal{T}^{\text{real}} \hat{f}_{h+1}$ is feasible. The result then follows from Lemma 3 of [57], since the constraint on the regression problem restricts the version space. \square

Lemma B.3. Assume that data in $\mathfrak{D}_{\text{sim}}$ is generated as in Assumption 5 via the procedure of Lemma C.3 run with some parameter β , and γ satisfies

$$2V_{\max}^2 \epsilon_{\text{sim}}^2 + \frac{43V_{\max}^2 \beta^2}{dH} \cdot \log \frac{8H|\mathcal{F}_h|}{\delta} + 6V_{\max}^2 \beta \sqrt{\frac{\log \frac{8H|\mathcal{F}_h|}{\delta}}{dH}} \leq \gamma.$$

Then with probability at least $1 - \delta$ we have, for each $h \in [H]$:

1. $\mathcal{T}^{\text{real}} \hat{f}_{h+1}$ is feasible for (B.2).
2. The set of feasible f for (B.2) is a subset of

$$\{f \in \mathcal{F} : \mathbb{E}^{\text{sim}, \pi_{\text{exp}}^{\text{sim}}} [(f_h(s_h, a_h) - \mathcal{T}^{\text{sim}} \hat{f}_{h+1}(s_h, a_h))^2] \leq 2\gamma\}.$$

Proof. By Lemma C.1, we have that with probability at least $1 - \delta/2H$,

$$\frac{1}{T_{\text{sim}}} \sum_{t=1}^{T_{\text{sim}}} (\mathcal{T}^{\text{real}} \hat{f}_{h+1}(\tilde{s}_h^t, \tilde{a}_h^t) - \tilde{f}_h(\tilde{s}_h^t, \tilde{a}_h^t))^2 \leq 2V_{\max}^2 \epsilon_{\text{sim}}^2 + \frac{512V_{\max}^2}{T_{\text{sim}}} \cdot \log \frac{8H|\mathcal{F}_h|}{\delta} + V_{\max}^2 \sqrt{\frac{2 \log \frac{4H|\mathcal{F}_h|}{\delta}}{T_{\text{sim}}}}.$$

By Lemma C.3, we have $\frac{12dH}{\beta^2} \leq T_{\text{sim}}$, which implies

$$\frac{1}{T_{\text{sim}}} \sum_{t=1}^{T_{\text{sim}}} (\mathcal{T}^{\text{real}} \hat{f}_{h+1}(\tilde{s}_h^t, \tilde{a}_h^t) - \tilde{f}_h(\tilde{s}_h^t, \tilde{a}_h^t))^2 \leq 2V_{\max}^2 \epsilon_{\text{sim}}^2 + \frac{43V_{\max}^2 \beta^2}{dH} \cdot \log \frac{8H|\mathcal{F}_h|}{\delta} + V_{\max}^2 \beta \sqrt{\frac{\log \frac{4H|\mathcal{F}_h|}{\delta}}{6dH}}.$$

Part 1 then follows given our assumption on γ .

To bound the feasible set for (B.2) we appeal to Lemma C.2 which states that with probability at least $1 - \delta/2H$ we have that the feasible set of (B.2) is a subset of

$$\left\{ f_h \in \mathcal{F}_h : \mathbb{E}^{\text{sim}, \pi_{\text{exp}}^{\text{sim}}} [(f_h(s_h, a_h) - \mathcal{T}^{\text{sim}} \hat{f}_{h+1}(s_h, a_h))^2] \leq \gamma + 18V_{\max}^2 \sqrt{\frac{\log \frac{8H|\mathcal{F}_h|}{\delta}}{T_{\text{sim}}}} \right\}.$$

Again using that $\frac{12dH}{\beta^2} \leq T_{\text{sim}}$, we have have that this is a subset of

$$\begin{aligned} & \left\{ f_h \in \mathcal{F}_h : \mathbb{E}^{\text{sim}, \pi_{\text{exp}}^{\text{sim}}} [(f_h(s_h, a_h) - \mathcal{T}^{\text{sim}} \hat{f}_{h+1}(s_h, a_h))^2] \leq \gamma + 18V_{\max}^2 \beta \sqrt{\frac{\log \frac{8H|\mathcal{F}_h|}{\delta}}{12dH}} \right\} \\ & \subseteq \left\{ f_h \in \mathcal{F}_h : \mathbb{E}^{\text{sim}, \pi_{\text{exp}}^{\text{sim}}} [(f_h(s_h, a_h) - \mathcal{T}^{\text{sim}} \hat{f}_{h+1}(s_h, a_h))^2] \leq 2\gamma \right\} \end{aligned}$$

where the inclusion follows from our assumption on γ . The result then follows from a union bound. \square

B.2 Performance of Full-Rank sim Policies in real

Lemma B.4. Consider policies $\{\pi_{\text{exp}}^h\}_{h=1}^H$, and assume that

$$\lambda_{\min} \left(\mathbf{A}_{\pi_{\text{exp}, h}^s}^s \right) \geq \bar{\lambda}_{\min}, \quad \forall h \in [H] \tag{B.4}$$

and that π_{exp}^h plays actions uniformly at random for $h' > h$. Let $\pi_{\text{exp}} = \text{unif}(\{\pi_{\text{exp}}^h\}_{h=1}^H)$. Then, for any π , $\kappa > 0$, $\gamma > 0$, $h \in [H]$, and $\mathcal{Z}' \subseteq \mathcal{S} \times \mathcal{A}$, we have

$$w_h^{\text{real}, \pi}(\mathcal{Z}') \leq \frac{4H\gamma A^{k^*-2}}{\kappa} \cdot w_h^{\text{real}, \pi_{\text{exp}}}(\mathcal{Z}') + 4\kappa,$$

where

$$\xi := 2\sqrt{\frac{A}{\bar{\lambda}_{\min}}} \left(\frac{d}{\gamma} + H\epsilon_{\text{sim}} \right) \quad \text{and} \quad k^* := \lceil \frac{\log 1/\kappa}{\log 1/\xi} \rceil.$$

Proof. Denote

$$\tilde{\mathcal{Z}}_{h+1} := \{(s, a) : \phi^r(s, a)^\top (\Lambda_{\pi_{\text{exp}, h+1}}^r)^{-1} \phi^r(s, a) > \gamma\}$$

for some $\gamma > 0$. We have

$$\begin{aligned} w_{h+1}^{\text{real}, \pi_{\text{exp}}^h}(\tilde{\mathcal{Z}}_{h+1}) &= \mathbb{E}^{\text{real}, \pi_{\text{exp}}^h}[\mathbb{I}\{(s_{h+1}, a_{h+1}) \in \tilde{\mathcal{Z}}_{h+1}\}] \\ &\stackrel{(a)}{\leq} \mathbb{E}^{\text{real}, \pi_{\text{exp}}^h} \left[\frac{\phi^r(s_{h+1}, a_{h+1})^\top (\Lambda_{\pi_{\text{exp}, h+1}}^r)^{-1} \phi^r(s_{h+1}, a_{h+1})}{\gamma} \cdot \mathbb{I}\{(s_{h+1}, a_{h+1}) \in \tilde{\mathcal{Z}}_{h+1}\} \right] \\ &\leq \mathbb{E}^{\text{real}, \pi_{\text{exp}}^h} \left[\frac{\phi^r(s_{h+1}, a_{h+1})^\top (\Lambda_{\pi_{\text{exp}, h+1}}^r)^{-1} \phi^r(s_{h+1}, a_{h+1})}{\gamma} \right] \\ &= \frac{1}{\gamma} \cdot \text{tr} \left(\mathbb{E}^{\text{real}, \pi_{\text{exp}}^h} [\phi^r(s_{h+1}, a_{h+1}) \phi^r(s_{h+1}, a_{h+1})^\top] (\Lambda_{\pi_{\text{exp}, h+1}}^r)^{-1} \right) \\ &= \frac{d}{\gamma} \end{aligned}$$

where (a) follows since for all $(s, a) \in \tilde{\mathcal{Z}}_{h+1}$, we have $1 < \phi^r(s, a)^\top (\Lambda_{\pi_{\text{exp}, h+1}}^r)^{-1} \phi^r(s, a) / \gamma$. By Lemma A.2, we then have that

$$w_{h+1}^{\text{sim}, \pi_{\text{exp}}^h}(\tilde{\mathcal{Z}}_{h+1}) \leq \frac{d}{\gamma} + H\epsilon_{\text{sim}}. \quad (\text{B.5})$$

Let $\tilde{\mathcal{S}}_{h+1} := \{s : \exists a \text{ s.t. } (s, a) \in \tilde{\mathcal{Z}}_{h+1}\}$ and note that

$$\begin{aligned} w_{h+1}^{\text{sim}, \pi_{\text{exp}}^h}(\tilde{\mathcal{Z}}_{h+1}) &= \mathbb{E}^{\text{sim}, \pi_{\text{exp}}^h} \left[\int_{\tilde{\mathcal{S}}_{h+1}} \sum_{a: (s, a) \in \tilde{\mathcal{Z}}_{h+1}} \pi_{\text{exp}}^h(a | s, h+1) dP_h^{\text{sim}}(s | s_h, a_h) \right] \\ &\geq \frac{1}{A} \mathbb{E}^{\text{sim}, \pi_{\text{exp}}^h} \left[\int_{\tilde{\mathcal{S}}_{h+1}} d\mu_h^s(s)^\top \phi^s(s_h, a_h) \right] \\ &= \frac{1}{A} \mathbb{E}^{\text{sim}, \pi_{\text{exp}}^h} [P_h^{\text{sim}}(\tilde{\mathcal{S}}_{h+1} | s_h, a_h)] \\ &\geq \frac{1}{A} \mathbb{E}^{\text{sim}, \pi_{\text{exp}}^h} [P_h^{\text{sim}}(\tilde{\mathcal{S}}_{h+1} | s_h, a_h)^2] \end{aligned}$$

where we have used the fact that $\pi_{\text{exp}}^h(a | s, h+1) = 1/A$ for all (s, a) by assumption, and define $P_h^{\text{sim}}(\tilde{\mathcal{S}}_{h+1} | s, a) := \mathbb{P}^{\text{sim}}[s_{h+1} \in \tilde{\mathcal{S}}_{h+1} | s_h = s, a_h = a] = \int_{\tilde{\mathcal{S}}_{h+1}} d\mu_h^s(s)^\top \phi^s(s, a)$, where the last equality follows from the definition of a linear MDP. Letting $\mu_h^s(\tilde{\mathcal{S}}_{h+1}) := \int_{\tilde{\mathcal{S}}_{h+1}} d\mu_h^s(s)$, note that:

$$\begin{aligned} \frac{1}{A} \mathbb{E}^{\text{sim}, \pi_{\text{exp}}^h} [P_h^{\text{sim}}(\tilde{\mathcal{S}}_{h+1} | s_h, a_h)^2] &= \frac{1}{A} \mu_h^s(\tilde{\mathcal{S}}_{h+1})^\top \mathbb{E}^{\text{sim}, \pi_{\text{exp}}^h} [\phi^s(s_h, a_h) \phi^s(s_h, a_h)^\top] \mu_h^s(\tilde{\mathcal{S}}_{h+1}) \\ &= \frac{1}{A} \mu_h^s(\tilde{\mathcal{S}}_{h+1})^\top \Lambda_{\pi_{\text{exp}, h}}^s \mu_h^s(\tilde{\mathcal{S}}_{h+1}) \\ &\geq \frac{\bar{\lambda}_{\min}}{A} \|\mu_h^s(\tilde{\mathcal{S}}_{h+1})\|_2^2, \end{aligned}$$

where the last inequality follows from (B.4). Combining this with (B.5), we have

$$\frac{d}{\gamma} + H\epsilon_{\text{sim}} \geq \frac{\bar{\lambda}_{\min}}{A} \|\mu_h^s(\tilde{\mathcal{S}}_{h+1})\|_2^2.$$

Now note that, for any $z \in \mathcal{S} \times \mathcal{A}$:

$$P_h^{\text{sim}}(\tilde{\mathcal{S}}_{h+1} | z) = \int_{\tilde{\mathcal{S}}_{h+1}} dP_h^{\text{sim}}(s | z) = \left(\int_{\tilde{\mathcal{S}}_{h+1}} d\mu_h^s(s) \right)^\top \phi^s(z) \leq \|\mu_h^s(\tilde{\mathcal{S}}_{h+1})\|_2$$

and we also have that $P_h^{\text{sim}}(\tilde{\mathcal{S}}_{h+1} \mid z) \geq P_h^{\text{real}}(\tilde{\mathcal{S}}_{h+1} \mid z) - \epsilon_{\text{sim}}$ under Assumption 1. Putting this together we have that for all $z \in \mathcal{S} \times \mathcal{A}$:

$$P_h^{\text{real}}(\tilde{\mathcal{S}}_{h+1} \mid z) \leq \sqrt{\frac{A}{\bar{\lambda}_{\min}} \left(\frac{d}{\gamma} + H\epsilon_{\text{sim}} \right)} + \epsilon_{\text{sim}}.$$

Note that we can always take $\epsilon_{\text{sim}} \leq 1$, and will always have $\bar{\lambda}_{\min} \leq 1$. This implies that $\epsilon_{\text{sim}} \leq \sqrt{\frac{A}{\bar{\lambda}_{\min}} \left(\frac{d}{\gamma} + H\epsilon_{\text{sim}} \right)}$. Thus,

$$P_h^{\text{real}}(\tilde{\mathcal{S}}_{h+1} \mid z) \leq 2\sqrt{\frac{A}{\bar{\lambda}_{\min}} \left(\frac{d}{\gamma} + H\epsilon_{\text{sim}} \right)} =: \xi.$$

Coverage of π_{exp} in real. Let $k^* := \lceil \frac{\log 1/\kappa}{\log 1/\xi} \rceil$, so that $\xi^{k^*} \leq \kappa$. Let $\bar{\mathcal{Z}}_h := (\mathcal{S} \times \mathcal{A}) \setminus \tilde{\mathcal{Z}}_h$. Fix some $\mathcal{Z}' \subseteq (\mathcal{S} \times \mathcal{A})$, $h \in [H]$, and policy π .

Consider some $z \in \bar{\mathcal{Z}}_h$, and some $\mathcal{S}' \subseteq \mathcal{S}$. Then note that⁵

$$\begin{aligned} P_h^{\text{real}}(\mathcal{S}' \mid z) &= \boldsymbol{\mu}_h^r(\mathcal{S}')^\top \boldsymbol{\phi}^r(z) = \boldsymbol{\mu}_h^r(\mathcal{S}')^\top (\boldsymbol{\Lambda}_{\pi_{\text{exp}}^{h-1}, h}^r)^{1/2} (\boldsymbol{\Lambda}_{\pi_{\text{exp}}^{h-1}, h}^r)^{-1/2} \boldsymbol{\phi}^r(z) \\ &\leq \|\boldsymbol{\mu}_h^r(\mathcal{S}')\|_{\boldsymbol{\Lambda}_{\pi_{\text{exp}}^{h-1}, h}^r} \|\boldsymbol{\phi}^r(z)\|_{(\boldsymbol{\Lambda}_{\pi_{\text{exp}}^{h-1}, h}^r)^{-1}} \\ &\leq \sqrt{\gamma} \|\boldsymbol{\mu}_h^r(\mathcal{S}')\|_{\boldsymbol{\Lambda}_{\pi_{\text{exp}}^{h-1}, h}^r} \end{aligned}$$

where the last inequality follows from the definition of $\bar{\mathcal{Z}}_h$. Note, though, that

$$\|\boldsymbol{\mu}_h^r(\mathcal{S}')\|_{\boldsymbol{\Lambda}_{\pi_{\text{exp}}^{h-1}, h}^r}^2 = \mathbb{E}^{\text{real}, \pi_{\text{exp}}^{h-1}} [(\boldsymbol{\mu}_h^r(\mathcal{S}')^\top \boldsymbol{\phi}^r(z_h))^2] = \mathbb{E}^{\text{real}, \pi_{\text{exp}}^{h-1}} [P_h^{\text{real}}(\mathcal{S}' \mid z_h)^2].$$

This implies that for all $z \in \bar{\mathcal{Z}}_h$,

$$\mathbb{E}^{\text{real}, \pi_{\text{exp}}^{h-1}} [P_h^{\text{real}}(\mathcal{S}' \mid z_h)^2] \geq \frac{1}{\gamma} \cdot P_h^{\text{real}}(\mathcal{S}' \mid z)^2.$$

For $h' < h$, define

$$\mathcal{S}_{h', i} := \{s : w_h^{\text{real}, \pi}(\mathcal{Z}' \mid s_{h'} = s) \in [2^{-i+1}, 2^{-i}]\}$$

for $w_h^{\text{real}, \pi}(\mathcal{Z} \mid s_{h'} = s) := \mathbb{P}^{\text{real}, \pi}[z_h \in \mathcal{Z} \mid s_{h'} = s]$. Note that we then have $w_h^{\text{real}, \pi}(\mathcal{Z}' \mid \mathcal{S}_{h', i}) \in [2^{-i+1}, 2^{-i}]$. By what we have just shown, we have that for $z \in \bar{\mathcal{Z}}_{h'}$

$$\mathbb{E}^{\text{real}, \pi_{\text{exp}}^{h'-1}} [P_{h'}^{\text{real}}(\mathcal{S}_{h'+1, i} \mid z_{h'})^2] \geq \frac{1}{\gamma} \cdot P_{h'}^{\text{real}}(\mathcal{S}_{h'+1, i} \mid z)^2$$

which implies that

$$\mathbb{E}^{\text{real}, \pi_{\text{exp}}^{h'-1}} [P_{h'}^{\text{real}}(\mathcal{S}_{h'+1, i} \mid z_{h'})] \geq \frac{1}{\gamma} \cdot P_{h'}^{\text{real}}(\mathcal{S}_{h'+1, i} \mid z)^2. \quad (\text{B.6})$$

⁵If $\boldsymbol{\Lambda}_{\pi_{\text{exp}}^{h-1}, h}^r$ is not invertible, we can repeat this argument with $\boldsymbol{\Lambda}_{\pi_{\text{exp}}^{h-1}, h}^r + \lambda I$ and take $\lambda \rightarrow 0$.

