Provable Hierarchy-Based Meta-Reinforcement Learning

Kurtland Chua
Princeton University

Qi Lei
New York University

Jason D. Lee
Princeton University

Abstract

Hierarchical reinforcement learning (HRL) has seen widespread interest as an approach to tractable learning of complex modular behaviors. However, existing works either assume access to expert-constructed hierarchies, or use hierarchy-learning heuristics with no provable guarantees. To address this gap, we analyze HRL in the meta-RL setting, where a learner learns latent hierarchical structure during meta-training for use in a downstream task. We consider a tabular setting where natural hierarchical structure is embedded in the transition dynamics. Analogous to supervised meta-learning theory, we provide "diversity conditions" which, together with a tractable optimism-based algorithm, guarantee sample-efficient recovery of this natural hierarchy. Furthermore, we provide regret bounds on a learner using the recovered hierarchy to solve a meta-test task. Our bounds incorporate common notions in HRL literature such as temporal and state/action abstractions, suggesting that our setting and analysis capture important features of HRL in practice.

1 INTRODUCTION

Reinforcement learning (RL) has demonstrated tremendous successes in many domains (Schulman et al., 2015; Vinyals et al., 2019; Schrittwieser et al., 2020), learning near-optimal policies despite limited supervision. Nevertheless, RL remains difficult to apply to problems requiring temporally extended planning and/or exploration (Ecoffet et al., 2021). A promising approach to this problem is hierarchical reinforcement learning (HRL), which has seen continued interest due to its appealing biological basis. In its most basic form, HRL seeks to decompose tasks into a sequence of skills, each of which is easier to learn individually than the full task. By restricting the agent to using learned skills, the search space over policies can be greatly reduced. Furthermore, learned skills can induce simpler state and/or action spaces, simplifying the learning problem. Finally, learned skills with useful semantic behavior can be reused for other related tasks, enabling transfer learning.

Naturally, a hierarchy-based learner is limited by the quality of skills that are made available and/or learned. Accordingly, many empirical works have proposed algorithms for online skill learning in the context of a single RL task (Nachum et al., 2019a, 2018). These approaches have been experimentally demonstrated to be effective at finding useful and interpretable skills. Other approaches consider the skill learning problem in the context of meta-RL (Frans et al., 2018), or in the reward-free setting (Eysenbach et al., 2018). Nevertheless, the heuristics and algorithms proposed in these empirical works do not provide any provable guarantees on the quality of learned skills.

On the other hand, theoretical analyses have mostly focused on how learners benefit from having access to skills. For example, Fruit and Lazaric (2017) provide a regret bound on learning with skills in the infinite-horizon average reward case. Meanwhile, in the meta-RL setting, Brunskill and Li (2014) consider the problem of finding and using skills in a continual learning setting and provide a sample complexity analysis. However, these analyses either sidestep the question of how the skills are obtained, or do not address the problem in a computationally tractable manner.

In this work, we aim to take a step towards providing provable guarantees for hierarchy learning through tractable algorithms. We focus on the meta-RL setting, in which a learner extracts skills from a set of provided tasks which are then used in a downstream task. We work in the tabular case, assuming the transition dynamics of the given tasks have shared hierarchical structure. This hierarchical structure comes in the form of a certain clustering in the state space, such that the inter-cluster connections vary between tasks. Our contributions are as follows:

1. **"Diversity conditions" ensuring hierarchy recovery.** We develop natural optimism-based coverage conditions ensuring that the aforementioned clusters

Proceedings of the 26th International Conference on Artificial Intelligence and Statistics (AISTATS) 2023, Valencia, Spain. PMLR: Volume 206.

and bottlenecks detectable by solving provided meta-training tasks.

2. **A tractable hierarchy-learning algorithm.** We provide an algorithm that provably learns the latent hierarchy from interactions, assuming the coverage conditions above. Our method has sample complexity scaling as $O(TKS)$ in the leading term compared to $O(TS^2A)$ for a brute-force method, where T are the number of tasks, S is the number of states, A is the number of actions, and $K \ll SA$ is the number of skills to learn. We also experimentally demonstrate our algorithm on toy settings in Section C.

3. **Regret bounds on downstream tasks.** We provide regret bounds for learners that apply the extracted hierarchy on downstream tasks. Furthermore, we show an exponential regret separation between hierarchy-based and hierarchy-oblivious learners for a family of task distributions, corroborating prevailing intuitions regarding when and why HRL helps. In our construction, hierarchy-based learners incur regret bounded by $O(\sqrt{H^2N})$ while hierarchy-oblivious learners incur worst-case regret $\Omega(2^{H/2}\sqrt{H^2N})$.

2 NOTATION

We write $[K] \coloneqq \{1, \dots, K\}$. Furthermore, we use the standard notations O, Θ, Ω to denote orders of growth, and $\tilde{O}, \tilde{\Theta}, \tilde{\Omega}$ to indicate suppressed logarithmic factors. We use $\delta(x)$ to denote the Dirac delta measure on x. Finally, given discrete probability measures P and Q, we define the total variation (TV) norm between P and Q to be $\|P - Q\|_{\mathrm{TV}} = \frac{1}{2}\sum_x |P(x) - Q(x)|$.

We work with finite-horizon Markov decision processes (MDPs), defined as a tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathbb{P}, r, H)$, where $\mathcal{S}$ is the set of states, $\mathcal{A}$ is the set of actions, $\mathbb{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ are the transition dynamics, $r : \mathcal{S} \times \mathcal{A} \to [0, 1]$ is the reward function, and H is the horizon. We assume stationary dynamics unless otherwise noted, in which case $\mathbb{P}^{(h)}$ is the dynamics at time step h. For constants relating to horizons, we will define $[H] \coloneqq \{0, \dots, H-1\}$. Given a policy $\pi : [H] \times \mathcal{S} \to \mathcal{A}$, we define the value functions

$$Q_h^\pi(s, a) \coloneqq \mathbb{E}\left[\sum_{k=h}^{H-1} r(s_k, a_k) \,\middle|\, (s_h, a_h) = (s, a)\right]$$
$$V_h^\pi(s) \coloneqq Q^\pi(s, \pi_k(s))$$

where $s_{k+1} \sim \mathbb{P}(\cdot \mid s_k, a_k)$. Furthermore, we write V^* and Q^* to denote optimal value functions obtained by maximizing over π (and are attained by some optimal policy π^*). When playing $\pi_1, \dots, \pi_N$ in $\mathcal{M}$, we define the regret as

$$\mathrm{Regret}_N(\mathcal{M}) \coloneqq \sum_{t=1}^{N} V_0^*(s_0) - V_0^{\pi_t}(s_0).$$

We use $\ominus$ to denote a terminal state. We let τ_π denote the (random) trajectory of (s, a, r, s') pairs generated by π. For a state s and length-H trajectory τ, we write $s \in \tau_\pi$ if $s_h = s$ for some $h \in [H]$. We define $(s, a) \in \tau_\pi$ similarly. Finally, given an MDP $\mathcal{M}$, $\mathcal{M}(2H)$ denotes a copy of $\mathcal{M}$ with a doubled horizon.

3 SETTING

The standard meta-RL setting is divided into two stages: *meta-training*, and *meta-testing*. During meta-training, the learner can access T MDPs $\{(\mathcal{S}, \mathcal{A}, \mathbb{P}_t, r_t, H)\}_{t\in[T]} = \{\mathcal{M}_t\}_{t\in[T]}$. Note that $\mathbb{P}_t$ and r_t are task-dependent. Subsequently, the learner is presented in the meta-test phase with an MDP $\mathcal{M}_{T+1} = (\mathcal{S}, \mathcal{A}, \mathbb{P}_{T+1}, r_{T+1}, H)$, within which the learner seeks to minimize its regret. We assume, without loss of generality, that the MDPs have a shared starting state s_0. The problem setting is summarized in Algorithm 1.

For meta-RL to succeed, the MDPs must have shared structure. As our focus is on hierarchical RL, we first define a notion of clustering:

Definition 3.1. A *hierarchical clustering $\mathcal{C}$ on* $(\mathcal{S}, \mathcal{A})$ is a tuple $(\mathcal{Z}, \mathrm{Ent}(\cdot), \mathrm{Ext}(\cdot))$, where $\mathcal{Z}$ is a partitioning of $\mathcal{S}$ into *clusters*, and for any cluster $Z \in \mathcal{Z}$:

- $\mathrm{Ent}(Z) \subseteq Z$ is a set of *entrances* into Z.
- $\mathrm{Ext}(Z) \subseteq Z \times \mathcal{A}$ is a set of *exits* from Z.

Accordingly, we define the *interior* of Z, denoted Z°, as $(Z \times \mathcal{A}) \setminus \mathrm{Ext}(Z)$. Finally, we write $\mathrm{Ent}(\mathcal{C})$ as the set of all entrances, and $\mathrm{Ext}(\mathcal{C})$ as the set of all exits. $\diamond$

Unless noted otherwise, we consider a single hierarchical clustering $\mathcal{C} = (\mathcal{Z}, \mathrm{Ent}(\cdot), \mathrm{Ext}(\cdot))$. We now move on to the desired notion of shared hierarchical structure:

Definition 3.2 (Latent Hierarchy)**.** We say that the tasks $\{\mathcal{M}_t\}_{t\in[T+1]}$ have a *shared latent hierarchy with respect to a hierarchical clustering $\mathcal{C}$* if for any $Z \in \mathcal{Z}$:

(a) For any $(s, a) \in Z^\circ$, $\mathbb{P}_t(\cdot \mid s, a)$ is independent of task $t \in [T + 1]$, and supported on Z.

(b) For any $(s, a) \in \mathrm{Ext}(Z)$, $\mathbb{P}_t(\cdot \mid s, a)$ is supported on $\mathrm{Ent}(\mathcal{C})$ for any $t \in [T + 1]$. Furthermore, there exists tasks $t, t' \le T$ such that $\mathbb{P}_t(\cdot \mid s, a) \neq \mathbb{P}_{t'}(\cdot \mid s, a)$. $\diamond$

Algorithm 1 The meta-RL setting.

Require: Meta-training MDPs $\{\mathcal{M}_t\}_{t\in[T]}$, meta-test MDP $\mathcal{M}_{\mathrm{Tg}}$.
1: **Meta-training**: Agent interacts with $\{\mathcal{M}_t\}$, extracts (algorithm-specific) structure.
2: **Meta-testing**: Agent interacts with $\mathcal{M}_{\mathrm{Tg}}$ conditioned on extracted structure.

At a high level, Definition 3.2 assumes that $\mathcal{S}$ can be partitioned into distinct clusters so that (a) dynamics within cluster interiors are task-independent and (b) exits serve as bottlenecks for reaching other clusters.

Example 3.1 (Gated Four-Room)**.** Figure 1 illustrates the family of gated four-room environments along with a sample task. In this environment, the agent has five available actions: `left`, `right`, `up`, `down`, and `stay`. Movement actions move the agent by one block in the specified direction if there is no wall/gate present between the starting and target positions; otherwise, the agent stays in place. For example, in the given configuration with a red gate, playing `up` from right below the red gate will cause no change to the agent state.

Within the family of tasks which vary gate configurations and which room center the agent spawns in, one can show that the environment has a shared latent hierarchy, where clusters are delineated by the colored gates. Cluster entrances are colored aqua, while exits are marked by arrows indicating the state and directional action for the exit.[1] ⌟

To see how Definition 3.2 captures intuitive notions of hierarchy in practical settings, we provide the following example:

Example 3.2 (The Alchemy benchmark)**.** Alchemy (Wang et al., 2021) is a recently proposed benchmark for meta-RL, where the agent needs to place a stone in a series of potions to obtain a desired appearance, as illustrated in Figure 2. Dipping a stone into a potion traverses an edge in a graph of possible stone appearances. We focus on task distributions that randomize the edges of this graph (i.e., potion positions and feasible stone appearances are fixed). Then, the set of obtainable MDPs has a latent hierarchy where dipping the stone into any of the potions is an exit. Indeed, other than potion dipping, all other actions (e.g., moving the stone around the room) have task-independent dynamics. ⌟

For our analysis, we need to define several quantities. In particular, we let $K \coloneqq |\mathrm{Ext}(\mathcal{C})|$, $L \coloneqq |\mathrm{Ent}(\mathcal{C})|$, and $M = \sup_{Z \in \mathcal{Z}} |\mathrm{Ext}(Z)|$. That is, K and L are the total number of exits and entrances, respectively, while M is the maximal number of exits from any cluster.

[1] Although we assume a fixed starting state s_0, we can model task-dependent initial states by assuming that any action from s_0 takes the agent to the true starting state. Any initial state must then be a cluster entrance, while (s_0, a) is an exit for any a.

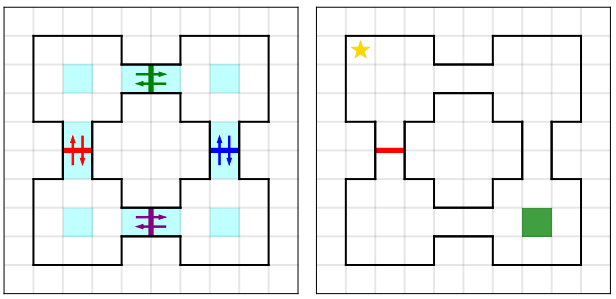

Figure 1: The gated four-room setting, with a green start square and starred goal.

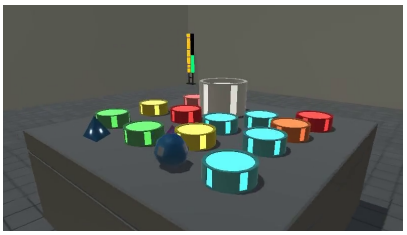

Figure 2: alchemy. placing stones in potions moves the agent through a latent graph of object properties.[2]

Query Model. We work in the online setting, where the agent interacts with the tasks by playing policies from the initial state s_0. During meta-training, we allow the agent to interact with the environments using an unbounded number of timesteps for each trajectory before resetting. We then compute query complexity in terms of the total number of timesteps spent in all tasks in total.

4 META-TRAINING ANALYSIS

In this section, we outline an algorithm for learning a shared latent hierarchy, as defined in Definition 3.2. As exits are bottlenecks between clusters, exit identification is a key part of the learning process. Recall that a key property of every exit in Definition 3.2 is that their dynamics is guaranteed to be different for at least two tasks. To quantify the number of samples needed to detect this change, we need to quantity the difference in dynamics:

Definition 4.1 (β-dynamics separation)**.** There exists $\beta > 0$ such that for any exit $e \in \mathrm{Ext}(\mathcal{C})$ and $t, t' \le T$,

$$\mathbb{P}_t(\cdot \mid e) \neq \mathbb{P}_{t'}(\cdot \mid e) \implies \|\mathbb{P}_t(\cdot \mid e) - \mathbb{P}_{t'}(\cdot \mid e)\|_{\mathrm{TV}} \ge \beta. \qquad \diamond$$

Via standard concentration results, β-dynamics separation allows for high-probability exit detection with $\tilde{O}(S/\beta^2)$ samples. This suggests a brute-force approach: for every (s, a), one can learn to reach (s, a) in every task t, learn dynamics estimates $\hat{\mathbb{P}}_t(\cdot \mid s, a)$, and perform comparisons to check for large deviations in TV norm. This can be done with query complexity $\tilde{O}(TS^2A/\beta^2)$. However, this ignores the reward functions associated with the meta-training MDPs. Indeed, under reasonable "coverage" assumptions in the next section making use of the reward functions, the query cost can be lowered to $\tilde{O}(TKS/\beta^2)$.

[2] Image from Wang et al. (2021), extracted from a larger figure with no other modifications (License).

4.1 Defining a Notion of Coverage

In supervised meta-learning, "diversity conditions" ensure that the meta-training tasks reveal the underlying latent structure (Tripuraneni et al., 2020; Du et al., 2020). We provide analogous conditions ensuring that $\mathcal{M}_1, \ldots, \mathcal{M}_T$ "cover" the latent hierarchy. Since solving $\max_\pi V^\pi(s_0)$ requires fewer samples than learning $\mathbb{P}$, we expect such conditions to provide sample complexity gains.

Visitation Probabilities and α-Importance. Minimally, exits need to be visited by optimal policies in meta-training tasks for coverage. We thus define the following notion:

Definition 4.2 (α-importance)**.** For any MDP $\mathcal{M}$ and state-action pair (s, a), let $\mathcal{M} \setminus (s, a)$ be a modified MDP such that (s, a) brings the agent to a terminal state with no reward. Then, we say that (s, a) is *α-important for $\mathcal{M}$* if $V_0^{\mathcal{M}\setminus(s,a),*}(s_0) < V_0^{\mathcal{M},*}(s_0) - \alpha$. ◇

Definition 4.2 allows us to quantify value gaps between policies that can use a particular state-action pair and those that cannot. To illustrate, consider the task in Figure 3. Note that any trajectory with non-zero return must contain the black arrow at least once. That is, the optimal value when the arrow is unavailable is 0, and thus the arrow has high importance. This suggests that near-optimal policies tend to visit state-action pairs with high-importance. We formalize this connection below, and defer its proof to Section A.

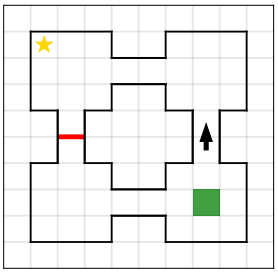

Figure 3: Illustrating Definition 4.2. Since any path to the goal contains the arrow, it has high importance.

Lemma 4.1. *If (s, a) is α-important for $\mathcal{M}$, then for any policy π with $V_0^*(s_0) - V_0^\pi(s_0) < \varepsilon$, then $P((s, a) \in \tau_\pi) > (\alpha - \varepsilon)/H$.*

A Preliminary Coverage Assumption? The previous discussion suggests a simple coverage condition: for any exit (s, a), assume that (1) it has high α-importance for two tasks, and (2) that is has different dynamics between these two tasks. Thus, solving the two tasks would ensure that (s, a) is visited sufficiently to form dynamics estimates, and the TV-norm difference between the estimates would be sufficiently large with high probability for exit detection. However, this condition excludes natural settings, including the earlier four-room example:

Example 4.1. In the gated four-room example, we can show that no exit can be covered under the preliminary condition above. Recall that gates are either open or closed; when a gate is closed, no optimal agent would attempt to use such a gate. In other words, an exit (s, a) cannot be α-important with $\alpha > 0$ when closed. Thus, if a gate has non-zero α-importance for two tasks, then it must have been open for both tasks, and thus the preliminary coverage assumption cannot hold. We illustrate this failure case in Figure 4. ⌟

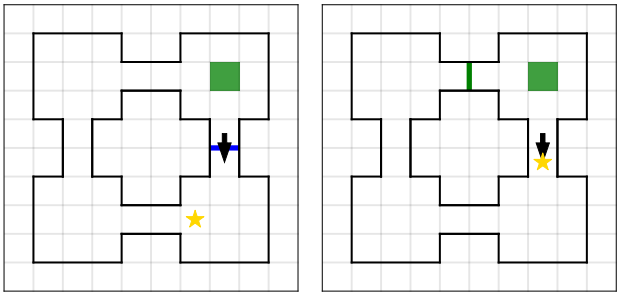

Figure 4: The black arrow is not α-significant for one of the tasks, and thus cannot be "covered" under the preliminary coverage condition.

An Alternative Coverage Mechanism. The proposed assumption fails because there are exits that only have non-zero α-importance for one configuration. For such exits, near-optimal policies can only see one dynamics configuration, making them insufficient for detecting exits.

As an alternative, consider the following hypothetical scenario in the context of Figure 4: an agent has solved both tasks, achieving optimal values V_1^* and V_2^*. Furthermore, note that in the process of solving the second task, the agent will have learned open-gate dynamics for the black arrow. If the agent were to relearn the first task *while borrowing the black arrow's dynamics from the second* (i.e. imagining the gate were open), it would obtain a new value $\hat{V}_1$. Note that $\hat{V}_1 \gg V_1^*$, as the path to the goal in the first task is shorter when the gate is open. Thus, it can reasonably conclude that the black arrow must have been an exit. We illustrate this process in Figure 5, where $\pi_{\mathcal{I}}$ is the optimal policy after "borrowing dynamics." The learner could then run $\pi_{\mathcal{I}}$ for exit detection.

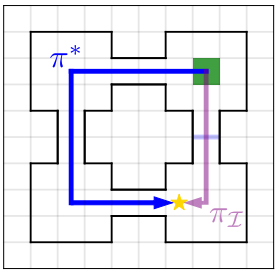

Figure 5: Using optimistic imagination for exit detection.

We refer to the counterfactual reasoning about the dynamics used above as *optimistic imagination*. Unlike the preliminary condition, optimistic imagination only requires that an exit be important for one task and induce a value gap in another when borrowing dynamics – a weaker condition in many cases. With the above intuition in mind, we now present the main coverage assumption.

Assumption 4.1 ((α, ζ)-coverage)**.** *Assume* $(\mathcal{M}_t)_{t\in[T]}$

have a latent hierarchy with respect to $\mathcal{C}$. Then, there exists $\alpha, \zeta > 0$ such that for any $\{e_1, \ldots, e_n\} \subseteq \mathrm{Ext}(\mathcal{C})$, there exists borrowing indices $b_1, \ldots, b_n \in [T]$ so that

(a) e_i is α-important for $\mathcal{M}_{b_i}$ for $i \in [n]$.

(b) For some $\mathcal{M}_t$ with $t \in [T]$, if we construct a new MDP $\bar{\mathcal{M}}_t = (\mathcal{S}, \mathcal{A}, \bar{\mathbb{P}}, r_t, H)$ via

$$\bar{\mathbb{P}}(\cdot \mid s, a) = \begin{cases} \mathbb{P}^{\mathcal{M}_{b_i}}(\cdot \mid s, a) & (s, a) = e_i \\ \mathbb{P}^{\mathcal{M}_t}(\cdot \mid s, a) & \textit{otherwise} \end{cases},$$

(i.e. we replace e_i's dynamics in $\mathcal{M}_t$ with those from $\mathcal{M}_{b_i}$), then $V^{\bar{\mathcal{M}}_t,*}(s_0) > V^{\mathcal{M}_t,*}(s_0) + \zeta$.

Informally, (α, ζ)-coverage ensures that while there are undiscovered exits, borrowing dynamics will result in an overestimated value.[3] Furthermore, the policy attaining this overestimated value must visit at least one exit with borrowed dynamics, leveraging a similar value-gap argument as with Lemma 4.1.

4.2 Why Should Optimistic Imagination Hold?

In this section, we provide a heuristic reason as for why the proposed coverage condition holds for many task distributions of interest. Fix any subset of exits $\{e_1, \ldots, e_n\}$ and any meta-training MDP $\mathcal{M}_t$. As our goal is to demonstrate that one of the exits can be discovered via optimistic imagination, we impose the following restriction:

Property 4.1. *For every e_i, there exists a distribution d_i^* such that $\mathbb{P}_t(\cdot \mid e_i) \neq d_i^*$ implies that e_i is 0-important for $\mathcal{M}_t$, for any $t \in [T]$.*

This property excludes the use of the preliminary coverage condition. As it is reasonable to assume that these exits have high importance for at least one task (i.e. Assumption 4.1(a)), Property 4.1 implies d_i^* is learnable from interactions. Therefore, all that remains is to construct an MDP such that borrowing d_i^* for all $i \in [n]$ within the MDP guarantees the value gap in Assumption 4.1(b).

To do so, we first outline a connection between value functions and the geometry of state visitation measures, as explored by Eysenbach et al. (2021) within the discounted infinite-horizon setting. Note that when one restricts to action-independent reward functions, we can write $V_0^{\mathcal{M},\pi}(s_0) = H \rho_{M,\pi}^\top r$, where r is the vectorized reward function and

$$\rho_{\mathcal{M},\pi}(s) = \frac{1}{H} \sum_{h \in [H]} P^{\mathcal{M},\pi}(s_h = s \mid s_0)$$

is the *state visitation measure of π within $\mathcal{M}$*. An important property we will use is that for any fixed $\mathcal{M}$, $\{\rho_{\mathcal{M},\pi}\}$ is a convex subset of the probability simplex on $\mathcal{S}$.

Given this, we now show that performing optimistic imagination within $\mathcal{M}_t$ results in a non-zero value gap in many cases, justifying Assumption 4.1(b). Define $\bar{\mathcal{M}}_t$ to be the MDP with borrowed dynamics

$$\mathbb{P}^{\bar{\mathcal{M}}_t}(\cdot \mid s, a) = \begin{cases} d_i^* & (s, a) = e_i \\ \mathbb{P}_t(\cdot \mid s, a) & \text{otherwise} \end{cases}.$$

Since $\|\mathbb{P}_t(\cdot \mid e) - d_i^*\|_{\mathrm{TV}} > \beta$ by Definition 4.1, heuristically, we expect the following to be true:

Assumption 4.2. *$\rho_{\bar{\mathcal{M}}_t,\pi^+} \notin \{\rho_{\mathcal{M}_t,\pi}\}$ for some π^+.*

Stated less formally, we expect there to be states that can be reached more easily in $\bar{\mathcal{M}}_t$ compared to $\mathcal{M}_t$. As a result, if we reward the agent for reaching these states in the borrowed MDP, we would obtain the desired reward gap.

Example 4.2. In Example 3.1, this corresponds to the fact that the agent can spend more time in states immediately past open gates compared to if they were closed. ⌟

Continuing from above, we can formally construct a reward function that has the desired value gap. More specifically, since $\{\rho_{\mathcal{M}_t,\pi}\}$ is convex, we can invoke the hyperplane separation theorem to demonstrate the existence of a reward vector $\bar{r}$ for which

$$V_0^{\bar{\mathcal{M}}_t,*}(s_0) \geq H \rho_{\bar{\mathcal{M}}_t,\pi^+}^\top \bar{r} > H \sup_\pi \rho_{\mathcal{M}_t,\pi}^\top \bar{r} = V_0^{\mathcal{M}_t,*}(s_0).$$

Thus, there exists a non-zero value gap between $\mathcal{M}_t$ and $\bar{\mathcal{M}}_t$ with reward function r, as assumed by Assumption 4.1.

4.3 Algorithm Outline

In this section, we outline our exit detection algorithm. Our procedure consists of a task-solving phase, a reward-free phase, and an exit detection phase. We also illustrate our steps in the context of Figure 5, outline the algorithm in Algorithm 2, and provide the full algorithm in Section A.1. In addition, we experimentally demonstrate successful exit detection via our proposed approach in Section C.

4.3.1 Phase I: Task-Specific Dynamics Learning

The first step in our algorithm is to solve each individual meta-training task using UCBVI (Azar et al., 2017) to obtain near-optimal policies. Via Lemma 4.1, data obtained from these policies can then be used to estimate the dynamics of all α-important state-action pairs. This allows the learner to borrow high-quality estimates of high-importance exit dynamics when applying optimistic imagination. To illustrate the contribution of this portion of the algorithm in the context of Figure 4, note that solving the first task allows us to learn the dynamics of the blue gate when open, as Figure 6 shows.

[3] Note that the coverage condition does not consider the preliminary condition at all. Incorporating this condition can weaken Assumption 4.1, so that it only has to hold for subsets of exits not covered under the preliminary condition. However, our algorithm can trivially accomodate this combined assumption.