Fix $z \in \bar{\mathcal{Z}}_{h'}$. Note that

$$\begin{aligned}
w_h^{\text{real}, \pi}(\mathcal{Z}' \mid z_{h'} = z) &= \mathbb{E}_{s \sim P_{h'}^{\text{real}}(\cdot \mid z)}[w_h^{\text{real}, \pi}(\mathcal{Z}' \mid s_{h'+1} = s)] \\
&= \sum_{i=1}^{\infty} \mathbb{E}_{s \sim P_{h'}^{\text{real}}(\cdot \mid z)}[w_h^{\text{real}, \pi}(\mathcal{Z}' \mid s_{h'+1} = s) \cdot \mathbb{I}\{s \in \mathcal{S}_{h'+1, i}\}] \\
&\leq \sum_{i=1}^{\infty} 2^{-i+1} P_{h'}^{\text{real}}(\mathcal{S}_{h'+1, i} \mid z) \\
&= \sum_{i=1}^{\lfloor \log 4/\kappa \rfloor} 2^{-i+1} P_{h'}^{\text{real}}(\mathcal{S}_{h'+1, i} \mid z) + \kappa \\
&\leq \sum_{i=1}^{\lfloor \log 4/\kappa \rfloor} 2^{-i+1} P_{h'}^{\text{real}}(\mathcal{S}_{h'+1, i} \mid z) \cdot \mathbb{I}\{P_{h'}^{\text{real}}(\mathcal{S}_{h'+1, i} \mid z) \geq \kappa\} + 3\kappa \\
&\leq 2 \sum_{i=1}^{\lfloor \log 4/\kappa \rfloor} \mathbb{E}_{s \sim \lambda_i}[w_h^{\text{real}, \pi}(\mathcal{Z}' \mid s_{h'+1} = s)] P_{h'}^{\text{real}}(\mathcal{S}_{h'+1, i} \mid z) \cdot \mathbb{I}\{P_{h'}^{\text{real}}(\mathcal{S}_{h'+1, i} \mid z) \geq \kappa\} + 3\kappa
\end{aligned}$$

for any $\lambda_i \in \Delta_{\mathcal{S}_{h'+1, i}}$. Note also that, since $\pi_{\text{exp}}^{h'-1}$ plays randomly for all $h'' \geq h'$, we have:

$$w_h^{\text{real}, \pi_{\text{exp}}^{h'-1}}(\mathcal{Z}' \mid s_{h'+1} = s) \geq \frac{1}{A^{h-h'}} \cdot w_h^{\text{real}, \pi}(\mathcal{Z}' \mid s_{h'+1} = s),$$

since with probability $1/A^{h-h'}$ on any given episode, $\pi_{\text{exp}}^{h'-1}$ will play the same sequence of actions as π from steps h' to h . It follows that we can bound the above as:

$$\begin{aligned}
&\leq 2A^{h-h'} \cdot \sum_{i=1}^{\lfloor \log 4/\kappa \rfloor} \mathbb{E}_{s \sim \lambda_i}[w_h^{\text{real}, \pi_{\text{exp}}^{h'-1}}(\mathcal{Z}' \mid s_{h'+1} = s)] P_{h'}^{\text{real}}(\mathcal{S}_{h'+1, i} \mid z) \cdot \mathbb{I}\{P_{h'}^{\text{real}}(\mathcal{S}_{h'+1, i} \mid z) \geq \kappa\} + 3\kappa \\
&\stackrel{(a)}{\leq} \frac{2A^{h-h'} \gamma}{\kappa} \cdot \sum_{i=1}^{\lfloor \log 4/\kappa \rfloor} \mathbb{E}_{s \sim \lambda_i}[w_h^{\text{real}, \pi_{\text{exp}}^{h'-1}}(\mathcal{Z}' \mid s_{h'+1} = s)] \mathbb{E}^{\text{real}, \pi_{\text{exp}}^{h'-1}}[P_{h'}^{\text{real}}(\mathcal{S}_{h'+1, i} \mid z_{h'})] \mathbb{I}\{P_{h'}^{\text{real}}(\mathcal{S}_{h'+1, i} \mid z) \geq \kappa\} + 3\kappa \\
&\leq \frac{2\gamma A^{h-h'}}{\kappa} \cdot \sum_{i=1}^{\lfloor \log 4/\kappa \rfloor} \mathbb{E}_{s \sim \lambda_i}[w_h^{\text{real}, \pi_{\text{exp}}^{h'-1}}(\mathcal{Z}' \mid s_{h'+1} = s)] \cdot w_{h'+1}^{\text{real}, \pi_{\text{exp}}^{h'-1}}(\mathcal{S}_{h'+1, i}) + 3\kappa \\
&\stackrel{(b)}{=} \frac{2\gamma A^{h-h'}}{\kappa} \cdot \sum_{i=1}^{\lfloor \log 4/\kappa \rfloor} \sum_{s \in \mathcal{S}_{h'+1, i}} w_h^{\text{real}, \pi_{\text{exp}}^{h'-1}}(\mathcal{Z}' \mid s_{h'+1} = s) w_{h'+1}^{\text{real}, \pi_{\text{exp}}^{h'-1}}(s) + 3\kappa \\
&\leq \frac{2\gamma A^{h-h'}}{\kappa} \cdot w_h^{\text{real}, \pi_{\text{exp}}^{h'-1}}(\mathcal{Z}') + 3\kappa
\end{aligned}$$

where (a) follows from (B.6) and since $P_{h'}^{\text{real}}(\mathcal{S}_{h'+1, i} \mid z) \geq \kappa$, and (b) follows choosing $\lambda_i(s) = w_{h'+1}^{\text{real}, \pi_{\text{exp}}^{h'-1}}(s)/w_{h'+1}^{\text{real}, \pi_{\text{exp}}^{h'-1}}(\mathcal{S}_{h'+1, i}) \cdot \mathbb{I}\{s \in \mathcal{S}_{h'+1, i}\}$. We therefore have that, for all $z \in \bar{\mathcal{Z}}_{h'}$:

$$w_h^{\text{real}, \pi}(\mathcal{Z}' \mid z_{h'} = z) \leq \frac{2\gamma A^{h-h'}}{\kappa} \cdot w_h^{\text{real}, \pi_{\text{exp}}^{h'-1}}(\mathcal{Z}') + 3\kappa. \quad (\text{B.7})$$

Controlling events. Consider events $\mathcal{E} := \{z_h \in \mathcal{Z}'\}$ and $\mathcal{E}_{h'} := \{z_{h'} \in \bar{\mathcal{Z}}_{h'}\}$. We then have

$$\begin{aligned}
w_h^{\text{real}, \pi}(\mathcal{Z}') &= \mathbb{P}^{\text{real}, \pi}[\mathcal{E}] \\
&= \mathbb{P}^{\text{real}, \pi}[\mathcal{E} \cap \mathcal{E}_{h-1}] + \mathbb{P}^{\text{real}, \pi}[\mathcal{E} \cap \mathcal{E}_{h-1}^c] \\
&= \sum_{h'=h-k^*+1}^h \mathbb{P}^{\text{real}, \pi}[\mathcal{E} \cap \mathcal{E}_{h'-1} \cap \bigcap_{i=h'}^{h-1} \mathcal{E}_i^c] + \mathbb{P}^{\text{real}, \pi}[\mathcal{E} \cap \mathcal{E}_{h-k^*-1} \cap \bigcap_{i=h-k^*}^{h-1} \mathcal{E}_i^c] \\
&\leq \sum_{h'=h-k^*+1}^h \mathbb{P}^{\text{real}, \pi}[\mathcal{E} \cap \mathcal{E}_{h'-1}] + \mathbb{P}^{\text{real}, \pi}[\mathcal{E} \cap \mathcal{E}_{h-k^*-1} \cap \bigcap_{i=h-k^*}^{h-1} \mathcal{E}_i^c].
\end{aligned}$$

We now analyze each of these terms. First, note that

$$\mathbb{P}^{\text{real},\pi}[\mathcal{E} \cap \mathcal{E}_{h'-1}] = \mathbb{P}^{\text{real},\pi}[\mathcal{E} \mid \mathcal{E}_{h'-1}] \mathbb{P}^{\text{real},\pi}[\mathcal{E}_{h'-1}] \leq \mathbb{P}^{\text{real},\pi}[\mathcal{E} \mid \mathcal{E}_{h'-1}] = w_h^{\text{real},\pi}(\mathcal{Z}' \mid z_{h'-1} \in \bar{\mathcal{Z}}_{h'-1}).$$

We can then bound

$$w_h^{\text{real},\pi}(\mathcal{Z}' \mid z_{h'-1} \in \bar{\mathcal{Z}}_{h'-1}) \leq \frac{2\gamma A^{h-h'-1}}{\kappa} \cdot w_h^{\text{real},\pi_{\text{exp}}^{h'-2}}(\mathcal{Z}') + 3\kappa$$

where the inequality follows from (B.7). For the second term, we have

$$\begin{aligned} \mathbb{P}^{\text{real},\pi}[\mathcal{E} \cap \mathcal{E}_{h-k^*-1} \cap \bigcap_{i=h-k^*}^{h-1} \mathcal{E}_i^c] &\leq \mathbb{P}^{\text{real},\pi}[\mathcal{E} \cap \bigcap_{i=h-k^*}^{h-1} \mathcal{E}_i^c] \\ &= \mathbb{P}^{\text{real},\pi}[\mathcal{E} \mid \bigcap_{i=h-k^*}^{h-1} \mathcal{E}_i^c] \cdot \prod_{j=1}^{k^*} \mathbb{P}^{\text{real},\pi}[\mathcal{E}_{h-j}^c \mid \bigcap_{i=h-k^*}^{h-j-1} \mathcal{E}_i^c]. \end{aligned}$$

Note, however, that $\mathbb{P}^{\text{real},\pi}[\mathcal{E} \mid \bigcap_{i=h-k^*}^{h-1} \mathcal{E}_i^c] \leq \xi$ and $\mathbb{P}^{\text{real},\pi}[\mathcal{E}_{h-j}^c \mid \bigcap_{i=h-k^*}^{h-j-1} \mathcal{E}_i^c] \leq \xi$ for all j . We therefore can bound the above as

$$\xi^{k^*+1} \leq \kappa.$$

Altogether, then, we have that

$$w_h^{\text{real},\pi}(\mathcal{Z}') \leq \sum_{h'=h-k^*+1}^h \frac{2\gamma A^{h-h'-1}}{\kappa} \cdot w_h^{\text{real},\pi_{\text{exp}}^{h'-2}}(\mathcal{Z}') + 4\kappa.$$

Furthermore, since $\pi_{\text{exp}} = \text{unif}(\{\pi_{\text{exp}}^h\}_{h=1}^H)$, we have $w_h^{\text{real},\pi_{\text{exp}}^{h'-2}}(\mathcal{Z}') \leq H w_h^{\text{real},\pi_{\text{exp}}}(\mathcal{Z}')$, so we conclude that

$$w_h^{\text{real},\pi}(\mathcal{Z}') \leq \frac{4H\gamma A^{k^*-2}}{\kappa} \cdot w_h^{\text{real},\pi_{\text{exp}}}(\mathcal{Z}') + 4\kappa.$$

□

B.3 Proof of Unconstrained Upper Bound

Theorem 2. Assume that one of the two conditions is met:

1. For each h , π_{exp}^h plays actions uniformly at random for $h' > h$,

$$\lambda_{\min}(\Lambda_{\pi_{\text{exp},h}^s}) \geq \bar{\lambda}_{\min}, \quad (\text{B.8})$$

and

$$T \geq c \cdot \frac{V_{\max}^4 H^4 d^2 A^{2(k^*-2)} \log(2H|\mathcal{F}|/\delta)}{\epsilon^4 \epsilon_{\text{sim}}^2},$$

for

$$k^* = \lceil \frac{\log_A \frac{64HV_{\max}}{\epsilon}}{\log_A 1/\xi} \rceil, \quad \xi = 2\sqrt{\frac{2HA}{\bar{\lambda}_{\min}}} \cdot \epsilon_{\text{sim}}.$$

2. $\epsilon_{\text{sim}} \leq \epsilon/4H^2$ and

$$T \geq \frac{16H^2 \log \frac{4}{\delta}}{\epsilon^2}.$$

Then with probability at least $1 - \delta$, Algorithm 1 returns a $\hat{\pi}$ such that $V_0^{\text{real},\pi^{\text{real},*}} - V_0^{\text{real},\hat{\pi}} \leq \epsilon$.

Proof. We consider each of the conditions above.

Condition 1. First, note that by our assumption on π_{exp} and applying Lemma B.4 with $\kappa = \frac{\epsilon}{64HV_{\max}}$ and $\gamma = \frac{d}{H\epsilon_{\text{sim}}}$, for any π and $\mathcal{Z}' \subseteq \mathcal{S} \times \mathcal{A}$, we have

$$w_h^{\text{real}, \pi}(\mathcal{Z}') \leq \frac{256dHV_{\max}A^{k^*-2}}{\epsilon\epsilon_{\text{sim}}} \cdot w_h^{\text{real}, \pi_{\text{exp}}}(\mathcal{Z}') + \frac{\epsilon}{16HV_{\max}}$$

for

$$k^* = \lceil \frac{\log_A \frac{64HV_{\max}}{\epsilon}}{\log_A 1/\xi} \rceil, \quad \xi = 2\sqrt{\frac{2HA}{\lambda_{\min}}} \cdot \epsilon_{\text{sim}}.$$

By Lemma B.1 we then have that, with probability at least $1 - \delta^6$,

$$\begin{aligned} V_0^{\text{real}, \pi^{\text{real}, *}} - V_0^{\text{real}, \hat{\pi}} &\leq \frac{256dHV_{\max}A^{k^*-2}}{\epsilon\epsilon_{\text{sim}}} \cdot 4H\sqrt{\frac{256V_{\max}^2 \log(2H|\mathcal{F}|/\delta)}{T}} + \epsilon/4 \\ &\leq \epsilon/2 \end{aligned}$$

where the last inequality follows under our condition on T .

Condition 2. By Lemma A.3, we have that $V_0^{\text{real}, *} - V_0^{\text{real}, \pi^{\text{sim}, *}} \leq 2H^2\epsilon_{\text{sim}}$. Thus, if $\epsilon_{\text{sim}} \leq \epsilon/4H^2$, we have $V_0^{\text{real}, *} - V_0^{\text{real}, \pi^{\text{sim}, *}} \leq \epsilon/2$.

Concluding the Proof. By what we have shown, as long as one of our conditions is met, we will have that with probability at least $1 - \delta/2$, there exists $\pi \in \{\pi^{\hat{f}}, \pi^{\text{sim}, *}\}$ such that $V_0^{\text{real}, *} - V_0^{\text{real}, \pi} \leq \epsilon/2$. Denote this policy as $\tilde{\pi}$.

Note that $V_0^{\text{real}, \pi} = \mathbb{E}^{\text{real}, \pi}[\sum_{h=0}^{H-1} r_h]$ and that $\sum_{h=0}^{H-1} r_h \in [0, H]$ almost surely. Consider playing π for $T/4$ episodes in real and let R^i denote the total return of the i th episode. Let

$$\hat{V}_0^{\pi} := \frac{4}{T} \sum_{i=1}^{T/4} R^i.$$

By Hoeffding's inequality we have that, with probability at least $1 - \delta/4$:

$$|\hat{V}_0^{\pi} - V_0^{\text{real}, \pi}| \leq H\sqrt{\frac{4 \log \frac{4}{\delta}}{T}}.$$

Thus, if

$$T \geq \frac{16H^2 \log \frac{4}{\delta}}{\epsilon^2}, \tag{B.9}$$

we have that $|\hat{V}_0^{\tilde{\pi}} - V_0^{\text{real}, \tilde{\pi}}| \leq \epsilon/2$. Union bounding over this for both $\pi \in \{\pi^{\hat{f}}, \pi^{\text{sim}, *}\}$, we have that with probability at least $1 - \delta/2$:

$$V_0^{\text{real}, \hat{\pi}} \geq \hat{V}_0^{\hat{\pi}} - \epsilon/4 \geq \hat{V}_0^{\tilde{\pi}} - \epsilon/4 \geq V_0^{\text{real}, \tilde{\pi}} - \epsilon/2.$$

It follows that

$$V_0^{\text{real}, *} - V_0^{\text{real}, \hat{\pi}} \leq V_0^{\text{real}, *} - V_0^{\text{real}, \tilde{\pi}} + \epsilon/2 \leq \epsilon.$$

The proof follows from a union bound and our condition on T (note that (B.9) is satisfied in both cases). □

⁶Note that, while Lemma B.1 applies to the constrained regression setting, this is equivalent to the unconstrained regression setting considered here if we choose γ large enough so that the constraint is vacuous.

Proof of Theorem 1. We first assume that $\zeta \leq \frac{\lambda_{\min}^*}{4d}$, for ζ the input regularization value given to Algorithm 5 by Algorithm 1, and Condition 1 of Theorem 2, and show that in this case A^{k^*-2} is at most polynomial in problem parameters.