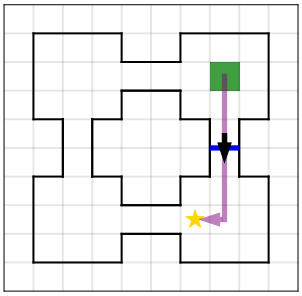

Figure 6: Phase I contribution to learning $\pi_{\mathcal{I}}$ in Figure 5. Solving other tasks allows the agent to see the open gate.

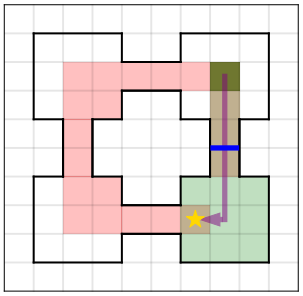

Figure 7: Phase II. Optimal policy state coverage (red) is insufficient for learning $\pi_{\mathcal{I}}$, which requires data from the green region.

4.3.2 Phase II: Reward-Free RL

To perform optimistic imagination, the learner not only needs to be able to borrow high-quality dynamics models from other tasks, but also be able to simulate state-action pairs that are not being borrowed. Since only exits can change between tasks, non-exit state-action pairs can be simulated using data *from any task*. Formally, we learn a single "template" dynamics model $\hat{\mathbb{P}}_0$ by applying reward-free RL (Jin et al., 2020) to a single task.

To further illustrate the necessity of this phase, Figure 7 highlights the state coverage of near-optimal policies in red. In particular, most states beyond the blue gate are not covered by these policies. However, accurately evaluating the imagined policy $\pi_{\mathcal{I}}$ in Figure 5 not only requires seeing the blue gate when opened, but also knowing the highlighted green region in Figure 7. In this context, Phase II allows the learner to also simulate the green region's dynamics, and evaluate the value of $\pi_{\mathcal{I}}$ under optimistic borrowing.

4.3.3 Phase III: Exit Detection

Having completed Phases I and II, the learner now has sufficient estimates to perform optimistic imagination. This is done by modifying value iteration (VI) to optimistically borrow dynamics estimates from other tasks.

Formally, we first partition $\mathcal{S} \times \mathcal{A}$ into "known exits" (initially empty) and "candidates" (initially $\mathcal{S} \times \mathcal{A}$). For every (s, a)-pair and timestep of the Bellman backup in a task $\mathcal{M}_t$ during VI, we either use $\hat{\mathbb{P}}_t(\cdot \mid s, a)$ if (s, a) is a known exit[4], or any of the estimates $\hat{\mathbb{P}}_b(\cdot \mid s, a)$ for $b \in \{0, \ldots, T\}$ otherwise. Since only exits can change dynamics, this procedure only results in significant value overestimation if an undiscovered exit's dynamics is borrowed from a different task. We can show that modifying VI in this way performs an optimistic search over policies and MDPs, where Assumption 4.1(b) guarantees the feasibility of an MDP with a detectable overestimated value. As with α-importance, this value gap implies that the policy $\pi_{\mathcal{I}}$ obtained by this procedure visits an undetermined exit whose dynamics are borrowed during VI. Upon learning the dynamics of this discovered exit in every task and marking it as a known exit, we continue the process until no task has siginificantly overestimated values.

[4]Since (s, a) is a known exit, we know that (s, a)'s dynamics are task-dependent, so we have to use the estimates for task t.

Algorithm 2 An outline of our hierarchy learning procedure during the meta-training phase.

Require: Meta-training MDPs $\mathcal{M}_1, \ldots, \mathcal{M}_T$
(Phase I) Apply UCBVI to learn all MDPs to appropriate level of accuracy (dependent on α-importance)
(Phase II) Apply reward-free RL from Jin et al. (2020) to $\mathcal{M}_1$ (or any other arbitrarily chosen meta-training MDP).
while True **do**
  (Phase III) Choose one of the meta-training MDPs, and apply value iteration (modified to perform optimistic imagination/borrowing using estimates from prior phases)
  if value returned is much higher compared to UCBVI applied to the same task **then**
    Run policy from value iteration and form new dynamics estimate, compare with previous estimates using TV-norm to find new exit
    Learn the dynamics of the new exit in all tasks
  end if
  if no value overestimation after going through all T MDPs since last found exit **then**
    break from while loop
  end if
end while
output Set of found exits, transition dynamics estimates

4.4 Meta-Training Guarantee

We outline our meta-training guarantee in this section. In particular, we can show that the algorithm in the previous section can be used to implement a particularly useful oracle for downstream tasks. In the following, we will define two new auxiliary states, `GOAL` and `FAIL`.

Definition 4.3 (Hierarchy oracle)**.** An *ε-suboptimal hierarchy oracle* is an oracle such that given $(x, f, r, \tilde{H})$, where

- $x \in \mathrm{Ent}(\mathcal{C})$ is some starting state,
- $f : \mathrm{Ext}(\mathcal{C}) \to \{\texttt{GOAL}, \texttt{FAIL}\}$ is an exit-labeling function,
- r is a reward function, and
- $\tilde{H} \leq H$ is a horizon,

the oracle returns an ε-suboptimal policy for the tuple-induced MDP $(\mathcal{S} \cup \{\texttt{GOAL}, \texttt{FAIL}\}, \mathcal{A}, \mathbb{P}_f, r, \tilde{H})$ with starting state x and dynamics $\mathbb{P}_f$ given by

$$\mathbb{P}_f(\cdot \mid s, a) = \begin{cases} \delta(f(s,a)) & (s,a) \in \mathrm{Ext}(\mathcal{C}) \\ \delta(s) & s \in \{\texttt{GOAL}, \texttt{FAIL}\} \\ \mathbb{P}_t(\cdot \mid s, a) & \text{otherwise, any } t \in [T] \end{cases}. \quad \diamond$$

To illustrate how the hierarchy oracle can be used, consider the problem of trying to reach a particular exit e within the current cluster, given that the agent had just entered the cluster and is at entrance x. For MDPs consistent with the shared latent hierarchy, this is a useful skill as exits are bottlenecks to other clusters. The hierarchy oracle solves this subproblem when queried with f that labels e as GOAL and all other exits as FAIL, and setting the reward to $r(s,a) = \mathbb{1}[s = \texttt{GOAL}]$. Observe that the clusters of $\mathcal{C}$ are disconnected in the induced MDP solved by the hierarchy oracle, and thus the policy will stay within a single cluster while solving the subproblem. As an added bonus, if it is unknown a priori that e and x belong to the same cluster, the value predicted by the oracle for the induced MDP can be used to determine reachability.[5] Our guarantee ensures that this hierarchy oracle is implementable:

Theorem 4.1 (Meta-training guarantee, informal)**.** *Under Assumption 4.1 and other assumptions in Section A.3, the data obtained from the algorithm in Section 4.3 allows for:*

(a) implementing an ε-suboptimal hierarchy oracle, and

(b) determining, for every $s \in \mathrm{Ent}(\mathcal{S})$, the reachable exits in the cluster containing s,

simultaneously with probability at least $1 - p$. Furthermore, this is achieved with query complexity

$$\tilde{O}\left[T\left(\frac{KL}{\alpha \min(\zeta,\beta)^2} + \frac{KS}{\alpha\zeta^2} + \frac{SA}{\min(\alpha,\zeta)^2} + \frac{KS^2A}{\alpha}\right) + \frac{S^4A}{\min(\varepsilon,\zeta)} + \frac{S^2A}{\min(\varepsilon,\zeta)^2}\right] \mathrm{poly}(H).$$

As a point of comparison, we have the following guarantee on brute-force hierarchy learning:

Theorem 4.2. *The brute-force approach outlined in Section A.6, under Assumption 4.1(a), determines the set of exits with high probability and query complexity*

$$\tilde{O}\left[T\left(\frac{S^2A}{\alpha\beta^2} + \frac{SA}{\alpha^2} + \frac{S^4A}{\alpha}\right)\right] \mathrm{poly}(H).$$

When α, β, and ζ are of the same order, we see that the proposed method incurs a smaller query complexity compared to a brute force learner that has only learned the exits. We provide proofs of both results in Section A, along with all other necessary assumptions and full algorithm details.

[5] More specifically, the predicted value can only be non-zero if exit e is reachable from x without moving to other clusters.

5 META-TEST ANALYSIS

In this section, we bound the regret of an agent using a hierarchy oracle during meta-testing. We first characterize a family of tasks for which one can achieve improved regret bounds. Furthermore, we provide sufficient conditions ensuring that using the hierarchy incurs low suboptimality.

5.1 Assumptions

Recall that we evaluate the regret an agent incurs on $\mathcal{M}_{T+1} = (\mathcal{S}, \mathcal{A}, \mathbb{P}_{T+1}, r_{T+1}, H)$. For meta-learning to succeed, we need to restrict $\mathcal{M}_{T+1}$, as the hierarchy cannot be compatible with all possible downstream tasks. We have the following sufficient condition for compatibility with the learned hierarchy:

Assumption 5.1 (Task Compatibility)**.** *There exists a cluster Z^* such that r_{T+1} is supported on $Z^* \cup \mathrm{Ext}(\mathcal{C})$. Furthermore, there exists an optimal policy π^* satisfying*

(a) *Conditioned on $s_h \in Z^*$, we have that $(s_{h'}, a_{h'}) \notin \mathrm{Ext}(Z^*)$ for $h' \geq h$ almost surely.*

(b) *The number of exits encountered by π^* is bounded by H_{eff} with probability ζ.*

Hierarchical compatibility. Intuitively, Condition (a) and the reward assumption in Assumption 5.1 suggests that the task can be decomposed into (1) navigating exits to reach Z^* and (2) optimally collecting rewards within Z^*. As the learner is able to use the hierarchy oracle to reach exits and plan within clusters, the hierarchy oracle reduces the complexity of exploration in the two phases. We note that decomposing a task in the manner described above is characteristic of the goal-conditioned RL setting, which has been studied extensively in recent empirical works (Nachum et al., 2019a; Levy et al., 2018; Nachum et al., 2018), and involves an agent navigating to a particular goal location.

Temporal Abstraction. As noted above, Condition (a) allows the agent to abstract away the planning process so that it only needs to either select an exit to use, or remain within the current cluster for the rest of the episode. Thus, using the hierarchy oracle reduces the *effective planning horizon* of the agent. Condition (b) quantifies this reduction. Informally, we expect that most tasks of interest are solvable via a short sequence of skills that can be correctly executed with high probability by a near-optimal agent.

Hierarchical Suboptimality. Restricting the learner to executing hierarchical policies can greatly reduce the search space. While this reduction leads to improved regret bounds, this also incurs approximation error, as the optimal policy may not lie in this restricted class. We refer to this error as *hierarchical suboptimality*. We will show that hierarchical suboptimality is controllable with certain properties

of $\mathbb{P}_{T+1}$, which require the following notions of reaching times:

Definition 5.1 (Reaching times)**.** Let s be a starting state and g be a goal state, both belonging to a single cluster Z, and fix a horizon $\tilde{H}$. Furthermore, for a policy π, let $(s_0, s_1, \ldots, s_{\tilde{H}})$ be the (random) states visited by rolling out π for $\tilde{H}$ steps from s, so that $s_0 = s$. We then define the (random) truncated reaching time $T^{\pi}_{\tilde{H}}(s, g)$ as

$$T^{\pi}_{\tilde{H}}(s, g) \coloneqq \min\{h \mid s_h = g, (s_{0:h}) \subseteq Z\} \cup \{\tilde{H}\}.$$

Thus, $T^{\pi}_{\tilde{H}}(s, g)$ is the time to reach g from s, while remaining in the same cluster (truncated to $\tilde{H}$). Additionally, we write

$$T^{*}_{\tilde{H}}(s, g) \coloneqq \inf_{\pi} \mathbb{E}\left[T^{\pi}_{\tilde{H}}(s, g)\right]$$
$$T^{\min}(s, g) \coloneqq \inf_{\pi} \min\{h \mid P(T^{\pi}_{\infty}(s, g) = h) > 0\}.$$

In particular, $T^{\min}$ is the minimum time for which it is possible to reach g from s. ◇

We now proceed with describing the relevant property:

Definition 5.2 (Regular and low-variance dynamics)**.** There exists $\eta, \kappa, \gamma > 0$ such that for any cluster Z, states $s, g \in Z$, and horizon $\tilde{H} < H$,

(a) ((η, κ)-unreliability) For any deterministic policy π with $\mathbb{E}[T^{\pi}_{\tilde{H}}(s, g)] - T^{*}_{\tilde{H}}(s, g) < \eta$, $T^{\pi}_{\tilde{H}}(s, g)$ has a sub-Gaussian upper tail with variance proxy $\kappa^2 \mathbb{E}[T^{\pi}_{\tilde{H}}(s, g)]^2$.

(b) (γ-goal-reaching suboptimality) $T^{*}_{\tilde{H}}(s, g) \leq (1 + \gamma) T^{\min}(s, g)$. ◇

Since (η, κ)-unreliability only considers near-optimal deterministic policies, the condition can informally be thought of as quantifying the randomness in reaching time due to the inherent randomness in the dynamics. On the other hand, γ-goal-reaching suboptimality measures whether near-optimal goal-reaching policies nearly achieve the minimum possible reaching time. Deterministic environments satisfies these conditions with $\eta = \infty$ and $\kappa = \gamma = 0$. We provide an extended discussion of why these quantities control hierarchical suboptimality in Section B.5. Note that the guarantees of Definition 5.2 scales with the cluster width, and thus we have the following final assumption:

Assumption 5.2. *There exists a $W > 0$ such that for any cluster Z and $s, g \in Z$, $T^{\min}(s, g) \leq W$, and $H_{\mathrm{eff}} W \ll H$.*

Assumption 5.2 limits the length of the subtasks within each cluster. This is consistent with hierarchy-based methods in practice, with skills only being executed for a limited amount of time. This width bound, together with Assumption 5.1, suggests that π^* requires $O(H_{\mathrm{eff}} W)$ timesteps to reach Z^* with high probability, which is much smaller than H by assumption.[6]

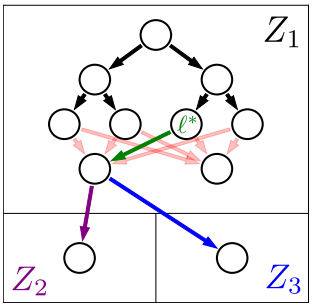

Figure 8: The hard instance in Theorem 5.2. Randomization of ℓ^* forces hierarchy-oblivious learners to explore the whole tree.

5.2 Meta-test Regret Guarantee

Assumption 5.1 allows us to consider a "high-level" MDP wherein entrances serve as abstracted states and exits are abstracted actions, which can be solved by any arbitrarily chosen RL algorithm. We formalize this in Section B. In our case, using regret bounds on EULER by Zanette and Brunskill (2019), we can obtain the following regret bound:

Theorem 5.1. *We work under Assumptions 5.1 and 5.2. Furthermore, assume that the learner has access to an ε-suboptimal hierarchy oracle as guaranteed by Theorem 4.1, where $\varepsilon < \eta$. Then, a learner that applies the procedure in Section B.2 to $\mathcal{M}_{\mathrm{Tg}}$, with high probability, incurs regret*

$$\mathrm{Regret}(N) \lesssim \sqrt{H^2 H_{\mathrm{eff}} W L M N} + N \varepsilon_{\mathrm{subopt}},$$

where

$$\varepsilon_{\mathrm{subopt}} \coloneqq \zeta H + (1 + H_{\mathrm{eff}} + \kappa\sqrt{H_{\mathrm{eff}}})\varepsilon + \left[\gamma H_{\mathrm{eff}} + \kappa(1 + \gamma)\sqrt{H_{\mathrm{eff}}}\right] W.$$

We note that the hierarchical suboptimality term $\varepsilon_{\mathrm{subopt}}$ decomposes into three terms corresponding to errors from temporal abstraction, meta-training, and noise, respectively. Furthermore, the irreducible hierarchical suboptimality (i.e. when $\varepsilon = 0$) tends to zero as $\gamma, \kappa, \zeta \to 0$. In particular, environments with deterministic dynamics within clusters do not incur hierarchical suboptimality. We prove this regret bound in Section B.

When does knowing the hierarchy help? Consider the binary tree environment in Figure 8. All of the leaves take the learner to a state with exits with probability $1/2$, with the exception of a special leaf ℓ^* that does so with probability $(1/2) + \varepsilon$. Rewards can only be collected upon performing one of the exits (blue/purple). To achieve low regret, a learner has to quickly identity ℓ^* and the correct exit.

We consider the set of task distributions indexed by ℓ^* that randomize the reward-granting exit. Knowing the hierarchy

[6] In practice, the task horizon is often much longer than what is needed to solve the task, which is consistent with this assumption.

amounts to knowing ℓ^*, reducing the exploration problem to determining the correct exit. However, a hierarchy-oblivious learner needs to explore the tree, leading to regret that is exponential in the tree depth. Formally, we have the following result:

Theorem 5.2. *There exists a family of task distributions such that any hierarchy-oblivious learner incurs expected regret lower bounded by $\Omega(2^{W/2}\sqrt{H^2N})$ on at least one task distribution. In contrast, a learner with access to a 0-suboptimal[7] hierarchy oracle incurs regret bounded by $O(\sqrt{H^2N})$ with high probability, over any sampled task from any of the task distributions.*

We prove this result in Section B.4.3, using a result by Domingues et al. (2021) which shows that the set of binary tree subproblems above form a set of minimax instances for any RL algorithm. This separation result suggests that hierarchy-based learners gain in situations where temporally extended exploratory behaviors are needed. This corroborates the experimental findings of Nachum et al. (2019b), which attributes the benefits of hierarchical RL to improved exploration.

6 RELATED WORK

Hierarchical reinforcement learning has been studied extensively (Sutton et al., 1999; Parr and Russell, 1998; Dietterich et al., 1998; Vezhnevets et al., 2017). An early approach formalizing the use of hierarchies in RL is the options framework (Sutton et al., 1999), which fixes a finite set of available skills/options. Since then, a large body of work has focused on designing methods for learning and adapting these options during the learning process (McGovern and Barto, 2001; Menache et al., 2002; Şimşek and Barto, 2004; Mann et al., 2014). Of particular note is the work of Frans et al. (2018), which learns a finite set of neural network sub-skills in the meta-RL setting. On the other hand, Laplacian-based option discovery in Machado et al. (2017, 2018) defines options using proto-value functions (Mahadevan, 2005), which captures global features of the state space. In more theoretical directions, Fruit and Lazaric (2017); Brunskill and Li (2014) provide regret and sample complexity bounds, respectively, for learning with options. In particular, Brunskill and Li (2014) first prove a combined sample complexity bound on the meta-learning process, assuming the existence of a small option set that is sufficient for near-optimal behavior, and subsequently propose an option discovery algorithm inspired by their PAC bound. In contrast, our work starts with a natural but specific notion of shared structure, which we show can be leveraged for provable option learning, and provide conditions quantifying their usefulness for downstream tasks. Additionally, Mann and Mannor (2014) demonstrate that options can improve the convergence rate of approximate value iteration.

More recent empirical works have studied hierarchy learning beyond the options framework in a wide variety of settings. Nachum et al. (2019a); Levy et al. (2018) provide algorithms for learning hierarchies based on goal-conditioned policies, reducing the learning problem to choosing subgoals. Nachum et al. (2018) considers a more general case when learned representations are used to map states to goals. Other works such as Co-Reyes et al. (2018); Eysenbach et al. (2018); Sharma et al. (2019) provide intrinsic objectives for learning hierarchies without rewards.

Closely related is the work of Wen et al. (2020), which introduces a similar latent structure on the state space, and provides sufficient conditions ensuring sample-efficient and tractable hierarchy-based learning. However, their works hinges on prior knowledge of the latent structure, while a major focus of our work is discovering the structure itself from interactions.

Several heuristics have been proposed in prior work for the detection of useful bottlenecks in MDPs (McGovern and Barto, 2001; Menache et al., 2002; Şimşek and Barto, 2004; Şimşek et al., 2005). In particular, Şimşek and Barto (2004) define the notion of *access states*, states which maximize short-term novelty of future states. Although exits as defined in Definition 3.2 meet this heuristic, we note that cluster interiors may also contain bottlenecks also meet these conditions. In contrast, our algorithm would not detect these bottlenecks; however, this behavior is desirable as such bottlenecks are unimportant for meta-learning.

7 CONCLUSION

We have demonstrated that certain natural coverage conditions allow for learning useful hierarchies from tasks. Interesting future directions include analyzing hierarchy-based multi-task RL and extending the ideas in this work to continuous state and/or action spaces. Another interesting direction would be to provide sample-efficient algorithms for learning additional structures that can be imposed on the learned hierarchy, such as cluster equivalences as in Wen et al. (2020).

Societal Impact Statement. As our work provides theoretical analyses of existing algorithms, we do not foresee any negative social impacts arising from the misuse of the paper's contribution.

Acknowledgements

KC is supported by a National Science Foundation Graduate Research Fellowship, Grant DGE-2039656. JDL acknowledges support of the the Sloan Research Fellowship, NSF CCF 2002272, NSF IIS 2107304, and NSF CAREER Award

[7] We use a 0-suboptimal hierarchy oracle for the separation result for ease of presentation.

2144994. Additionally, we thank Aurick Zhou for discussions and feedback.

References

Azar, M. G., Osband, I., and Munos, R. (2017). Minimax regret bounds for reinforcement learning. In *International Conference on Machine Learning*, pages 263–272. PMLR.

Brunskill, E. and Li, L. (2014). Pac-inspired option discovery in lifelong reinforcement learning. In *International conference on machine learning*, pages 316–324. PMLR.

Co-Reyes, J., Liu, Y., Gupta, A., Eysenbach, B., Abbeel, P., and Levine, S. (2018). Self-consistent trajectory autoencoder: Hierarchical reinforcement learning with trajectory embeddings. In *International Conference on Machine Learning*, pages 1009–1018. PMLR.

Dietterich, T. G. et al. (1998). The maxq method for hierarchical reinforcement learning. In *ICML*, volume 98, pages 118–126. Citeseer.

Domingues, O. D., Ménard, P., Kaufmann, E., and Valko, M. (2021). Episodic reinforcement learning in finite mdps: Minimax lower bounds revisited. In *Algorithmic Learning Theory*, pages 578–598. PMLR.

Du, S. S., Hu, W., Kakade, S. M., Lee, J. D., and Lei, Q. (2020). Few-shot learning via learning the representation, provably. *arXiv preprint arXiv:2002.09434*.

Ecoffet, A., Huizinga, J., Lehman, J., Stanley, K. O., and Clune, J. (2021). First return, then explore. *Nature*, 590(7847):580–586.

Eysenbach, B., Gupta, A., Ibarz, J., and Levine, S. (2018). Diversity is all you need: Learning skills without a reward function. In *International Conference on Learning Representations*.

Eysenbach, B., Salakhutdinov, R., and Levine, S. (2021). The information geometry of unsupervised reinforcement learning. *arXiv preprint arXiv:2110.02719*.

Frans, K., Ho, J., Chen, X., Abbeel, P., and Schulman, J. (2018). Meta learning shared hierarchies. In *International Conference on Learning Representations*.

Fruit, R. and Lazaric, A. (2017). Exploration-exploitation in mdps with options. In *Artificial Intelligence and Statistics*, pages 576–584. PMLR.

Jin, C., Krishnamurthy, A., Simchowitz, M., and Yu, T. (2020). Reward-free exploration for reinforcement learning. In *International Conference on Machine Learning*, pages 4870–4879. PMLR.

Levy, A., Konidaris, G., Platt, R., and Saenko, K. (2018). Learning multi-level hierarchies with hindsight. In *International Conference on Learning Representations*.

Machado, M. C., Bellemare, M. G., and Bowling, M. (2017). A laplacian framework for option discovery in reinforcement learning. In *International Conference on Machine Learning*, pages 2295–2304. PMLR.

Machado, M. C., Rosenbaum, C., Guo, X., Liu, M., Tesauro, G., and Campbell, M. (2018). Eigenoption discovery through the deep successor representation. In *International Conference on Learning Representations*.

Mahadevan, S. (2005). Proto-value functions: Developmental reinforcement learning. In *Proceedings of the 22nd international conference on Machine learning*, pages 553–560.

Mann, T., Mankowitz, D., and Mannor, S. (2014). Time-regularized interrupting options (trio). In *International Conference on Machine Learning*, pages 1350–1358. PMLR.

Mann, T. and Mannor, S. (2014). Scaling up approximate value iteration with options: Better policies with fewer iterations. In *International conference on machine learning*, pages 127–135. PMLR.

McGovern, A. and Barto, A. G. (2001). Automatic discovery of subgoals in reinforcement learning using diverse density. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 361–368.

Menache, I., Mannor, S., and Shimkin, N. (2002). Q-cut—dynamic discovery of sub-goals in reinforcement learning. In *European Conference on Machine Learning*, pages 295–306. Springer.

Nachum, O., Gu, S., Lee, H., and Levine, S. (2018). Near-optimal representation learning for hierarchical reinforcement learning. In *International Conference on Learning Representations*.

Nachum, O., Gu, S., Lee, H., and Levine, S. (2019a). Data-efficient hierarchical reinforcement learning. In *32nd Conference on Neural Information Processing Systems (NeurIPS 2018)*, pages 3303–3313. Curran Associates, Inc.

Nachum, O., Tang, H., Lu, X., Gu, S., Lee, H., and Levine, S. (2019b). Why does hierarchy (sometimes) work so well in reinforcement learning? *arXiv preprint arXiv:1909.10618*.

Parr, R. and Russell, S. (1998). Reinforcement learning with hierarchies of machines. *Advances in neural information processing systems*, pages 1043–1049.

Schrittwieser, J., Antonoglou, I., Hubert, T., Simonyan, K., Sifre, L., Schmitt, S., Guez, A., Lockhart, E., Hassabis, D., Graepel, T., et al. (2020). Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588(7839):604–609.

Schulman, J., Levine, S., Abbeel, P., Jordan, M., and Moritz, P. (2015). Trust region policy optimization. In *Interna-*

tional conference on machine learning, pages 1889–1897. PMLR.

Sharma, A., Gu, S., Levine, S., Kumar, V., and Hausman, K. (2019). Dynamics-aware unsupervised discovery of skills. In *International Conference on Learning Representations*.

Şimşek, Ö. and Barto, A. G. (2004). Using relative novelty to identify useful temporal abstractions in reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*, page 95.

Şimşek, Ö., Wolfe, A. P., and Barto, A. G. (2005). Identifying useful subgoals in reinforcement learning by local graph partitioning. In *Proceedings of the 22nd international conference on Machine learning*, pages 816–823.

Sutton, R. S., Precup, D., and Singh, S. (1999). Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence*, 112(1-2):181–211.

Tripuraneni, N., Jordan, M., and Jin, C. (2020). On the theory of transfer learning: The importance of task diversity. *Advances in Neural Information Processing Systems*, 33.

Vezhnevets, A. S., Osindero, S., Schaul, T., Heess, N., Jaderberg, M., Silver, D., and Kavukcuoglu, K. (2017). Feudal networks for hierarchical reinforcement learning. In *International Conference on Machine Learning*, pages 3540–3549. PMLR.

Vinyals, O., Babuschkin, I., Czarnecki, W. M., Mathieu, M., Dudzik, A., Chung, J., Choi, D. H., Powell, R., Ewalds, T., Georgiev, P., et al. (2019). Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, 575(7782):350–354.