First, by Lemma C.7 we have that, under the assumption that $\zeta \leq \frac{\lambda_{\min}^*}{4d}$, the policy π_{\exp}^h given by the uniform mixture of policies returned by Algorithm 5 will, with probability at least $1 - \delta$, satisfy $\lambda_{\min}(\mathbf{A}_{\pi_{\exp}^h}^s) \geq \frac{\lambda_{\min}^*}{8d}$ under Assumption 3. Plugging $\bar{\lambda}_{\min} \leftarrow \frac{\lambda_{\min}^*}{8d}$ into Theorem 2, we have that $\xi = 2\sqrt{\frac{16dHA}{\lambda_{\min}^*}} \cdot \epsilon_{\text{sim}}$. Now note that

$$A^{k^*-2} \leq A^{\frac{\log_A 64HV_{\max}/\epsilon}{\log_A 1/\xi}} = \left(\frac{64HV_{\max}}{\epsilon} \right)^{1/\log_A 1/\xi}.$$

It then suffices that we show $1/\log_A 1/\xi \leq 1 \iff 1/A \geq \xi$. However, this is clearly met by our condition on ϵ_{sim} . Thus, as long as

$$T \geq c \cdot \frac{V_{\max}^6 H^6 d^2 \log(2HT|\mathcal{F}|/\delta)}{\epsilon^6 \epsilon_{\text{sim}}^2},$$

by Theorem 2 we have that $\hat{\pi}$ is ϵ -optimal.

Now, if $\epsilon_{\text{sim}} \leq \epsilon/4H^2$ and $T \geq \frac{16H^2 \log 4/\delta}{\epsilon}$, we also have that $\hat{\pi}$ is ϵ -optimal, by Theorem 2. Thus, in the first case, we at most will require

$$T \geq c \cdot \frac{V_{\max}^6 H^{10} d^2 \log(2HT|\mathcal{F}|/\delta)}{\epsilon^8}$$

to produce a policy that is ϵ -optimal, since otherwise we will be in the second case.

It remains to justify the assumption that $\zeta \leq \frac{\lambda_{\min}^*}{4d}$. Note that the condition of (4.1) is only required in the first case. Furthermore, if $\epsilon_{\text{sim}} \leq \epsilon/4H^2$ we will be in the second case. Thus, in the first case, we will have

$$\frac{\epsilon}{4H^2} \leq \epsilon_{\text{sim}} \leq \frac{\lambda_{\min}^*}{64dHA^3}.$$

Rearranging this we obtain that, to be in the first case, we have

$$\frac{16dA^3\epsilon}{H} \leq \lambda_{\min}^*$$

By our choice of $\zeta = \frac{4A^3\epsilon}{H}$, we then have that $\zeta \leq \frac{\lambda_{\min}^*}{4d}$. By Lemma C.7 and our choice of ζ , we have that Oracle 4.2 is called at most $\text{poly}(d, H, \epsilon^{-1}, \log \frac{1}{\delta})$ times, and we call the oracle of Oracle 4.1 only H times. The result follows from a union bound and rescaling δ . \square

B.4 Reducing the Version Space

As we noted, in general, given that we do not assume that ϕ^r is unknown, $\log |\mathcal{F}|$ could be significantly greater than the dimension. One might hope that, given access to \mathcal{M}^{sim} , we can reduce this dependence somewhat. We next show that this is possible given access to the following *constrained* regression oracle.

Oracle B.1 (Constrained Regression Oracle). We assume access to a regression oracle such that, for any h and datasets $\{(s^t, a^t, y^t)\}_{t=1}^T$ and $\{(\tilde{s}^t, \tilde{a}^t, \tilde{y}^t)\}_{t=1}^{\tilde{T}}$, we can compute:

$$\hat{f}_h = \arg \min_{f \in \mathcal{F}_h} \sum_{t=1}^T (f(s^t, a^t) - y^t)^2 \quad \text{s.t.} \quad \sum_{t=1}^{\tilde{T}} (f(\tilde{s}^t, \tilde{a}^t) - \tilde{y}^t)^2 \leq \gamma.$$

While in general the oracle of Oracle B.1 cannot be reduced to the oracle of Oracle 4.1, under certain conditions on \mathcal{F} this is possible. Given this oracle, we have the following result.

Theorem 3. Assume that $\epsilon_{\text{sim}} \leq \frac{\lambda_{\min}^*}{64dHA^3}$. Then if

$$T \geq \tilde{\mathcal{O}} \left(\frac{d^2 H^{16}}{\epsilon^8} \cdot \log \frac{H|\tilde{\mathcal{F}}|}{\delta} \right),$$

with probability at least $1 - \delta$, Algorithm 4 returns policy $\hat{\pi}$ such that $V_0^{\text{real}, \pi^{\text{real}, *}} - V_0^{\text{real}, \hat{\pi}} \leq \epsilon$, where

$$\tilde{\mathcal{F}} := \left\{ f \in \mathcal{F} : \sup_{\pi} (\mathbb{E}^{\text{sim}, \pi} [f_h(s_h, a_h) - \mathcal{T}^{\text{sim}} f_{h+1}(s_h, a_h)])^2 \leq \alpha \cdot \epsilon_{\text{sim}}^2 \right\}$$

for $\alpha = \tilde{\mathcal{O}}(AdH^3 \cdot \log^2 \frac{\log |\mathcal{F}|/\delta}{\epsilon_{\text{sim}}})$. Furthermore, the computation oracles of Oracle 4.2 and Oracle B.1 are called at most $\text{poly}(d, A, H, \epsilon^{-1}, \log \frac{|\mathcal{F}|}{\delta})$ times.

Theorem 3 shows that, rather than paying for the full complexity of \mathcal{F} , we can pay only for the subset of \mathcal{F} that is Bellman-consistent on \mathcal{M}^{sim} .

B.4.1 Algorithm and Proof

Algorithm 4 sim-to-real transfer via simulated exploration (SIM2EXPLORE)

- 1: **input:** tolerance ϵ , confidence δ , budget T , Q -value function class \mathcal{F}
 - 2: $\Pi_{\text{exp}}^h \leftarrow \text{LEARNEXPOLICIES}(\mathcal{M}^{\text{sim}}, \delta, \frac{4A^3\epsilon}{H}, h)$ for all $h \in [H]$
 - 3: $\iota \leftarrow \mathcal{O}(\log_2 \frac{V_{\max} AdH}{\epsilon})$
 - 4: **for** $\ell = 1, 2, \dots, \iota$ **do**
 - 5: $\bar{\epsilon}^\ell \leftarrow 2^{-\ell}, T^\ell \leftarrow T/2\iota, \gamma^\ell \leftarrow 10V_{\max}^2(\bar{\epsilon}^\ell)^2$
 - 6: Run exploration procedure of Lemma C.3 with $\beta_\ell \leftarrow \frac{\gamma^\ell}{20V_{\max}^2 \log \frac{8H|\mathcal{F}|}{\delta}}$ to obtain $\mathfrak{D}_{\text{sim}}^\ell$
 - 7: $\hat{\pi}^\ell \leftarrow \text{EXPLOREREAL}(\{\text{unif}(\Pi_{\text{exp}}^h)\}_{h \in [H]}, T^\ell, \mathfrak{D}_{\text{sim}}^\ell, \gamma^\ell)$ (Algorithm 3)
 - 8: $\hat{V}_0^{\pi^\ell} \leftarrow$ average return running $\hat{\pi}^\ell$ in real $T^\ell/2$ times
 - 9: **return** $\hat{\pi} \leftarrow \arg \max_{\ell \in [\iota]} \hat{V}_0^{\pi^\ell}$
-

Theorem 4. Assume that one of the two conditions is met:

1. For each h , π_{exp}^h plays actions uniformly at random for $h' > h$,

$$\lambda_{\min} \left(\Lambda_{\pi_{\text{exp}, h}^h}^s \right) \geq \bar{\lambda}_{\min}, \quad (\text{B.10})$$

and

$$T \geq c \cdot \frac{V_{\max}^4 H^4 d^2 A^{2(k^*-2)} \iota \log(16H|\tilde{\mathcal{F}}|/\delta)}{\epsilon^4 \epsilon_{\text{sim}}^2},$$

for

$$k^* = \lceil \frac{\log_A \frac{64HV_{\max}}{\epsilon}}{\log_A 1/\xi} \rceil, \quad \xi = 2\sqrt{\frac{2HA}{\bar{\lambda}_{\min}} \cdot \epsilon_{\text{sim}}}$$

and

$$\begin{aligned} \tilde{\mathcal{F}} &:= \left\{ f \in \mathcal{F} : \sup_{\pi} (\mathbb{E}^{\text{sim}, \pi} [f_h(s_h, a_h) - \mathcal{T}^{\text{sim}} f_{h+1}(s_h, a_h)])^2 \right. \\ &\quad \left. \leq c \left(\log \frac{\log \frac{32H|\mathcal{F}|}{\delta}}{V_{\max} \epsilon_{\text{sim}}^2} + 1 \right) AdH V_{\max}^2 \log \frac{48d \log \frac{32H|\mathcal{F}|}{\delta}}{V_{\max} \epsilon_{\text{sim}}^2} \cdot \epsilon_{\text{sim}}^2 \right\}. \end{aligned}$$

2. $\epsilon_{\text{sim}} \leq \epsilon/16H^2$ and

$$T \geq c \cdot \frac{H^2 \iota \log \frac{16\iota}{\delta}}{\epsilon^2}.$$

Then with probability at least $1 - \delta$, Algorithm 4 returns a policy $\hat{\pi}$ such that $V_0^{\text{real}, \pi^{\text{real}, *}} - V_0^{\text{real}, \hat{\pi}} \leq \epsilon$.

Proof. We break the proof into two cases.

Case 1: $\epsilon_{\text{sim}} \geq \epsilon/16H^2$. Let $\bar{\ell} = \lfloor \log_2 \epsilon_{\text{sim}}^{-1} \rfloor$ and note that $\bar{\ell} \leq \iota$ in this case and that this is a deterministic quantity. Further, note that $\gamma^{\bar{\ell}} \in [10V_{\text{max}}^2 \epsilon_{\text{sim}}^2, 40V_{\text{max}}^2 \epsilon_{\text{sim}}^2]$ and $\bar{\epsilon}^{\bar{\ell}} \in [\epsilon_{\text{sim}}, 2\epsilon_{\text{sim}}]$. Note that by our assumption on π_{exp} and applying Lemma B.4 with $\kappa = \frac{\epsilon}{64HV_{\text{max}}}$ and $\gamma = \frac{d}{H\epsilon_{\text{sim}}}$, for any π and $\mathcal{Z}' \subseteq \mathcal{S} \times \mathcal{A}$, we have

$$w_h^{\text{real}, \pi}(\mathcal{Z}') \leq \frac{256dHV_{\text{max}}A^{k^*-2}}{\epsilon\epsilon_{\text{sim}}} \cdot w_h^{\text{real}, \pi_{\text{exp}}}(\mathcal{Z}') + \frac{\epsilon}{16HV_{\text{max}}}$$

for

$$k^* = \lceil \frac{\log_A \frac{64HV_{\text{max}}}{\epsilon}}{\log_A 1/\xi} \rceil, \quad \xi = 2\sqrt{\frac{2HA}{\lambda_{\min}}} \cdot \epsilon_{\text{sim}}.$$

By Lemma B.1, as long as $\beta^{\bar{\ell}}$ and $\gamma^{\bar{\ell}}$ satisfy

$$2V_{\text{max}}^2 \epsilon_{\text{sim}}^2 + \frac{43V_{\text{max}}^2 \beta_{\bar{\ell}}^2}{dH} \cdot \log \frac{8H|\mathcal{F}_h|}{\delta} + 6V_{\text{max}}^2 \beta_{\bar{\ell}} \sqrt{\log \frac{8H|\mathcal{F}_h|}{\delta}} \leq \gamma^{\bar{\ell}}, \quad (\text{B.11})$$

we have that with probability at least $1 - 2\delta$,

$$V_0^{\text{real}, \pi^{\text{real}, *}} - V_0^{\text{real}, \hat{\pi}^{\bar{\ell}}} \leq \frac{256dHV_{\text{max}}A^{k^*-2}}{\epsilon\epsilon_{\text{sim}}} \cdot 4H \sqrt{\frac{256V_{\text{max}}^2 \log(4H|\tilde{\mathcal{F}}^{\bar{\ell}}|/\delta)}{T^{\bar{\ell}}}} + \epsilon/4$$

where

$$\tilde{\mathcal{F}}^{\bar{\ell}} := \{f \in \mathcal{F} : \mathbb{E}^{\text{sim}, \pi_{\text{exp}}}[(f_h(s_h, a_h) - \mathcal{T}^{\text{sim}} f_{h+1}(s_h, a_h))^2] \leq 2\gamma^{\bar{\ell}}, \forall h \in [H]\}.$$

However, since $V_{\text{max}}^2 \epsilon_{\text{sim}} \leq \frac{1}{10}\gamma^{\bar{\ell}}$, and by our choice of $\beta^{\bar{\ell}} = \frac{\gamma^{\bar{\ell}}}{20V_{\text{max}}^2 \log \frac{8H|\mathcal{F}|}{\delta}}$, we see that (B.11) is met, so the conclusion holds. Note that, by Lemma C.5, we have that with probability at least $1 - \delta$:

$$\begin{aligned} \tilde{\mathcal{F}}^{\bar{\ell}} &\subseteq \left\{ f \in \mathcal{F} : \sup_{\pi} (\mathbb{E}^{\text{sim}, \pi}[f_h(s_h, a_h) - \mathcal{T}^{\text{sim}} f_{h+1}(s_h, a_h)])^2 \right. \\ &\quad \left. \leq \left(4 \log \frac{1}{\beta_{\bar{\ell}}} + 6\right) A \cdot \left[48dH \log \frac{48d}{\beta_{\bar{\ell}}^2} \cdot 2\gamma^{\bar{\ell}} + V_{\text{max}}^2 \sqrt{96dH \log \frac{48d}{\beta_{\bar{\ell}}^2} \log \frac{1}{\delta} \cdot \beta_{\bar{\ell}}}\right] \right\} \\ &\subseteq \left\{ f \in \mathcal{F} : \sup_{\pi} (\mathbb{E}^{\text{sim}, \pi}[f_h(s_h, a_h) - \mathcal{T}^{\text{sim}} f_{h+1}(s_h, a_h)])^2 \right. \\ &\quad \left. \leq c \left(\log \frac{\log \frac{8H|\mathcal{F}|}{\delta}}{V_{\text{max}}^2 \epsilon_{\text{sim}}^2} + 1 \right) AdHV_{\text{max}}^2 \log \frac{48d \log \frac{8H|\mathcal{F}|}{\delta}}{V_{\text{max}}^2 \epsilon_{\text{sim}}^2} \cdot \epsilon_{\text{sim}}^2 \right\} \\ &=: \tilde{\mathcal{F}} \end{aligned}$$

where the second inclusion follows from our setting of $\beta_{\bar{\ell}}$, and bounds on $\gamma^{\bar{\ell}}$.

Since $T^{\bar{\ell}} \leftarrow T/2\iota$, it follows that if

$$T \geq c \cdot \frac{d^2 H^4 V_{\text{max}}^4 A^{2(k^*-2)} \iota \log(4H|\tilde{\mathcal{F}}|/\delta)}{\epsilon^4 \epsilon_{\text{sim}}^2},$$

then we have that $V_0^{\text{real}, \pi^{\text{real}, *}} - V_0^{\text{real}, \hat{\pi}^{\bar{\ell}}} \leq \epsilon/2$.

Case 2: $\epsilon_{\text{sim}} \leq \epsilon/16H^2$. By Lemma B.5 and our choice of $T_{\text{sim}}^{\bar{\ell}}$, we have that with probability at least $1 - \delta$,

$$V_0^{\text{real}, *} - V_0^{\text{real}, \hat{\pi}^{\iota}} \leq 6H \left(2 \log \frac{20V_{\text{max}}^2 \log \frac{8H|\mathcal{F}|}{\delta}}{\gamma^{\iota}} + 3 \right) \cdot \sqrt{192AdH \log \frac{960dV_{\text{max}}^2 \log \frac{8H|\mathcal{F}|}{\delta}}{\gamma^{\iota}}} \cdot \gamma^{\iota} + 4H^2 \epsilon_{\text{sim}}.$$

By our choice of $\iota = \mathcal{O}(\log_2 \frac{V_{\text{max}} AdH}{\epsilon})$ and since $\gamma^{\iota} = 10V_{\text{max}}^2 (\bar{\epsilon}^{\iota})^2 = 10V_{\text{max}}^2 \cdot 2^{-2\iota}$, we can bound $V_0^{\text{real}, *} - V_0^{\text{real}, \hat{\pi}^{\iota}} \leq \epsilon/2$.

Completing the Proof. In either case, we have that with probability at least $1 - \delta$, there exists some $\hat{i} \in [\iota]$ such that $V_0^{\text{real},\star} - V_0^{\text{real},\hat{\pi}^{\hat{i}}} \leq \epsilon/2$.

Note that $V_0^{\text{real},\pi} = \mathbb{E}^{\text{real},\pi}[\sum_{h=0}^{H-1} r_h]$ and that $\sum_{h=0}^{H-1} r_h \in [0, H]$ almost surely. Consider playing π for n episodes in real and let R^i denote the total return of the i th episode. Let

$$\hat{V}_0^\pi := \frac{1}{n} \sum_{i=1}^n R^i.$$

By Hoeffding's inequality we have that, with probability at least $1 - \delta/\iota$:

$$|\hat{V}_0^\pi - V_0^{\text{real},\pi}| \leq H \sqrt{\frac{\log \frac{2\iota}{\delta}}{n}}.$$

Thus, if

$$n \geq \frac{16H^2 \log \frac{2\iota}{\delta}}{\epsilon^2},$$

we have that $|\hat{V}_0^\pi - V_0^{\text{real},\pi}| \leq \epsilon/2$. However, as we run each $\pi \in \hat{\Pi}^\ell$ $T_\ell/2 = T/2\iota$ times, and in either case we assume $T \geq \frac{c\iota H^2}{\epsilon^2} \cdot \log \frac{4\iota}{\delta}$, this will be met. Union bounding over this for all $\hat{\pi}^\ell$, we have that with probability at least $1 - \delta$:

$$V_0^{\text{real},\hat{\pi}} \geq \hat{V}_0^{\hat{\pi}} - \epsilon/4 \geq \hat{V}_0^{\hat{\pi}^{\hat{i}}} - \epsilon/4 \geq V_0^{\text{real},\hat{\pi}^{\hat{i}}} - \epsilon/2.$$

It follows that

$$V^{\text{real},\star} - V_0^{\text{real},\hat{\pi}} \leq V^{\text{real},\star} - V_0^{\text{real},\hat{\pi}^{\hat{i}}} + \epsilon/2 \leq \epsilon.$$

The result then follows from a union bound and rescaling δ . \square

Proof of Theorem 3. The argument follows analogously to the proof of Theorem 1, but using Theorem 4 in place of Theorem 2. The bound on the number of oracle calls follows from Lemma C.3 and our choice of β_ℓ . \square

Lemma B.5. *With probability at least $1 - \delta$, for some ℓ , we have*

$$\begin{aligned} V_0^{\text{sim},\star} - V_0^{\text{sim},\hat{\pi}^\ell} &\leq 6H \left(2 \log \frac{20V_{\max}^2 \log \frac{8H|\mathcal{F}|}{\delta}}{\gamma^\ell} + 3 \right) \cdot \sqrt{192AdH \log \frac{960dV_{\max}^2 \log \frac{8H|\mathcal{F}|}{\delta}}{\gamma^\ell}} \cdot \gamma^\ell, \\ V_0^{\text{real},\star} - V_0^{\text{real},\hat{\pi}^\ell} &\leq 6H \left(2 \log \frac{20V_{\max}^2 \log \frac{8H|\mathcal{F}|}{\delta}}{\gamma^\ell} + 3 \right) \cdot \sqrt{192AdH \log \frac{960dV_{\max}^2 \log \frac{8H|\mathcal{F}|}{\delta}}{\gamma^\ell}} \cdot \gamma^\ell + 4H^2 \epsilon_{\text{sim}}. \end{aligned}$$

Proof. By Lemma C.4 we have, with probability at least $1 - \delta$,

$$\begin{aligned} V_0^{\text{sim},\star} - V_0^{\text{sim},\hat{\pi}^\ell} &\leq 2H \left(2 \log \frac{1}{\beta_\ell} + 3 \right) \cdot \left[\beta_\ell \sqrt{512V_{\max}^2 A \log \frac{8H|\mathcal{F}|}{\delta}} + \sqrt{96AdH \log \frac{48d}{\beta_\ell^2}} \cdot \gamma^\ell \right. \\ &\quad \left. + \sqrt{2AV_{\max}^2 \sqrt{96dH \log \frac{48d}{\beta_\ell^2} \log \frac{2}{\delta}} \cdot \beta_\ell} \right] \\ &\leq 6H \left(2 \log \frac{20V_{\max}^2 \log \frac{8H|\mathcal{F}|}{\delta}}{\gamma^\ell} + 3 \right) \cdot \sqrt{192AdH \log \frac{960dV_{\max}^2 \log \frac{8H|\mathcal{F}|}{\delta}}{\gamma^\ell}} \cdot \gamma^\ell \end{aligned}$$

where the second inequality holds by our setting of β_ℓ .

We have

$$V_0^{\text{real},\star} - V_0^{\text{real},\hat{\pi}^t} = V_0^{\text{real},\star} - V_0^{\text{real},\hat{\pi}^{\text{sim},\star}} + V_0^{\text{real},\hat{\pi}^{\text{sim},\star}} - V_0^{\text{real},\hat{\pi}^{\text{sim},\star}} + V_0^{\text{sim},\hat{\pi}^{\text{sim},\star}} - V_0^{\text{sim},\hat{\pi}^{\text{sim},\star}} - V_0^{\text{sim},\hat{\pi}^t} + V_0^{\text{sim},\hat{\pi}^t} - V_0^{\text{real},\hat{\pi}^t}.$$

By Lemma A.3, we can bound

$$V_0^{\text{real},*} - V_0^{\text{real},\pi^{\text{sim},*}} \leq 2H^2\epsilon_{\text{sim}}$$

and by Lemma A.1 we can bound

$$V_0^{\text{real},\pi^{\text{sim},*}} - V_0^{\text{sim},\pi^{\text{sim},*}} \leq H^2\epsilon_{\text{sim}}, \quad V_0^{\text{sim},\hat{\pi}^\ell} - V_0^{\text{real},\hat{\pi}^\ell} \leq H^2\epsilon_{\text{sim}}.$$

Combining this with our bound on $V_0^{\text{sim},*} - V_0^{\text{sim},\hat{\pi}^\ell}$ gives the result. \square

C Learning in sim

In this section we provide additional supporting lemmas for our main results and in particular, we focus on linear in sim. In Appendix C.1 we provide several technical results critical to showing that sim can be utilized to restrict the version space, as is done in Theorem 4. In order to restrict the version space using sim, sufficiently rich data must be collected from sim, and in Appendix C.2 we provide results on this data collection. Finally, in Appendix C.3 we provide a procedure to compute the exploration policies in sim which we ultimately transfer to real.

In Appendices C.1 and C.2, we let hypothesis \tilde{f} and \hat{f} be defined recursively as:

$$\tilde{f}_h := \arg \min_{f_h \in \mathcal{F}_h} \frac{1}{T_{\text{sim}}} \sum_{t=1}^{T_{\text{sim}}} (f_h(\tilde{s}_h^t, \tilde{a}_h^t) - \tilde{r}_h^t - \max_{a'} \hat{f}_{h+1}(\tilde{s}_{h+1}^t, a'))^2.$$

and $\hat{f}_h \in \mathcal{F}_h$ some hypothesis satisfying

$$\frac{1}{T_{\text{sim}}} \sum_{t=1}^{T_{\text{sim}}} (\hat{f}_h(\tilde{s}_h^t, \tilde{a}_h^t) - \tilde{f}_h(\tilde{s}_h^t, \tilde{a}_h^t))^2 \leq \gamma$$

for parameter $\gamma > 0$.

In Appendix C.1 we make the following assumption on the data generating process.

Assumption 5. Consider the dataset $\mathcal{D}_{\text{sim}} = \{(\tilde{s}_0^t, \tilde{a}_0^t, \tilde{r}_0^t, \dots, \tilde{s}_{H-1}^t, \tilde{a}_{H-1}^t, \tilde{r}_{H-1}^t)\}_{t=1}^{T_{\text{sim}}}$. We assume that episode t in \mathcal{D}_{sim} was generated by playing an \mathcal{F}_{t-1} -measurable policy $\tilde{\pi}_{\text{exp}}^t$, and denote $\pi_{\text{exp}}^{\text{sim}} = \text{unif}(\{\tilde{\pi}_{\text{exp}}^t\}_{t=1}^{T_{\text{sim}}})$.

We provide a specific instantiation of $\pi_{\text{exp}}^{\text{sim}}$ in Appendix C.2. In Appendix C.3, we provide a procedure for learning a set of policies which induce full-rank covariates in sim, a crucial piece in obtaining good exploration performance in real.

C.1 Regularizing with Data from sim

Lemma C.1. With probability at least $1 - \delta$:

$$\frac{1}{T_{\text{sim}}} \sum_{t=1}^{T_{\text{sim}}} (\mathcal{T}^{\text{real}} \hat{f}_{h+1}(\tilde{s}_h^t, \tilde{a}_h^t) - \tilde{f}_h(\tilde{s}_h^t, \tilde{a}_h^t))^2 \leq 2V_{\text{max}}^2 \epsilon_{\text{sim}}^2 + \frac{512V_{\text{max}}^2}{T_{\text{sim}}} \cdot \log \frac{4|\mathcal{F}_h|}{\delta} + V_{\text{max}}^2 \sqrt{\frac{2 \log \frac{2|\mathcal{F}_h|}{\delta}}{T_{\text{sim}}}}.$$

Proof. First, note that $\mathcal{T}^{\text{real}} \hat{f}_{h+1} \in \mathcal{F}_h$ by Assumption 4.

By Azuma-Hoeffding and a union bound, we have that, with probability at least $1 - \delta$, for each $f, f' \in \mathcal{F}_h$,

$$\begin{aligned} \frac{1}{T_{\text{sim}}} \sum_{t=1}^{T_{\text{sim}}} (f_h(\tilde{s}_h^t, \tilde{a}_h^t) - f'_h(\tilde{s}_h^t, \tilde{a}_h^t))^2 &\leq \frac{1}{T_{\text{sim}}} \sum_{t=1}^{T_{\text{sim}}} \mathbb{E}^{\text{sim}, \tilde{\pi}_{\text{exp}}^t} [(f_h(\tilde{s}_h, \tilde{a}_h) - f'_h(\tilde{s}_h, \tilde{a}_h))^2] + V_{\text{max}}^2 \sqrt{\frac{2 \log |\mathcal{F}_h| / \delta}{T_{\text{sim}}}} \\ &= \mathbb{E}^{\text{sim}, \pi_{\text{exp}}^{\text{sim}}} [(f_h(s_h, a_h) - f'_h(s_h, a_h))^2] + V_{\text{max}}^2 \sqrt{\frac{2 \log |\mathcal{F}_h| / \delta}{T_{\text{sim}}}}. \end{aligned}$$

In particular, this implies that

$$\frac{1}{T_{\text{sim}}} \sum_{t=1}^{T_{\text{sim}}} (\mathcal{T}^{\text{real}} \hat{f}_{h+1}(\tilde{s}_h^t, \tilde{a}_h^t) - \tilde{f}_h(\tilde{s}_h^t, \tilde{a}_h^t))^2 \leq \mathbb{E}^{\text{sim}, \pi_{\text{exp}}^{\text{sim}}} [(\mathcal{T}^{\text{real}} \hat{f}_{h+1}(s_h, a_h) - \tilde{f}_h(s_h, a_h))^2] + V_{\text{max}}^2 \sqrt{\frac{2 \log |\mathcal{F}_h| / \delta}{T_{\text{sim}}}}.$$

We can bound

$$\begin{aligned} \mathbb{E}^{\text{sim}, \pi_{\text{exp}}^{\text{sim}}} [(\mathcal{T}^{\text{real}} \hat{f}_{h+1}(s_h, a_h) - \tilde{f}_h(s_h, a_h))^2] &\leq \underbrace{2\mathbb{E}^{\text{sim}, \pi_{\text{exp}}^{\text{sim}}} [(\mathcal{T}^{\text{real}} \hat{f}_{h+1}(s_h, a_h) - \mathcal{T}^{\text{sim}} \hat{f}_{h+1}(s_h, a_h))^2]}_{(a)} \\ &\quad + \underbrace{2\mathbb{E}^{\text{sim}, \pi_{\text{exp}}^{\text{sim}}} [(\mathcal{T}^{\text{sim}} \hat{f}_{h+1}(s_h, a_h) - \tilde{f}_h(s_h, a_h))^2]}_{(b)}. \end{aligned}$$

To bound (a), we note that

$$\begin{aligned} \mathcal{T}^{\text{real}} \hat{f}_{h+1}(s_h, a_h) - \tilde{f}_h(s_h, a_h) &= \mathbb{E}^{\text{real}}[\max_{a'} \hat{f}_{h+1}(s', a') \mid s, a] - \mathbb{E}^{\text{sim}}[\max_{a'} \hat{f}_{h+1}(s', a') \mid s, a] \\ &= \sum_{s'} (P_h^{\text{real}}(s' \mid s, a) - P_h^{\text{sim}}(s' \mid s, a)) \cdot \max_{a'} \hat{f}_{h+1}(s', a') \\ &\leq V_{\text{max}} \cdot \sum_{s'} |P_h^{\text{real}}(s' \mid s, a) - P_h^{\text{sim}}(s' \mid s, a)| \\ &\leq V_{\text{max}} \epsilon_{\text{sim}} \end{aligned}$$

where the last inequality follows under Assumption 1. This gives that (a) $\leq 2V_{\text{max}}^2 \epsilon_{\text{sim}}^2$. To bound (b), we apply Lemma 3 of [57], which gives that with probability at least $1 - \delta$,

$$(b) \leq \frac{512V_{\text{max}}^2}{T_{\text{sim}}} \cdot \log \frac{4|\mathcal{F}_h|}{\delta}.$$

Combining these with a union bound gives the result. \square

Lemma C.2. *Consider the set*

$$\hat{\mathcal{F}}_h := \left\{ f_h \in \mathcal{F}_h : \frac{1}{T_{\text{sim}}} \sum_{t=1}^{T_{\text{sim}}} (f_h(\tilde{s}_h^t, \tilde{a}_h^t) - \tilde{f}_h(\tilde{s}_h^t, \tilde{a}_h^t))^2 \leq \gamma \right\}.$$

Then with probability $1 - 2\delta$ we have

$$\hat{\mathcal{F}}_h \subseteq \left\{ f_h \in \mathcal{F}_h : \mathbb{E}^{\text{sim}, \pi_{\text{exp}}^{\text{sim}}} [(f_h(s_h, a_h) - \mathcal{T}^{\text{sim}} \hat{f}_{h+1}(s_h, a_h))^2] \leq \gamma + 18V_{\text{max}}^2 \sqrt{\frac{\log \frac{4|\mathcal{F}_h|}{\delta}}{T_{\text{sim}}}} \right\}.$$

Proof. By Azuma-Hoeffding, we have that with probability at least $1 - \delta$, for each $f_h, f'_h \in \mathcal{F}_h$,

$$\mathbb{E}^{\text{sim}, \pi_{\text{exp}}^{\text{sim}}} [(f_h(s_h, a_h) - f'_h(s_h, a_h))^2] - V_{\text{max}}^2 \sqrt{\frac{2 \log |\mathcal{F}_h| / \delta}{T_{\text{sim}}}} \leq \frac{1}{T_{\text{sim}}} \sum_{t=1}^{T_{\text{sim}}} (f_h(\tilde{s}_h^t, \tilde{a}_h^t) - f'_h(\tilde{s}_h^t, \tilde{a}_h^t))^2$$

which implies in particular that, for any $f_h \in \mathcal{F}_h$,

$$\mathbb{E}^{\text{sim}, \pi_{\text{exp}}^{\text{sim}}} [(f_h(s_h, a_h) - \tilde{f}_h(s_h, a_h))^2] - V_{\text{max}}^2 \sqrt{\frac{2 \log |\mathcal{F}_h| / \delta}{T_{\text{sim}}}} \leq \frac{1}{T_{\text{sim}}} \sum_{t=1}^{T_{\text{sim}}} (f_h(\tilde{s}_h^t, \tilde{a}_h^t) - \tilde{f}_h(\tilde{s}_h^t, \tilde{a}_h^t))^2.$$

We can write

$$\begin{aligned} \mathbb{E}^{\text{sim}, \pi_{\text{exp}}^{\text{sim}}} [(f_h(s_h, a_h) - \tilde{f}_h(s_h, a_h))^2] &= \mathbb{E}^{\text{sim}, \pi_{\text{exp}}^{\text{sim}}} [(f_h(s_h, a_h) - \mathcal{T}^{\text{sim}} \hat{f}_{h+1}(s_h, a_h))^2] + \mathbb{E}^{\text{sim}, \pi_{\text{exp}}^{\text{sim}}} [(\tilde{f}_h(s_h, a_h) - \mathcal{T}^{\text{sim}} \hat{f}_{h+1}(s_h, a_h))^2] \\ &\quad - 2\mathbb{E}^{\text{sim}, \pi_{\text{exp}}^{\text{sim}}} [(\tilde{f}_h(s_h, a_h) - \mathcal{T}^{\text{sim}} \hat{f}_{h+1}(s_h, a_h))(f_h(s_h, a_h) - \mathcal{T}^{\text{sim}} \hat{f}_{h+1}(s_h, a_h))] \\ &\geq \mathbb{E}^{\text{sim}, \pi_{\text{exp}}^{\text{sim}}} [(f_h(s_h, a_h) - \mathcal{T}^{\text{sim}} \hat{f}_{h+1}(s_h, a_h))^2] \\ &\quad - 2\mathbb{E}^{\text{sim}, \pi_{\text{exp}}^{\text{sim}}} [(\tilde{f}_h(s_h, a_h) - \mathcal{T}^{\text{sim}} \hat{f}_{h+1}(s_h, a_h))(f_h(s_h, a_h) - \mathcal{T}^{\text{sim}} \hat{f}_{h+1}(s_h, a_h))]. \end{aligned}$$

By Lemma 3 of [57], with probability at least $1 - \delta$,

$$\mathbb{E}^{\text{sim}, \pi_{\text{exp}}^{\text{sim}}}[(\tilde{f}_h(s_h, a_h) - \mathcal{T}^{\text{sim}} \hat{f}_{h+1}(s_h, a_h))^2] \leq \frac{256V_{\max}^2}{T_{\text{sim}}} \cdot \log \frac{2|\mathcal{F}_h|}{\delta}.$$