Wang, J. X., King, M., Porcel, N., Kurth-Nelson, Z., Zhu, T., Deck, C., Choy, P., Cassin, M., Reynolds, M., Song, F., et al. (2021). Alchemy: A structured task distribution for meta-reinforcement learning. *arXiv preprint arXiv:2102.02926*.

Wen, Z., Precup, D., Ibrahimi, M., Barreto, A., Van Roy, B., and Singh, S. (2020). On efficiency in hierarchical reinforcement learning. *Advances in Neural Information Processing Systems*, 33.

Zanette, A. and Brunskill, E. (2019). Tighter problem-dependent regret bounds in reinforcement learning without domain knowledge using value function bounds. In *International Conference on Machine Learning*, pages 7304–7312. PMLR.

A META-TRAINING PROOFS

A.1 Algorithm

In this section, we provide the complete algorithm for exit detection with optimistic imagination. For readability, we separate the three phases.

A.1.1 Phase I: Task-Specific Learning

Algorithm 3 Exit Detection, Phase I: Task-Specific Learning

Require: Tasks $\mathcal{M}_1, \ldots, \mathcal{M}_T$, N_{UCBVI} UCBVI iterations, N_{TS} policy samples, threshold $N_{\mathrm{thresh}}^{\mathrm{TS}}$

for all $t \in [T]$ **do**
  $\mathcal{D}^t \leftarrow \varnothing$.
  Obtain policy set $\Phi^t \leftarrow \mathrm{UCBVI}(\mathcal{M}_t, N_{\mathrm{UCBVI}})$.
  for all $n = 1, \ldots, N_{\mathrm{TS}}$ **do**
    Sample $\pi \sim \mathrm{Unif}\,(\Phi^t)$.
    Play π in $\mathcal{M}_t$, add all (s, a, s') pairs to $\mathcal{D}^t$, get sum of rewards $\hat{V}^{(n)}$.
  end for
  Form estimated dynamics model $\hat{\mathbb{P}}_t$ from $\mathcal{D}_t$.
  Form optimal value estimate $\hat{V}_t \leftarrow \dfrac{1}{N_{\mathrm{TS}}} \displaystyle\sum_{n=1}^{N_{\mathrm{TS}}} \hat{V}^{(n)}$
  $N_t(s, a) \leftarrow |\{(x, u, x') \in \mathcal{D}_t \mid x = s, u = a\}|$.
  for all $(s, a) \in \mathcal{S} \times \mathcal{A}$ **do**
    if $N_t(s, a) < N_{\mathrm{thresh}}^{\mathrm{TS}}$ **then**
      $\hat{\mathbb{P}}_t(\cdot \mid s, a) \leftarrow 0$.
    end if
  end for
end for

output dynamics estimates $\hat{\mathbb{P}}_t$ and value estimates $\hat{V}_t$ for $t \in [T]$.

A.1.2 Phase II: Learning Reference Dynamics

Algorithm 4 Exit Detection, Phase II: Learning Reference Dynamics

Require: MDP $\mathcal{M}$, $N_{\mathrm{EULER}}^{\mathrm{RF}}$ Euler iterations, N_{RF} policy samples

Set policy class $\Psi \leftarrow \varnothing$ and dataset $\mathcal{D}_{\mathrm{RF}} \leftarrow \varnothing$
for all $g \in \mathcal{S}$ **do**
  Create MDP $\mathcal{M}_g$ from $\mathcal{M}$ with horizon $2H$ and $P(\ominus \mid g, a) = 1$ for any a.
  $r_g(s, a) \leftarrow \mathbb{1}[s = g]$ for any $(s, a) \in (\mathcal{S} \cup \{\ominus\}) \times \mathcal{A}$.
  $\Phi^g \leftarrow \textsc{Euler}(\mathcal{M}_g, r_g, N_{\mathrm{RF}}^{\mathrm{EULER}})$
  $\pi_h(\cdot \mid g) \leftarrow \mathrm{Unif}\,(\mathcal{A})$ for $\pi \in \Phi^g$, $h \in [H]$.
  Add policies in Φ^g to Ψ.
end for
for all $n = 1, \ldots, N_{\mathrm{RF}}$ **do**
  Sample $\pi \sim \mathrm{Unif}\,(\Psi)$.
  Play π in $\mathcal{M}$ and obtain trajectory $(s_0, a_0, \ldots, s_{2H})$.
  Sample $h \sim \mathrm{Unif}\,([2H])$ and add (s_h, a_h, s_{h+1}) to $\mathcal{D}_{\mathrm{RF}}$.
end for

output reference dynamics $\hat{\mathbb{P}}_0$ formed from $\mathcal{D}_{\mathrm{RF}}$.

A.1.3 Phase III: Exit Detection

Algorithm 5 Exit Detection, Phase III: Exit Detection

Require: $N_{\mathrm{ED}}, N_{\mathrm{thresh}}^{\mathrm{ED}}, N_{\mathrm{EULER}}^{\mathrm{EL}}, N_{\mathrm{EL}}$ policy samples
Initialize $\textsc{IsExit}[s,a] \leftarrow \textsc{False}$ for $(s,a) \in \mathcal{S} \times \mathcal{A}$.
while True **do**
  for all $t \in [T]$ **do**
    $\hat{\mathbb{P}}_0(\cdot \mid s,a) \leftarrow \hat{\mathbb{P}}_t(\cdot \mid s,a)$ for $(s,a) \in \mathcal{S} \times \mathcal{A}$ with $\textsc{IsExit}[s,a]$.
    $\tilde{V}^t, \tilde{Q}^t \leftarrow \textsc{OptImgVI}(\hat{\mathbb{P}}_0, (\hat{\mathbb{P}}_1, \ldots, \hat{\mathbb{P}}_T), r_t, \textsc{IsExit})$
    if $\tilde{V}_0^t(s_0) - \hat{V}_t > (2/3)\zeta$ **then**
      Run greedy policy with respect to $\tilde{Q}$ N_{ED} times and form estimate $\hat{\mathbb{P}}$ for (s,a) pairs visited at least N'_{thresh} times.
      for all $(s,a) \in \mathcal{S} \times \mathcal{A}$ with $\hat{\mathbb{P}}(\cdot \mid s,a) \not\equiv 0$ **do**
        if $(\exists t \in [T])\ \hat{\mathbb{P}}_t(\cdot \mid s,a) \not\equiv 0$ and $\left\|\hat{\mathbb{P}}(\cdot \mid s,a) - \hat{\mathbb{P}}_t(\cdot \mid s,a)\right\|_{\mathrm{TV}} > \beta/2$ **then**
          $\textsc{IsExit}[s,a] \leftarrow$ True
          $\hat{\mathbb{P}}_t(\cdot \mid s,a) \leftarrow \textsc{Learn-Exit}(\mathcal{M}_t, (s,a), N_{\mathrm{EULER}}^{\mathrm{EL}}, N_{\mathrm{EL}})$ **for all** $t \in [T]$
        end if
      end for
    end if
    if no new exits found after passing through T tasks since last found exit **then**
      Break loop
    end if
  end for
end while
output $\textsc{IsExit}$

Algorithm 6 **B**orrowing **O**ptimistically **A**cross **T**asks during **V**alue **I**teration (BOAT-VI)

Require: Reference dynamics $\hat{\mathbb{P}}_0$, Estimated dynamics $(\hat{\mathbb{P}}_1, \ldots, \hat{\mathbb{P}}_T)$, Reward function r, Table $\textsc{IsExit}[\mathcal{S} \times \mathcal{A}]$
$\hat{V}_H(s) \leftarrow 0$ for $s \in \mathcal{S}$.
for all $h = H-1, \ldots, 0$ **do**
  for all $(s,a) \in \mathcal{S} \times \mathcal{A}$ **do**
    if $\textsc{IsExit}[s,a]$ **then**
      $\hat{Q}_h(s,a) \leftarrow r(s,a) + \hat{\mathbb{P}}_0 \hat{V}_{h+1}(s,a)$
    else
      $\hat{Q}_h(s,a) \leftarrow r(s,a) + \max_{t=0,\ldots,T} \hat{\mathbb{P}}_t \hat{V}_{h+1}(s,a)$
    end if
  end for
  $\hat{V}_h(s) \leftarrow \max_a \hat{Q}_h(s,a)$ for $s \in \mathcal{S}$.
end for
output $\hat{V}, \hat{Q}$

Algorithm 7 Exit-learning subroutine

Require: MDP $\mathcal{M}$, exit (s,a), $N_{\mathrm{EULER}}^{\mathrm{EL}}$ EULER iterations, N_{EL} policy samples
Create MDP $\tilde{\mathcal{M}}$ from $\mathcal{M}$ so $P(\ominus \mid s,a) = 1$.
$\tilde{r}(s',a') \leftarrow \mathbb{1}\left[(s',a') = (s,a)\right]$ for any $(s',a') \in (\mathcal{S} \cup \{\ominus\}) \times \mathcal{A}$.
$\Psi \leftarrow \textsc{Euler}(\tilde{\mathcal{M}}, \tilde{r}, N_{\mathrm{EULER}}^{\mathrm{EL}})$
for all $n = 1, \ldots, N_{\mathrm{EL}}$ **do**
  Sample $\pi \sim \mathrm{Unif}\,(\Psi)$.
  Play π in $\mathcal{M}$ and obtain trajectory $(s_0, a_0, \ldots, s_H)$.
end for
output reference dynamics $\hat{\mathbb{P}}(\cdot \mid s,a)$ formed from all trajectory data.

A.2 Subroutines from Prior Work

In order to keep our work self-contained, we also include the following subroutines from prior work:

Algorithm 8 Computing Q-values with bonuses for UCBVI from Azar et al. (2017).

Require: Bonus algorithm `bonus`, Data $\mathcal{H}$

Compute, for all $(x, a, y) \in \mathcal{S} \times \mathcal{A} \times \mathcal{S}$,
$N_k(x, a, y) = \sum_{(x', a', y') \in \mathcal{H}} \mathbb{I}(x' = x, a' = a, y' = y)$
$N_k(x, a) = \sum_{y \in \mathcal{S}} N_k(x, a, y)$
$N'_{k,h}(x, a) = \sum_{(x_{i,h}, a_{i,h}, x_{i,h+1}) \in \mathcal{H}} \mathbb{I}(x_{i,h} = x, a_{i,h} = a)$
Let $\mathcal{K} = \{(x, a) \in \mathcal{S} \times \mathcal{A},\ N_k(x, a) > 0\}$
Estimate $\widehat{P}_k(y|x, a) = \frac{N_k(x,a,y)}{N_k(x,a)}$ for all $(x, a) \in \mathcal{K}$
Initialize $V_{k,H+1}(x) = 0$ for all $(x, a) \in \mathcal{S} \times \mathcal{A}$
for $H = H, H - 1, \ldots, 1$ **do**
  for $(X, a) \in \mathcal{S} \times \mathcal{A}$ **do**
    if $(x, a) \in \mathcal{K}$ **then**
      $b_{k,h}(x, a) = \texttt{bonus}(\widehat{P}_k, V_{k,h+1}, N_k, N'_{k,h})$
      $Q_{k,h}(x, a) = \min\big(Q_{k-1,h}(x, a), H,$
        $R(x, a) + (\widehat{P}_k V_{k,h+1})(x, a) + b_{k,h}(x, a)\big)$
    else
      $Q_{k,h}(x, a) = H$
    end if
    $V_{k,h}(x) = \max_{a \in \mathcal{A}} Q_{k,h}(x, a)$
  end for
end for

output Q-values $Q_{k,h}$

Algorithm 9 UCBVI from Azar et al. (2017)

Initialize data $\mathcal{H} = \varnothing$
for episode $k = 1, 2, \ldots, K$ **do**
  Compute $Q_{k,h}$ via Algorithm 8
  for step $h = 1, \ldots, H$ **do**
    Take action $a_{k,h} = \arg\max_a Q_{k,h}(x_{k,h}, a)$
    Update $\mathcal{H} = \mathcal{H} \cup (x_{k,h}, a_{k,h}, x_{k,h+1})$
  end for
end for

Algorithm 10 EULER from Zanette and Brunskill (2019).

Input: $\delta' = \frac{1}{7}\delta$, $b_k^r(s,a) = \sqrt{\frac{2\widehat{\text{Var}}R(s,a)\ln\frac{4SAT}{\delta'}}{n_k(s,a)}} + \frac{7\ln\frac{4SAT}{\delta'}}{3(n_k(s,a)-1)}$, $\phi(s,a) = \sqrt{\frac{2\widehat{\text{Var}}_{\hat{p}_k(s,a)}(\overline{V}^{\tilde{\pi}_k}_{t+1k})\ln\frac{4SAT}{\delta'}}{n_k(s,a)}} + \frac{H\ln\frac{4SAT}{\delta'}}{3(n_k(s,a)-1)}$,
$B_p = H\sqrt{2\ln\frac{(4SAT)}{\delta'}}, B_v = \sqrt{2\ln\frac{(4SAT)}{\delta'}}, J = H\ln\frac{(4SAT)/\delta'}{3}$.
for $k = 1, 2, \ldots$ **do**
  for $t = H, H-1, \ldots, 1$ **do**
    for $s \in \mathcal{S}$ **do**
      for $a \in \mathcal{A}$ **do**
        $\hat{p} = \frac{p_{sum}(\cdot,s,a)}{n_k(s,a)}$
        $b_k^{pv} = \phi(\hat{p}(s,a), \overline{V}_{t+1}) + \frac{1}{\sqrt{n(s,a)}}\left(\frac{4J+B_p}{\sqrt{n_k(s,a)}} + B_v\|\overline{V}_{t+1} - \underline{V}_{t+1}\|_{2,\hat{p}}\right)$
        $Q(a) = \min\{H-t, \hat{r}_k(s,\tilde{\pi}_k(s,t)) + b_k^r(s,a) + \hat{p}^\top\overline{V}_{t+1} + b_k^{pv}\}$
      end for
      $\tilde{\pi}_k(s,t) = \text{argmax}_a\, Q(a)$
      $\overline{V}_t(s) = Q(\tilde{\pi}_k(s,t))$
      $b_k^{pv} = \phi(\hat{p}(s,\tilde{\pi}_k(s,t)), \underline{V}_{t+1}) + \frac{1}{\sqrt{n(s,\tilde{\pi}_k(s,t))}}\left(\frac{4J+B_p}{\sqrt{n_k(s,\tilde{\pi}_k(s,t))}} + B_v\|\overline{V}_{t+1} - \underline{V}_{t+1}\|_{2,\hat{p}}\right)$
      $\underline{V}_t(s) = \max\{0, \hat{r}_k(s,\tilde{\pi}_k(s,t)) - b_k^r(s,\tilde{\pi}_k(s,t)) + \hat{p}^\top\underline{V}_{t+1} - b_k^{pv}\}$
    end for
  end for
  Evaluate policy $\tilde{\pi}_k$ and update MLE estimates $\hat{p}(\cdot,\cdot)$ and $\hat{r}(\cdot,\cdot)$
end for

A.3 Other Assumptions and Relevant Definitions

The remaining assumptions quantify the reachability of certain states. First, we have the following assumption, which in effect ensures that one can reach most states regardless of exit configuration in the meta-training tasks:

Assumption A.1 (Non-limiting exit configurations)**.** *Let $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathbb{P}, H)$ be any reward-free environment with time-varying dynamics*

$$\mathbb{P}^{(h)}(\cdot \mid s, a) = \mathbb{P}_{t(h,s,a)}(\cdot \mid s, a) \quad \textit{for some } t : [H] \times \mathcal{S} \times \mathcal{A} \to [T].$$

Then, there exists $C > 1$ such that for any s and $t \in [T]$,

$$\max_{\pi} P_{\mathcal{M}}(s \in \tau_\pi) \leq C \max_{\pi} P_{\mathcal{M}_t}(s \in \tau_\pi)$$

Intuitively, the assumption states that the reachability of a state in $\mathcal{M}_t$ would not be significantly improved even under an optimal configuration of the exits. Therefore, running reward-free RL on one of the meta-training tasks is sufficient for learning all non-exit (s, a) pairs.

Remark A.1. We note that Assumption A.1 is restrictive in that it requires that every state be roughly reachable in any of the meta-training MDPs. This may not hold in practice, e.g., consider a four-room environment where one of the rooms is blocked off for one of the tasks. However, this can be weakened to requiring that s be reachable in at least one of N arbitrarily chosen meta-training tasks. This would require that the algorithm run Phase II over N meta-training tasks, which results in a benign increase in the query complexity of the algorithm, so long as N is a constant much smaller than T. We focus on the $N = 1$ case for ease of presentation. ⌟

To simplify the presentation of the rest of the assumptions, we recall the following definition of δ-significance in Jin et al. (2020):

Definition A.1. A state s is *δ-significant* if $\max_\pi P(s \in \tau_\pi) \geq \delta$. Additionally, we say that (s, a) is δ-significant if s is δ-significant. ◇

Note that we have modified the definition to remove the dependence on the timestep $h \in [H]$. This is because the dynamics are stationary, and thus it does not matter when s is visited in a trajectory.

Note that Assumption A.1 implies that a reachable entrance in one task is reachable in all tasks (i.e., is significant in the sense of Definition A.1). Therefore, we can quantify the minimum level of significance:[8]

Definition A.2 (ρ-significant entrances)**.** For any $s \in \mathrm{Ent}(\mathcal{S})$, s is ρ-significant for any task. ◇

Finally, we want to quantify the reachability of every exit from any entrance in the same cluster, assuming that it is indeed reachable:

Definition A.3 (In-cluster exit reachability)**.** Fix any cluster Z, entry $s \in \mathrm{Ent}(Z)$, and exit $(g, a) \in \mathrm{Ext}(Z)$. Consider the reward-free environment $\mathcal{M}_t|_Z = (Z, A, \mathbb{P}_t|_Z, H)$, where $\mathbb{P}_t|_Z$ is the restriction of $\mathbb{P}_t$ to $Z \times \mathcal{A}$ and the starting state is s. Then, if g has nonzero significance in $\mathcal{M}_t|_Z$ for any $t \in [T]$, then it is ε_0-significant. ◇

The requirement that the assumption hold for any $t \in [T]$ is without loss of generality since non-exit dynamics do not change.

A.4 Verifying Exit Detection

In this section, we demonstrate that the algorithm in Section A.1 can successfully discover $\mathrm{Ext}(\mathcal{C})$ with high probability. Formally, we have the following result:

Theorem A.1 (Provable exit detection)**.** *Assume we run the algorithm in Section A.1 with the parameter choices given in Table 1. Then, with probability at least $1 - p$, the algorithm returns an array* IsExit *satisfying:*

$$\{(s, a) \mid \textsc{IsExit}[s, a]\} = \mathrm{Ext}(\mathcal{C}).$$

To prove this result, we proceed with a phase-by-phase analysis of the algorithm in Section A.1, which we then compile into proof of the desired result.

[8] One can weaken this assumption in a way that is compatible with the weakened form of Assumption A.1.

Parameter	Value
N_{UCBVI}	$\frac{H^2SA}{\min(\alpha,\zeta)^2}\log^2\frac{HSAT}{p}$
$N_{\text{thresh}}^{\text{TS}}$	$S\max\left(\frac{H^4}{\zeta^2},\frac{1}{\beta^2}\right)\log\frac{SAHT}{p\alpha\min(\beta,\zeta)}$
N_{TS}	$S\max\left(\frac{H^5}{\alpha\zeta^2},\frac{H}{\alpha\beta^2}\right)\log\frac{SAHT}{p\alpha\min(\beta,\zeta)}+\frac{H^2}{\min(\alpha,\zeta)^2}\log\frac{SAT}{p}$
$N_{\text{EULER}}^{\text{RF}}$	$\frac{H^2S^4A}{\min(\rho\min(\varepsilon,\varepsilon_0),\zeta/C)}\log^3\frac{HSA}{p}$
N_{RF}	$\frac{H^5S^2A}{\min(\rho\min(\varepsilon,\varepsilon_0)^2,\zeta^2/C)}\log\frac{A}{p}$
$N_{\text{thresh}}^{\text{ED}}$	$\frac{S}{\beta^2}\log\frac{SAH}{p\zeta\beta}$
N_{ED}	$\frac{HKS}{\zeta\beta^2}\log\frac{SAH}{p\zeta\beta}+\frac{H^2K^2}{\zeta^2}\log\frac{K}{p}$
$N_{\text{thresh}}^{\text{EL}}$	$L\max\left(\frac{H^4}{\zeta^2},\frac{1}{\beta^2}\right)\log\frac{CSAHT}{p\alpha\min(\beta,\zeta)}$
$N_{\text{EULER}}^{\text{EL}}$	$\frac{CH^3S^2A}{\alpha}\log^3\left(\frac{HSAT}{p}\right)$
N_{EL}	$L\max\left(\frac{CH^5}{\alpha\zeta^2},\frac{CH}{\alpha\beta^2}\right)\log\frac{CSAHT}{p\alpha\min(\beta,\zeta)}+\frac{C^2H^2}{\alpha^2}\log\frac{SAT}{p}$

Table 1: Table of parameters for the results in Theorem A.1. Since $K \le SA$ and $L \le S$, the agent does not need to know K or L in advance, at the expense of a worse sample complexity bound.

A.4.1 Phase I Analysis

First, we prove that during Phase I, Algorithm 3 sufficiently visits all relevant exits and that all value estimates are sufficiently close. Formally, we have the following result:

Proposition A.1. *Set*

$$N_{\text{thresh}}^{\text{TS}} = \Omega\left[S\max\left(\frac{H^4}{\zeta^2},\frac{1}{\beta^2}\right)\log\frac{SAHTN_{\text{TS}}}{p}\right],$$

and consider the following procedure applied to one of the meta-training tasks $\mathcal{M}_t$:

1. UCBVI *is run for*

$$N_{\text{UCBVI}} = \Omega\left(\frac{H^2SA}{\min(\alpha,\zeta)^2}\log^2\frac{HSAT}{p}\right)$$

iterations, generating policies $\pi_1^{(t)},\dots,\pi_{N_{\text{UCBVI}}}^{(t)}$.

2. *The learner uniformly samples*

$$N_{\mathrm{TS}} = \Omega\left(\frac{H}{\alpha} N_{\mathrm{thresh}}^{\mathrm{TS}} + \frac{H^2}{\min(\alpha,\zeta)^2}\log\frac{TK}{p}\right)$$

policies from the previous step, runs each policy in $\mathcal{M}_t$, and obtains a dataset of transitions $\mathcal{D}_t$ and returns $\hat{V}^{(1)},\dots,\hat{V}^{(N_{\mathrm{TS}})}$.

Then, with probability at least $1 - p/3T$,

(a) *We have the regret bound*

$$V_0^*(s_0) - \frac{1}{N_{\mathrm{UCBVI}}}\sum_{k=1}^{N_{\mathrm{UCBVI}}} V_0^{\pi_k^{(t)}}(s_0) < \frac{\zeta}{6}.$$

(b) *The set of obtained returns satisfy*

$$\left|\frac{1}{N_{\mathrm{TS}}}\sum_{i=1}^{N_{\mathrm{TS}}}\hat{V}^{(i)} - \frac{1}{N}\sum_{k=1}^{N_{\mathrm{UCBVI}}} V_0^{\pi_k^{(t)}}(s_0)\right| < \frac{\zeta}{6}.$$

(c) *If (s,a) is α-important for $\mathcal{M}_t$, then $N_t(s,a) \geq N_{\mathrm{thresh}}$.*

(d) *For every (s,a) pair such that $N_t(s,a) \geq N_{\mathrm{thresh}}$,*

$$\sup_{f:\mathcal{S}\to[0,H]}\left|\left[(\hat{\mathbb{P}}_t - \mathbb{P}_t)f\right](s,a)\right| < \min\left(\frac{\zeta}{24H}, \frac{\beta H}{2}\right).$$

To prove the above result, we first recall the following regret bound on UCBVI, as proven by Azar et al. (2017):

Lemma A.1 (UCBVI regret bound)**.** *For sufficiently large N, with probability at least $1 - p/6$,*

$$V_0^*(s_0) - \frac{1}{N}\sum_{k=1}^{N} V_0^{\pi_k}(s_0) \lesssim \sqrt{\frac{H^2SA}{N}}\log\left(\frac{HSAN}{p}\right).$$

As we will see later on, with our choice of N_{UCBVI}, we obtain the desired regret bound in (a). Additionally, by Hoeffding's inequality, the average of N_{TS} returns concentrates around the desired quantity with high probability, proving (b). Thus, all that remains is ensuring that every α-important exit is sufficiently visited, and thus their dynamics are sufficiently well-estimated.

Recall from the main text that the key step is demonstrating that a near-optimal policy for a task must visit its α-important states with non-negligible probability:

Lemma A.2. *Let (s,a) be α-important for $\mathcal{M}$, and let π be an ε-suboptimal policy for $\varepsilon < \alpha$. Then,*

$$P((s,a) \in \tau_\pi) > \frac{1}{H}(\alpha - \varepsilon).$$

Proof. By α-importance,

$$\begin{aligned}\alpha &\leq V_0^{\mathcal{M},*}(s_0) - V_0^{\mathcal{M}^{\setminus(s,a)},*}(s_0) \leq \left[V_0^{\mathcal{M},*}(s_0) - V_0^{\mathcal{M},\pi}(s_0)\right] + \left[V_0^{\mathcal{M},\pi}(s_0) - V_0^{\mathcal{M}^{\setminus(s,a)},*}(s_0)\right] \\ &\leq \left[V_0^{\mathcal{M},\pi}(s_0) - V_0^{\mathcal{M}^{\setminus(s,a)},*}(s_0)\right] + \varepsilon.\end{aligned}$$

Therefore, by applying Lemma A.17 and noting that $\{\Delta \cap \tau_\pi \neq \varnothing\} = \{(s,a) \in \tau_\pi\}$, we obtain the desired result. □

Through the prior result, we can relate the UCBVI regret bound to the probability that a randomly chosen UCBVI-generated policy visits an α-important state:

Lemma A.3. *Let (s, a) be α-important for $\mathcal{M}$. Assume that* UCBVI*, when run for*

$$N_{\text{UCBVI}} = \Omega\left(\frac{H^2SA}{\alpha^2}\log^2\frac{HSA}{p}\right)$$

iterations, generates policies $\pi_1, \ldots, \pi_N$. If we sample π uniformly from these policies and let τ be the (random) trajectory generated by this randomly selected policy, then

$$P((s, a) \in \tau) > \frac{\alpha}{2H},$$

conditioned on the high probability event in Lemma A.1.

Proof. By Lemma A.2, for any fixed π, we can write

$$P((s, a) \in \tau_\pi) \geq \frac{1}{H}(\alpha - [V_0^*(s_0) - V_0^\pi(s_0)])_+,$$

where $x_+ = x\mathbb{1}[x > 0]$. Then, since π is chosen randomly from the policies generated by UCBVI,

$$\begin{aligned} P((s, a) \in \tau) &= \frac{1}{N}\sum_{k=1}^N P((s, a) \in \tau_{\pi_k}) \geq \frac{1}{HN}\sum_{k=1}^N(\alpha - [V_0^*(s_0) - V_0^{\pi_k}(s_0)])_+ \\ &\geq \frac{1}{H}\left[\alpha - \frac{1}{N}\sum_{k=1}^N V_0^*(s_0) - V_0^{\pi_k}(s_0)\right]. \end{aligned}$$

Therefore, by applying the regret bound in Lemma A.1 and the choice of N, we find that

$$P((s, a) \in \tau) > \frac{\alpha}{2H}.$$

□

With all of the above intermediate results, we can now prove Proposition A.1.

Proof of Proposition A.1. Throughout this proof, we condition on the high-probability event in Lemma A.1, instantiated to occur with probability at least $p/12T$.

(a) By the choice of N_{UCBVI},

$$N_{\text{UCBVI}} \gtrsim \frac{H^2SA}{\zeta^2}\log^2\frac{HSAT}{p},$$

and thus, we obtain the desired bound by plugging this value into the regret bound provided by Lemma A.1.

(b) Note that $(\hat{V}^{(i)})$ are i.i.d., bounded in $[0, H]$, and for any $i \in [N_{\text{TS}}]$,

$$\mathbb{E}\left[\hat{V^{(i)}}\right] = \frac{1}{N}\sum_{k=1}^N V_0^{\pi_k^{(t)}}(s_0).$$

Therefore, by applying Hoeffding's inequality, with probability at least $1 - p/12T$,

$$\left|\frac{1}{N_{\text{TS}}}\sum_{i=1}^{N_{\text{TS}}}\hat{V}^{(i)} - \frac{1}{N_{\text{UCBVI}}}\sum_{k=1}^{N_{\text{UCBVI}}} V_0^{\pi_k^{(t)}}(s_0)\right| \lesssim \sqrt{\frac{H^2}{N_{\text{TS}}}\log\frac{T}{p}}$$

The result immediately follows from the fact that $N_{\text{TS}} \gtrsim (H^2/\zeta^2)\log(T/p)$.

(c) The result simply follows from Lemma A.16 instantiated with failure probability $1 - p/12T$, together with the choice of $N_{\text{thresh}}^{\text{TS}}$.

(d) With the choice of N_{UCBVI}, the conclusion of Lemma A.1 can be made to hold with probability at least $1 - p/24T$. Fix an α-important exit (s, a) for $\mathcal{M}_t$, so that the probability that (s, a) is visited by the procedure is at least $\alpha/2H$. By Lemma A.15, sampling N_{TS} trajectories is sufficient to ensure that $N_t(s, a) \geq N_{\text{thresh}}^{\text{TS}}$ with probability at least $1 - p/24TK$. Therefore, by performing a union bound over the set of α-important exits (which contains at most K elements), $N_t(s, a) \geq N_{\text{thresh}}^{\text{TS}}$ for any α-important exit with probability at least $1 - p/24T$. Thus, overall, this event occurs with probability at least $1 - p/12T$.

Since each part fails with probability at most $p/12T$, the overall failure probability is at most $p/3T$, the desired result. □

A.4.2 Phase II Analysis

In this section, we provide guarantees on the dataset $\mathcal{D}_{\mathrm{RF}}$ obtained by performing reward-free RL in Algorithm 4. Formally, we have the following high-probability result:

Proposition A.2. *For any $\delta > 0$ and failure probability p, if Algorithm 4 is run with parameters*

$$N_{\mathrm{EULER}}^{\mathrm{RF}} = O\left(\frac{H^2S^2A}{\delta}\log^3\frac{HSA}{p}\right)$$
$$N_{\mathrm{RF}} = O\left[\max\left(\frac{C}{\zeta^2}, \frac{1}{\rho\min(\varepsilon,\varepsilon_0)^2}\right)H^5S^2A\log\frac{A}{p}\right].$$

Then, with probability at least $1 - p/3$:

(a) The distribution μ generating each sample in $\mathcal{D}_{\mathrm{RF}}$ satisfies

$$s \in \mathcal{S} \text{ is } \delta\text{-significant in } \mathcal{M}_1(2H) \implies \max_{a,\pi}\frac{P((s,a)\in\tau_\pi)}{\mu(s,a)} \le 4SAH.$$

(b) The estimated dynamics model $\hat{\mathbb{P}}_0$ satisfies

$$\max_{f:\mathcal{S}\to[0,H]}\max_{\nu:\mathcal{S}\to\mathcal{A}}\mathbb{E}_{(s,a)\sim\mu}\left[\left|\left[(\hat{\mathbb{P}}-\mathbb{P})f\right](s,a)\right|^2\mathbb{1}\left[a=\nu(s)\right]\right]$$
$$\lesssim \min\left(\frac{\zeta^2}{4\cdot 24^2C}, \frac{\rho\min(\varepsilon,\varepsilon_0)^2}{16}\right)\frac{1}{H^3SA}.$$

The details of the proof of Proposition A.2 follow that of Jin et al. (2020), which we provide here for completeness. First, we adapt the regret bound from Zanette and Brunskill (2019) for any MDP and reward function used in Algorithm 4.

Lemma A.4. *For any $g \in \mathcal{S}$, running* EULER *in $\mathcal{M}_g$ for N iterations returns N policies $\pi_1, \ldots, \pi_N$ satisfying the regret bound*

$$V_0^*(s_0) - \frac{1}{N}\sum_{k=1}^N V_0^{\pi_k}(s_0) \lesssim \sqrt{4V_0^*(s_0)\frac{SA}{N}\log\frac{SAHN}{p}} + \frac{S^2AH^2}{N}\log^3\frac{SAHN}{p}$$

with probability at least $1 - p$.

Proof. Observe that

$$\begin{aligned}
&\frac{1}{NH}\sum_{k=1}^N\mathbb{E}_{\pi_k}\left[\left(\sum_{h=1}^{H-1}r(s_h,a_h) - V_0^{\pi_k}(s_0)\right)^2\middle|\, s_0\right] \\
&\quad\le \frac{2}{NH}\sum_{k=1}^N\mathbb{E}_{\pi_k}\left[\left(\sum_{h=1}^{H-1}r(s_h,a_h)\right)^2 + \left(V_0^{\pi_k}(s_0)\right)^2\middle|\, s_0\right] \\
&\quad\le \frac{2}{NH}\sum_{k=1}^N\mathbb{E}_{\pi_k}\left[\sum_{h=1}^{H-1}r(s_h,a_h) + V_0^{\pi_k}(s_0)\middle|\, s_0\right] \\
&\quad\le \frac{4}{H}V_0^*(s_0).
\end{aligned}$$

Therefore, by applying the regret bounds from Zanette and Brunskill (2019), we obtain the regret bound

$$V_0^*(s_0) - \frac{1}{N}\sum_{k=1}^N V_0^{\pi_k}(s_0) \lesssim \sqrt{4V_0^*(s_0)\frac{SA}{N}\log\frac{SAHN}{p}} + \frac{S^2A^2H^2}{N}\log^3\frac{SAHN}{p}$$

with probability at least $1 - p$. □

With the regret bound above, we now proceed to prove Proposition A.2.

Proof of Proposition A.2. (a) Fix a δ-significant $g \in \mathcal{S}$. Note that for r_g, $V_0^\pi(s_0) = P(g \in \tau_\pi)$ for any policy π. Therefore, via the regret bound from Lemma A.4 and the choice of $N_{\text{EULER}}^{\text{RF}}$, we obtain

$$\max_\pi P(g \in \tau_\pi) - \frac{1}{N_{\text{EULER}}^{\text{RF}}} \sum_{k=1}^{N_{\text{EULER}}^{\text{RF}}} P(g \in \tau_\pi) \le \frac{1}{2} \max_\pi P(g \in \tau_\pi)$$
$$\implies \max_\pi P(g \in \tau_\pi) \le \frac{2}{N_{\text{RF}}^{\text{EULER}}} \sum_{\pi \in \Phi_g} P(g \in \tau_\pi)$$

with probability at least $1 - p/2S$. Now, since $\pi(\cdot \mid g) \sim \text{Unif}(\mathcal{A})$, we have that for any a,

$$\max_\pi P((g,a) \in \tau_\pi) \le \frac{2A}{N_{\text{RF}}^{\text{EULER}}} \sum_{\pi \in \Phi_g} P((g,a) \in \tau_\pi).$$

Finally, by applying the same argument above across all δ-significant $g \in \mathcal{S}$, we have that for any (g,a),

$$\max_\pi P((g,a) \in \tau_\pi) \le \sum_{g \in S} \max_{a,\pi} P((g,a) \in \tau_\pi) \le 2SA \left[\frac{1}{SN_{\text{RF}}^{\text{EULER}}} \sum_{\pi \in \Psi} P((g,a) \in \tau_\pi) \right]$$

with probability at least $1 - p/2$. To complete the proof of (a), observe that

$$\frac{1}{SN_{\text{RF}}^{\text{EULER}}} \sum_{\pi \in \Psi} \frac{1}{2H} P((g,a) \in \tau_\pi) \le \mu(s,a) \implies \max_{s,a,\pi} \frac{P((s,a) \in \tau_\pi)}{\mu(s,a)} \le 4SAH,$$

since conditioned on $(g,a) \in \tau_\pi$, the probability that (g,a) is sampled is at least $1/2H$.

(b) The result follows by following the same proof of Lemma C.2 in Jin et al. (2020), with failure probability $p/2$. Note that the dynamics are stationary, and thus we do not need to perform a union bound over the time step $h \in [H]$. □

A.4.3 Phase III Analysis

Having analyzed the previous two phases, we now show that Algorithm 5 successfully finds all exits during Phase III. As part of this, we prove the following guarantee:

Proposition A.3. *Assume that Algorithm 5 is at Line 5, having just arrived at this step for the first time, or after finding a new exit. Let $E = \{(s,a) \mid \text{IsExit}[s,a]\}$. We assume:*

(a) $E \subseteq \text{Ext}(\mathcal{C})$.

(b) The high-probability events in Proposition A.1 (for any $t \in [T]$) and Proposition A.2 (for $\delta \le \zeta/24CH^2S$) both hold, providing estimators $\hat{\mathbb{P}}_0, \hat{\mathbb{P}}_1, \ldots, \hat{\mathbb{P}}_T$.

(c) For every $(s,a) \in E$ and $t \in [T]$, we have access to an estimator $\hat{\mathbb{P}}_t(\cdot \mid s,a)$ for $\mathbb{P}_t(\cdot \mid s,a)$ satisfying

$$\max_{f:\mathcal{S} \to [0,H]} \left| \left[(\hat{\mathbb{P}}_t - \mathbb{P}_t) f \right](s,a) \right| \le \min\left(\frac{\zeta}{24H}, \frac{\beta H}{2} \right).$$

Then, if $E = \text{Ext}(\mathcal{C})$, the algorithm terminates after passing through T tasks. Otherwise, if $E \neq \text{Ext}(\mathcal{C})$, the following events hold simultaneously with probability at least $1 - p/3K$:

(a) For one of the next T tasks that the algorithm inspects, there exists at least one $t \in [T]$ such that

$$\left| \tilde{V}^t - \hat{V}_t \right| > \frac{2}{3}\zeta.$$

(b) For the task in (a), running Lines 5–5 finds at least one $(s,a) \in \text{Ext}(\mathcal{C}) \setminus E$ (and only (s,a) pairs in this set), and learns an estimator $\hat{\mathbb{P}}_t(\cdot \mid s,a)$ for $\mathbb{P}_t(\cdot \mid s,a)$ satisfying

$$\max_{f:\mathcal{S} \to [0,H]} \left| \left[(\hat{\mathbb{P}}_t - \mathbb{P}_t) f \right](s,a) \right| \le \min\left(\frac{\zeta}{24H}, \frac{\beta H}{2} \right).$$

To prove the above result, we will consider the following special set of MDPs:

Definition A.4 (Imaginable MDPs)**.** Fix a task $\mathcal{M}_t$. Furthermore, let $E \subseteq \mathrm{Ext}(\mathcal{C})$. For any function $\mathcal{I} : \mathcal{S} \times \mathcal{A} \times [H] \to \{0, \dots, T\}$, we can construct an associated MDP $\mathcal{M}_{\mathcal{I}} = (\mathcal{S}, \mathcal{A}, \mathbb{P}_{\mathcal{I}}, r_t, H)$ via

$$\mathbb{P}_{\mathcal{I}}^{(h)}(\cdot \mid s, a) = \mathbb{P}_{\mathcal{I}(s,a,h)}(\cdot \mid s, a).$$

We define the set of imaginable MDPs to be the set $\mathbb{M}_t(E)$ to be the set of MDPs generated by any $\mathcal{I}$ satisfying

$$\mathcal{I}(s, a, h) \in \begin{cases} \{t\} & (s, a) \in E \\ \{0\} \cup \left\{k \;\middle|\; N^k(s, a) \geq N_{\mathrm{thresh}}\right\} & \text{otherwise} \end{cases}. \qquad \diamond$$

Informally, $\mathbb{M}_t(E)$ is the set of obtainable MDPs by borrowing dynamics for (s, a) pairs that are not known to be exits. This set is of particular interest in our analysis, since BOAT-VI performs a maximization over the MDPs in this set:

Lemma A.5 (Optimism)**.** *Assume the preconditions of Proposition A.3. Over the course of running Algorithm 6, the algorithm implicitly defines an index function $\mathcal{I} : \mathcal{S} \times \mathcal{A} \times [H] \to \{0, \dots, T\}$. This function $\mathcal{I}$ satisfies $\mathcal{M}_{\mathcal{I}} \in \mathbb{M}_t(E)$, and $\mathcal{M}_{\mathcal{I}}$ is a maximizer of*

$$\max_{\mathcal{M} \in \mathbb{M}_t(E)} \max_{\pi} \hat{V}_0^{\mathcal{M},\pi}(s_0).$$

Proof. To see that $\mathcal{M}_{\mathcal{I}} \in \mathbb{M}_t(E)$, note that if $(s, a) \in E$, then $\mathcal{I}_h(s, a) = t$ for any $h \in [H]$. Otherwise, note that although the maximum is over all indices, $\hat{\mathbb{P}}_k(s' \mid s, a) = 0$ for any s' if $N_t(s, a) < N_{\mathrm{thresh}}$. Therefore, since the estimated value function is always positive, the maximum is effectively only over any k with $N_k(s, a) \geq 0$. Thus, $\mathcal{M}_{\mathcal{I}} \in \mathbb{M}_t(E)$.

Now, we prove that $\mathcal{M}_{\mathcal{I}} = \mathcal{M}$ is a maximizer of the estimated value function, which we prove by induction. Let $\mathcal{I}'$ be another index function satisfying $\mathcal{M}' = \mathcal{M}_{\mathcal{I}'} \in \mathbb{M}_t(E)$. Clearly, $\hat{V}_H^{\mathcal{M},*}(s) = 0 \leq \hat{V}_H^{\mathcal{M}',*}(s)$. Then, for any $h \in [H]$ and (s, a),

$$\begin{aligned} \hat{Q}_h^{\mathcal{M},*}(s, a) &= r(s, a) + \hat{\mathbb{P}}_{\mathcal{I}(s,a,h)} \hat{V}_{h+1}^{\mathcal{M},*}(s, a) \\ &\geq r(s, a) + \hat{\mathbb{P}}_{\mathcal{I}'(s,a,h)} \hat{V}_{h+1}^{\mathcal{M},*}(s, a) \geq r(s, a) + \hat{\mathbb{P}}_{\mathcal{I}'(s,a,h)} \hat{V}_{h+1}^{\mathcal{M}',*}(s, a) \\ &= \hat{Q}_h^{\mathcal{M}',*}(s, a), \end{aligned}$$

where the first inequality follows from the definition of $\mathcal{I}$, and the second follows from the inductive hypothesis. Therefore, for any s,

$$\hat{V}_h^{\mathcal{M},*}(s) = \max_a \hat{Q}_h^{\mathcal{M},*}(s, a) \geq \max_a \hat{Q}_h^{\mathcal{M}',*}(s, a) = \hat{V}_h^{\mathcal{M}',*}(s)$$

Thus, by induction, $\hat{V}_0^{\mathcal{M},*}(s_0) \geq \hat{V}_0^{\mathcal{M}',*}(s_0)$. Since the argument applies for any $\mathcal{I}'$, we have shown the desired optimality result. □

Note that $\bar{\mathcal{M}}$ is contained in $\mathbb{M}_t(E)$ via our assumed preconditions, suggesting that the BOAT-VI should find an MDP with a sufficiently over-optimistic value. However, the maximization above makes use of estimated dynamics, and thus we need to prove that every MDP in $\mathbb{M}_t(E)$ is sufficiently well-estimated. We now show that the preconditions of Proposition A.3 are sufficient for estimation. To this end, we recall the performance difference lemma:

Lemma A.6 (Performance Difference)**.** *Fix two MDPs $\mathcal{M} = (\mathcal{S}, \mathcal{A}, r, \mathbb{P}, H)$ and $\mathcal{M}' = (\mathcal{S}, \mathcal{A}, r, \mathbb{P}', H)$. Then, for any policy π,*

$$V_0^{\mathcal{M}',\pi}(s_0) - V_0^{\mathcal{M},\pi}(s_0) = \mathbb{E}_{\mathcal{M},\pi}\left[\sum_{h=0}^{H-1} [(\mathbb{P}'_h - \mathbb{P}_h)\hat{V}_{h+1}](s_h, a_h) \;\middle|\; s_0\right].$$

We now present the estimation result:

Lemma A.7. *For any π and $t \in [T]$, let $V_0^{\mathcal{M},\pi}(s_0)$ be the value of a policy π in $\mathcal{M} \in \mathbb{M}_t(E)$, and $\hat{V}_0^{\mathcal{M},\pi}(s_0)$ an estimate using available quantities from the preconditions of Proposition A.3. Then,*

$$\sup_{t \in [T]} \sup_{\substack{\pi \\ \mathcal{M} \in \mathbb{M}_t(E)}} \left|\hat{V}_0^{\mathcal{M},\pi}(s_0) - V_0^{\mathcal{M},\pi}(s_0)\right| < \frac{\zeta}{6}.$$

Proof. Fix a $t \in [T]$, $\mathcal{M} \in \mathbb{M}_t(E)$ and policy π, with associated index function $\mathcal{I}$. Lemma A.6 implies that

$$\begin{aligned}\left|\hat{V}_0^{\mathcal{M},\pi}(s_0) - V_0^{\mathcal{M},\pi}(s_0)\right| &\le \sum_{h=0}^{H-1} \mathbb{E}_{\mathcal{M},\pi}\left[\left|\left[(\hat{\mathbb{P}}^{(h)} - \mathbb{P}^{(h)})\hat{V}_{h+1}^{\pi}\right](s_h, a_h)\right|\right] \\ &\le \sum_{h=0}^{H-1}\sum_{(s,a)} \left|\left[(\hat{\mathbb{P}}^{(h)} - \mathbb{P}^{(h)})\hat{V}_{h+1}^{\pi}\right](s,a)\right| P_h^{\mathcal{M},\pi}(s,a).\end{aligned}$$

We now define the following sets:

$$\begin{aligned}U_\delta &= \{(s,a) \text{ is } \delta\text{-insignificant for } \mathcal{M}_1\} \\ B_h &= \{(s,a) \mid \mathcal{I}_h(s,a) \neq 0\} \setminus (E \cup U_\delta) \\ R_h &= \{(s,a) \mid \mathcal{I}_h(s,a) = 0\} \setminus U_\delta.\end{aligned}$$

Note that R_h is estimated via reference dynamics from Phase II, while B_h is estimated using task-specific dynamics from Phase I. Then, for a fixed h, we can decompose the inner sum above as

$$\begin{aligned}&\sum_{(s,a)} \left|\left[(\hat{\mathbb{P}}^{(h)} - \mathbb{P}^{(h)})\hat{V}_{h+1}^{\pi}\right](s,a)\right| P_h^{\mathcal{M},\pi}(s,a) \\ &\quad\le \underbrace{\sum_{(s,a)\in E} \left|\left[(\hat{\mathbb{P}}^{(h)} - \mathbb{P}^{(h)})\hat{V}_{h+1}^{\pi}\right](s,a)\right| P_h^{\mathcal{M},\pi}(s,a)}_{=:\text{(I)}} \\ &\qquad + \underbrace{\sum_{(s,a)\in R_h} \left|\left[(\hat{\mathbb{P}}^{(h)} - \mathbb{P}^{(h)})\hat{V}_{h+1}^{\pi}\right](s,a)\right| P_h^{\mathcal{M},\pi}(s,a)}_{=:\text{(II)}} \\ &\qquad + \underbrace{\sum_{(s,a)\in B_h} \left|\left[(\hat{\mathbb{P}}^{(h)} - \mathbb{P}^{(h)})\hat{V}_{h+1}^{\pi}\right](s,a)\right| P_h^{\mathcal{M},\pi}(s,a)}_{=:\text{(III)}} \\ &\qquad + \underbrace{\sum_{(s,a)\in U_\delta} \left|\left[(\hat{\mathbb{P}}^{(h)} - \mathbb{P}^{(h)})\hat{V}_{h+1}^{\pi}\right](s,a)\right| P_h^{\mathcal{M},\pi}(s,a)}_{=:\text{(IV)}}\end{aligned}$$

Note the inequality since $E \cap U_\delta$ is not necessarily disjoint. We now bound the four terms above separately.

Bounding (I): Dynamics Error from Known Exits. We first bound (I), which we note derives from errors in estimating the dynamics of known exits. Recall that by precondition (c) in Proposition A.3,

$$\sup_{f:\mathcal{S}\to[0,H]} \left|\left[(\hat{\mathbb{P}}_t - \mathbb{P}_t)f\right](s,a)\right| \le \frac{\zeta}{24H}.$$

Therefore,

$$\begin{aligned}\text{(I)} &= \sum_{(s,a)\in E} \left|\left[(\hat{\mathbb{P}}^{(h)} - \mathbb{P}^{(h)})\hat{V}_{h+1}^{\pi}\right](s,a)\right| P_h^{\mathcal{M},\pi}(s,a) \\ &= \sum_{(s,a)\in E} \left|\left[(\hat{\mathbb{P}}_t - \mathbb{P}_t)\hat{V}_{h+1}^{\pi}\right](s,a)\right| P_h^{\mathcal{M},\pi}(s,a) \le \frac{\zeta}{24H}\sum_{(s,a)\in E} P_h^{\mathcal{M},\pi}(s,a) \\ &\le \frac{\zeta}{24H}.\end{aligned}$$

Bounding (II): Reference Dynamics Error. Note that within R_h, $\mathbb{P}_h = \mathbb{P}_0$, which we estimate via $\hat{\mathbb{P}}_0$. Therefore, we bound the error resulting from using $\mathcal{D}_{\mathrm{RF}}$ to estimate $\mathbb{P}_0$. This part of the proof follows that of Jin et al. (2020). First, by

Cauchy-Schwarz,

$$\begin{aligned}\text{(II)} &= \sum_{(s,a)\in R_h} \left|\left[(\hat{\mathbb{P}}^{(h)} - \mathbb{P}^{(h)})\hat{V}^{\pi}_{h+1}\right](s,a)\right| P_h^{\mathcal{M},\pi}(s,a) \\ &= \sum_{(s,a)\in R_h} \left|\left[(\hat{\mathbb{P}}_0 - \mathbb{P}_0)\hat{V}^{\pi}_{h+1}\right](s,a)\right| P_h^{\mathcal{M},\pi}(s,a) \\ &\leq \left[\sum_{(s,a)\in R_h} \left|\left[(\hat{\mathbb{P}}_0 - \mathbb{P}_0)\hat{V}^{\pi}_{h+1}\right](s,a)\right|^2 P_h^{\mathcal{M},\pi}(s,a)\right]^{1/2}.\end{aligned}$$

Observe that $\hat{V}^{\pi}_{h+1}$ only depends on π through timesteps $h+1,\ldots,H-1$. Therefore,

$$\begin{aligned}&\sum_{(s,a)\in R_h} \left|\left[(\hat{\mathbb{P}}_0 - \mathbb{P}_0)\hat{V}^{\pi}_{h+1}\right](s,a)\right|^2 P_h^{\mathcal{M},\pi}(s,a) \\ &\quad\leq \max_{\nu:\mathcal{S}\to\mathcal{A}} \sum_{(s,a)\in R_h} \left|\left[(\hat{\mathbb{P}}_0 - \mathbb{P}_0)\hat{V}^{\pi}_{h+1}\right](s,a)\right|^2 P_h^{\mathcal{M},\pi}(s)\mathbb{1}\left[\nu(s)=a\right].\end{aligned}$$

By applying Assumption A.1,

$$\begin{aligned}P_h^{\mathcal{M},\pi}(s) \leq P^{\mathcal{M}}(s\in\tau_\pi) &\leq \max_\pi P^{\mathcal{M}}(s\in\tau_\pi) \leq C\max_\pi P^{\mathcal{M}_1}(s\in\tau_\pi) \\ &\leq C\max_\pi P^{\mathcal{M}_1(2H)}(s\in\tau_\pi) \leq 4CHSA\mu(s,a),\end{aligned}$$

where we have applied Assumption 4.1 to move from $\mathcal{M}$ to $\mathcal{M}_1$. Substituting into the earlier expression,

$$\begin{aligned}&\max_{\nu:\mathcal{S}\to\mathcal{A}} \sum_{(s,a)\in R_h} \left|\left[(\hat{\mathbb{P}}_0 - \mathbb{P}_0)\hat{V}^{\pi}_{h+1}\right](s,a)\right|^2 P_h^{\mathcal{M},\pi}(s)\mathbb{1}\left[\nu(s)=a\right] \\ &\quad\leq 4CHSA \max_{\nu:\mathcal{S}\to\mathcal{A}} \sum_{(s,a)\in R_h} \left|\left[(\hat{\mathbb{P}}_0 - \mathbb{P}_0)\hat{V}^{\pi}_{h+1}\right](s,a)\right|^2 \mathbb{1}\left[a=\nu(s)\right]\mu(s,a) \\ &\quad\leq 4CHSA \max_{\nu:\mathcal{S}\to\mathcal{A}} \sum_{s,a} \left|\left[(\hat{\mathbb{P}}_0 - \mathbb{P}_0)\hat{V}^{\pi}_{h+1}\right](s,a)\right|^2 \mathbb{1}\left[a=\nu(s)\right]\mu(s,a) \\ &\quad= 4CHSA \max_{\nu:\mathcal{S}\to\mathcal{A}} \mathbb{E}_{(s,a)\sim\mu}\left[\left|\left[(\hat{\mathbb{P}}_0 - \mathbb{P}_0)\hat{V}^{\pi}_{h+1}\right](s,a)\right|^2 \mathbb{1}\left[a=\nu(s)\right]\right].\end{aligned}$$

Thus, by applying the bound on the right-hand side provided by Proposition A.2,

$$\text{(II)} = \sum_{(s,a)\in R_h} \left|\left[(\hat{\mathbb{P}}^{(h)} - \mathbb{P}^{(h)})\hat{V}^{\pi}_{h+1}\right](s,a)\right| P_h^{\mathcal{M},\pi}(s,a) \leq \frac{\zeta}{24H}.$$

Bounding (III): Error from Task-Specific Dynamics. Recall that on B_h, $\mathbb{P}_h = \mathbb{P}_k$ for some $k \neq 0$. Thus, (III) is the error resulting from dynamics estimation in Algorithm 3. By following the same argument as that used to bound (I) and applying Proposition A.1, we find that

$$\sum_{(s,a)\in B_h} \left|\left[(\hat{\mathbb{P}}^{(h)} - \mathbb{P}^{(h)})\hat{V}^{\pi}_{h+1}\right](s,a)\right| P_h^{\mathcal{M},\pi}(s,a) \leq \frac{\zeta}{24H}.$$