We can therefore bound the final term as

$$\begin{aligned} & \mathbb{E}^{\text{sim}, \pi_{\text{exp}}^{\text{sim}}}[(\tilde{f}_h(s_h, a_h) - \mathcal{T}^{\text{sim}} \hat{f}_{h+1}(s_h, a_h))(f_h(s_h, a_h) - \mathcal{T}^{\text{sim}} \hat{f}_{h+1}(s_h, a_h))] \\ & \leq V_{\max} \cdot \mathbb{E}^{\text{sim}, \pi_{\text{exp}}^{\text{sim}}}[\tilde{f}_h(s_h, a_h) - \mathcal{T}^{\text{sim}} \hat{f}_{h+1}(s_h, a_h)] \\ & \leq V_{\max} \cdot \sqrt{\mathbb{E}^{\text{sim}, \pi_{\text{exp}}^{\text{sim}}}[(\tilde{f}_h(s_h, a_h) - \mathcal{T}^{\text{sim}} \hat{f}_{h+1}(s_h, a_h))^2]} \\ & \leq V_{\max} \cdot \sqrt{\frac{256V_{\max}^2}{T_{\text{sim}}} \cdot \log \frac{2|\mathcal{F}_h|}{\delta}}. \end{aligned}$$

Altogether then we have shown that, for any $f_h \in \mathcal{F}_h$, with probability at least $1 - 2\delta$:

$$\frac{1}{T_{\text{sim}}} \sum_{t=1}^{T_{\text{sim}}} (f_h(\tilde{s}_h^t, \tilde{a}_h^t) - \tilde{f}_h(\tilde{s}_h^t, \tilde{a}_h^t))^2 \geq \mathbb{E}^{\text{sim}, \pi_{\text{exp}}^{\text{sim}}}[(f_h(s_h, a_h) - \mathcal{T}^{\text{sim}} \hat{f}_{h+1}(s_h, a_h))^2] - 18V_{\max}^2 \sqrt{\frac{\log 2|\mathcal{F}_h|/\delta}{T_{\text{sim}}}}.$$

Thus, if

$$\frac{1}{T_{\text{sim}}} \sum_{t=1}^{T_{\text{sim}}} (f_h(\tilde{s}_h^t, \tilde{a}_h^t) - \tilde{f}_h(\tilde{s}_h^t, \tilde{a}_h^t))^2 \leq \gamma,$$

then

$$\mathbb{E}^{\text{sim}, \pi_{\text{exp}}^{\text{sim}}}[(f_h(s_h, a_h) - \mathcal{T}^{\text{sim}} \hat{f}_{h+1}(s_h, a_h))^2] \leq \gamma + 18V_{\max}^2 \sqrt{\frac{\log 2|\mathcal{F}_h|/\delta}{T_{\text{sim}}}}.$$

The result follows from a union bound. \square

C.2 Data Collection with CoverTraj

Lemma C.3. Consider running the COVERTRAJ algorithm of [65] for each $h \in [H]$ with parameters $m \leftarrow \lceil \log_2 1/\beta \rceil$ and $\gamma_i \leftarrow 2^i \cdot \beta$ for some $\beta \in [0, 1]$, and with REGMIN set to the policy optimization oracle of Oracle 4.2. Then this procedure collects

$$T_{\text{sim}} := H \cdot \sum_{i=1}^m \left\lceil \frac{24d}{2^i \cdot \beta^2} \log \frac{48d}{2^i \cdot \beta^2} \right\rceil$$

episodes, calls the policy optimization oracle at most T_{sim} times, and produces covariates $\Lambda_{h,i}$ and sets $\mathcal{X}_{h,i}$ such that, for each $i \in [m]$,

$$\sup_{\pi} w_h^{\text{sim}, \pi}(\mathcal{X}_{h,i}) \leq 2^{-i+1} \quad \text{and} \quad \phi^\top \Lambda_{h,i}^{-1} \phi \leq 2^{2i} \cdot \beta^2, \forall \phi \in \mathcal{X}_{h,i},$$

and $\sup_{\pi} w_h^{\text{sim}, \pi}(\mathcal{B}^d \setminus \cup_{i=1}^m \mathcal{X}_{h,i}) \leq \beta$. Furthermore, we have

$$\frac{12dH}{\beta^2} \leq T_{\text{sim}} \leq \frac{48dH}{\beta^2} \log \frac{48d}{\beta^2}.$$

Proof. Instantiating REGMIN with the oracle of Oracle 4.2, we have that Definition 5.1 of [65] is met with $C_1 = C_2 = 0$. Therefore, we have that at each stage i we collect exactly (using the precise form for K_i given in the appendix of [65])

$$K_i = \lceil 2^i \cdot \frac{24d}{\gamma_i^2} \log \frac{48 \cdot 2^i d}{\gamma_i^2} \rceil$$

episodes. The result then follows by Theorem 3 of [65]. \square

Lemma C.4. Consider running the procedure of Lemma C.3 to collect data. Then with probability at least $1 - 2\delta$, we have

$$\begin{aligned} V_0^{\text{sim},\star} - V_0^{\text{sim},\pi^{\hat{f}}} &\leq 2H \left(2 \log \frac{1}{\beta} + 3 \right) \cdot \left[\beta \sqrt{512 V_{\max}^2 A \log(4H|\mathcal{F}|/\delta)} + \sqrt{96AdH \log \frac{48d}{\beta^2}} \cdot \gamma \right. \\ &\quad \left. + \sqrt{2AV_{\max}^2 \sqrt{96dH \log \frac{48d}{\beta^2} \log \frac{1}{\delta}} \cdot \beta} \right]. \end{aligned}$$

Proof. By Lemma A.4:

$$\begin{aligned} V_0^{\text{sim},\star} - V_0^{\text{sim},\pi^{\hat{f}}} &\leq \max_{\pi \in \{\pi^{\hat{f}}, \pi^{\text{sim},\star}\}} \sum_{h=0}^{H-1} 2 \left| \mathbb{E}^{\text{sim},\pi} [\hat{f}_h(s_h, a_h) - \mathcal{T}^{\text{sim}} \hat{f}_{h+1}(s_h, a_h)] \right| \\ &\leq \max_{\pi \in \{\pi^{\hat{f}}, \pi^{\text{sim},\star}\}} \sum_{h=0}^{H-1} 2 \mathbb{E}^{\text{sim},\pi} [|\hat{f}_h(s_h, a_h) - \mathcal{T}^{\text{sim}} \hat{f}_{h+1}(s_h, a_h)|]. \end{aligned}$$

Denote $g(z_h) := |\hat{f}_h(s_h, a_h) - \mathcal{T}^{\text{sim}} \hat{f}_{h+1}(s_h, a_h)|$ and $\mathbf{\Lambda}_{h-1} = \sum_{i=1}^m \mathbf{\Lambda}_{h,i} + I$, for $\mathbf{\Lambda}_{h,i}$ collected as in Lemma C.3, and note that

$$\begin{aligned} \mathbb{E}^{\text{sim},\pi} [g(z_h)] &= \mathbb{E}^{\text{sim},\pi} \left[\int g(z) dP_h^\pi(z \mid z_{h-1}) \right] \\ &= \mathbb{E}^{\text{sim},\pi} \left[\int \int g(z) \pi(a \mid s) d\text{ad} \boldsymbol{\mu}_{h-1}^s(s)^\top \boldsymbol{\phi}^s(z_{h-1}) \right] \\ &= \mathbb{E}^{\text{sim},\pi} \left[\int \int g(z) \pi(a \mid s) d\text{ad} \boldsymbol{\mu}_{h-1}^s(s)^\top \mathbf{\Lambda}_{h-1}^{1/2} \mathbf{\Lambda}_{h-1}^{-1/2} \boldsymbol{\phi}^s(z_{h-1}) \right] \quad (\text{C.1}) \\ &\leq \mathbb{E}^{\text{sim},\pi} [\| \int \int g(z) \pi(a \mid s) d\text{ad} \boldsymbol{\mu}_{h-1}^s(s) \|_{\mathbf{\Lambda}_{h-1}} \cdot \| \boldsymbol{\phi}^s(z_{h-1}) \|_{\mathbf{\Lambda}_{h-1}^{-1}}] \\ &= \| \int \int g(z) \pi(a \mid s) d\text{ad} \boldsymbol{\mu}_{h-1}^s(s) \|_{\mathbf{\Lambda}_{h-1}} \cdot \mathbb{E}^{\text{sim},\pi} [\| \boldsymbol{\phi}^s(z_{h-1}) \|_{\mathbf{\Lambda}_{h-1}^{-1}}]. \end{aligned}$$

We bound each of these terms separately. First, we have

$$\begin{aligned} \mathbb{E}^{\text{sim},\pi} [\| \boldsymbol{\phi}^s(z_{h-1}) \|_{\mathbf{\Lambda}_{h-1}^{-1}}] &\leq \sum_{i=1}^m \max_{\boldsymbol{\phi} \in \mathcal{X}_{h-1,i}} \| \boldsymbol{\phi} \|_{\mathbf{\Lambda}_{h-1}^{-1}} \cdot \sup_{\pi} \mathbb{E}^{\text{sim},\pi} [\mathbb{I}\{\boldsymbol{\phi}^s(z_{h-1}) \in \mathcal{X}_{h-1,i}\}] \\ &\quad + \max_{\boldsymbol{\phi} \in \mathcal{B}^d \setminus \cup_{i=1}^m \mathcal{X}_{h-1,i}} \| \boldsymbol{\phi} \|_{\mathbf{\Lambda}_{h-1}^{-1}} \cdot \sup_{\pi} \mathbb{E}^{\text{sim},\pi} [\mathbb{I}\{\boldsymbol{\phi}^s(z_{h-1}) \in \mathcal{X}_{h-1,i}\}] \\ &\stackrel{(a)}{\leq} \sum_{i=1}^{\ell} \gamma_i \cdot 2^{-i+1} + \beta \\ &\leq (2m+1)\beta \end{aligned}$$

where (a) follows from Lemma C.3 and since $\| \boldsymbol{\phi} \|_{\mathbf{\Lambda}_{h-1}^{-1}} \leq 1$ always.

We turn now to bounding the first term. Note that

$$\begin{aligned}
& \left\| \int \int g(z) \pi(a | s) d\mathbf{a} d\boldsymbol{\mu}_{h-1}^s(s) \right\|_{\boldsymbol{\Lambda}_{h-1}} \\
&= \sqrt{\sum_{t=1}^{T_{\text{sim}}} \left(\int \int g(z) \pi(a | s) d\mathbf{a} d\boldsymbol{\mu}_{h-1}^s(s) \right)^\top \boldsymbol{\phi}_{h-1}^t }^2 \\
&= \sqrt{\sum_{t=1}^{T_{\text{sim}}} \mathbb{E} \pi[g(z_h) | z_{h-1}^t]^2} \\
&\leq \sqrt{\sum_{t=1}^{T_{\text{sim}}} \mathbb{E} \pi[g(z_h)^2 | z_{h-1}^t]} \\
&\stackrel{(a)}{\leq} \sqrt{A \cdot \sum_{t=1}^{T_{\text{sim}}} \mathbb{E}^{\pi_{\text{exp}}^{h-1,t}} [g(z_h)^2 | z_{h-1}^t]} \\
&= \sqrt{A \cdot \sum_{t=1}^{T_{\text{sim}}} \mathbb{E}^{\pi_{\text{exp}}^{h-1,t}} [(\hat{f}_h(s_h, a_h) - \mathcal{T}^{\text{sim}} \hat{f}_{h+1}(s_h, a_h))^2 | z_{h-1}^t]} \\
&\leq \sqrt{2A \cdot \sum_{t=1}^{T_{\text{sim}}} \mathbb{E}^{\pi_{\text{exp}}^{h-1,t}} [(\tilde{f}_h(s_h, a_h) - \mathcal{T}^{\text{sim}} \hat{f}_{h+1}(s_h, a_h))^2 | z_{h-1}^t] + 2A \cdot \sum_{t=1}^{T_{\text{sim}}} \mathbb{E}^{\pi_{\text{exp}}^{h-1,t}} [(\tilde{f}_h(s_h, a_h) - \hat{f}_h(s_h, a_h))^2 | z_{h-1}^t]} \\
&\stackrel{(b)}{\leq} \sqrt{512V_{\text{max}}^2 A \log(4H|\mathcal{F}|/\delta) + 2A \cdot \sum_{t=1}^{T_{\text{sim}}} \mathbb{E}^{\pi_{\text{exp}}^{h-1,t}} [(\tilde{f}_h(s_h, a_h) - \hat{f}_h(s_h, a_h))^2 | z_{h-1}^t]}
\end{aligned}$$

where (a) uses the fact that $\pi_{\text{exp}}^{h-1,t}$ plays actions randomly at step h and (b) holds with probability at least $1 - \delta$ by Lemma C.6. By Azuma-Hoeffding, we have with probability $1 - \delta$:

$$\begin{aligned}
\sum_{t=1}^{T_{\text{sim}}} \mathbb{E}^{\pi_{\text{exp}}^{h-1,t}} [(\tilde{f}_h(s_h, a_h) - \hat{f}_h(s_h, a_h))^2 | z_{h-1}^t] &\leq \sum_{t=1}^{T_{\text{sim}}} (\tilde{f}_h(s_h^t, a_h^t) - \hat{f}_h(s_h^t, a_h^t))^2 + \sqrt{2V_{\text{max}}^4 T_{\text{sim}} \log 1/\delta} \\
&\leq T_{\text{sim}} \gamma + \sqrt{2V_{\text{max}}^4 T_{\text{sim}} \log 1/\delta}
\end{aligned}$$

where the last inequality follows from the definition of \hat{f}_h .

Altogether then we have shown that, with probability at least $1 - 2\delta$:

$$V_0^{\text{sim}, \star} - V_0^{\text{sim}, \pi^{\hat{f}}} \leq 2H(2m+1)\beta \cdot \sqrt{512V_{\text{max}}^2 A \log(4H|\mathcal{F}|/\delta) + 2AT_{\text{sim}}\gamma + 2AV_{\text{max}}^2 \sqrt{2T_{\text{sim}} \log 1/\delta}}.$$

Using that $T_{\text{sim}} \leq \frac{48dH}{\beta^2} \log \frac{48d}{\beta^2}$ as given in Lemma C.3, we can bound this as

$$\begin{aligned}
&\leq 2H(2m+1) \left[\beta \sqrt{512V_{\text{max}}^2 A \log(4H|\mathcal{F}|/\delta)} + \sqrt{96AdH \log \frac{48d}{\beta^2}} \cdot \gamma \right. \\
&\quad \left. + \sqrt{2AV_{\text{max}}^2 \sqrt{96dH \log \frac{48d}{\beta^2} \log \frac{1}{\delta}} \cdot \beta} \right].
\end{aligned}$$

The result follows. \square

Lemma C.5. Assume that

$$\mathbb{E}^{\text{sim}, \pi_{\text{exp}}^{\text{sim}}} [(f_h(s_h, a_h) - \mathcal{T}^{\text{sim}} f_{h+1}(s_h, a_h))^2] \leq \gamma.$$

Then this implies that, with probability at least $1 - \delta$,

$$\begin{aligned} & \sup_{\pi} (\mathbb{E}^{\text{sim}, \pi} [f_h(s_h, a_h) - \mathcal{T}^{\text{sim}} f_{h+1}(s_h, a_h)])^2 \\ & \leq \left(4 \log \frac{1}{\beta} + 6\right) A \cdot \left[48d \log \frac{48d}{\beta^2} \cdot \gamma + V_{\max} \sqrt{96d \log \frac{48d}{\beta^2} \log \frac{1}{\delta} \cdot \beta}\right]. \end{aligned}$$

Therefore,

$$\begin{aligned} & \{f \in \mathcal{F} : \mathbb{E}^{\text{sim}, \pi_{\text{exp}}^{\text{sim}}} [(f_h(s_h, a_h) - \mathcal{T}^{\text{sim}} f_{h+1}(s_h, a_h))^2] \leq \gamma\} \\ & \subseteq \left\{f \in \mathcal{F} : \sup_{\pi} (\mathbb{E}^{\text{sim}, \pi} [f_h(s_h, a_h) - \mathcal{T}^{\text{sim}} f_{h+1}(s_h, a_h)])^2 \right. \\ & \quad \left. \leq \left(4 \log \frac{1}{\beta} + 6\right) A \cdot \left[48dH \log \frac{48d}{\beta^2} \cdot \gamma + V_{\max}^2 \sqrt{96dH \log \frac{48d}{\beta^2} \log \frac{1}{\delta} \cdot \beta}\right]\right\}. \end{aligned}$$

Proof. We follow a similar argument as the proof of Lemma C.4. Denoting $g(z_h) := f_h(s_h, a_h) - \mathcal{T}^{\text{sim}} f_{h+1}(s_h, a_h)$, by the same calculation as (C.1) we have

$$\mathbb{E}^{\text{sim}, \pi} [g(z_h)] \leq \left\| \int \int g(z) \pi(a | s) \text{d}a \text{d}\mu_{h-1}^s(s) \right\|_{\Lambda_{h-1}} \cdot \mathbb{E}^{\text{sim}, \pi} [\|\phi^s(z_{h-1})\|_{\Lambda_{h-1}^{-1}}]$$

and as in the proof of Lemma C.4, we can bound

$$\mathbb{E}^{\text{sim}, \pi} [\|\phi^s(z_{h-1})\|_{\Lambda_{h-1}^{-1}}] \leq (2m+1)\beta$$

and

$$\left\| \int \int g(z) \pi(a | s) \text{d}a \text{d}\mu_{h-1}^s(s) \right\|_{\Lambda_{h-1}} \leq \sqrt{A \cdot \sum_{t=1}^{T_{\text{sim}}} \mathbb{E}^{\pi_{\text{exp}}^{h-1, t}} [(f_h(s_h, a_h) - \mathcal{T}^{\text{sim}} f_{h+1}(s_h, a_h))^2 | z_{h-1}^t]}$$

By Azuma-Hoeffding, with probability at least $1 - \delta$ we can then bound

$$\begin{aligned} \sum_{t=1}^{T_{\text{sim}}} \mathbb{E}^{\pi_{\text{exp}}^{h-1, t}} [(f_h(s_h, a_h) - \mathcal{T}^{\text{sim}} f_{h+1}(s_h, a_h))^2 | z_{h-1}^t] & \leq T_{\text{sim}} \cdot \mathbb{E}^{\pi_{\text{exp}}^{\text{sim}}} [(f_h(s_h, a_h) - \mathcal{T}^{\text{sim}} f_{h+1}(s_h, a_h))^2] \\ & \quad + \sqrt{2V_{\max}^4 T_{\text{sim}} \log 1/\delta} \\ & \leq T_{\text{sim}} \gamma + \sqrt{2V_{\max}^4 T_{\text{sim}} \log 1/\delta} \end{aligned}$$

where the last inequality follows by assumption, and where $\pi_{\text{exp}}^{\text{sim}} = \text{unif}(\{\pi_{\text{exp}}^{h-1, t}\}_{t=1}^{T_{\text{sim}}})$. Altogether then, for all π , we have

$$\mathbb{E}^{\text{sim}, \pi} [f_h(s_h, a_h) - \mathcal{T}^{\text{sim}} f_{h+1}(s_h, a_h)] \leq (2m+1)\beta \cdot \sqrt{AT_{\text{sim}}\gamma + AV_{\max}^2 \sqrt{2T_{\text{sim}} \log 1/\delta}}.$$

Using that $T_{\text{sim}} \leq \frac{48dH}{\beta^2} \log \frac{48d}{\beta^2}$ as given in Lemma C.3, we can bound this as

$$\leq (2m+1) \sqrt{48AdH \log \frac{48d}{\beta^2} \cdot \gamma} + (2m+1) \sqrt{AV_{\max}^2 \sqrt{96dH \log \frac{48d}{\beta^2} \log \frac{1}{\delta} \cdot \beta}}.$$

The result follows from some algebra. \square

Lemma C.6. With probability at least $1 - \delta$, for each $h \in [H]$ simultaneously, we have

$$\sum_{t=1}^{T_{\text{sim}}} \mathbb{E}^{\text{sim}, \pi_{\text{exp}}^{h-1, t}} [(\tilde{f}_h(s_h, a_h) - \mathcal{T}^{\text{sim}} \hat{f}_{h+1}(s_h, a_h))^2 | s_{h-1}^t, a_{h-1}^t] \leq 256V_{\max}^2 \log(4H|\mathcal{F}|/\delta).$$

Proof. This follows from Lemma 3 of [57]. \square

Algorithm 5 Learn Exploration Policies in \mathcal{M}^{sim} (LEARNEXPOLICIES)

```
1: input: environment  $\mathcal{M}$ , confidence  $\delta$ , regularization  $\zeta$ , step  $h$ 
2:  $\mathbb{A}_{\mathcal{R}} \leftarrow$  policy optimization oracle of Oracle 4.2
3: for  $j = 1, 2, 3, \dots, \mathcal{O}(\log_2(\frac{d}{\zeta} \cdot \log \frac{1}{\delta} + \zeta^{-9} \cdot \log^{3/2} \frac{1}{\delta}))$  do
4:    $N_j \leftarrow \lceil 2^{j/3} \rceil - 1, K_j \leftarrow \lceil 2^{2j/3} \rceil, T_j \leftarrow (N_j + 1)K_j, \delta_j \leftarrow \frac{\delta}{4j^2}$ 
   // DYNAMICOEED algorithm from [64]
5:    $\Sigma_j, \Pi_j \leftarrow \text{DYNAMICOEED}(\Phi, N_j, K_j, \delta_j, \mathbb{A}_{\mathcal{R}})$  for  $\Phi(\Lambda_h) \leftarrow \text{tr}((\Lambda_h + \zeta \cdot I)^{-1})$ 
6:   if  $\lambda_{\min}(\Sigma_j) \geq 12544d \log \frac{4+64T_j}{\delta}$  and  $T_j \geq c \cdot \zeta^{-9} \cdot \log^{3/2} \frac{jT_j}{\delta}$  then
7:     break
8: return  $\Pi_j$ 
```

C.3 Learning Full-Rank Policies

We consider running the MINEIG algorithm (Algorithm 6) of [64] in sim. For a fixed h , we instantiate the setting of Appendix C of [64] with $\psi(\tau) = \phi(s_h, a_h)\phi(s_h, a_h)^\top$, $D = 1$, and $\mathbb{A}_{\mathcal{R}}$ the policy optimization oracle of Oracle 4.2 (and so $C_{\mathcal{R}} = 0$), and set $N = 1$ for MINEIG. We note that this algorithm is computationally efficient, given a policy optimization oracle.

Lemma C.7. For $\mathcal{M} \leftarrow \mathcal{M}^{\text{sim}}$, Algorithm 5 will call Oracle 4.2 at most $\tilde{\mathcal{O}}(\frac{d}{\zeta} \cdot \log \frac{1}{\delta} + \zeta^{-9} \cdot \log^{3/2} \frac{1}{\delta})$ times, and with probability at least $1 - \delta$, under Assumption 3 and if $\zeta \leq \frac{\lambda_{\min}^*}{4d}$, will return policies Π such that

$$\lambda_{\min} \left(\frac{1}{|\Pi|} \sum_{\pi \in \Pi} \Lambda_{\pi, h}^s \right) \geq \frac{\lambda_{\min}^*}{8d} \quad (\text{C.2})$$

and each $\pi \in \Pi$ plays actions randomly for $h' > h$.

Proof. We first argue that, if $\zeta \leq \frac{\lambda_{\min}^*}{4d}$, then with probability at least $1 - \delta$, (C.2) holds. Let \mathcal{E} denote the success event of each call to DYNAMICOEED, and note that by our choice of δ_j , we have $\mathbb{P}[\mathcal{E}] \geq 1 - \delta/2$. Let j^* denote the minimal value of j such that

$$\frac{\lambda_{\min}^*}{4d} T_j \geq 12544d \log \frac{4+64T_j}{\delta} \quad \text{and} \quad T_j \geq c \cdot \zeta^{-9} \cdot \log^{3/2} \frac{jT_j}{\delta}. \quad (\text{C.3})$$

By Lemma C.4 of [64] and if $\zeta \leq \frac{\lambda_{\min}^*}{4d}$, we then have that, on \mathcal{E} , $\lambda_{\min}(\Sigma_{j^*}) \geq \frac{\lambda_{\min}^*}{4d} T_{j^*}$, which implies that the termination criteria of Algorithm 5 will be met. By Lemma C.5 of [64], it follows that with probability at least $1 - \delta/2$, we have $\lambda_{\min}(\frac{1}{|\Pi_{j^*}|} \sum_{\pi \in \Pi_{j^*}} \Lambda_{\pi, h}^s) \geq \frac{\lambda_{\min}^*}{8d}$ (since $T_{j^*} = |\Pi_{j^*}|$), the desired conclusion.

Assume that Algorithm 5 terminates for some $j < j^*$. This implies that $\frac{\lambda_{\min}^*}{4d} T_j < 12544d \log \frac{4+64T_j}{\delta}$. However, in this case, we then have that

$$\lambda_{\min}(\Sigma_j) \geq 12544d \log \frac{4+64T_j}{\delta} \geq \frac{\lambda_{\min}^*}{4d} T_j.$$

From Lemma C.5 of [64], it then follows that with probability at least $1 - \delta/2$, we have $\lambda_{\min}(\frac{1}{|\Pi_j|} \sum_{\pi \in \Pi_j} \Lambda_{\pi, h}^s) \geq \frac{\lambda_{\min}^*}{8d}$.

It follows that, assuming T_j is large enough that (C.3) is met, and we are in the case when $\zeta \leq \frac{\lambda_{\min}^*}{4d}$ holds, then Algorithm 5 will terminate and return a set of policies satisfying (C.2), with probability at least $1 - \delta$. Note that $T_j = \mathcal{O}(2^j)$. Given that Algorithm 5 does not terminate until $j = \mathcal{O}(\log_2(\frac{d}{\zeta} \cdot \log \frac{1}{\delta} + \zeta^{-9} \cdot \log^{3/2} \frac{1}{\delta})) \geq \mathcal{O}(\log_2(\frac{d^2}{\lambda_{\min}^*} \cdot \log \frac{1}{\delta} + \zeta^{-9} \cdot \log^{3/2} \frac{1}{\delta}))$, we will have that T_j will be large enough that (C.3) is met, if $\zeta \leq \frac{\lambda_{\min}^*}{4d}$. The proof then follows since DYNAMICOEED calls Oracle 4.2 at most T_j times at round j , and the total sum of T_j is bounded as $\tilde{\mathcal{O}}(\frac{d}{\zeta} \cdot \log \frac{1}{\delta} + \zeta^{-9} \cdot \log^{3/2} \frac{1}{\delta})$ by the maximum of j , and since the actions chosen by $\pi \in \Pi$ for $h' > h$ are irrelevant for the operation of DYNAMICOEED, so they can be set to random. \square

D Lower Bound Proofs

D.1 Proof of Propositions 1, 3 and 4

Construction. Consider the following variation of the combination lock. We let the action space $\mathcal{A} = \{1, 2\}$, and assume there are two states, $\mathcal{S} = \{s_1, s_2\}$, and horizon H . We start in state s_1 . The sim dynamics are given as:

$$\begin{aligned} \forall h < H-1 : \quad & P_h^{\text{sim}}(s_1 | s_1, a_1) = 1, \quad P_h^{\text{sim}}(s_2 | s_1, a_2) = 1 \\ & P_{H-1}^{\text{sim}}(s_1 | s_1, a_1) = P_{H-1}^{\text{sim}}(s_2 | s_1, a_1) = P_{H-1}^{\text{sim}}(s_1 | s_1, a_2) = P_{H-1}^{\text{sim}}(s_2 | s_1, a_2) = 1/2 \\ \forall h \in [H] : \quad & P_h^{\text{sim}}(s_2 | s_2, a) = 1, a \in \{a_1, a_2\}. \end{aligned}$$

We define two real instances, $\mathcal{M}_1 := \mathcal{M}^{\text{real},1}$ and $\mathcal{M}_2 := \mathcal{M}^{\text{real},2}$, where for both we have:

$$\begin{aligned} \forall h < H-1 : \quad & P_h^{\text{real}}(s_1 | s_1, a_1) = 1, \quad P_h^{\text{real}}(s_2 | s_1, a_2) = 1 \\ \forall h \in [H] : \quad & P_h^{\text{real}}(s_2 | s_2, a) = 1, a \in \{a_1, a_2\} \end{aligned}$$

for \mathcal{M}_1 :

$$\begin{aligned} P_{H-1}^{\text{real}}(s_1 | s_1, a_1) &= 1/2 + \epsilon_{\text{sim}}, P_{H-1}^{\text{real}}(s_2 | s_1, a_1) = 1/2 - \epsilon_{\text{sim}}, \\ P_{H-1}^{\text{real}}(s_1 | s_1, a_2) &= 1/2 - \epsilon_{\text{sim}}, P_{H-1}^{\text{real}}(s_2 | s_1, a_2) = 1/2 + \epsilon_{\text{sim}}, \end{aligned}$$

and for \mathcal{M}_2 :

$$\begin{aligned} P_{H-1}^{\text{real}}(s_1 | s_1, a_1) &= 1/2 - \epsilon_{\text{sim}}, P_{H-1}^{\text{real}}(s_2 | s_1, a_1) = 1/2 + \epsilon_{\text{sim}}, \\ P_{H-1}^{\text{real}}(s_1 | s_1, a_2) &= 1/2 + \epsilon_{\text{sim}}, P_{H-1}^{\text{real}}(s_2 | s_1, a_2) = 1/2 - \epsilon_{\text{sim}}. \end{aligned}$$

Note then that $\mathcal{M}_1, \mathcal{M}_2$, and sim only differ at step $H-1$ in state s_1 . Furthermore, it is easy to see that both \mathcal{M}_1 and \mathcal{M}_2 satisfy Assumption 1 with misspecification ϵ_{sim} . It is easy to see that Assumption 2 holds as well with $d = 4$ since this is a tabular MDP, and furthermore Assumption 3 also holds with $\lambda_{\min}^* = 1/4$. We define the reward function as (note that this is deterministic, and the same for all instances):

$$\begin{aligned} \forall h \in [H] : \quad & r_h(s_1, a_2) = 1/2 + \epsilon_{\text{sim}}(1/2 - h/4H) \\ & r_H(s_1, a) = 1, a \in \{a_1, a_2\}, \end{aligned}$$

and all other rewards are taken to be 0.

In sim, we see that the optimal policy always plays a_2 . In both \mathcal{M}_1 and \mathcal{M}_2 , the optimal policy plays a_1 for all $h < H-1$, for \mathcal{M}_1 plays a_1 at $H-1$, and for \mathcal{M}_2 plays a_2 at $H-1$. Note that for both \mathcal{M}_1 and \mathcal{M}_2 , we have $V_0^* = 1/2 + \epsilon_{\text{sim}}$.

The most natural choice of \mathcal{F} would be the set of all tabular Q -value functions, however, this set is infinite, and would require a covering argument to incorporate. For simplicity, consider \mathcal{F}_H the set of functions mapping to $\{0, 1\}$, and \mathcal{F}_h the set of functions mapping to a finite set containing $\{0, 1/2 - \epsilon_{\text{sim}}, 1/2 + \epsilon_{\text{sim}}\} \cup \{1/2 + \epsilon_{\text{sim}}(1/2 - h'/4H)\}_{h'=0}^H$. Note that such a set satisfies Assumption 4 and we can construct it such that $\log |\mathcal{F}| \leq \mathcal{O}(H)$.

Lower Bound for Direct Policy Transfer (Proposition 3). We consider direct sim2real transfer with randomized exploration. In particular, as noted, the optimal policy in sim always plays a_2 , so we consider the ζ -greedy policy that at every state plays a_2 with probability $1 - \zeta$, and plays unif($\{a_1, a_2\}$) with probability ζ . Denote this policy as $\hat{\pi}$. We then wish to lower bound:

$$\inf_{\hat{\pi}} \sup_{i \in \{1,2\}} \mathbb{E}^{\mathcal{M}_i, \hat{\pi}} [V_0^{\mathcal{M}_i, \star} - V_0^{\mathcal{M}_i, \hat{\pi}}]$$

after running our procedure for T episodes. Note that on \mathcal{M}_1 , regardless of the actions $\hat{\pi}$ chooses in other states, we have

$$V_0^{\mathcal{M}_1, \star} - V_0^{\mathcal{M}_1, \hat{\pi}} \geq \frac{\epsilon_{\text{sim}}}{2} (1 - \hat{\pi}_{H-1}(a_1 | s_1)),$$

since the only way $\hat{\pi}$ can achieve a reward of $1/2 + \epsilon_{\text{sim}}$ is by playing a_1 in s_1 at step $H - 1$, and all other sequences of actions obtain a reward of at most $1/2 + \epsilon_{\text{sim}}/2$. Similarly for \mathcal{M}_2 we have

$$V_0^{\mathcal{M}_2, \star} - V_0^{\mathcal{M}_2, \hat{\pi}} \geq \frac{\epsilon_{\text{sim}}}{2} (1 - \hat{\pi}_{H-1}(a_2 \mid s_1)).$$

Using this, and replacing the max over $i \in \{1, 2\}$ with the average of them, we obtain

$$\begin{aligned} \inf_{\hat{\pi}} \sup_{i \in \{1, 2\}} \mathbb{E}^{\mathcal{M}_i, \hat{\pi}} [V_0^{\mathcal{M}_i, \star} - V_0^{\mathcal{M}_i, \hat{\pi}}] &\geq \inf_{\hat{\pi}} \frac{1}{2} \mathbb{E}^{\mathcal{M}_1, \hat{\pi}} \left[\frac{\epsilon_{\text{sim}}}{2} (1 - \hat{\pi}_{H-1}(a_1 \mid s_1)) \right] + \frac{1}{2} \mathbb{E}^{\mathcal{M}_2, \hat{\pi}} \left[\frac{\epsilon_{\text{sim}}}{2} (1 - \hat{\pi}_{H-1}(a_2 \mid s_1)) \right] \\ &= \frac{\epsilon_{\text{sim}}}{2} \left[1 - \frac{1}{2} \cdot \sup_{\hat{\pi}} \left(\mathbb{E}^{\mathcal{M}_1, \hat{\pi}} [\hat{\pi}_{H-1}(a_1 \mid s_1)] + \mathbb{E}^{\mathcal{M}_2, \hat{\pi}} [\hat{\pi}_{H-1}(a_2 \mid s_1)] \right) \right]. \end{aligned}$$