Bounding (IV): Error from δ-Insignificance. The remaining set of (s,a) pairs are those such that s is δ-insignificant in $\mathcal{M}_1$. Note that

$$\begin{aligned}\text{(IV)} &= \sum_{(s,a)\in U_\delta} \left|\left[(\hat{\mathbb{P}}^{(h)} - \mathbb{P}^{(h)})\hat{V}^{\pi}_{h+1}\right](s,a)\right| P_h^{\mathcal{M},\pi}(s,a) \leq H\sum_{(s,a)\in U_\delta} P_h^{\mathcal{M},\pi}(s,a) \\ &= H\sum_{s\in U_\delta} P_h^{\mathcal{M},\pi}(s).\end{aligned}$$

As a result,

$$P_h^{\mathcal{M},\pi}(s) \le P^{\mathcal{M}}(s \in \tau_\pi) \le \max_\pi P^{\mathcal{M}}(s \in \tau_\pi) \le C \max_\pi P^{\mathcal{M}_1}(s \in \tau_\pi)$$
$$\le C\delta.$$

By setting $\delta = \zeta/24CH^2S$ when performing reward-free RL in Phase II, we thus find that

$$\sum_{(s,a)\in U_\delta} \left| \left[(\hat{\mathbb{P}}^{(h)} - \mathbb{P}^{(h)}) \hat{V}_{h+1}^{\pi} \right](s,a) \right| P_h^{\mathcal{M},\pi}(s,a) \le \frac{\zeta}{24H}.$$

Concluding. By combining the bounds on (I) through (IV) and summing across $h = 0, \dots, H-1$, we find that

$$\left| \hat{V}_0^{\mathcal{M},\pi}(s_0) - V_0^{\mathcal{M},\pi}(s_0) \right| \le \frac{\zeta}{6}.$$

Note that this argument simultaneously applies to any such $\mathcal{M}$; therefore, the desired conclusion follows. □

The prior estimation result, together with Assumption 4.1, suggests that BOAT-VI should find an MDP that sufficiently overestimates the value of the task so long as not all exits have been found. This ensures that the exit-finding routine is triggered. Formally,

Lemma A.8. *Assume the preconditions of Proposition A.3, and that $E \neq \mathrm{Ext}(\mathcal{C})$. Additionally, let $t \in [T]$ be the task with a ζ-overoptimistic value when borrowing exits $\mathrm{Ext}(\mathcal{C}) \setminus E$. Finally, let $\tilde{V}^t$ be the value function returned by Algorithm 6 on $\mathcal{M}_t$. Then,*

$$\tilde{V}_0^t(s_0) - \hat{V}_t > \frac{2}{3}\zeta.$$

Proof. Throughout this proof, we omit the timestep 0 and the initial state s_0 for brevity. Define $\mathcal{M}^*$ and π^* to be the maximizers of

$$\max_{\mathcal{M}\in\mathbb{M}_t(E)} \max_\pi V^{\mathcal{M},\pi}.$$

Furthermore, let $\bar{\mathcal{M}}$ be the imagined MDP guaranteed by Assumption 4.1 on top of $\mathcal{M}_t$, such that $V^{\bar{\mathcal{M}},*} > V^{\mathcal{M}_t,*} + \zeta$. Then, we have that

$$\hat{V}^{\mathcal{M},\pi} - V^{\mathcal{M}_t,*} = \underbrace{(\hat{V}^{\mathcal{M},\pi} - \hat{V}^{\mathcal{M}^*,\pi^*})}_{\ge 0} + \underbrace{(V^{\hat{\mathcal{M}}^*,\pi^*} - V^{\mathcal{M}^*,\pi^*})}_{>-\zeta/6}$$
$$+ \underbrace{(V^{\mathcal{M}^*,\pi^*} - V^{\bar{\mathcal{M}},\bar{\pi}})}_{\ge 0} + \underbrace{(V^{\bar{\mathcal{M}},\bar{\pi}} - V^{\mathcal{M}_t,*})}_{>\zeta}$$
$$> \frac{5}{6}\zeta.$$

Furthermore,

$$V^{\mathcal{M}_t,*} - \hat{V}_t = \underbrace{\left(V^{\mathcal{M}_t,*} - \frac{1}{N_{\text{UCBVI}}} \sum_{k=1}^{N_{\text{UCBVI}}} V^{\mathcal{M}_t,\pi_k^{(t)}} \right)}_{\ge 0} + \underbrace{\left(\frac{1}{N_{\text{UCBVI}}} \sum_{k=1}^{N_{\text{UCBVI}}} V^{\mathcal{M}_t,\pi_k^{(t)}} - \hat{V}_t \right)}_{\ge -\zeta/6},$$

where the first follows from optimality, while the second follows from Proposition A.2. Thus, putting the two inequalities together,

$$\tilde{V}^t - \hat{V}_t > \frac{2}{3}\zeta.$$ □

While the prior algorithm ensures that at least one of the tasks will trigger the exit condition, the actual task that triggers the condition may not be the same one invoked in the proof above. Nevertheless, we can prove that the trigger condition ensures that the algorithm will find a new exit.

Lemma A.9. *Assume the preconditions of Proposition A.3, and that $E \neq \mathrm{Ext}(\mathcal{C})$. Let $t \in [T]$ be a task such that the value estimate $\tilde{V}^t$ returned by Algorithm 6 satisfies $\tilde{V}_0^t(s_0) - \hat{V}_t > (2/3)\zeta$, and let π be the optimal policy for $\tilde{V}$. Then, there exists $(s,a) \in \mathrm{Ext}(\mathcal{C}) \setminus E$ such that for some $t' \neq t$,*

(a) $N_{t'}(s,a) \geq N_{\mathrm{thresh}}^{\mathrm{TS}}$ *and* $\mathbb{P}_t(\cdot \mid s,a) \neq \mathbb{P}_{t'}(\cdot \mid s,a)$.

(b) $P^{M_t}((s,a) \in \tau_\pi) > \zeta/6KH$.

Proof. Let $\mathcal{M}$ be the implicit MDP defined by Algorithm 6 in the process of computing $\tilde{V}^t$. We will prove a value gap between $\mathcal{M}$ and $\mathcal{M}_t$, which implies that π must visit state-action pairs with imagined dynamics.

Note that

$$\begin{aligned} V_0^{\mathcal{M},\pi}(s_0) - V_0^{\mathcal{M}_t,*}(s_0) = &\underbrace{\left[V_0^{\mathcal{M},\pi}(s_0) - \tilde{V}_0^t(s_0)\right]}_{\geq -\zeta/6} + \underbrace{\left[\tilde{V}_0^t(s_0) - \hat{V}_t\right]}_{\geq (2/3)\zeta} \\ &+ \underbrace{\left[\hat{V}_t - \frac{1}{N_{\mathrm{UCBVI}}} \sum_{k=1}^{N_{\mathrm{UCBVI}}} V_0^{\mathcal{M}_t,\pi_k^{(t)}}(s_0)\right]}_{\geq -\zeta/6} \\ &+ \underbrace{\left[\frac{1}{N_{\mathrm{UCBVI}}} \sum_{k=1}^{N_{\mathrm{UCBVI}}} V_0^{\mathcal{M}_t,\pi_k^{(t)}}(s_0) - V_0^{\mathcal{M}_t,*}(s_0)\right]}_{\geq -\zeta/6}, \end{aligned}$$

where the first inequality comes from Lemma A.7 and the last two inequalities come from Proposition A.1. Thus, $V_0^{\mathcal{M},\pi}(s_0) - V_0^{\mathcal{M}_t,*}(s_0) \geq \zeta/6$.

We now leverage this value gap to show that π must use some exit (s,a) whose dynamics in $\mathcal{M}$ have been modified from $\mathcal{M}_t$ with some probability. Formally, define the set $\Delta = \left\{(s,a,h) \;\middle|\; \mathbb{P}_t(\cdot \mid s,a) \neq \mathbb{P}_h^{\mathcal{M}}(\cdot \mid s,a)\right\}$. By construction, $\Delta \subseteq \mathrm{Ext}(\mathcal{C}) \times [H]$, and for any $(s,a,h) \in \Delta$, there exists t' such that $N_{t'}(s,a) \geq N_{\mathrm{thresh}}^{\mathrm{TS}}$ and $\mathbb{P}_{t'}(\cdot \mid s,a) = \mathbb{P}_h^{\mathcal{M}}(\cdot \mid s,a)$. Furthermore, Δ must be non-empty, as otherwise, $\mathbb{P}_t = \mathbb{P}_h^M$ for all h, and thus $V_0^{\mathcal{M},\pi}(s_0) \leq V_0^{\mathcal{M}_t,*}(s_0)$, a contradiction. Therefore, by applying Lemma A.17, we find that

$$\frac{\zeta}{6H} < P^{\mathcal{M}_t}(\tau_\pi \cap \Delta \neq \varnothing) \leq \sum_{\{(s,a) \mid (s,a,h) \in \Delta\}} P^{\mathcal{M}_t}((s,a) \in \tau_\pi),$$

which implies the desired result, as $\{(s,a) \mid (s,a,h) \in \Delta\}$ has at most K elements. □

Because of Lemma A.9, we simply need to run π enough times and threshold at the number of samples needed to reliably determine which (s,a) pairs have an $O(\beta)$ change in TV distance between tasks.

Lemma A.10. *We work in the setting of Lemma A.9. Set*

$$N_{\mathrm{thresh}}^{\mathrm{ED}} = \Omega\left(\frac{S}{\beta^2} \log \frac{SAHN_{\mathrm{ED}}}{p}\right) \quad \textit{and} \quad N_{\mathrm{ED}} = \Omega\left(\frac{HK}{\zeta} N_{\mathrm{thresh}}^{\mathrm{ED}} + \frac{H^2K^2}{\zeta^2} \log \frac{K}{p}\right).$$

Then, if we execute π within $\mathcal{M}_t$ N_{ED} and let $N(s,a)$ be the number of times that (s,a) is played in this process, then with probability at least $1 - p/6K$, the following hold:

(a) For the (s,a) pair and task t' in Lemma A.9, $N(s,a) \geq N_{\mathrm{thresh}}^{\mathrm{ED}}$ and

$$\left\|\hat{\mathbb{P}}(\cdot \mid s,a) - \hat{\mathbb{P}}_{t'}(\cdot \mid s,a)\right\|_{\mathrm{TV}} > \frac{\beta}{2}.$$

(b) For any $(s,a) \notin \mathrm{Ext}(\mathcal{C})$ with $N(s,a) \geq N_{\mathrm{thresh}}^{\mathrm{ED}}$ and t' with $N_{t'}(s,a) \geq 0$,

$$\left\|\hat{\mathbb{P}}_t(\cdot \mid s,a) - \hat{\mathbb{P}}_{t'}(\cdot \mid s,a)\right\|_{\mathrm{TV}} \leq \frac{\beta}{2}.$$

Proof. By the choice of N_{ED} and the lower bound $P^{\mathcal{M}_t}((s,a) \in \tau_\pi) > \zeta/6KH$ from Lemma A.9, we guarantee that $N(s,a) \geq N_{\mathrm{thresh}}^{\mathrm{ED}}$ with probability at least $1 - p/12K$. Furthermore, due to the choice of $N_{\mathrm{thresh}}^{\mathrm{ED}}$, with probability at least $1 - p/12K$, we have that for any (s,a) with $N(s,a) \geq N_{\mathrm{thresh}}^{\mathrm{ED}}$,

$$\left\|\hat{\mathbb{P}}(\cdot \mid s,a) - \mathbb{P}_t(\cdot \mid s,a)\right\|_{\mathrm{TV}} < \frac{\beta}{4},$$

by applying Lemma A.16. We condition on these two events simultaneously for the rest of the proof, which occurs with probability at least $1 - p/6K$.

We now prove each part separately. For brevity, we omit (s,a) wherever it is understood.

(a) By applying the triangle inequality,

$$\|\mathbb{P}_t - \mathbb{P}_{t'}\|_{\mathrm{TV}} \leq \left\|\mathbb{P}_t - \hat{\mathbb{P}}\right\|_{\mathrm{TV}} + \left\|\hat{\mathbb{P}} - \hat{\mathbb{P}}_{t'}\right\|_{\mathrm{TV}} + \left\|\hat{\mathbb{P}}_{t'} - \mathbb{P}_{t'}\right\|_{\mathrm{TV}} \leq \left\|\hat{\mathbb{P}} - \hat{\mathbb{P}}_{t'}\right\|_{\mathrm{TV}} + \frac{\beta}{2}.$$

Therefore, by lower bounding the left-hand side using β-dynamics separation in Definition 4.1, we find that

$$\left\|\hat{\mathbb{P}}(\cdot \mid s,a) - \hat{\mathbb{P}}_{t'}(\cdot \mid s,a)\right\|_{\mathrm{TV}} > \frac{\beta}{2}.$$

(b) The triangle inequality implies that

$$\left\|\hat{\mathbb{P}} - \hat{\mathbb{P}}_{t'}\right\|_{\mathrm{TV}} \leq \underbrace{\left\|\hat{\mathbb{P}} - \mathbb{P}_t\right\|_{\mathrm{TV}}}_{\leq \beta/4} + \underbrace{\|\mathbb{P}_t - \mathbb{P}_{t'}\|_{\mathrm{TV}}}_{=0} + \underbrace{\left\|\mathbb{P}_{t'} - \hat{\mathbb{P}}_{t'}\right\|_{\mathrm{TV}}}_{\leq \beta/4} \leq \frac{\beta}{2},$$

where the bound on the first term is provided by Proposition A.1. □

The prior result demonstrates that if the exit-finding condition is detected at any point by the algorithm, then the algorithm finds a previously undiscovered exit in $\mathrm{Ext}(\mathcal{C})$. At this point, all that remains is to ensure that the algorithm sufficiently learns the dynamics of the newly-found exit in all of the meta-training tasks.

Lemma A.11. *Fix an $(s,a) \in \mathrm{Ext}(\mathcal{C})$, which was found via exit detection, and let*

$$N_{\mathrm{thresh}}^{\mathrm{EL}} = \Omega\left[L \max\left(\frac{H^4}{\zeta^2}, \frac{1}{\beta^2}\right) \log \frac{SAHN_{\mathrm{EL}}T}{p}\right]$$

Assume we run the exit-learning subroutine with

$$N_{\mathrm{EULER}}^{\mathrm{EL}} = \Omega\left[\frac{CS^2AH^3}{\alpha} \log^3\left(\frac{SAHT}{p}\right)\right] \quad \textit{and} \quad N_{\mathrm{EL}} = \Omega\left(\frac{CH}{\alpha} N_{\mathrm{thresh}}^{\mathrm{EL}} + \frac{C^2H^2}{\alpha^2} \log \frac{SAT}{p}\right).$$

in each of the tasks. Then, with probability at least $1 - p/6K$,

$$\max_{f:\mathcal{S}\to[0,H]} \left|\left[(\hat{\mathbb{P}}_t - \mathbb{P}_t)f\right](s,a)\right| \leq \min\left(\frac{\zeta}{24H}, \frac{\beta H}{2}\right)$$

for every $t \in [T]$.

Proof. Fix a task $t \in [T]$. Note that Assumption 4.1 implies that (s,a) is (α/H)-significant for some task. Then, (s,a) must be (α/CH)-significant for all of the other tasks by Assumption A.1.

By applying Lemma A.4, the set of policies found by the exit-learning subroutine for every task $t \in [T]$ satisfies

$$\begin{aligned}
&\max_\pi P((s,a) \in \tau_\pi) - \frac{1}{N_{\mathrm{EULER}}^{\mathrm{EL}}} \sum_{k=1}^{N_{\mathrm{EULER}}^{\mathrm{EL}}} P((s,a) \in \tau_{\pi_k}) \\
&\lesssim \sqrt{\max_\pi P((s,a) \in \tau_\pi) \frac{SA}{N_{\mathrm{EULER}}^{\mathrm{EL}}} \log \frac{SAHTN_{\mathrm{EULER}}^{\mathrm{EL}}}{p}} + \frac{S^2AH^2}{N_{\mathrm{EULER}}^{\mathrm{EL}}} \log^3 \frac{SAHTN_{\mathrm{EULER}}^{\mathrm{EL}}}{p}
\end{aligned}$$

with probability at least $1 - p/18TK$. By setting

$$N_{\text{EULER}}^{\text{EL}} \gtrsim \frac{CS^2AH^3}{\alpha} \log^3\left(\frac{HSAT}{p}\right),$$

we thus find that for any $t \in [T]$,

$$\frac{\alpha}{2CH} \leq \frac{1}{2} \max_{\pi} P((s,a) \in \tau_\pi) \leq \frac{1}{N_{\text{EULER}}^{\text{EL}}} \sum_{\pi \in \Phi_t(s,a)} P((s,a) \in \tau_\pi).$$

Note that the right-hand side is exactly the probability that the trajectory of a randomly chosen policy in $\Phi_t(s,a)$ contains (s,a) in the trajectory. Therefore, by applying Lemma A.15, playing

$$N_{\text{EL}} = \Omega\left(\frac{CH^2 N_{\text{thresh}}^{\text{EL}}}{\alpha} + \frac{C^2H^4}{\alpha^2} \log\frac{SAT}{p}\right)$$

is sufficient to guarantee that we obtain at least $N_{\text{thresh}}^{\text{EL}}$ samples from $\mathbb{P}_t(\cdot \mid s,a)$ with probability at least $1 - p/18TK$. Since $(s,a) \in \text{Ext}(\mathcal{C})$, $\mathbb{P}_t(\cdot \mid s,a)$ (and by extension, $\hat{\mathbb{P}}_t(\cdot \mid s,a)$) is supported on $\text{Ent}(\mathcal{S})$. Therefore, we can modify the proof in Lemma A.16 so that with probability at least $1 - p/18TK$ we get the bound

$$\begin{aligned}\max_{f:\mathcal{S}\to[0,H]} \left|\left[(\hat{\mathbb{P}}_t - \mathbb{P}_t)f\right](s,a)\right| &\leq \max_{f:\text{Ent}(\mathcal{S})\to[0,H]} \left|\left[(\hat{\mathbb{P}}_t - \mathbb{P}_t)f\right](s,a)\right| \\ &\leq \min\left(\frac{\zeta}{24H}, \frac{\beta H}{2}\right)\end{aligned}$$

with $N_{\text{thresh}}^{\text{EL}}$ depending linearly on L instead of S.

Note that by performing a union-bound, all events occur with probability at least $1 - p/6TK$. Performing a second union-bound over all of the available tasks results in the desired failure probability. □

At this point, we have effectively proven the second half of our Phase III guarantee. All that remains is to prove that if $\{(s,a) \mid \text{IsExit}[s,a]\} = \text{Ext}(\mathcal{C})$, then the algorithm terminates without triggering the exit-finding condition.

Lemma A.12. *Assume that $E = \{(s,a) \mid \text{IsExit}[s,a]\} = \text{Ext}(\mathcal{C})$. Then, under the preconditions of Proposition A.3, every task satisfies*

$$\left|\tilde{V}^t(s_0) - \hat{V}_t\right| \leq \frac{2}{3}\zeta.$$

Proof. Fix a task $t \in [T]$. Once $E = \text{Ext}(\mathcal{C})$, then $\mathbb{M}_t(E) = \{\mathcal{M}_t\}$, since the only (s,a)-dynamics that can be substituted from other tasks are those of non-exits, which do not change between tasks. Therefore, by Lemma A.5, $\tilde{V}^t(s_0) = \hat{V}_0^{\mathcal{M},*}(s_0)$. Finally, by applying Lemma A.7, we thus find that

$$\tilde{V}^t(s_0) - \hat{V}_t = \left[\hat{V}_0^{\mathcal{M},*}(s_0) - V_0^{\mathcal{M},*}(s_0)\right] + \left[V_0^{\mathcal{M},*}(s_0) - \hat{V}_t\right] < \frac{\zeta}{3}.$$

The desired result follows since the argument holds for any task t. □

A.4.4 Proof of Theorem A.1

In this section, we compile the guarantees provided by each of the three phases into a proof of Theorem A.1.

Proof of Theorem A.1. We condition on the following high-probability events:

(a) Proposition A.1 guarantees for all $\mathcal{M}_t$ with $t \in [T]$.

(b) Proposition A.2.

Via a union-bound, this holds with probability at least $1 - (2/3)p$.

To prove the theorem, we provide an induction-based analysis of Phase III. In particular, we will show that while $E = \{(s, a) \mid \textsc{IsExit}[s, a]\} \subsetneq \mathrm{Ext}(\mathcal{C})$, Phase III will add at least one state-action pair to E that belongs to $\mathrm{Ext}(\mathcal{C}) \setminus E$.

Formally, let F_k denote the internal state of the algorithm after it has added k state-action pairs. Note that with $k = 0$, $\{(s, a) \mid \textsc{IsExit}[s, a]\}$ in F_k is empty. Thus, F_k satisfies the preconditions of Proposition A.3, which in turn implies that the algorithm adds a new state-action pair in $\mathrm{Ext}(\mathcal{C})$ and sufficiently learns its dynamics for all tasks with probability at least $1 - p/3K$. In short, the internal state of the algorithm at time F_1 also satisfies the preconditions of Proposition A.3 with probability at least $1 - p/3K$. More generally, Proposition A.3 ensures that if F_k satisfies the preconditions of Proposition A.3, then so does F_{k+1}. Therefore, with probability at least $1 - p/3$, F_K satisfies the preconditions of Proposition A.3, which necessarily implies that $\{(s, a) \mid \textsc{IsExit}[s, a]\} = \mathrm{Ext}(\mathcal{C})$ in F_K, and thus the algorithm exits as desired. By performing a union bound, all this occurs with probability at least $1 - p$. □

A.5 Proving the Meta-Training Guarantee

Having demonstrated that $\mathrm{Ext}(\mathcal{C})$ can be successfully recovered by interacting with the environment, we now show that the data can also be used to determine exit reachability and implement the hierarchy oracle.

We formally state our main result here:

Theorem A.2. *Assume that $\mathcal{M}_1, \ldots, \mathcal{M}_T$ have a latent hierarchy with respect to $(\{Z_c\}, \mathrm{Ent}(\cdot), \mathrm{Ext}(\cdot))$, and assume that these tasks satisfy the (α, ζ)-coverage condition in Assumption 4.1. Furthermore, we assume the additional assumptions in Section A.3. Then, by running the algorithm in Section A.1 with the parameters in Table 1, with probability at least $1 - p$, the collected data can be used to implement the following:*

(a) An ε-suboptimal hierarchy oracle.

(b) A function $\mathrm{AvExt}(s) : \mathrm{Ent}(\mathcal{S}) \to \mathcal{P}(\mathrm{Ext}(\mathcal{C}))$ such that, given $s \in \mathrm{Ent}(Z_s)$, returns $\mathrm{Ext}(Z_s)$.

The algorithm achieves both of these with query complexity

$$\tilde{O}\left[\frac{S^4A}{\min(\rho\min(\varepsilon,\varepsilon_0),\zeta/C)} + \frac{S^2A}{\min(\rho\min(\varepsilon,\varepsilon_0)^2,\zeta^2/C)} + T\left(\frac{SA}{\min(\alpha,\zeta)^2} + \frac{KS}{\zeta\beta^2} + \frac{K^2}{\zeta^2} + \frac{CKS^2A}{\alpha} + \frac{CKL}{\alpha\max(\zeta,\beta)^2}\right)\right]\mathrm{poly}(H).$$

A.5.1 Implementing the Hierarchy Oracle

We first show that we can implement the hierarchy oracle in this section. In particular, we have the following result:

Proposition A.4. *Let $\mathcal{M}$ be the MDP corresponding to the tuple $(x, f, r, \tilde{H})$ as described in Definition 4.3. Then, given $(x, f, r, \tilde{H})$, we can form the following estimator for $\mathbb{P}_f$:*

$$\hat{\mathbb{P}}_f(\cdot \mid s, a) = \begin{cases} \delta(f(s,a)) & (s,a) \in \mathrm{Ext}(\mathcal{C}) \\ \delta(s) & s \in \{\texttt{GOAL}, \texttt{FAIL}\} \\ \hat{\mathbb{P}}_0(\cdot \mid s, a) & \textit{otherwise} \end{cases},$$

where $\hat{\mathbb{P}}_0$ is the estimator obtained from Phase II in Section A.1. Assuming that the high-probability event in Proposition A.2 holds for $\delta \le \rho\varepsilon/2SH^2$, value iteration using $\hat{\mathbb{P}}_f$ returns a policy π such that $V_0^{\mathcal{M},*}(x) - V_0^{\mathcal{M},\pi}(x_0) \le \varepsilon$.

Throughout the rest of this section, we fix the tuple $(x, f, r, \tilde{H})$ and the corresponding MDP $\mathcal{M}$. Furthermore, we write Z for the cluster containing x.

To prove Proposition A.4, we will show that $\mathcal{M}$ can be sufficiently simulated so that the value of any policy can be reasonably estimated. Given this simulation result, we can then show that value iteration finds the desired policy. This simulation result depends on the following intermediate result, which provides insight as to why Phase II data is sufficient:

Lemma A.13. *For any $s^* \in Z$,*

$$\left[\max_\pi P^{\mathcal{M}_1}(x \in \tau_\pi)\right]\left[\max_\pi P^{\mathcal{M}}(s^* \in \tau_\pi)\right] \le \max_\pi P^{\mathcal{M}_1(2H)}(s^* \in \tau_\pi)$$

Proof. First, we note that there exists an MDP such that $P^{\mathcal{M}_1(2H)}(s^* \in \tau_\pi)$ is the corresponding value function. In particular, modifying $\mathcal{M}_1(2H)$ so that any action from s^* leads to a terminal state and defining $r(s,a) = \mathbb{1}[s = s^*]$ results in such an MDP.

Now, let π_x and π_{s^*} be the policies achieving

$$\max_\pi P^{\mathcal{M}_1}(x \in \tau_\pi) \quad \text{and} \quad \max_\pi P^{\mathcal{M}}(s^* \in \tau_\pi),$$

respectively. Consider the concatenation of π_x and π_{s^*} into a history-dependent policy that runs π_x until the agent reaches s, and switches to π_{s^*} thereafter. This policy reaches s with probability at least

$$\left[\max_\pi P^{\mathcal{M}_1}(x \in \tau_\pi)\right]\left[\max_\pi P^{\mathcal{M}}(s^* \in \tau_\pi)\right].$$

within the modified MDP described above. Since the optimal value among all policies is achieved by a history-independent policy, we obtain the desired inequality. □

Informally, the prior result states that if x is reachable within horizon H, then any state reachable from x within Z is also reachable in $\mathcal{M}_1$ within a $2H$ horizon. Therefore, performing reward-free RL with horizon $2H$ during Phase II provides coverage over all clusters. Now, we prove the simulation result.