Since $\hat{\pi}_{H-1}(a_1 \mid s_1) = 1 - \hat{\pi}_{H-1}(a_2 \mid s_1)$, we have

$$\begin{aligned} \mathbb{E}^{\mathcal{M}_1, \hat{\pi}} [\hat{\pi}_{H-1}(a_1 \mid s_1)] + \mathbb{E}^{\mathcal{M}_2, \hat{\pi}} [\hat{\pi}_{H-1}(a_2 \mid s_1)] &= 1 + \mathbb{E}^{\mathcal{M}_1, \hat{\pi}} [\hat{\pi}_{H-1}(a_1 \mid s_1)] - \mathbb{E}^{\mathcal{M}_2, \hat{\pi}} [\hat{\pi}_{H-1}(a_1 \mid s_1)] \\ &\leq 1 + \text{TV}(\mathbb{P}^{\mathcal{M}_1, \hat{\pi}}, \mathbb{P}^{\mathcal{M}_2, \hat{\pi}}) \\ &\leq 1 + \sqrt{\frac{1}{2} \text{KL}(\mathbb{P}^{\mathcal{M}_1, \hat{\pi}} \parallel \mathbb{P}^{\mathcal{M}_2, \hat{\pi}})} \end{aligned}$$

where TV denotes the total-variation distance, KL the KL-divergence, and the last inequality follows from Pinsker's inequality. We therefore have

$$\inf_{\hat{\pi}} \sup_{i \in \{1, 2\}} \mathbb{E}^{\mathcal{M}_i, \hat{\pi}} [V_0^{\mathcal{M}_i, \star} - V_0^{\mathcal{M}_i, \hat{\pi}}] \geq \frac{\epsilon_{\text{sim}}}{4} \left(1 - \sqrt{\frac{1}{2} \text{KL}(\mathbb{P}^{\mathcal{M}_1, \hat{\pi}} \parallel \mathbb{P}^{\mathcal{M}_2, \hat{\pi}})} \right).$$

Now note that, since \mathcal{M}_1 and \mathcal{M}_2 only differ at state s_1 and step $H - 1$, we have

$$\begin{aligned} \text{KL}(\mathbb{P}^{\mathcal{M}_1, \hat{\pi}} \parallel \mathbb{P}^{\mathcal{M}_2, \hat{\pi}}) &= \mathbb{E}^{\mathcal{M}_1, \hat{\pi}} [T_{H-1}(s_1, a_1)] \text{KL}(P_{H-1}^{\mathcal{M}_1}(\cdot \mid s_1, a_1) \parallel P_{H-1}^{\mathcal{M}_2}(\cdot \mid s_1, a_1)) \\ &\quad + \mathbb{E}^{\mathcal{M}_1, \hat{\pi}} [T_{H-1}(s_1, a_2)] \text{KL}(P_{H-1}^{\mathcal{M}_1}(\cdot \mid s_1, a_2) \parallel P_{H-1}^{\mathcal{M}_2}(\cdot \mid s_1, a_2)), \end{aligned}$$

where $T_{H-1}(s_1, a_i)$ denotes the total number of visits to (s_1, a_i) at step $H - 1$ after T episodes (see e.g. [59]). We have

$$\begin{aligned} \text{KL}(P_{H-1}^{\mathcal{M}_1}(\cdot \mid s_1, a_1) \parallel P_{H-1}^{\mathcal{M}_2}(\cdot \mid s_1, a_1)) &= \text{KL}(P_{H-1}^{\mathcal{M}_1}(\cdot \mid s_1, a_2) \parallel P_{H-1}^{\mathcal{M}_2}(\cdot \mid s_1, a_2)) \\ &= \frac{1}{4} \log \frac{1/4}{3/4} + \frac{3}{4} \log \frac{3/4}{1/4} \leq \frac{3}{5} \end{aligned}$$

where the last inequality holds as long as $\epsilon_{\text{sim}} \leq 1/6$. Note that the only way for a policy to reach s_1 at step $H - 1$ is to play action a_1 $H - 1$ consecutive times. Since $\hat{\pi}$ only plays a_1 at any given step with probability $\zeta/2$, it follows that the probability that $\hat{\pi}$ reaches s_1 at step $H - 1$ on any given episode is only $(\zeta/2)^{H-1}$. Thus,

$$\begin{aligned} \text{KL}(\mathbb{P}^{\mathcal{M}_1, \hat{\pi}} \parallel \mathbb{P}^{\mathcal{M}_2, \hat{\pi}}) &\leq \frac{3}{5} \left(\mathbb{E}^{\mathcal{M}_1, \hat{\pi}} [T_{H-1}(s_1, a_1)] + \mathbb{E}^{\mathcal{M}_1, \hat{\pi}} [T_{H-1}(s_1, a_2)] \right) \\ &= \frac{3}{5} \mathbb{E}^{\mathcal{M}_1, \hat{\pi}} [T_{H-1}(s_1)] \\ &= \frac{3}{5} \left(\frac{\zeta}{2} \right)^{H-1} \cdot T. \end{aligned}$$

We thus have:

$$\inf_{\hat{\pi}} \sup_{i \in \{1, 2\}} \mathbb{E}^{\mathcal{M}_i, \hat{\pi}} [V_0^{\mathcal{M}_i, \star} - V_0^{\mathcal{M}_i, \hat{\pi}}] \geq \frac{\epsilon_{\text{sim}}}{4} \left(1 - \sqrt{\frac{3}{10} \left(\frac{\zeta}{2} \right)^{H-1} \cdot T} \right)$$

and we therefore have $\inf_{\hat{\pi}} \sup_{i \in \{1, 2\}} \mathbb{E}^{\mathcal{M}_i, \hat{\pi}} [V_0^{\mathcal{M}_i, \star} - V_0^{\mathcal{M}_i, \hat{\pi}}] \geq \epsilon_{\text{sim}}/8$ unless

$$T \geq \frac{5}{6} \cdot \left(\frac{2}{\zeta} \right)^{H-1}.$$

Lower Bound for ζ -Greedy Without sim (Proposition 1). In order to quantify the performance of a ζ -greedy algorithm, we must specify how it chooses \hat{f} when it has not yet observed any samples from a given (s, a, h) . Following the lead of Theorem 2 of [11], to avoid an overly optimistic or pessimistic initialization, we assume that the replay buffer is initialized with a single sample from each (s, a, h) . Note that the conclusion would hold with other initializations, however, e.g. initializing $\hat{f}_h(s, a) = 0$ or randomly if we have no observations from (s, a, h) .

Assume that the observation from $(s_1, a_1, H-1)$ transitions to s_2 , which occurs with probability at least $1/4$. In this case, we then have that, for each h , $\hat{f}_h^0(s_1, a_2) \geq \hat{f}_h^0(s_1, a_1)$. Thus, following the ζ -greedy policy, we have that $\pi_h^0(a_1 | s_1) \leq 1/2$. Denote this event on \mathcal{E}_0 . Furthermore, the only way we will have $\hat{f}_h^0(s_1, a_2) < \hat{f}_h^0(s_1, a_1)$ is if we visit $(s_1, a_1, H-1)$ again and observe a transition to s_1 . For this to occur, however, we must play action a_1 $H-1$ times consecutively which, in this case, will occur with probability at most $\max\{1/2, \zeta/2\}^{H-1} \leq 1/2^{H-1}$.

Following the argument in the direct policy transfer case, we have

$$\begin{aligned} \inf_{\hat{\pi}} \sup_{i \in \{1,2\}} \mathbb{E}^{\mathcal{M}_i, \hat{\pi}} [V_0^{\mathcal{M}_i, \star} - V_0^{\mathcal{M}_i, \hat{\pi}}] &\geq \inf_{\hat{\pi}} \sup_{i \in \{1,2\}} \frac{1}{4} \mathbb{E}^{\mathcal{M}_i, \hat{\pi}} [V_0^{\mathcal{M}_i, \star} - V_0^{\mathcal{M}_i, \hat{\pi}} | \mathcal{E}_0] \\ &\geq \frac{\epsilon_{\text{sim}}}{16} \left(1 - \sqrt{\frac{3}{10} \mathbb{E}^{\mathcal{M}_1} [T_{H-1}(s_1) | \mathcal{E}_0]} \right) \end{aligned}$$

where $\mathbb{E}^{\mathcal{M}_1} [T_{H-1}(s_1) | \mathcal{E}_0]$ is the expected number of visitations to $(s_1, H-1)$ after T episodes of running the ζ -greedy policy. We can rewrite

$$\mathbb{E}^{\mathcal{M}_1} [T_{H-1}(s_1) | \mathcal{E}_0] = \sum_{t=1}^T \mathbb{E}^{\mathcal{M}_1} [\mathbb{I}\{s_{H-1} = s_1\} | \mathcal{E}_0].$$

Let \mathcal{E} be the event that we have reached $(s_1, H-1)$ in the first T rounds. Then,

$$\begin{aligned} \mathbb{E}^{\mathcal{M}_1} [\mathbb{I}\{s_{H-1} = s_1\} | \mathcal{E}_0] &= \mathbb{E}^{\mathcal{M}_1} [\mathbb{I}\{s_{H-1} = s_1\} | \mathcal{E}, \mathcal{E}_0] \mathbb{P}^{\mathcal{M}_1} [\mathcal{E} | \mathcal{E}_0] + \mathbb{E}^{\mathcal{M}_1} [\mathbb{I}\{s_{H-1} = s_1\} | \mathcal{E}^c, \mathcal{E}_0] \mathbb{P}^{\mathcal{M}_1} [\mathcal{E}^c | \mathcal{E}_0] \\ &\leq \mathbb{P}^{\mathcal{M}_1} [\mathcal{E} | \mathcal{E}_0] + \mathbb{E}^{\mathcal{M}_1} [\mathbb{I}\{s_{H-1} = s_1\} | \mathcal{E}^c, \mathcal{E}_0]. \end{aligned}$$

By what we have just argued, we have $\mathbb{P}^{\mathcal{M}_1} [\mathcal{E} | \mathcal{E}_0] \leq T \cdot \frac{1}{2^{H-1}}$, and $\mathbb{E}^{\mathcal{M}_1} [\mathbb{I}\{s_{H-1} = s_1\} | \mathcal{E}^c, \mathcal{E}_0] \leq \frac{1}{2^{H-1}}$. Thus, $\mathbb{E}^{\mathcal{M}_1} [T_{H-1}(s_1) | \mathcal{E}_0] \leq \frac{2T^2}{2^{H-1}}$. It follows that,

$$\inf_{\hat{\pi}} \sup_{i \in \{1,2\}} \mathbb{E}^{\mathcal{M}_i, \hat{\pi}} [V_0^{\mathcal{M}_i, \star} - V_0^{\mathcal{M}_i, \hat{\pi}}] \geq \frac{\epsilon_{\text{sim}}}{16} \left(1 - \sqrt{\frac{3}{10} \frac{2T^2}{2^{H-1}}} \right)$$

and we therefore have $\inf_{\hat{\pi}} \sup_{i \in \{1,2\}} \mathbb{E}^{\mathcal{M}_i, \hat{\pi}} [V_0^{\mathcal{M}_i, \star} - V_0^{\mathcal{M}_i, \hat{\pi}}] \geq \epsilon_{\text{sim}}/32$ unless

$$T \geq \sqrt{\frac{5}{8}} \cdot 2^{H-1}.$$

Upper Bound for Exploration Policy Transfer (Proposition 4). To obtain an upper bound for Algorithm 1, we can apply Theorem 1, so long as

$$\epsilon_{\text{sim}} \leq \frac{\lambda_{\min}^*}{64dHA^3}.$$

Note that in our setting we have $d = 4$, $A = 2$, $\lambda_{\min}^* = 1/4$, so this condition reduces to $\epsilon_{\text{sim}} \leq \frac{1}{8192H}$. Taking \mathcal{F} to simply be the set of Q -functions defined above (so $V_{\max} = H$), Theorem 1 then gives that with probability at least $1 - \delta$, Algorithm 1 learns an ϵ -optimal policy as long as $T \geq c \cdot \frac{H^{17}}{\epsilon^8} \cdot \log \frac{H}{\delta}$.

D.2 Proof of Proposition 5

We define three MDPs: \mathcal{M}^{sim} , and two possible real MDPs, $\mathcal{M}_1 := \mathcal{M}^{\text{real},1}$ and $\mathcal{M}_2 := \mathcal{M}^{\text{real},2}$. In all cases we have states $\mathcal{S} = \{s_1, s_2\}$, actions $\mathcal{A} = \{a_1, a_2, a_3, a_4\}$, and $H = 2$, and set the starting state to s_1 . We define

$$P_1^{\text{sim}}(s_1 | s_1, a_1) = 1, \quad P_1^{\text{sim}}(s_1 | s_1, a) = P_1^{\text{sim}}(s_2 | s_1, a) = 1/2, a \in \{a_2, a_3, a_4\}.$$

For both \mathcal{M}_1 and \mathcal{M}_2 , we have:

$$P_1^{\text{real}}(s_1 | s_1, a_1) = 1, P_1^{\text{real}}(s_1 | s_1, a_4) = P_1^{\text{real}}(s_2 | s_1, a_4) = 1/2$$

for \mathcal{M}_1 , we have

$$P_1^{\text{real}}(s_2 | s_1, a_2) = 1 + \epsilon_{\text{sim}}, P_1^{\text{real}}(s_1 | s_1, a_2) = 1 - \epsilon_{\text{sim}}, P_1^{\text{real}}(s_1 | s_1, a_3) = P_1^{\text{real}}(s_2 | s_1, a_3) = 1/2$$

and for \mathcal{M}_2 ,

$$P_1^{\text{real}}(s_2 | s_1, a_3) = 1 + \epsilon_{\text{sim}}, P_1^{\text{real}}(s_1 | s_1, a_3) = 1 - \epsilon_{\text{sim}}, P_1^{\text{real}}(s_1 | s_1, a_2) = P_1^{\text{real}}(s_2 | s_1, a_2) = 1/2.$$

We take the reward to be 0 everywhere, except $r_2(s_2, a) = 1$ for all a .

Note that each of these can be represented as a linear MDP in $d = 2$ dimensions, so Assumption 2 holds. In particular, for \mathcal{M}^{sim} we can take:

$$\begin{aligned} \phi^s(s, a_1) &= e_1, \phi^s(s, a) = e_2, a \in \{a_2, a_3, a_4\}, s \in \mathcal{S}, \\ \mu_1^s(s_1) &= [1, 1/2], \mu_1^s(s_2) = [0, 1/2]. \end{aligned}$$

For \mathcal{M}_1 we can instead take:

$$\begin{aligned} \phi^r(s, a_1) &= e_1, \phi^r(s, a) = [1/2, 1/2], a \in \{a_3, a_4\}, s \in \mathcal{S}, \\ \phi^r(s, a_2) &= [1/2 - \epsilon_{\text{sim}}, 1/2 + \epsilon_{\text{sim}}], s \in \mathcal{S}, \\ \mu_1^r(s_1) &= [1, 0], \mu_1^r(s_2) = [0, 1]. \end{aligned}$$

\mathcal{M}_2 follows similarly with the role of a_2 and a_3 flipped.

It is easy to see that Assumption 1 is met on this instance for both choices of $\mathcal{M}^{\text{real}}$. On \mathcal{M}^{sim} , the policy π_{exp} which in every states plays action a_1 with probability 1/2 and action a_4 with probability 1/2 satisfies $\lambda_{\min}(\mathbb{E}^{\text{sim}, \pi_{\text{exp}}}[\phi^s(s_h, a_h)\phi^s(s_h, a_h)^\top]) \geq 1/2$ (which shows that Assumption 3 holds).

Note, however, that π_{exp} does not play action a_2 or a_3 . As \mathcal{M}_1 and \mathcal{M}_2 differ only on a_2 and a_3 , playing π_{exp} will not allow for \mathcal{M}_1 and \mathcal{M}_2 to be distinguished. As a_2 is the optimal action on \mathcal{M}_1 and a_3 the optimal action on \mathcal{M}_2 , it follows that playing π_{exp} will not allow for the identification of the optimal policy on \mathcal{M}_1 and \mathcal{M}_2 . This can be formalized identically to Appendix D.1, yielding the stated result.

E Experimental Details

E.1 Didactic Tabular Example

Consider the following variation of the combination lock. We let the action space $\mathcal{A} = \{1, 2\}$, and assume there are two states, $\mathcal{S} = \{s_1, s_2\}$, and horizon H . We start in state s_1 . The sim dynamics are given as:

$$\begin{aligned} \forall h < H - 1 : \quad & P_h^{\text{sim}}(s_1 | s_1, a_1) = 1, \quad P_h^{\text{sim}}(s_2 | s_1, a_2) = 1 \\ & P_{H-1}^{\text{sim}}(s_1 | s_1, a_1) = 1/4, P_{H-1}^{\text{sim}}(s_2 | s_1, a_1) = 3/4, P_{H-1}^{\text{sim}}(s_2 | s_1, a_2) = 1 \\ \forall h \in [H] : \quad & P_h^{\text{sim}}(s_2 | s_2, a) = 1, a \in \{a_1, a_2\}, \end{aligned}$$

and the real dynamics are given as:

$$\begin{aligned} \forall h < H - 1 : \quad & P_h^{\text{real}}(s_1 | s_1, a_1) = 1, \quad P_h^{\text{real}}(s_2 | s_1, a_2) = 1 \\ & P_{H-1}^{\text{real}}(s_1 | s_1, a_1) = 3/4, P_{H-1}^{\text{real}}(s_2 | s_1, a_1) = 1/4, P_{H-1}^{\text{real}}(s_2 | s_1, a_2) = 1 \\ \forall h \in [H] : \quad & P_h^{\text{real}}(s_2 | s_2, a) = 1, a \in \{a_1, a_2\}. \end{aligned}$$

Note that these only differ on (s_1, a_1) at $h = H - 1$, and we have $\epsilon_{\text{sim}} = 1/2$. We define the reward function as (note that this is deterministic, and the same for both sim and real):

$$\begin{aligned} \forall h \in [H] : \quad & r_h(s_1, a_2) = 1/8 - h/8H \\ & r_H(s_1, a) = 1/5, a \in \{a_1, a_2\}, \end{aligned}$$

and all other rewards are taken to be 0.

The intuition for this example is as follows. In both sim and real, the only way the agent can get reward is to either end up in state s_1 at step H , or to take action a_2 in state s_1 at any point. In sim, the probability of ending up in state s_1 at step H , even if the optimal sequence of actions to do this is taken, is only $1/4$, due to the final transition, and thus the average reward obtained by the policy which aims to end up in s_1 is only $1/4$. In contrast, if we take action a_2 in s_1 , we will always collect reward of at least $3/8$ (and the earlier we take action a_2 the more reward we collect, up to $1/2$). Thus, in sim the optimal thing to do in s_1 is always to play a_2 . However, if we play a_2 even once, we will transition out of s_1 and never return, so there is no chance we will reach s_1 at step H .

In real, the transitions at the final step are flipped, so that now the probability of finishing in s_1 , if we take the optimal sequences of actions to do this, is $3/4$, and the expected reward for this is then also $3/4$. Since the reward for taking a_2 in s_1 does not change, and is bounded as $1/2$, then in real the optimal policy is to seek to end up in s_1 at the final step.

The challenge with ending up in s_1 at the end is that it requires playing action a_1 at every step. In this sense it is then a classic combination lock instance, and randomized exploration will fail, requiring $\Omega(2^H)$ episodes to reach the final state (since the probability of randomly taking a_1 at every state decreases exponentially with the horizon). Similarly, if we transfer the optimal policy from sim to real, it will never take action a_1 , so will never reach s_1 at the end, and if we transfer the optimal policy from sim with some random exploration, it will fail for the same reason random exploration from scratch fails.

However, note that we can transfer a policy from sim that is able to reach s_1 at the second-to-last step with probability 1, i.e. the policy that takes action a_1 at every step. Thus, if in sim we aim to learn exploration policies that can traverse the MDP, and we transfer these exploration policies, they will transfer, and will allow us to easily reach s_1 at the final step, and quickly determine that it is indeed the optimal thing in real.

We provide additional experimental results on this instance in Appendix E.1.

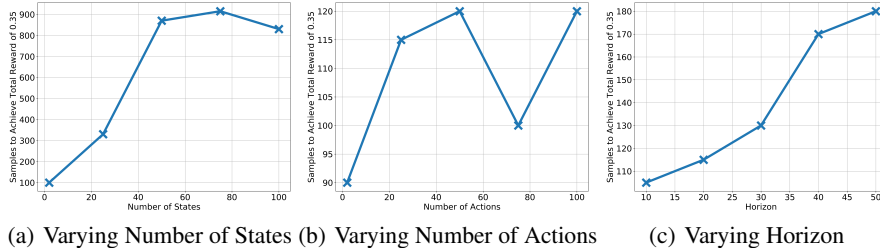


Figure 5: Performance of Exploration Policy Transfer on instance from Section 5.2, varying number of states, actions, and horizon. We plot the number of samples required to achieve a reward of 0.35, which is approximately solving the task. All results are averaged across 20 trials. When increasing the number of states, we add additional 0-reward states (i.e. states given in yellow in Figure 2), and when adding additional actions we add additional low-reward actions (i.e. actions that have the same behavior as action a_2 in Figure 2). We observe that increasing the number of states and horizon increases the number of samples needed, while increasing the number of actions does not substantially. We emphasize, however, that this is for a *particular* example, and this scaling may not be the same for all examples—Theorem 1, however, gives an upper bound on *all* examples.

E.2 Practical Algorithm Details

The core of our work is to decouple the optimal policy training from exploration strategies in reinforcement learning fine-tuning. Specifically, we propose a framework that uses a set of diverse exploration policies to collect samples from the environment. These exploration policies are fixed while we run off policy RL updates on the collected samples to extract an optimal policy. Our theoretical derivation suggests that this decoupling can improve sample efficiency and overall learning performance.

Algorithm 6 sim2real transfer using OS for exploration and SAC for optimization

```
1: Input: Simulator  $\mathcal{M}^{\text{sim}}$ , real environment  $\mathcal{M}^{\text{real}}$ , simulator training budget  $N$ , exploration
   reward balancing  $\alpha$ , reward threshold  $\epsilon$ , exploration set size  $n$ .
2: Pre-train Exploration Policies in  $\mathcal{M}^{\text{sim}}$ :
3: Initialize  $\Pi_{\text{exp}} = \{\pi_\theta(\cdot|z) | z \in \{1 \dots n\}\}$ 
4: Initialize discriminator  $D_\phi$ 
5: for  $i = 1$  to  $N$  do ▷ Learn diverse exploration policies
6:   Sample latent  $z \sim \text{unif}(1, n)$  and initial state  $s_0$ .
7:   for  $t = 1$  to  $\text{max\_steps\_per\_episode}$  do
8:     Sample action  $a_t \sim \pi_\theta(a_t | s_t, z)$ .
9:     Step environment:  $s_{t+1} \sim p(s_{t+1} | s_t, a_t)$ .
10:    Compute discriminator score  $d_t = D(s_{t+1}, z)$ 
11:    Compute exploration reward  $r_e(s_{t+1}, z) = \log \frac{\exp(d_t)}{\sum_{z'} \exp(d(s_{t+1}, z'))}$ .
12:    if  $R_\pi \geq \epsilon$  then
13:      Compute reward  $r_t = r(s_t, a_t) + \alpha \cdot r_e(s_{t+1}, z)$ .
14:    else
15:      Compute reward  $r_t = r(s_t, a_t)$ 
16:    Let  $\mathcal{D} \leftarrow \mathcal{D} \cup \{(s_t, a_t, r_t, s_{t+1}, z)\}$ .
17:    Update  $\pi_\theta$  to maximize  $J_\pi$  with SAC.
18:    Update  $\phi$  to maximize  $J_u$ ,  $\phi \leftarrow \phi + \eta \nabla_\phi \mathbb{E}_{s, z \sim \mathcal{D}} [\log D_\phi(s, z)]$ 
19:  Compute  $R_\pi = \sum_t r_t$ 
20: Explore in  $\mathcal{M}^{\text{real}}$  and Estimate Optimal Policy :
21: Initialize SAC agent (either from scratch or to weights of optimal sim policy).
22: while not converged do
23:   Sample  $z \sim \text{unif}(1, n)$ , play  $\pi_\theta(\cdot | z)$  in  $\mathcal{M}^{\text{real}}$ , add data to replay buffer of SAC.
24:   Roll out SAC policy for one step, perform standard SAC update.
```

Our framework is complementary to (a) RL works on diversity or exploration that generate diverse policies and (b) off policy RL algorithms that optimize for policies. One can plug in (a) to extract a set of exploration policy from a simulator and use them for data collect in the real world but use (b) to optimize for the final policy. The design choice to use simulator to extract a set of exploration policies where each policy is not necessarily optimizing for the task at hand marks our distinction from previous works in (a) and (b).

We provide a practical instantiation of our framework using an approach inspired by One Solution is Not All You Need (OS) [30] to extract exploration policies and Soft Actor Critic (SAC) [30] to optimize for the optimal policy. We details the instantiation in Algorithm 6. OS trains a set of policy to optimize not only the task reward but also a discriminator reward where the discriminator encourages each policy to achieve different state. Unlike OS which carefully balances the task and exploration rewards to ensure all policies have a chance at solving the desired task, we emphasize only on having diverse policies. With a known sim2real gap, we posit that some sub-optimal policies that are not solving the task in the simulator is actually helpful for exploration in the real world, which allows us to simplify the balance between task and exploration. We uses standard off-shelf SAC update to optimize for the policy.

E.3 TychoEnv sim2sim Experiment Details

For the TychoEnv experiment we run a variant of Algorithm 6. We set $n = 20$, and set the reward to $r_t^i = (1 - \alpha_i)r + \alpha_i r_e$ where we vary α_i from 0 to 0.5. While we use a sparse reward in $\mathcal{M}^{\text{real}}$, to speed up training in \mathcal{M}^{sim} we use a dense reward that penalizes the agent for its distance to the target. We train in \mathcal{M}^{sim} for 7M steps to obtain exploration policies. Rather than simply transferring the converged version of the exploration policies trained in \mathcal{M}^{sim} , we found it most effective to save the weights of the policies throughout training, and transfer all of these policies. As the majority of these policies do not collect any reward in \mathcal{M}^{sim} , we run an initial filtering stage where we identify several policies from this set that find reward (this can be seen in Figure 4 with the initial region of 0 reward). We then run SAC in $\mathcal{M}^{\text{real}}$, initialized from scratch, feeding in the data collected by these refined exploration policies into the replay buffer. We found it most effective to only inject data from the

Hyperparameter	Value
reward balance α (OS)	$\{\frac{1}{38}i - \frac{1}{38} : i = 1, 2, \dots, 20\}$
learning rate	0.0003
Q update magnitude τ	0.005
discount γ	0.99
batch size	2048
steps per episode	45
replay buffer size	5×10^6
training steps N (in $\mathcal{M}^{\text{real}}$)	7×10^7

Table 1: Hyperparameters used in Tycho training and finetuning

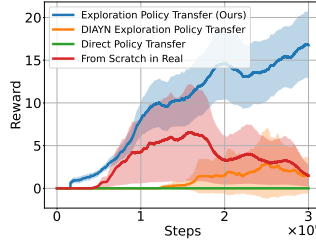


Figure 6: Additional results on Tycho, including baselines training from scratch in $\mathcal{M}^{\text{real}}$, and training exploration policies in \mathcal{M}^{sim} with reward as stated above but with $\alpha_i = 1$ (which is equivalent to simply training exploration policies with DIAYN [14]). As can be seen, while training from scratch in $\mathcal{M}^{\text{real}}$ is able to learn, it learns at a much slower rate than exploration policy transfer, and achieves a much lower final value. Furthermore, training the exploration policies to maximize a mix of the task and diversity reward yields a substantial gain over simply training them to be diverse.

exploration policies in the replay buffer on episodes where they observe reward. We run vanilla SAC with $\text{UTD} = 3$ and target entropy of -3 . We rely on the implementation of SAC from stable-baselines3 [51].

For direct policy transfer, we train a policy to convergence in \mathcal{M}^{sim} that solves the task (using SAC), and then transfer this single policy, otherwise following the same procedure as above.

In $\mathcal{M}^{\text{real}}$, our reward is chosen to have a value of 50 if the end effector makes contact with the ball, and otherwise 0. If the robot successfully makes contact with the ball the episode terminates. To generate a realistic transfer environment, we change the control frequency (doubling it in $\mathcal{M}^{\text{real}}$) and the action bounds.

For both methods, we run the \mathcal{M}^{sim} training procedure 4 times, and then with each of these run it in $\mathcal{M}^{\text{real}}$ twice. Error bars in our plot denote one standard error.

All experiments were run on two Nvidia V100 GPUs, and 32 Intel(R) Xeon(R) CPU E5-2620 v4 @ 2.10GHz CPUs. Additional hyperparameters are given in Table 1.

We provide results on several additional baselines for the Tycho setup in Figure 6.

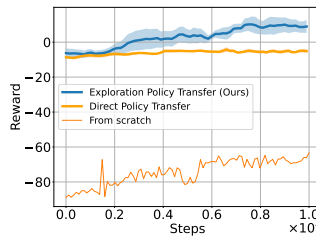


Figure 7: Results on Franka sim2real experiment, comparing to training from scratch in real.

Hyperparameter	Value
reward balance α (OS)	0.5
reward threshold ϵ (OS)	-16
learning rate	0.0003
Q update magnitude τ	0.005
discount γ	0.99
batch size	256
steps per episode	45
replay buffer size	1×10^6
training steps N	2×10^7

Table 2: Hyperparameters used in Franka training and finetuning

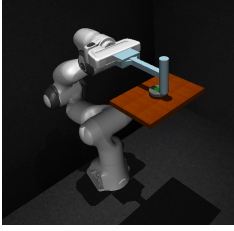


Figure 8: Franka Hammering Task Setup

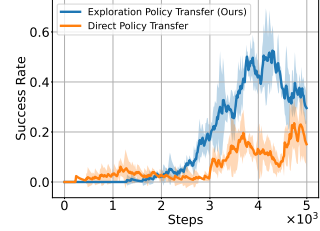


Figure 9: Results on sim2sim Transfer in Franka Simulator

E.4 sim2sim Transfer on Franka Emika Panda Robot Arm

We next turn to the Franka Emika Panda robot arm [17], for which we use a realistic custom simulator built using the MuJoCo simulation engine [61]. We consider a hammering task, where the Franka arm holds a hammer, and the goal is to hammer a nail into the board (see Figure 8). Success is obtained when the nail is fully inserted. We simulate sim2real transfer by setting $\mathcal{M}^{\text{real}}$ to be a version of the simulator with nail location and stiffness significantly beyond the range seen during training in \mathcal{M}^{sim} .

We compare exploration policy transfer with direct sim2real policy transfer. Unlike the Tycho experiment, where we trained policies from scratch in $\mathcal{M}^{\text{real}}$ and simply used the policies trained in \mathcal{M}^{sim} to explore, here we initialize the task policy in $\mathcal{M}^{\text{real}}$ to $\pi^{\text{sim},*}$, which we then finetune on the data collected in $\mathcal{M}^{\text{real}}$ by running SAC. For direct sim2real transfer, we collect data in $\mathcal{M}^{\text{real}}$ by simply rolling out $\pi^{\text{sim},*}$ and feeding this data to the replay buffer of SAC. For exploration policy transfer, we train an ensemble of $n = 10$ exploration policies in \mathcal{M}^{sim} and run these policies in $\mathcal{M}^{\text{real}}$, again feeding this data to the replay buffer of SAC to finetune $\pi^{\text{sim},*}$. During training in \mathcal{M}^{sim} , we utilize domain randomization for both methods, randomizing nail stiffness, location, radius, mass, board size, and damping.

The results of this experiment are shown in Figure 9. We see that, while direct policy transfer is able to learn, it learns at a significantly slower rate than our exploration policy transfer approach, and achieves a much smaller final success rate.

E.5 Franka sim2real Experiment Details

We use Algorithm 6 to train a policy on the Franka robot with $n = 15$.

The reward of the pushing task is given by:

$$r(s_t, a_t) = -\|\mathbf{p}_{\text{ee}} - \mathbf{p}_{\text{obj}}\|^2 - \|\mathbf{p}_{\text{obj}} - \mathbf{p}_{\text{goal}}\|^2 + \mathbb{I}_{\mathbf{p}_{\text{obj}} - \mathbf{p}_{\text{goal}} \leq 0.025} - \mathbb{I}_{\mathbf{p}_{\text{obj}} \text{ offtable}} \quad (\text{E.1})$$

where \mathbf{p}_{goal} is the desired position of the puck by the edge of the surface.

The network architecture of the actor and critic networks are identical, consisting of a 2-layer MLP, each of size 256 and ReLU activations.

We use stable-baselines3 [51] for our SAC implementation, using all of their default hyperparameters. The implementation of OS is built on top of this SAC implementation. Values of hyperparameters

are shown in Table 2. Gaussian noise with mean 0 and standard deviation 0.005 meters is added in simulation to the position of the puck. Hyperparameters are identical between exploration policy transfer and direct transfer methods.

For finetuning in real, we start off by sampling exclusively from the buffer used during simulation. Then, as finetuning proceeds, we gradually start taking more samples from the real buffer, with the proportion of samples taken from sim equal to $1 - s/3000$, where s is the current number of steps. After 3000 steps, all samples are taken from the real buffer.

Experiments were run using a standard Nvidia RTX 4090 GPU. Training in simulation takes about 3 hours, while finetuning was ran for about 90 minutes.

In Figure 7, we provide results on this setup running the additional baseline of training a policy from scratch in real. As can be seen, this is significantly worse than either transfer method.

NeurIPS Paper Checklist

1. Claims

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

Answer: [\[Yes\]](#)

Justification: We validate all our claims with theoretical results and experiments.

Guidelines:

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

2. Limitations

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

Answer: [\[Yes\]](#)

Justification: Please see Discussion section.

Guidelines:

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

3. Theory Assumptions and Proofs

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

Answer: [\[Yes\]](#)

Justification: All results are precisely proved in the supplemental, and all assumptions clearly stated.

Guidelines:

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

4. Experimental Result Reproducibility

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

Answer: [\[Yes\]](#)

Justification: To the extent possible, given that we are working with real-world systems, we have described our setup and implementation details.

Guidelines:

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

5. Open access to data and code

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

Answer: [No]

Justification: We have not currently released our code but hope to in the future.

Guidelines:

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

6. Experimental Setting/Details

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

Answer: [Yes]

Justification: We have stated all parameters and algorithm details to the best of our knowledge (please see Experimental Details section in supplemental).

Guidelines:

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

7. Experiment Statistical Significance

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

Answer: [Yes]

Justification: For all our experimental results, we provide error bars corresponding to 1 standard error.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

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

8. Experiments Compute Resources

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

Answer: [Yes]

Justification: Please see Experimental Details section in appendix.

Guidelines:

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

9. Code Of Ethics

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

Answer: [Yes]

Justification: This paper does not violate any ethical guidelines.

Guidelines:

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

10. Broader Impacts

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

Answer: [NA]

Justification: This work is related to the advancement of our fundamental understanding of machine learning. As such, we do not believe there are any direct societal impacts from this work.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

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

Answer: [NA]

Justification: We are not releasing high-risk models.

Guidelines:

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

12. Licenses for existing assets

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

Answer: [Yes]

Justification: We have cited the creators of the code used in this project.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset’s creators.

13. **New Assets**

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

Answer: [NA]

Justification:

Guidelines:

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

14. **Crowdsourcing and Research with Human Subjects**

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

Answer: [NA]

Justification: This work does not involve human subjects research.

Guidelines:

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

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

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

Answer: [NA]

Justification: This work does not involve human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.