Lemma A.14. *Assume that the Phase II guarantee in Proposition A.2 is instantiated for $\delta \le \rho\varepsilon/4SH^2$. Then, if V^π is the value of π under $\mathcal{M}$, and $\hat{V}^\pi$ is its corresponding estimate under $\hat{\mathbb{P}}_f$, then*

$$\left|\hat{V}_0^\pi(x) - V_0^\pi(x)\right| \le \frac{\varepsilon}{2}.$$

Proof. The proof follows similarly to that of Lemma A.7. By the performance difference lemma,

$$\begin{aligned}\left|\hat{V}_0^\pi(s) - V_0^\pi(s)\right| &\le \sum_{h=0}^{\tilde{H}-1} \mathbb{E}_{\mathcal{M},\pi}\left[\left|\left[\left(\hat{\mathbb{P}}_f - \mathbb{P}_f\right)\hat{V}_{h+1}^\pi\right](s_h,a_h)\right|\right] \\ &\le \sum_{h=0}^{\tilde{H}-1}\sum_{(s,a)}\left|\left[\left(\hat{\mathbb{P}}_f - \mathbb{P}_f\right)\hat{V}_{h+1}^\pi\right](s,a)\right| P_h^\pi(s,a).\end{aligned}$$

Observe that if $s \in \mathcal{S} \setminus Z$, $P_h^\pi(s,a) = 0$ for any π. Furthermore, since the dynamics within $\{\texttt{GOAL}, \texttt{FAIL}\}$ are known, $(\hat{\mathbb{P}}_f - \mathbb{P}_f)\hat{V}_{h+1}^\pi(s,a) = 0$ for $s \in \{\texttt{GOAL}, \texttt{FAIL}\}$. Therefore, we can restrict the sum to be over $Z \times \mathcal{A}$.

Now, let Z_δ denote the set of δ-significant (s,a) pairs in $Z \times \mathcal{A}$ from x, for some δ to be determined. For a fixed $h \in [\tilde{H}]$, we can decompose the inner sum as

$$\begin{aligned}&\sum_{(s,a)}\left|\left[\left(\hat{\mathbb{P}}_f - \mathbb{P}_f\right)\hat{V}_{h+1}^\pi\right](s,a)\right| P_h^\pi(s,a) \\ &\quad\le \underbrace{\sum_{(s,a)\in Z_\delta}\left|\left[\left(\hat{\mathbb{P}}_f - \mathbb{P}_f\right)\hat{V}_{h+1}^\pi\right](s,a)\right| P_h^\pi(s,a)}_{\text{(I)}} + \underbrace{\sum_{(s,a)\notin Z_\delta}\left|\left[\left(\hat{\mathbb{P}}_f - \mathbb{P}_f\right)\hat{V}_{h+1}^\pi\right](s,a)\right| P_h^\pi(s,a)}_{\text{(II)}}.\end{aligned}$$

Bounding (II): Error from δ-Insignificance By the definition of δ-significance,

$$\text{(II)} = \sum_{(s,a)\notin Z_\delta}\left|\left[\left(\hat{\mathbb{P}}_f - \mathbb{P}_f\right)\hat{V}_{h+1}^\pi\right](s,a)\right| P_h^\pi(s,a) \le H\sum_{s\notin Z_\delta} P_h^\pi(s) \le HS\delta \le \frac{\varepsilon}{4H},$$

where the last inequality follows from setting $\delta = \varepsilon/4SH^2$.

Bounding (I): Reference Dynamics Error. By the Cauchy-Schwarz inequality,

$$\begin{aligned}\text{(I)} &= \sum_{(s,a)\in Z_\delta} \left|\left[\left(\hat{\mathbb{P}}_f - \mathbb{P}_f\right)\hat{V}^\pi_{h+1}\right](s,a)\right| P^\pi_h(s,a) \\ &\le \left[\sum_{(s,a)\in Z_\delta} \left|\left[\left(\hat{\mathbb{P}}_f - \mathbb{P}_f\right)\hat{V}^\pi_{h+1}\right](s,a)\right|^2 P^\pi_h(s,a)\right]^{1/2}.\end{aligned}$$

Then,

$$\begin{aligned}&\sum_{(s,a)\in Z_\delta} \left|\left[\left(\hat{\mathbb{P}}_f - \mathbb{P}_f\right)\hat{V}^\pi_{h+1}\right](s,a)\right|^2 P^\pi_h(s,a) \\ &\quad\le \max_{\nu:\mathcal{S}\to\mathcal{A}} \sum_{(s,a)\in Z_\delta} \left|\left[\left(\hat{\mathbb{P}}_f - \mathbb{P}_f\right)\hat{V}^\pi_{h+1}\right](s,a)\right|^2 P^\pi_h(s)\mathbb{1}\left[\nu(s)=a\right].\end{aligned}$$

Since x is ρ-significant in $\mathcal{M}_1(2H)$ by Definition A.2, Lemma A.13 together with δ-significance in $\mathcal{M}$ implies $\rho\delta$-significance in $\mathcal{M}_1(2H)$. Therefore,

$$P^\pi_h(s) \le \max_\pi P^{\mathcal{M}}(s\in\tau_\pi) \le \frac{1}{\rho}\max_\pi P^{\mathcal{M}_1(2H)}(s\in\tau_\pi) \le \frac{4HSA}{\rho}\mu(s,a),$$

where the last inequality follows by part (a) of the Phase II guarantee in Proposition A.2. Substituting into the prior expression,

$$\begin{aligned}&\max_{\nu:\mathcal{S}\to\mathcal{A}} \sum_{(s,a)\in Z_\delta} \left|\left[\left(\hat{\mathbb{P}}_f - \mathbb{P}_f\right)\hat{V}^\pi_{h+1}\right](s,a)\right|^2 P^\pi_h(s)\mathbb{1}\left[\nu(s)=a\right] \\ &\quad\le \frac{4HSA}{\rho}\max_{\nu:\mathcal{S}\to\mathcal{A}} \sum_{(s,a)\in Z_\delta} \left|\left[\left(\hat{\mathbb{P}}_f - \mathbb{P}_f\right)\hat{V}^\pi_{h+1}\right](s,a)\right|^2 \mathbb{1}\left[\nu(s)=a\right]\mu(s,a) \\ &\quad\le \frac{4HSA}{\rho}\max_{\nu:\mathcal{S}\to\mathcal{A}} \mathbb{E}_{(s,a)\sim\mu}\left[\left|\left[\left(\hat{\mathbb{P}}_f - \mathbb{P}_f\right)\hat{V}^\pi_{h+1}\right](s,a)\right|^2 \mathbb{1}\left[\nu(s)=a\right]\right].\end{aligned}$$

Thus by applying part (b) of the Phase II guarantee in Proposition A.2, we have that

$$\text{(I)} \le \sqrt{\frac{4HSA}{\rho}\max_{\nu:\mathcal{S}\to\mathcal{A}} \mathbb{E}_{(s,a)\sim\mu}\left[\left|\left[\left(\hat{\mathbb{P}}_f - \mathbb{P}_f\right)\hat{V}^\pi_{h+1}\right](s,a)\right|^2 \mathbb{1}\left[\nu(s)=a\right]\right]} \le \frac{\varepsilon}{4H}.$$

Concluding. By combining the bounds on (I) and (II), we obtain the desired result. □

With this estimation result, we can now prove Proposition A.4.

Proof of Proposition A.4. Let π be the policy found by value iteration using $\hat{\mathbb{P}}_f$, which achieves the maximal value in the corresponding MDP. Then, by Lemma A.14

$$V^*_0(x) - V^\pi_0(x) \le \underbrace{\left[V^*_0(s_0) - \hat{V}^{\pi^*}_0(s_0)\right]}_{\le\varepsilon/2} + \underbrace{\left[\hat{V}^{\pi^*}_0(s_0) - \hat{V}^{\hat{\pi}}_0(s_0)\right]}_{\le 0} + \underbrace{\left[\hat{V}^{\hat{\pi}}_0(s_0) - V^{\hat{\pi}}_0(s_0)\right]}_{\le\varepsilon/2} \le \varepsilon.$$ □

A.5.2 Determining Available Exits

In this section, we prove that we can determine the set of available exits. We have the following formal result:

Proposition A.5. *Assume access to the ε-suboptimal hierarchy oracle from the previous section and that the guarantee in Theorem A.1 holds. Then, we can implement the function* $\mathrm{AvExt}(s) : \mathrm{Ent}(\mathcal{S}) \to \mathcal{P}(\mathrm{Ext}(\mathcal{C}))$ *which, given* $s \in \mathrm{Ent}(Z_s)$*, returns the subset of* $\mathrm{Ext}(Z_s)$ *that is reachable from* s*.*

Proof. Fix an input $x \in \mathrm{Ent}(\mathcal{S})$, which we assume belongs to some cluster Z_x. It suffices to demonstrate that we can implement $\mathbb{1}[e \in Z_x]$ for any fixed $e \in \mathrm{Ent}(\mathcal{S})$. Define

$$f_e(s,a) = \begin{cases} \texttt{GOAL} & (s,a) = e \\ \texttt{FAIL} & \text{otherwise} \end{cases}$$

and $r_e(s,a) = \mathbb{1}[(s,a) = e]$. By Definition A.3, the MDP $\mathcal{M}$ corresponding to the tuple (x, f_e, r_e, H) has optimal value $V^* = \varepsilon_0 \mathbb{1}[e \in Z_x \wedge e \text{ reachable from } x]$. Additionally, by Lemma A.14, $|V_0^\pi(s) - \hat{V}_0^\pi(x)| \le \varepsilon/2$. We now proceed by cases. If $e \notin Z_x$ or e is not reachable from x, then $V_0^\pi(x) = 0$ for any policy π, and thus value iteration can only find a policy π with $\hat{V}_0^\pi(x) \le \varepsilon_0/3$. Otherwise, for $e \in Z_x$, $V_0^*(x) = \varepsilon_0$, and thus value iteration necessarily must find a π with $\hat{V}_0^\pi(x) \ge 2\varepsilon_0/3$. Putting these together, if $\hat{V}$ is the optimal estimated value in $\mathcal{M}$, then

$$\mathbb{1}[e \in Z_x] = \mathbb{1}\left[\hat{V} \ge \frac{2}{3}\varepsilon_0\right].$$

Note that this is implementable for all $e \in \mathrm{Ext}(\mathcal{C})$ (and returns a subset of $\mathrm{Ext}(Z_x)$ for the query above) since the set of exits are already known. □

A.5.3 Finalizing the Guarantee: Query Complexity

In this section, we finalize the proof of the meta-training guarantee by computing the query complexity.

Proof of Theorem A.2. As demonstrated by Proposition A.4 and Proposition A.5, running the algorithm in Section A.1 with the parameters in Table 1 provides the desired guarantees with probability at least $1 - p$.

To compute the query complexity, observe that we perform the following number of trajectories while executing the algorithm in Section A.1.

$$O\left[T(N_{\text{UCBVI}} + N_{\text{TS}}) + N_{\text{EULER}}^{\text{RF}} + N_{\text{RF}} + KN_{\text{ED}} + TK(N_{\text{EULER}}^{\text{EL}} + N_{\text{EL}})\right].$$

Ignoring terms that do not depend on T or ε, we obtain the claim. □

A.6 Brute-Force Learning of the Hiearchy

Algorithm 11 Brute-force learning of the latent hierarchy.

Require: $(\mathcal{M}_1, \ldots, \mathcal{M}_T)$, N_{EULER} iterations, N policy samples, threshold N_{thresh}
for all $t \in [T], s \in \mathcal{S}$ **do**
  Create MDP $\mathcal{M}_t^s$ so $P(\ominus \mid s, a) = 1$ for any a.
  $\tilde{r}_s(s', a') \leftarrow \mathbb{1}[s' = s]$.
  $\Psi_t^s \leftarrow \text{EULER}(\mathcal{M}_t^s, r, N_{\text{EULER}})$
end for
for all $t \in [T], s \in \mathcal{S}, a \in \mathcal{A}$ **do**
  Modify policies in Ψ_t^s to play a on s.
  for all $n \in [N]$ **do**
    Sample $\pi \sim \text{Unif}(\Psi_t^s)$.
    Play π in $\mathcal{M}_t$, collect sample (s, a, s_n') if (s,a) is encountered
  end for
  $N_t(s,a) \leftarrow$ number of times (s,a) is encountered above.
  $\hat{\mathbb{P}}_t(\cdot \mid s, a) \leftarrow$ estimate of (s,a) dynamics in t.
end for
output $\left\{(s,a) \;\middle|\; (\exists t \ne t') \left\|\hat{\mathbb{P}}_t - \hat{\mathbb{P}}_{t'}\right\|_{\text{TV}} > \beta/2, \min(N_t(s,a), N_{t'}(s,a)) \ge N_{\text{thresh}}\right\}$.

Theorem A.3. *Assume that Algorithm 11 is run with parameters satisfying*

$$N_{\text{EULER}} = \Omega\left(\frac{CH^3S^2A}{\alpha}\log^3\frac{SAHT}{p}\right)$$

and

$$N_{\text{thresh}} = \Omega\left(\frac{S}{\beta^2}\log\frac{SAHNT}{p}\right) \quad \textit{and} \quad N = \Omega\left(\frac{CH}{\alpha}N_{\text{thresh}} + \frac{C^2H^2}{\alpha^2}\log\frac{SAT}{p}\right)$$

Then, the set returned by the algorithm is exactly $\text{Ext}(\mathcal{C})$ *with probability at least* $1-p$*. Furthermore, the algorithm achieves this result with query complexity*

$$\tilde{O}\left[T\left(\frac{CS^4A}{\alpha} + \frac{CS^2A}{\alpha\beta^2}\right)\right]\text{poly}(H).$$

Proof. For any $(s,a) \in \text{Ext}(\mathcal{C})$, Lemma A.2 implies that s is α/H-significant for some task $t \in [T]$. Therefore, s is α/CH-significant for any task $t \in [T]$, by Assumption A.1.

Now, by an argument similar to that used in the proof of Lemma A.2, we have that with probability at least $1 - p/3T$, the choice of N_{EULER} implies

$$\frac{1}{N_{\text{EULER}}}\sum_{\pi\in\Psi_t^s} P^{\mathcal{M}_t}(s \in \tau_\pi) \geq \frac{\alpha}{2CH}$$

for any exit $(s,a) \in \text{Ext}(\mathcal{C})$ and a fixed task $t \in [T]$. Therefore, by a union-bound over the tasks, the same guarantee holds for all tasks simultaneously with probability at least $1 - p/3$.

Now, for any fixed (α/CH)-significant (s,a) pair, sampling from Ψ_t^s at least N times guarantees that with probability at least $1 - p/3SAT$, $N_t(s,a) \geq N_{\text{thresh}}$. Therefore, once again performing the necessary union-bound, we obtain the same result uniformly over any (α/CH)-significant (s,a) and $t \in [T]$ with probability at least $1 - p/3$.

Finally, for a fixed (s,a) and t, the estimator for $\mathbb{P}_t(\cdot \mid s,a)$ satisfies the property that when $N(s,a) > 0$,

$$\left\|\hat{\mathbb{P}}_t(\cdot \mid s,a) - \mathbb{P}_t(\cdot \mid s,a)\right\|_{\text{TV}} \leq \sqrt{\frac{H^2S}{N_t(s,a)}\log\frac{SAHNT}{p}} + \frac{HS}{N_t(s,a)}\log\frac{SAHNT}{p}$$

with probability at least $1 - p/3SAT$, using an argument similar to that used in Lemma A.16. Again, by a union bound, the same guarantee holds for any (s,a) and $t \in [T]$. In particular, for any (s,a) with $N_t(s,a) \geq N_{\text{thresh}}$,

$$\left\|\hat{\mathbb{P}}_t(\cdot \mid s,a) - \mathbb{P}_t(\cdot \mid s,a)\right\|_{\text{TV}} \leq \frac{\beta}{4}.$$

Therefore, by a similar argument to Lemma A.10, the following are true:

(a) If $(s,a) \in \text{Ext}(\mathcal{C})$, then there exists t, t' for which

$$\left\|\hat{\mathbb{P}}_t(\cdot \mid s,a) - \hat{\mathbb{P}}_{t'}(\cdot \mid s,a)\right\|_{\text{TV}} > \frac{\beta}{2}.$$

(b) If $(s,a) \notin \text{Ext}(\mathcal{C})$, then for any $t \neq t'$ with $N_t(s,a), N_{t'}(s,a) \geq N_{\text{thresh}}$,

$$\left\|\hat{\mathbb{P}}_t(\cdot \mid s,a) - \hat{\mathbb{P}}_t(\cdot \mid s,a)\right\|_{\text{TV}} \leq \frac{\beta}{2},$$

Putting everything together, we see that the set returned by Algorithm 11 is exactly $\text{Ext}(\mathcal{C})$, with probability at least $1 - p$. □

A.7 Technical Lemmas

Lemma A.15. *Let* $X_1, \ldots, X_M$ *be i.i.d.* $\text{Ber}(p)$ *random variables. Then, if*

$$M = \Omega\left(\frac{N}{p} + \frac{1}{p^2}\log\frac{1}{\delta}\right),$$

then with probability at least $1 - \delta$*,*

$$\sum_{i=1}^{M} \mathbb{1}\left[X_i = 1\right] \geq N.$$

Proof. By applying Hoeffding's inequality,

$$\begin{aligned} P\left(\sum_{i=1}^{M} \mathbb{1}\left[X_i = 1\right] < N\right) &= P\left(\frac{1}{M}\sum_{i=1}^{M} \mathbb{1}\left[X_i = 1\right] - p < \frac{N}{M} - p\right) \\ &= P\left(\frac{1}{M}\sum_{i=1}^{M} \mathbb{1}\left[X_i = 0\right] - (1-p) > p - \frac{N}{M}\right) \\ &\leq \exp\left[-2M\left(p - \frac{N}{M}\right)^2\right] \end{aligned}$$

Setting the final expression to the failure probability δ and solving, we obtain the quadratic inequality

$$p^2M^2 - \left(2Np + \frac{1}{2}\log\frac{1}{\delta}\right)M + N^2 \geq 0.$$

Finally, via solving this inequality for M, we find that

$$M \geq \frac{2N}{p} + \frac{1}{2p^2}\log\frac{1}{\delta}$$

is sufficient to guarantee the desired event with failure probability δ, as desired. □

Lemma A.16 (Dynamics estimation error bound)**.** *Fix a policy π, MDP with stationary dynamics $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathbb{P}, r, H)$, and $N \in \mathbb{N}$. Assume that π is played N times in $\mathcal{M}$, and all transitions are used to form an estimator $\hat{\mathbb{P}}(\cdot \mid s, a)$ using empirical averages. For any $(s, a) \in \mathcal{S} \times \mathcal{A}$, let $N(s, a)$ be the number of times (s, a) is encountered in this process. Then, with probability at least $1 - p$, any (s, a) with $N(s, a) > 0$ satisfies*

$$\sup_{f:\mathcal{S}\to[0,H]} \left|\left[\left(\hat{\mathbb{P}} - \mathbb{P}\right) f\right](s, a)\right| \leq \sqrt{\frac{H^2S}{N(s,a)}\log\frac{SAHN}{p}} + \frac{HS}{N(s,a)}\log\frac{SAHN}{p}.$$

Proof. Assume that the obtained samples are given by $\{(s_k, a_k, s'_k) \mid k \in [HN]\}$, so that $(s_{Hn+r}, a_{Hn+r}, s'_{Hn+r+1})$ is the r^{th} time step in the n^{th} execution of π in $\mathcal{M}$ for any $0 \leq n \leq N-1$ and $0 \leq r \leq H-1$.

Fix any $(s, a) \in \mathcal{S} \times \mathcal{A}$, and assume that $(s_{(j)}, a_{(j)}, s'_{(j)})$ is the j^{th} sample from $\mathbb{P}(\cdot \mid s, a)$. Furthermore, let $m_j(s, a)$ denote the index at which the j^{th} sample is obtained. We claim that for any $s^* \in \mathcal{S}$ and $0 < M \leq HT$,

$$\begin{aligned} &\left|\frac{1}{M}\sum_{j=1}^{M} \mathbb{1}\left[m_j(s, a) \leq HT\right]\left(\mathbb{1}\left[s'_{(j)} = s^*\right] - \mathbb{P}(s^* \mid s, a)\right)\right| \\ &\qquad \leq \sqrt{\frac{\mathbb{P}(s' \mid s, a)}{M}\log\frac{S}{\delta}} + \frac{1}{M}\log\frac{S}{\delta}. \end{aligned}$$

Let $\mathcal{F}_i$ be defined as the σ-algebra induced by the set of random variables

$$\left\{\left(m_j(a), \mathbb{1}\left[s'_{(j)} = s^*\right]\right) \;\middle|\; j \leq i\right\}.$$

Clearly, $(\mathcal{F}_i)$ is a filtration such that the j^{th} term in the sum above is measurable with respect to $\mathcal{F}_j$. Furthermore, observe that

$$\begin{aligned} &\mathbb{E}\left[\mathbb{1}\left[m_j(s, a) \leq HT\right]\left(\mathbb{1}\left[s'_{(j)} = s^*\right] - \mathbb{P}(s^* \mid s, a)\right) \;\middle|\; \mathcal{F}_{j-1}\right] \\ &\quad = \mathbb{E}\left[\mathbb{1}\left[s'_{(j)} = s^*\right] - \mathbb{P}(s^* \mid s, a) \;\middle|\; \mathcal{F}_{j-1}, m_j(s, a) \leq HT\right] P(m_j(s, a) \leq HT) \\ &\quad = 0. \end{aligned}$$

Therefore, the random variables in the sum forms martingale difference sequence. Furthermore, the sequence is bounded in $[-1, 1]$, and satisfies

$$\begin{aligned}
&\operatorname{Var}\left[\mathbb{1}\left[m_j(s,a) \le HT\right]\left(\mathbb{1}\left[s'_{(j)} = s^*\right] - \mathbb{P}(s^* \mid s,a)\right) \;\middle|\; \mathcal{F}_{j-1}\right] \\
&\quad = \mathbb{E}\left[\operatorname{Var}\left[\mathbb{1}\left[s'_{(j)} = s^*\right] - \mathbb{P}(s^* \mid s,a) \;\middle|\; \mathcal{F}_{j-1}, m_j(s,a) \le HT\right] \;\middle|\; \mathcal{F}_{j-1}\right] \\
&\quad \le \mathbb{P}(s^* \mid s,a).
\end{aligned}$$

Therefore, by applying Azuma-Bernstein, we have that

$$\begin{aligned}
&\left|\frac{1}{M}\sum_{j=1}^{M}\mathbb{1}\left[m_j(s,a) \le HT\right]\left(\mathbb{1}\left[s'_{(j)} = s^*\right] - \mathbb{P}(s^* \mid s,a)\right)\right| \\
&\qquad \le \sqrt{\frac{2\mathbb{P}(s' \mid s,a)}{M}\log\frac{SAHN}{\delta}} + \frac{2}{M}\log\frac{SAHN}{\delta}.
\end{aligned}$$

with probability at least $1 - p/SAHN$.

By applying a union bound on (s, a, s^*) and M, we thus have that with probability at least $1 - p$,

$$\begin{aligned}
&\left|\frac{1}{M}\sum_{j=1}^{M}\mathbb{1}\left[m_j(s,a) \le HT\right]\left(\mathbb{1}\left[s'_{(j)} = s^*\right] - \mathbb{P}(s^* \mid s,a)\right)\right| \\
&\qquad \le \sqrt{\frac{2\mathbb{P}(s' \mid s,a)}{M}\log\frac{SAHN}{\delta}} + \frac{2}{M}\log\frac{SAHN}{\delta}
\end{aligned}$$

holds for any (s, a, s^*) and M. Conditioned on this event, we thus have that for any (s, a) with $N(s, a) > 0$,

$$\begin{aligned}
\left\|\hat{\mathbb{P}}_t(\cdot \mid s,a) - \mathbb{P}_t(\cdot \mid s,a)\right\|_{\mathrm{TV}} &= \frac{1}{2}\sum_{s' \in \mathcal{S}}\left|\hat{\mathbb{P}}_t(s' \mid s,a) - \mathbb{P}_t(s' \mid s,a)\right| \\
&\lesssim \sum_{s' \in \mathcal{S}}\sqrt{\frac{\mathbb{P}(s' \mid s,a)}{N(s,a)}\log\frac{SAHN}{\delta}} + \frac{S}{N(s,a)}\log\frac{SAHN}{\delta} \\
&\lesssim \sqrt{\frac{S}{N(s,a)}\log\frac{SAHN}{\delta}} + \frac{S}{N(s,a)}\log\frac{SAHN}{\delta}.
\end{aligned}$$

The final result follows simply by noting that

$$\left|\left[(\hat{\mathbb{P}}_t - \mathbb{P}_t)f\right](s,a)\right| \lesssim \left\|\hat{\mathbb{P}}_t(\cdot \mid s,a) - \mathbb{P}_t(\cdot \mid s,a)\right\|_{\mathrm{TV}} \|f\|_\infty. \qquad \square$$

Lemma A.17. *Fix two MDPs $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathbb{P}, r, H)$ and $\mathcal{M}' = (\mathcal{S}, \mathcal{A}, \mathbb{P}', r, H)$. Let Δ denote the subset of $\mathcal{S} \times \mathcal{A} \times [H]$ for which $\mathbb{P}_h(\cdot \mid s,a) \ne \mathbb{P}'_h(\cdot \mid s,a)$. Then, for any policy π,*

$$V_0^{\mathcal{M}',\pi}(s_0) - V_0^{\mathcal{M},*}(s_0) > \rho \implies P_{\mathcal{M}}\left(\tau_\pi \cap \Delta \ne \varnothing\right) = P_{\mathcal{M}'}\left(\tau_\pi \cap \Delta \ne \varnothing\right) > \frac{\rho}{H}.$$

Proof. Write $q = P_{\mathcal{M}'}\left(\tau_\pi \cap \Delta \ne \varnothing\right)$. Note that $V_0^{\mathcal{M}',\pi}(s_0)$ can be decomposed as

$$\begin{aligned}
V_0^{\mathcal{M}',\pi}(s_0) &= q\mathbb{E}_{\mathcal{M}'}\left[\sum_{h=0}^{H-1} r_h(s_h,a_h) \;\middle|\; \tau_\pi \cap \Delta \ne \varnothing\right] \\
&\qquad + (1-q)\mathbb{E}_{\mathcal{M}'}\left[\sum_{h=0}^{H-1} r_h(s_h,a_h) \;\middle|\; \tau_\pi \cap \Delta = \varnothing\right] \\
&\le qH + (1-q)\mathbb{E}_{\mathcal{M}'}\left[\sum_{h=0}^{H-1} r_h(s_h,a_h) \;\middle|\; \tau_\pi \cap \Delta = \varnothing\right].
\end{aligned}$$

Since $\mathbb{P}$ and $\mathbb{P}'$ agree on $(\mathcal{S} \times \mathcal{A} \times [H]) \setminus \Delta$, the dynamics of $\mathcal{M}$ and $\mathcal{M}'$ agree up until π performs an action in Δ, and thus

$$P_{\mathcal{M}}\left(\tau_\pi \cap \Delta \neq \varnothing\right) = P_{\mathcal{M}'}\left(\tau_\pi \cap \Delta \neq \varnothing\right)$$
$$\mathbb{E}_{\mathcal{M}}\left[\sum_{h=0}^{H-1} r_h(s_h, a_h) \,\middle|\, \tau_\pi \cap \Delta = \varnothing\right] = \mathbb{E}_{\mathcal{M}'}\left[\sum_{h=0}^{H-1} r_h(s_h, a_h) \,\middle|\, \tau_\pi \cap \Delta = \varnothing\right]$$

Furthermore,

$$(1-q)\mathbb{E}\left[\sum_{h=0}^{h-1} r_h(s_h, a_h) \,\middle|\, \tau_\pi \cap \delta = \varnothing\right] \leq V_0^{\mathcal{M},\pi}(s_0) \leq V_0^{\mathcal{M},*}(s_0).$$

Putting everything together,

$$V_0^{\mathcal{M}',\pi}(s_0) \leq qH + V_0^{\mathcal{M},*}(s_0) \implies q > \frac{\rho}{H}.$$ □

B META-TEST PROOFS

We now provide an analysis of the regret incurred by a learner using an approximately learned hierarchy at meta-test time. We first show that the hierarchy oracle from the source tasks can provide useful temporally extended behavior. We then show that using these policies results in bounded suboptimality and achieves a better regret bound compared to standard UCB-VI.

Throughout this section, we fix an optimal π^* satisfying the conditions of Assumption 5.1. Furthermore, we assume that we have access to a hierarchy oracle that provides ε-suboptimal policies as defined in Definition 4.3.

B.1 Using the Hierarchy Oracle

In this section, we show that the hierarchy oracle can be used to implement two useful behaviors: (1) reaching exits and (2) behaving optimally within a cluster.

B.1.1 Near-Optimal Goal Reaching

Assume that the agent is currently at a state $z \in \{s_0\} \cup \mathrm{Ent}(\mathcal{S})$ at time step h, and intends to exit the current cluster Z via exit $g = (s^*, a^*) \in \mathrm{Ext}(Z)$. We obtain a policy implementing the high-level intent as follows:

(1) Define the termination for any $(s, a) \in \mathrm{Ext}(\mathcal{C})$ as:

$$f_g(s,a) \coloneqq \begin{cases} \texttt{GOAL} & (s,a) = g \\ \texttt{FAIL} & \text{otherwise} \end{cases}$$

(2) Define reward as $r_{\texttt{GOAL}}(s,a) \coloneqq \mathbb{1}\left[s = \texttt{GOAL}\right]$

(3) Provide $(z, f_g, r_{\texttt{GOAL}}, H-h)$ to the hierarchy oracle and obtain a policy $\pi_{z,g,h}$.

For simplicity, we will write $T^{\mathrm{hier}}_{H-h}(z,g)$ for $T^{\pi_{z,g,h}}_{H-h}(z,g)$ throughout our analysis. The following proposition quantifies the performance of the obtained policy:

Proposition B.1. *T^{hier} satisfies the following inequality:*

$$\mathbb{E}\left[T^{\mathrm{hier}}_{H-h}(z,g)\right] \le T^*_{H-h}(z,g) + \varepsilon.$$

Proof. Due to the definition of $\mathbb{P}_{f_g}$ and r, observe that for any π,

$$V^\pi_0(z) = \mathbb{E}\left[\sum_{h=0}^{H-h} r(s_h, a_h) \;\middle|\; s_0 = z\right] = (H-h) - \mathbb{E}\left[T^\pi_{H-h}(z,g)\right].$$

Therefore,

$$\begin{aligned} (H-h) - T^*_{H-h}(z,g) - \varepsilon &\le (H-h) - \mathbb{E}\left[T^{\mathrm{hier}}_{H-h}(z,g)\right] \\ \implies \mathbb{E}\left[T^{\mathrm{hier}}_{H-h}(z,g)\right] &\le T^*_{H-h}(z,e) + \varepsilon. \end{aligned}$$

□

B.1.2 Near-Optimal Within-Cluster Behavior

Assume that the agent is currently at a state $z \in \{s_0\} \cup \mathrm{Ent}(\mathcal{S})$ at time h, and intends to remain in the current cluster Z while maximizing a given reward function r. We obtain a policy for this high-level intent as follows:

(1) Define transition dynamics for any $(s,a) \in \mathrm{Ext}(Z)$ as $\mathbb{P}(\cdot \mid s,a) = \delta(\texttt{FAIL})$.

(2) Provide $\mathbb{P}$, r, and planning horizon $H-h$ to the hierarchy oracle, and obtain a policy π.

B.2 Formal Learning Procedure

In this section, we describe the procedure for learning a policy using the oracle-provided policies described in the previous section. Formally, we construct a surrogate MDP whose dynamics are determined by $\mathcal{M}$ and the oracle. We can then apply any tabular learning method to this new MDP (in our case, EULER), obtaining a policy in the surrogate MDP that readily translates into a policy in $\mathcal{M}$.

The components defining the surrogate $\mathcal{M}_{\mathrm{hl}} = (\mathcal{Z}, \mathcal{G}, \mathbb{P}_{\mathrm{hl}}, R_{\mathrm{hl}}, H_{\mathrm{eff}})$ are as follows:

Meta-state space $\mathcal{Z}$. We set

$$\mathcal{Z} \coloneqq \left(\mathrm{Ent}(\mathcal{S}) \times \left\{0, \dots, \bar{H}+1\right\}\right) \cup \{\ominus\},$$

where $\bar{H}$ is a high-probability bound on the time to move through H_{eff} exits (to be determined later). We incorporate the time step into the meta-state to ensure that both the dynamics and reward are computable from the state information (ensuring that $\mathcal{M}_{\mathrm{hl}}$ is indeed an MDP).

Meta-action space $\mathcal{G}$. Given a current meta-state (s, h) where $s \in Z$, the available meta-actions $\mathcal{G}$ can be identified with $\mathrm{Ext}(Z) \cup \{\ominus\}$.

Algorithm 12 Performing a Meta-Transition

Require: $(z, g) \in \mathcal{Z} \times \mathcal{G}$
{Executes the desired meta-transition in the original MDP $\mathcal{M}$.}
if $z = \ominus$ **then**
  Return $\ominus$
else if $z = (s, h)$ **then**
  if $h \le \bar{H}$ **then**
    if $s \in Z^*$ or $g = \ominus$ **then**
      Execute within-cluster policy from oracle until termination.
      Return $\ominus$
    else
      Execute $\pi_{z,g,h}$ obtained from oracle until g is performed or $h = \bar{H}$.
      $s', h' \leftarrow$ current state and time step
      if g was performed **then**
        Return (s', h')
      else
        Return $(s, \bar{H}+1)$
      end if
    end if
  else
    if $s \in Z^*$ or $g = \ominus$ **then**
      Return $\ominus$
    else
      Return (s, h)
    end if
  end if
end if

Meta-dynamics $\mathbb{P}_{\mathrm{hl}}$. Fix $(z, g) \in \mathcal{Z} \times \mathcal{G}$ for some $z \neq \ominus$, so that $z = (s, h)$. We consider the procedure in Algorithm 12 for generating the meta-dynamics. Intuitively, we execute a meta-action $g \neq \ominus$ by running the oracle-provided policy until the learner encounters g, or has acted for $\bar{H}$ timesteps in the current episode. On the other hand, if $g = \ominus$, the agent executes the oracle-provided ε-suboptimal policy that remains within the current cluster and acts for $H - h$ timesteps.

Formally, the next state z' is given by

$$z' = \begin{cases} \ominus & s \in Z^* \text{ or } g = \ominus \\ (s', h') & h \le \bar{H} \\ (s, h) & \text{otherwise} \end{cases},$$

where s' and h' are generated given $T^{\text{hier}}_{H-h}(s, g)$ as

$$h' \mid T^{\text{hier}}_{H-h}(s,g) = \min(h + T^{\text{hier}}_{H-h}(s,g), \bar{H}+1)$$
$$s' \mid h' \sim \begin{cases} \mathbb{P}(\cdot \mid g) & h' \le \bar{H} \\ \delta(s) & \text{otherwise} \end{cases}.$$

Note that the learner can only execute meta-actions while $h \le \bar{H}$. Furthermore, given access to $\mathcal{M}$, one can easily simulate the dynamics of $\mathcal{M}_{\text{hl}}$.

Meta-reward R_{hl}. Fix $((s,h),g) \in \mathcal{Z} \times \mathcal{G}$. Recall that the reward function of $\mathcal{M}$ is supported on $\text{Ext}(\mathcal{C}) \cup (Z^*)^\circ$. Thus, this reward function can be lifted onto $\mathcal{M}_{\text{hl}}$. Formally, we define the following reward function:

$$R_{\text{hl}}(z,g) = \begin{cases} W_h(s) & z = (s,h),\, s \in Z^* \text{ and } h \le \bar{H} \\ r(g) & z = (s,h),\, s \notin Z^* \text{ and } h' \le \bar{H} \\ 0 & \text{otherwise} \end{cases},$$

where $W_h(s)$ is the random sum of rewards obtained by playing a within-cluster policy starting from s' for the rest of the episode. Note that R_{hl} depends on $\mathbb{P}_{\text{hl}}$ and is thus random. Furthermore, this reward function is consistent with how meta-transitions are performed in Algorithm 12.

Meta-horizon H_{eff}. Recall that there exists an optimal policy that encounters at most H_{eff} exits with high probability. Accordingly, we limit the learner to being able to choose H_{eff} high-level actions, which recall can be choices of exits.

Solving $\mathcal{M}_{\text{hl}}$. To obtain the desired policy, we apply EULER to $\mathcal{M}_{\text{hl}}$. By the construction in Algorithm 12, the policy set returned by EULER easily translates into policies on $\mathcal{M}$. Furthermore, the value of this policy is the same on both MDPs.

B.3 Proving the Regret Bound

Having defined the procedure for learning a policy using the hierarchy, we now proceed with the regret analysis. Our analysis proceeds by constructing a policy expressible in $\mathcal{M}_{\text{hl}}$ that achieves near-optimal returns by imitating the high-level decisions made by π^*. We then use this policy as a comparator policy when applying EULER regret bounds to $\mathcal{M}_{\text{hl}}$.

To formally construct the desired comparator policy, we need to first define the notion of a meta-history, which contains the set of high-level decisions made by any policy:

Definition B.1. Fix a policy π, which given some horizon L, generates a (random) trajectory $(s_0, a_0, \ldots, s_L)$. Let $\text{Ext}(\pi)$ be the number of exits performed in the trajectory, i.e.

$$\text{Ext}(\pi) = \sum_{h=0}^{L-1} \mathbb{1}\left[(s_h, a_h) \in \text{Ext}(\mathcal{C})\right].$$

The meta-history $\mathcal{H}_{\text{hl}}(\pi)$ corresponding to this trajectory is the sequence

$$(z_0, g_0, z_1, g_1, \ldots, z_{\text{Ext}(\pi)}) = (s_{i_0}, (s_{j_0}, a_{j_0}), s_{i_1}, (s_{j_1}, a_{j_1}) \ldots, s_{i_{\text{Ext}(\pi)}}),$$

where

$$i_n \coloneqq \begin{cases} 0 & n = 0 \\ j_{n-1} + 1 & \text{otherwise} \end{cases}$$
$$j_n \coloneqq \min_{h = i_n, \ldots, L-1} \mathbb{1}\left[(s_h, a_h) \in \text{Ext}(\mathcal{C})\right].$$

Note that $z_i \in \text{Ent}(\mathcal{S})$ and $g_i \in \text{Ext}(\mathcal{C})$ for all $i = 0, \ldots, \text{Ext}(\pi)$. We omit π in writing $\mathcal{H}_{\text{hl}}$ when the underlying policy π is understood. $\diamond$

Informally, $\mathcal{H}_{\text{hl}}$ tracks all entrances and exits contained in a trajectory generated by π. We define the length of a meta-history $\mathcal{H}_{\text{hl}}$, denoted as $|\mathcal{H}_{\text{hl}}|$, as the number of exits contained in $\mathcal{H}_{\text{hl}}$.

B.3.1 Policy Construction

We now proceed with constructing the desired policy. Intuitively, the comparator imitates the distribution over $\mathcal{H}_{\mathrm{hl}}(\pi^*)$, conditioned on $|\mathcal{H}_{\mathrm{hl}}(\pi^*)| \leq H_{\mathrm{eff}}$. To see why this is sufficient for near-optimality, recall that the reward on $\mathcal{M}_{\mathrm{Tg}}$ is supported on $\mathrm{Ext}(Z^*) \cup (Z^*)^\circ$. Consequently, by imitating the distribution over meta-histories, the policy is expected to obtain roughly the same sum of rewards in expectation from the exits. Therefore, all that remains is to ensure that the learner collects roughly the same sum of rewards from Z^*, which is the same as ensuring that this policy does not take too long to reach Z^*.

Construction. Let $\mathcal{H}$ be the running meta-history, containing $k \leq H_{\mathrm{eff}}$ actions. The optimal policy induces a distribution $q(\cdot \mid \mathcal{H})$ over $\mathcal{A}_{\mathrm{hl}}$ representing the next exit it takes[9]. We then define π as

$$\pi(\cdot \mid z, \mathcal{H}) = \begin{cases} q(\cdot \mid \mathcal{H}) & z = (s,h), h < \bar{H} \\ \ominus & \text{otherwise.} \end{cases}$$

Observe that π terminates the episode upon reaching $\bar{H}$. Furthermore, this policy is dependent on the meta-history. However, since $\mathcal{M}_{\mathrm{hl}}$ is an MDP, there exists a stationary policy that achieves at least the same value.

B.3.2 Suboptimality Analysis

In this section, we prove that π achieves bounded suboptimality. Rather than analyzing π directly in $\mathcal{M}_{\mathrm{hl}}$, we construct a new $\tilde{\mathcal{M}}_{\mathrm{hl}}$ and $\tilde{\pi}$ to better track the meta-history. In particular, conditioned on the event that π requires more than $\bar{H}$ time steps to execute, then the agent would not be able to imitate the full meta-history generated by π^*, even after having performed less than H_{eff} exits.

Constructing a surrogate for analysis. We now formalize the construction of the surrogate MDP $\tilde{\mathcal{M}}_{\mathrm{hl}}$ and the policy $\tilde{\pi}$ corresponding to π in this MDP. To obtain $\tilde{\mathcal{M}}_{\mathrm{hl}}$, we redefine the dynamics from $\mathcal{M}_{\mathrm{hl}}$ so that $s' \mid h' \sim \mathbb{P}(\cdot \mid g)$ in $\tilde{\mathcal{M}}_{\mathrm{hl}}$. In effect, we allow the policy to continue performing transitions beyond $\bar{H}$, although without any reward. Accordingly, we define $\tilde{\pi}$ as $\tilde{\pi}(\cdot \mid z, \mathcal{H}) = q(\cdot \mid \mathcal{H})$. The following lemma formalizes how $\tilde{\pi}$ and $\tilde{\mathcal{M}}_{\mathrm{hl}}$ have desirable properties for the analysis:

Lemma B.1 (Surrogate Policy Characterization)**.** *Let μ^* denote the distribution of $\mathcal{H}_{\mathrm{hl}}(\pi^*) \mid |\mathcal{H}_{\mathrm{hl}}(\pi^*)| \leq H_{\mathrm{eff}}$ in $\mathcal{M}_{\mathrm{Tg}}$, and $\tilde{\mu}$ the distribution of $\mathcal{H}(\tilde{\pi})$ in $\tilde{\mathcal{M}}_{\mathrm{hl}}$. Then, $(1-\zeta)\mu^* \leq \tilde{\mu}$.*

Proof. Let ν^* be the distribution induced by the following procedure:

(1) Sample a meta-history from the distribution $\mathcal{H}_{\mathrm{hl}}(\pi^*) \mid |\mathcal{H}_{\mathrm{hl}}(\pi^*)| > H_{\mathrm{eff}}$.

(2) Truncate the obtained meta-history to length H_{eff}.

It is easy to see from the definition of $\tilde{\pi}$ that $\tilde{\mu} = (1-\zeta)\mu^* + \zeta\nu^*$. The desired result follows. □

Thus, we have indeed shown the desired property that $\tilde{\pi}$ properly tracks the (truncated) meta-history generated by π^*. To justify performing our analysis on $(\tilde{\mathcal{M}}_{\mathrm{hl}}, \tilde{\pi})$, we have the following result, which shows that any result on the value of the pair above applies to the value of π in $\mathcal{M}_{\mathrm{hl}}$.

Lemma B.2. *As constructed above, $V_0^{\tilde{\mathcal{M}}_{\mathrm{hl}},\tilde{\pi}}(s_0) = V_0^{\mathcal{M}_{\mathrm{hl}},\pi}(s_0)$.*

Proof. We write $\mathcal{M} \coloneqq \mathcal{M}_{\mathrm{hl}}$ and $\mathcal{M}' \coloneqq \tilde{\mathcal{M}}_{\mathrm{hl}}$. Similarly, we write $\pi' \coloneqq \tilde{\pi}$. We proceed by proving a chain of equalities. ($V^{\pi',\mathcal{M}'}(s_0) = V^{\pi,\mathcal{M}'}(s_0)$). We omit $\mathcal{M}'$ in this part of the argument for clarity. By the performance difference lemma, we have that for any $k \in [H_{\mathrm{eff}}]$ and $z \in \mathcal{S}_{\mathrm{hl}}$,

$$V_0^{\pi}(s_0) - V_0^{\pi'}(s_0) = \sum_{j=0}^{H_{\mathrm{eff}}-1} \mathbb{E}_{z \sim d_j^{\pi}} \left[A_j^{\pi'}(z, \pi) \right].$$

[9]The distribution q can return $\ominus$ if the learner stays in the cluster until episode termination.

Let $\Delta \coloneqq \left\{ z \in \mathcal{S}_{\mathrm{hl}} \;\middle|\; z = (s,h), s \notin Z^*, h \geq \bar{H} \right\}$, which is the set on which π and π' disagree. Observe that for any π and k, $V_k^\pi(z) = 0$ for any $z \in \Delta$, and thus $A_k^{\pi'}(z,\pi) = 0$ for all such states. For any other z, $A_k^{\pi'}(z,\pi)$ is clearly 0. Thus, we obtain the desired result.

($V^{\pi,\mathcal{M}'}(s_0) = V^{\pi,\mathcal{M}}(s_0)$) We omit π in this part of the argument for clarity. Using the simulation lemma,

$$V_0^{\mathcal{M}}(s_0) - V_0^{\mathcal{M}'}(s_0) = \sum_{j=0}^{H_{\mathrm{eff}}-1} \mathbb{E}_{(z,g)\sim d_j^{\mathcal{M}'}} \left[[(\mathbb{P}_{\mathcal{M}} - \mathbb{P}_{\mathcal{M}'}) V_{j+1}^{\mathcal{M}}](z,g) \right].$$

Observe that the behavior of the two MDPs are identical conditioned on $h' \leq \bar{H}$. On the other hand, conditioned on $h' > \bar{H}$, π can no longer receive rewards from either MDP. Therefore, $[(\mathbb{P}_{\mathcal{M}} - \mathbb{P}_{\mathcal{M}'}) V_j^{\mathcal{M}}](z,g) = 0$ for any j, z, g by decomposing the relevant expectations along the two events. We thus obtain the desired result. □

Analyzing the surrogate. With the results above, we now proceed to analyze the difference in values

$$V_0^{\mathcal{M}_{\mathrm{Tg}},*}(s_0) - V_0^{\tilde{\mathcal{M}}_{\mathrm{hl}},\tilde{\pi}}(s_0),$$

which then implies the desired suboptimality result. First, we have the following lemma characterizing the time $\tilde{\pi}$ requires to fully execute a given meta-history in the base MDP $\mathcal{M}$:

Lemma B.3. *Fix any $\mathcal{H}_{\mathrm{hl}} = (z_0, g_0, \dots)$ such that $|\mathcal{H}_{\mathrm{hl}}| \leq H_{\mathrm{eff}}$. Furthermore, define the sequence of reaching times*

$$T_0 \coloneqq 0 \quad \textit{and} \quad T_k \coloneqq T_{k-1} + T^{\mathrm{hier}}_{H-T_{k-1}}(z_{k-1}, u_{k-1}).$$

We define $T^{\mathrm{hier}}(\mathcal{H}_{\mathrm{hl}})$ to be the time required by the hierarchy to execute $\mathcal{H}_{\mathrm{hl}}$, which is formally given by $T_{|\mathcal{H}_{\mathrm{hl}}|}$ in the sequence above. Then,

(a) $\mathbb{E}\left[T^{\mathrm{hier}}(\mathcal{H}_{\mathrm{hl}})\right] \leq [1 + (1+\gamma)W + \varepsilon] H_{\mathrm{eff}}$.

(b) *Let $\sigma^2 \coloneqq \kappa^2 [(1+\gamma)W + \varepsilon]^2 H_{\mathrm{eff}}$. Then, for any $t > 0$,*

$$P\left(T^{\mathrm{hier}}(\mathcal{H}_{\mathrm{hl}}) \geq [1 + (1+\gamma)W + \varepsilon] H_{\mathrm{eff}} + t\right) \leq e^{-t^2/2\sigma^2}.$$

Proof. We prove the two parts separately:

(a) We will prove via induction that $\mathbb{E}[T_k] \leq k[1 + (1+\gamma)W + \varepsilon]$. For any k and T_{k-1},

$$\begin{aligned}
\mathbb{E}\left[T^{\mathrm{hier}}_{H-T_{k-1}}(z_{k-1}, g_{k-1}) \;\middle|\; T_{k-1} \right] &\leq \mathbb{E}\left[T^{*}_{H-T_{k-1}}(z_{k-1}, g_{k-1}) \;\middle|\; T_{k-1} \right] + \varepsilon \\
&= 1 + \mathbb{E}\left[T^{*}_{H-T_{k-1}}(z_{k-1}, s(g_{k-1})) \;\middle|\; T_{k-1} \right] + \varepsilon \\
&\leq 1 + (1+\gamma)W + \varepsilon,
\end{aligned}$$

where the first inequality uses properties of the hierarchy oracle, while the final inequality follows by combining Definition 5.2(b) and Assumption 5.2. Therefore, by linearity and the tower property of expectation,

$$\begin{aligned}
\mathbb{E}[T_k] &= \mathbb{E}[T_{k-1}] + \mathbb{E}\left[T^{\mathrm{hier}}_{H-T_{k-1}}(z_{k-1}, g_{k-1}) \right] \\
&= \mathbb{E}[T_{k-1}] + \mathbb{E}\left[\mathbb{E}\left[T^{\mathrm{hier}}_{H-T_{k-1}}(z_{k-1}, g_{k-1}) \;\middle|\; T_{k-1} \right] \right] \\
&\leq \mathbb{E}[T_{k-1}] + 1 + (1+\gamma)W + \varepsilon.
\end{aligned}$$

The desired result then follows by induction.

(b) Let $B_k \coloneqq k[1 + (1+\gamma)W + \varepsilon]$ and $f_k(t) \coloneqq \mathbb{E}\left[T^{\mathrm{hier}}_{H-t}(z_k, g_k)\right]$. Note that for any k and t, $B_{k-1} + f_{k-1}(t) \leq B_k$, by following the argument in (a). Therefore, for any $\lambda > 0$,

$$\begin{aligned}
&\mathbb{E}\left[\exp\left\{\lambda\left(T_k - B_k\right)\right\}\right] \\
&\quad = \mathbb{E}\left[\mathbb{E}\left[\exp\left\{\lambda\left(T_k - B_k\right)\right\} \mid T_{k-1}\right]\right] \\
&\quad \leq \mathbb{E}\left[\mathbb{E}\left[\exp\left\{\lambda\left(T_{k-1} + T^{\mathrm{hier}}_{H-T_{k-1}}(z_{k-1}, g_{k-1}) - B_{k-1} - f_{k-1}(T_{k-1})\right)\right\} \;\middle|\; T_{k-1}\right]\right],
\end{aligned}$$

where the last inequality uses the monotonicity of the exponential function. Therefore, by applying the sub-Gaussian condition given in Definition 5.2,

$$\begin{aligned}
&\mathbb{E}\left[\exp\left\{\lambda\left(T_k - B_k\right)\right\}\right] \\
&\quad\leq \mathbb{E}\left[\exp\left\{\lambda\left(T_{k-1} - B_{k-1}\right)\right\}\right. \\
&\qquad\qquad \left.\mathbb{E}\left[\exp\left\{\lambda\left(T^{\text{hier}}_{H-T_{k-1}}(z_{k-1}, g_{k-1}) - f_{k-1}(T_{k-1})\right)\right\}\,\middle|\, T_{k-1}\right]\right] \\
&\quad\leq \mathbb{E}\left[\exp\left\{\lambda(T_{k-1} - B_{k-1})\right\}\right]\exp\left[\lambda^2 C^2/2\right],
\end{aligned}$$

where we have used the fact that $T^{\text{hier}}_{H-T_{k-1}}(z_{k-1}, s(g_{k-1}))$ has a sub-Gaussian upper tail with variance proxy

$$\begin{aligned}
C^2 &= \kappa^2 \mathbb{E}\left[T^{\pi}_{H-T_{k-1}}(z_{k-1}, s(g_{k-1}))\,\middle|\, T_{k-1}\right]^2 \\
&\leq \kappa^2\left[(1+\gamma)W + \varepsilon\right]^2.
\end{aligned}$$

Note that we have once again used the properties of the hierarchy oracle, and Assumption 5.2. Therefore, by induction, $\mathbb{E}\left[\exp\left\{\lambda\left(T_k - B_k\right)\right\}\right] \leq \mathbb{E}\left[\lambda^2(\sqrt{k}C)^2/2\right]$, from which the desired tail bound follows by making use of Chernoff's inequality. □

As we have shown that $\tilde{\pi}$ closely tracks the meta-history of π^* and have analyzed the distribution of time it takes to execute a given meta-history, we can now analyze its suboptimality:

Lemma B.4. *There exists a policy π expressible in $\mathcal{M}_{\text{hl}}$ such that*

$$V_0^{\mathcal{M}_{\text{Tg}},*}(s_0) - V_0^{\mathcal{M}_{\text{Tg}},\pi}(s_0) \lesssim (1 + H_{\text{eff}} + \kappa\sqrt{H_{\text{eff}}})\varepsilon + \left[\gamma H_{\text{eff}} + \kappa(1+\gamma)\sqrt{H_{\text{eff}}}\right]W + \zeta H.$$

Proof. Assume that π^* generates a (random) meta-history of length N given by $\mathcal{H}_{\text{hl}} = (z_0, g_0, z_1, g_1, \ldots, z_N)$. Furthermore, let T^* denote the (random) time π^* takes to reach z_N. Then, given $\mathcal{H}_{\text{hl}}$ and T^*, observe that we can write

$$V_0^*(s_0) = \mathbb{E}\left[R^*_{T^*}(\mathcal{H}_{\text{hl}})\right], \quad \text{where } R^*_T(\mathcal{H}_{\text{hl}}) \coloneqq V^*_T(z_N)\mathbb{1}\left[z_N \in Z^*\right] + \sum_{k=0}^{N-1} r(g_k),$$

using the assumptions on the reward function and condition (a) in Assumption 5.1. Subsequently, letting E be the event $\{N \leq H_{\text{eff}}\}$, we can bound the right-hand side as

$$V_0^*(s_0) = \mathbb{E}\left[R^*_{T^*}(\mathcal{H}_{\text{hl}})\right] \leq (1-\zeta)\mathbb{E}\left[R^*_{T^*}(\mathcal{H}_{\text{hl}}) \mid E\right] + \zeta H,$$

where we have used Assumption 5.1 to bound the probability that $N > H_{\text{eff}}$.

Our goal for the rest of this proof is to transform the expectation on the right-hand side into a form that lower bounds $V_0^{\pi}(s_0)$. To this end, we define

$$R^{\text{hier}}_T(\mathcal{H}_{\text{hl}}) \coloneqq V^{\text{hier}}_T(z_N)\mathbb{1}\left[z_N \in Z^*\right] + \sum_{k=0}^{N-1} r(g_k),$$

and the sequence of times

$$T_0 = 0 \quad \text{and} \quad T_k \coloneqq T_{k-1} + T^{\text{hier}}_{H-T_{k-1}}(z_k, g_k).$$

Note that R^{hier} and T_N are analogous to R^* and T^*, respectively. Then, letting F be the event $\left\{T_N \leq \bar{H}\right\}$, note that

$$\begin{aligned}
V_0^*(s_0) &\leq (1-\zeta)\mathbb{E}\left[R^*_{T^*} \mid E\right] + \zeta H \\
&= (1-\zeta)\mathbb{E}\left[R^*_{T^*} - R^{\text{hier}}_{T_N} + R^{\text{hier}}_{T_N} \,\middle|\, E\right] + \zeta H \\
&\leq \underbrace{\mathbb{E}\left[\left(R^*_{T^*} - R^{\text{hier}}_{T_N}\right)\mathbb{1}\left[F\right] \,\middle|\, E\right]}_{\text{(I)}} + \underbrace{(1-\zeta)\mathbb{E}\left[R^{\text{hier}}_{T_N}\mathbb{1}\left[F\right] \,\middle|\, E\right]}_{\text{(II)}} + \left[\zeta + P\left(F^C \mid E\right)\right]H.
\end{aligned}$$

We bound (I) and (II) separately.

Bounding (I). Let G be the event $E \cap \{z_N \in Z^*\}$. Then, if we define

$$T_{\min} = \sum_{k=0}^{N-1} T_{\min}(z_k, u_k) \le N(W+1)$$

as the minimum time needed to execute $\mathcal{H}_{\text{hl}}$, we then have that

$$\begin{aligned}
\mathbb{E}\left[\left(R^*_{T^*} - R^{\text{hier}}_{T_N}\right)\mathbb{1}\left[F\right] \,\middle|\, E\right] &\le \mathbb{E}\left[R^*_{T^*} - R^{\text{hier}}_{T_N} \mid E\right] \\
&\le \mathbb{E}\left[V^*_{T^*}(z_N) - V^{\text{hier}}_{T_N}(z_n) \,\middle|\, G\right] \\
&\le \mathbb{E}\left[V^*_{T_{\min}}(z_N) - V^{\text{hier}}_{T_N}(z_N) \,\middle|\, G\right] \\
&\le \int_0^H P\left(V^*_{T_{\min}}(z_N) - V^{\text{hier}}_{T_N}(z_N) > \alpha \,\middle|\, G\right)\, \mathrm{d}\alpha.
\end{aligned}$$

Note that the bound on $T_{\min}$ follows from Assumption 5.2. To convert the different in values into a difference of times, observe that if $T_N - T_{\min} \le \alpha - \varepsilon$, then

$$\begin{aligned}
V^*_{T_{\min}}(z_N) - V^{\text{hier}}_{T_N}(z_N) &= V^*_{T_{\min}}(z_N) - V^*_{T_N}(z_N) + V^*_{T_N}(z_N) - V^{\text{hier}}_{T_N}(z_N) \\
&\le (\bar{T} - T_{\min}) + \varepsilon \\
&\le \alpha.
\end{aligned}$$

Therefore,

$$\begin{aligned}
&\int_0^H P\left(V^*_{T_{\min}}(z_N) - V^{\text{hier}}_{T_N}(z_N) > \alpha \,\middle|\, G\right)\, \mathrm{d}\alpha \\
&\le \int_0^H P(T_N - T_{\min} > \alpha - \varepsilon \mid G)\, \mathrm{d}\alpha \\
&\le \mathbb{E}\left[\int_0^H P(T_N - T_{\min} > \alpha - \varepsilon \mid \mathcal{H}_{\text{hl}})\, \mathrm{d}\alpha \,\middle|\, G\right] \\
&\le \mathbb{E}\left[\int_0^H P(T_N - [1 + (1+\gamma)W + \varepsilon]H_{\text{eff}} > \alpha - \varepsilon - H_{\text{eff}}(\gamma W + \varepsilon) \mid \mathcal{H}_{\text{hl}})\, \mathrm{d}\alpha \,\middle|\, G\right] \\
&\le \varepsilon + H_{\text{eff}}(\gamma W + \varepsilon) + \mathbb{E}\left[\int_0^\infty P(T_N - [1 + (1+\gamma)W + \varepsilon]H_{\text{eff}} > \alpha \mid \mathcal{H}_{\text{hl}})\, \mathrm{d}\alpha \,\middle|\, G\right] \\
&\lesssim \varepsilon + H_{\text{eff}}(\gamma W + \varepsilon) + \kappa[(1+\gamma)W + \varepsilon]\sqrt{H_{\text{eff}}},
\end{aligned}$$

where the final inequality integrates the tail bound provided in Lemma B.3. Overall, we have that by rearranging,

$$\text{(I)} \lesssim (1 + H_{\text{eff}} + \kappa\sqrt{H_{\text{eff}}})\varepsilon + \left[\gamma H_{\text{eff}} + \kappa(1+\gamma)\sqrt{H_{\text{eff}}}\right] W.$$

Bounding (II). By the characterization of $\tilde{\pi}$ in Lemma B.1,

$$(1-\zeta)\mathbb{E}\left[R^{\text{hier}}_{T_N}(\mathcal{H}_{\text{hl}}(\pi^*))\mathbb{1}\left[F\right] \,\middle|\, E\right] \le \mathbb{E}\left[R^{\text{hier}}_{T_N}(\mathcal{H}_{\text{hl}}(\tilde{\pi}))\mathbb{1}\left[F\right]\right] \le V^{\tilde{\pi}}_0(s_0),$$

where the final inequality uses the fact that $\bar{R}_{\bar{T}}(\mathcal{H}_{\text{hl}}(\tilde{\pi}))$ is the return of $\tilde{\pi}$ in $\tilde{\mathcal{M}}_{\text{hl}}$, given F.

Concluding. Putting all of the previous bounds together, we find that

$$\begin{aligned}
V^*_0(s_0) \le V^{\bar{\pi}}_0(s_0) &+ (1 + H_{\text{eff}} + \kappa\sqrt{H_{\text{eff}}})\varepsilon + \left[\gamma H_{\text{eff}} + \kappa(1+\gamma)\sqrt{H_{\text{eff}}}\right] W \\
&+ \left[\zeta + P\left(F^C \,\middle|\, E\right)\right] H.
\end{aligned}$$

By setting $\bar{H}$ to

$$\bar{H} = H_{\text{eff}}[1 + (1+\gamma)W + \varepsilon] + \kappa[(1+\gamma)W + \varepsilon]\sqrt{2H_{\text{eff}}\log\frac{1}{\zeta}} \ll H,$$

sub-Gaussian tail bounds on T_N implies that $P\left(F^C \mid E\right) \leq \zeta$. Finally, by Lemma B.2,

$$V_0^{\tilde{\mathcal{M}}_{\mathrm{hl}},\tilde{\pi}}(s_0) = V_0^{\mathcal{M}_{\mathrm{hl}},\pi}(s_0) = V_0^{\mathcal{M}_{\mathrm{Tg}},\pi},$$

where the last equality follows by the construction of $\mathcal{M}_{\mathrm{hl}}$. We thus obtain the desired suboptimality bound. □

B.3.3 Regret Analysis

As earlier suggested, we now make use of $\tilde{\pi}$ as a comparator policy in order to prove a regret bound on a learner making use of the procedure outlined in Section B.2.

Theorem B.1. *Assume that* EULER *generates policies $\pi_1, \ldots, \pi_N$ on $\mathcal{M}_{\mathrm{hl}}$, as constructed in Section B.2. Then, we have the following regret bound:*

$$\sum_{k=1}^{N} V_0^*(s_0) - V_0^{\pi_k}(s_0) \lesssim \sqrt{H^2 \bar{H} LMN} + N\varepsilon_{\mathrm{subopt}},$$

where

$$\varepsilon_{\mathrm{subopt}} \coloneqq (1 + H_{\mathrm{eff}} + \kappa\sqrt{H_{\mathrm{eff}}})\varepsilon + \left[\gamma H_{\mathrm{eff}} + \kappa(1+\gamma)\sqrt{H_{\mathrm{eff}}}\right] W + \zeta H.$$

Proof. Throughout the proof, we consider applying EULER to $\mathcal{M}_{\mathrm{hl}}$ where the rewards are scaled by $1/H$ to ensure that rewards are bounded in $[0,1]$. As a result, we can bound $\mathcal{G} \leq 1$ in the EULER regret bound in Zanette and Brunskill (2019), since the sum of rewards in $\mathcal{M}_{\mathrm{hl}}$ is also the sum of rewards in $\mathcal{M}$, and scaling by $1/H$ gives the desired bound on $\mathcal{G}$. Therefore,

$$\sum_{k=1}^{N} V_0^{*,\mathcal{M}_{\mathrm{hl}}}(s_0) - V_0^{\pi_k}(s_0) \lesssim H\sqrt{\frac{1}{H_{\mathrm{eff}}} \bar{H} LMH_{\mathrm{eff}} N} = \sqrt{H^2 \bar{H} LMN}.$$

Furthermore,

$$V_0^*(s_0) - V_0^{*,\mathcal{M}_{\mathrm{hl}}}(s_0) \leq V_0^*(s_0) - V_0^{\pi}(s_0) + V_0^{\pi,\mathcal{M}_{\mathrm{hl}}}(s_0) - V_0^{*,\mathcal{M}_{\mathrm{hl}}}(s_0) \leq \varepsilon_{\mathrm{subopt}}.$$

We thus obtain the desired result. □

B.4 An Exponential Regret Separation for a Hierarchy-Oblivious Learner

In this section, we provide proof of the exponential regret separation between a hierarchical learner and a learner oblivious to the hierarchy. The overall idea behind our proof is the reduction of solving the family of minimax instances described in Domingues et al. (2021) to a particular family of task distributions.

B.4.1 The Hard Task Distribution Family

In this section, we describe the family of task distributions that forces any meta-training-oblivious learner to incur exponential regret. For any string s, we write $|s|$ for its length.

We now define the family of binary tree room MDPs $\mathbb{M}_W$ of depth W. We index a member of this family by a tuple (ℓ^*, a^*, e^*), where ℓ^* is a binary string of length $W-1$, and $a^*, e^* \in \{0,1\}$. The MDP $\mathcal{M}_{(\ell^*,a^*,e^*)} = (\mathcal{S}, \mathcal{A}, \mathbb{P}_{(\ell^*,a^*,e^*)}, r, H)$ corresponding to this tuple is constructed as follows:

State Space $\mathcal{S}$. We create a root state s_{root}, $2^W - 1$ states indexed by binary strings of length at most $W-1$ collected into a set $T = \{s_0, s_1, s_{00}, s_{01}, \ldots\}$, a gate state s_{gate}, and terminal states $\ominus_{\mathrm{trap}}$, `GOAL`, `FAIL`.

Action Space $\mathcal{A}$. The set of available actions at every state is the set $\{0,1\}$.

Transition Dynamics $\mathbb{P}_{(\ell^*,a^*,e^*)}$. We define the dynamics as follows:

$$\mathbb{P}_{(\ell^*,a^*,e^*)}(\cdot \mid s,a) = \begin{cases} \delta(s_a) & s = s_{\text{root}} \\ \delta(s_{ta}) & s = s_t \in T, |t| < W-1 \\ b\delta(s_{\text{gate}}) + (1-b)\delta(\ominus_{\text{trap}}) & s = s_t \in T, |t| = W-1, \\ & s \neq s_{\ell^*}, b \sim \text{Ber}\,(1/2) \\ b\delta(s_{\text{gate}}) + (1-b)\delta(\ominus_{\text{trap}}) & s = s_{\ell^*}, b \sim \text{Ber}\,(1/2 + \varepsilon \mathbb{1}\,[a = a^*]) \\ \delta(\texttt{GOAL}) & s = s_{\text{gate}}, a = e^* \\ \delta(\texttt{FAIL}) & s = s_{\text{gate}}, a \neq e^*. \end{cases}$$

Reward Function r. The reward function is $r(s,a) = \mathbb{1}\,[s = \texttt{GOAL}] + \mathbb{1}\,[s = s_{\text{gate}}, a = a^*]$.

Having described all the components of every member of $\mathbb{M}_W$, all that remains is to construct the family of task distributions $\mathbb{T}_W$. Each member of this family will be indexed by (ℓ^*, a^*), where ℓ^* and a^* are as described above. Then, the task distribution $\mathcal{T}_{(\ell^*,a^*)} \in \mathbb{T}_W$ chooses uniformly within the set $\left\{\mathcal{M}_{(\ell^*,a^*,0)}, \mathcal{M}_{(\ell^*,a^*,1)}\right\}$. Note that this implicitly defines the latent hierarchy so that the clusters are $\{s_{\text{root}}, s_{\text{gate}}, \ominus_{\text{trap}}\} \cup T$, $\{\texttt{GOAL}\}$, and $\{\texttt{FAIL}\}$. Furthermore, the set of exits for the first cluster is $\{(s_{\text{gate}}, 0), (s_{\text{gate}}, 1)\}$.

B.4.2 A Family of Hard Instances

In this section, we describe the family of hard instances which we reduce to solving the task distribution above. Intuitively, if an algorithm incurs low regret throughout $\mathbb{M}_W$, then it must be able to quickly find a policy to reliable reach the gate state s_{gate} for any MDP in the family.

Constructing the hard instances. Accordingly, we define a new MDP family $\mathbb{N}_W$, which now is only indexed by (ℓ^*, a^*), and is constructed similarly as any member of $\mathbb{M}_W$, but ignoring states outside $\{s_{\text{root}}, s_{\text{gate}}, \ominus_{\text{trap}}\} \cup T$. Additionally, we redefine the reward function r for any member to be $r(s,a) \coloneqq \mathbb{1}\,[s = s_{\text{gate}}]$. We note that this is exactly the set of hard tasks used to prove a minimax regret bound in Domingues et al. (2021).

The lower bound. We state the lower bound result from Domingues et al. (2021), in a slightly more restricted form for ease of proof and presentation. In particular, we consider the following more restricted definition of an algorithm:

Definition B.2. Let $\mathcal{H}_n$ be the trajectory data generated by playing a policy π_n in an MDP $\mathcal{M}$. That is, $\mathcal{H}_n =$

Algorithm 13 The reduction $\mathcal{P}_{\mathcal{A}}$ of learning $\mathbb{N}_W$ to learning $\mathbb{M}_W$ in Section B.4.3.

Require: $\mathcal{M} \in \mathbb{N}_W$
Initialize $\mathcal{H}_0 = \varnothing$
for all $n \in [N]$ **do**
  Obtain $\pi_n = \mathcal{A}(\mathcal{H}_0, \ldots, \mathcal{H}_{n-1})$.
  Play π_n in $\mathcal{M}$, get history $\mathcal{G}_n = ((s_0, a_0, r_0, s_1), \ldots, (s_{H-1}, a_{H-1}, r_{H-1}, s_H))$.
  if $s_{W+1} = s_{\text{gate}}$ **then**
    $s'_{W+1} \leftarrow s_{W+1}$
    for all $h = W+1, \ldots, H-1$ **do**
      if $h = W$ **then**
        $s'_{h+1} \leftarrow \texttt{GOAL}$ if $a_h = 1$ else $\texttt{FAIL}$.
        $r'_h \leftarrow \mathbb{1}\,[a_h = 1]$
      else
        $s'_{h+1} \leftarrow s'_h$, $r'_h \leftarrow r_h$
      end if
      Replace (s_h, a_h, r_h, s_{h+1}) with $(s'_h, a_h, r'_h, s'_{h+1})$ in $\mathcal{G}_n$
    end for
  end if
  $\mathcal{H}_n \leftarrow \mathcal{G}_n$.
end for

$((s_0, a_0, r_0, s_1), (s_1, a_1, r_1, s_2), \ldots, (s_{H-1}, a_{H-1}, r_{H-1}, s_H))$, where s_0 and a_0 are fixed, $r_h = r(s_h, a_h)$, and $s_{h+1} \sim \mathbb{P}_{\mathcal{M}}(\cdot \mid s_h, a_h)$. Additionally, we set $\mathcal{H}_0 = \varnothing$.

Then, a *valid algorithm* $\mathcal{A}$ for our purposes is one which, for the n^{th} episode, outputs a deterministic, non-stationary policy π that is solely a function of the current state and $\bigcup_{i=1}^{n-1} \mathcal{H}_i$. That is, $\mathcal{A}$ does not output policies that adapt to the current running episode. $\diamond$

We again emphasize that this restriction is not necessary but that many algorithms nevertheless satisfy this condition (including UCBVI and EULER). We then have the following hardness result:

Theorem B.2 (Domingues et al. (2021), Theorem 9, restated)**.** *Assume that $W \geq 2$ and $H \geq 3W$. Then, for every algorithm $\mathcal{A}$, there exists an MDP $\mathcal{M} \in \mathbb{N}_W$ such that*

$$\mathbb{E}_{\mathcal{M},\mathcal{A}}\left[\sum_{n=1}^{N} V_0^*(s_{\text{root}}) - V_0^{\pi_n}(s_{\text{root}})\right] \gtrsim 2^{W/2}\sqrt{H^2 N}.$$

B.4.3 Proving the Hardness Result

We now use the hardness result in the previous section to demonstrate that no algorithm can incur sub-exponential regret in W on all tasks in $\mathbb{M}_W$. We do so by proving that an algorithm solving all tasks in $\mathbb{M}_W$ can be used to construct an algorithm for solving all tasks in $\mathbb{N}_W$.

Formally, let $\mathcal{A}$ be any algorithm for learning any MDP in $\mathbb{M}_W$. We then construct an algorithm $\mathcal{P}_{\mathcal{A}}$ for learning any MDP in $\mathbb{N}_W$ as in Algorithm 13.

Given this reduction, we aim to prove the following result:

Proposition B.2. *For any $\mathcal{M}_{(\ell^*,a^*)} \in \mathcal{N}_W$, we have that*

$$\mathbb{E}_{\mathcal{M}_{(\ell^*,a^*)},\mathcal{P}_{\mathcal{A}}}\left[\sum_{n=1}^{N} V_0^*(s_{\text{root}}) - V_0^{\pi_n}(s_{\text{root}})\right] \leq \mathbb{E}_{\mathcal{M}_{(\ell^*,a^*,1)},\mathcal{A}}\left[\sum_{n=1}^{N} V_0^*(s_{\text{root}}) - V_0^{\pi_n}(s_{\text{root}})\right].$$

To prove this result, we first prove that $\mathcal{P}_{\mathcal{A}}$ can simulate $\mathcal{M}_{(\ell^*,a^*,1)}$:

Lemma B.5. *For any n, the distribution over $(\mathcal{H}_0, \ldots, \mathcal{H}_n)$ induced by running Algorithm 13 over $\mathcal{M}_{(\ell^*,a^*)} \in \mathbb{N}_W$ is equal to that induced by running $\mathcal{A}$ over $\mathcal{M}_{(\ell^*,a^*,1)} \in \mathbb{M}_W$.*

Proof. We proceed by induction. The result holds trivially for $n = 0$.

Now, assume that the result holds for some n. We condition on the histories $(\mathcal{H}_0, \ldots, \mathcal{H}_n)$ Then, note that both algorithms play the same policy π_{n+1}, since $\mathcal{P}_{\mathcal{A}}$ uses $\mathcal{A}$ to obtain the next policy. As a result, by the construction of $\mathcal{M}_{(\ell^*,a^*)}$ and $\mathcal{M}_{(\ell^*,a^*,1)}$, the distribution over (s_h, a_h, r_h, s_{h+1}) are equal for $h \leq W$. Furthermore, Lines $6-14$ simulates the dynamics of $\mathcal{M}_{(\ell^*,a^*,1)}$ conditioned on $s_{W+1} = s_{\text{gate}}$, while conditioned on $s_{W+1} = \ominus_{\text{trap}}$, the dynamics of the two MDPs are the same. Therefore, conditioned on any $(\mathcal{H}_0, \ldots, \mathcal{H}_n)$, the distribution over $\mathcal{H}_{n+1}$ induced by the two algorithms are also the same. Thus, the claim holds by induction. $\square$

Finally, we can prove Proposition B.2.

Proof of Proposition B.2. Throughout this proof, we omit the starting state s_{root} and the timestep 0 in the value. We prove the result by induction. Clearly, the result holds for $N = 0$.

Assume that the bound holds for some N. Then, we have that

$$\begin{aligned}
&\mathbb{E}_{\mathcal{M}_{(\ell^*,a^*)},\mathcal{P}_{\mathcal{A}}}\left[\sum_{n=1}^{N+1} V^* - V^{\pi_n}\right] \\
&\quad= \mathbb{E}_{\mathcal{M}_{(\ell^*,a^*)},\mathcal{P}_{\mathcal{A}}}\left[\sum_{n=1}^{N} V^* - V^{\pi_n}\right] + \mathbb{E}_{\mathcal{M}_{(\ell^*,a^*)},\mathcal{P}_{\mathcal{A}}}\left[V^* - V^{\pi_{N+1}}\right] \\
&\quad\leq \mathbb{E}_{\mathcal{M}_{(\ell^*,a^*,1)},\mathcal{A}}\left[\sum_{n=1}^{N} V^* - V^{\pi_n}\right] + \mathbb{E}_{\mathcal{M}_{(\ell^*,a^*)},\mathcal{P}_{\mathcal{A}}}\left[\mathbb{E}\left[V^* - V^{\pi_{N+1}} \mid (\mathcal{H}_0, \ldots, \mathcal{H}_N)\right]\right],
\end{aligned}$$

where the final inequality uses the inductive hypothesis and the tower property of expectation. Now, recall from Lemma B.5 that

$$\begin{aligned}&\mathbb{E}_{\mathcal{M}_{(\ell^*,a^*)},\mathcal{P}_{\mathcal{A}}}\left[\mathbb{E}\left[V^* - V^{\pi_{N+1}} \mid (\mathcal{H}_0,\dots,\mathcal{H}_N)\right]\right]\\ &\quad= \mathbb{E}_{\mathcal{M}_{(\ell^*,a^*,1)},\mathcal{A}}\left[\mathbb{E}\left[V^* - V^{\pi_{N+1}} \mid (\mathcal{H}_0,\dots,\mathcal{H}_N)\right]\right].\end{aligned}$$

We emphasize that the value functions are still with respect to $\mathcal{M}_{(\ell^*,a^*)}$. However, for any policy π output by $\mathcal{A}$,

$$\begin{aligned}V^* - V^{\pi} &= \mathbb{E}_{\mathcal{M}(\ell^*,a^*),\pi}\left[(H-W-1)\mathbb{1}\left[s_{W+1} \neq s_{\text{gate}}\right]\right]\\ &\le \mathbb{E}_{\mathcal{M}(\ell^*,a^*),\pi}\left[(H-W-1)\mathbb{1}\left[s_{W+1} \neq s_{\text{gate}} \text{ or } a_{W+1} \neq 1\right]\right]\\ &\le \mathbb{E}_{\mathcal{M}(\ell^*,a^*,1),\pi}\left[(H-W-1)\mathbb{1}\left[s_{W+1} \neq s_{\text{gate}} \text{ or } a_{W+1} \neq 1\right]\right].\end{aligned}$$

Note that the right-hand side is the regret in $\mathcal{M}(\ell^*, a^*, 1)$ for playing π. Therefore, since both algorithms play the same policy π_{N+1}, we thus obtain the desired result by induction. □

With Proposition B.2, we can now formally state and prove the separation result:

Theorem B.3. *There exists a task distribution $\mathcal{T}_{(\ell^*,a^*)} \in \mathcal{T}_W$ such that an algorithm $\mathcal{A}$, without access to the meta-training tasks (and thus without access to the hierarchy), incurs expected regret lower bounded as*

$$\mathbb{E}_{\mathcal{M}\sim\mathcal{T}_{(\ell^*,a^*)}}\left[\text{Regret}_N(\mathcal{M},\mathcal{A})\right] \gtrsim 2^{W/2}\sqrt{H^2N}.$$

On the other hand, for any task distribution in the family, the hierarchy-based learner $\mathcal{P}$ in Section B.2, with access to a 0-suboptimal hierarchy oracle, achieves regret bounded by $\sqrt{H^2N}$ with high probability on any sampled task.

Proof. Fix any algorithm $\mathcal{A}$. Using Theorem B.2, there exists $\mathcal{M}_{(\ell^*,a^*)}$ such that

$$\mathbb{E}_{\mathcal{M}_{(\ell^*,a^*)},\mathcal{P}_{\mathcal{A}}}\left[\sum_{n=1}^{N} V_0^*(s_{\text{root}}) - V_0^{\pi_n}(s_{\text{root}})\right] \gtrsim 2^{W/2}\sqrt{H^2N}$$

Thus, by Proposition B.2,

$$\mathbb{E}_{\mathcal{M}_{(\ell^*,a^*,1)},\mathcal{A}}\left[\sum_{n=1}^{N} V_0^*(s_{\text{root}}) - V_0^{\pi_n}(s_{\text{root}})\right] \gtrsim 2^{W/2}\sqrt{H^2N}.$$

Note that the proof in Proposition B.2 can be extended for $\mathcal{M}_{(\ell^*,a^*,0)}$ with appropriate modifications to $\mathcal{P}_{\mathcal{A}}$, and thus the same inequality holds. Consequently,

$$\mathbb{E}_{\mathcal{M}\sim\mathcal{T}_{(\ell^*,a^*)}}\left[\text{Regret}_N(\mathcal{A})\right] \gtrsim 2^{W/2}\sqrt{H^2N}.$$

On the other hand, with access to the 0-suboptimal hierarchy oracle, observe that the learner only has to plan at timesteps 0 and $W+1$, allowing us to obtain tighter bounds (as $\mathcal{S}_{\text{hl}}$ is smaller than the construction in Section B.2). Furthermore, the suboptimality of planning with the hierarchy oracle is 0 for any task distribution in the family. We thus obtain the desired bound. □

B.5 A Discussion of Definition 5.2

In this section, we discuss why the values defined in Definition 5.2 control the suboptimality of the hierarchical learner. In particular, we provide examples of MDPs that satisfy Assumption 5.1, and are thus in a sense tasks that are "compatible with the hierarchy", but nevertheless force a hierarchy-based learner to incur $O(H)$ suboptimality.

B.5.1 (η,κ)-unreliability

Consider the MDP in Figure 9 with horizon $H+2$ and two actions a^* and a_1. The optimal policy chooses a^* at every step, achieving a value of $H - O(1)$, since

$$\begin{aligned}V_{H+1}^*(s_0) &= \frac{1}{2}H + \frac{1}{2}V_H^*(s_1) = \frac{1}{2}H + \frac{1}{4}(H-1) + \frac{1}{4}V_{H-1}^*(s_2)\\ &= H\sum_{h=1}^{H}\frac{1}{2^h} - \frac{1}{2}\sum_{h=1}^{H}\frac{h}{2^h} = H - O(1).\end{aligned}$$

Now, assume that the MDP has a latent hierarchy so that the set of exits are given by (t_i, a) for any $i \in [H]$ and $a \in \mathcal{A}$. Clearly, the optimal hierarchy-based learner would always choose (t_0, a^*) or (t_0, a_1) as its high-level action. However, if the agent fails to transition to t_0 at the first timestep due to stochasticity, it will go to the end of the chain, back to s_0 and try a^* once more. This is because it already has set a meta-action, and *does not replan until an exit is performed*. Thus, the optimal agent on the meta-MDP achieves a value of $H/2$, and is therefore $O(H)$-suboptimal, even with a 0-suboptimal hierarchy oracle.

Intuitively, hierarchy-based learners as formulated in Section B.2 fail on the MDP in Figure 9 because such learners commit to a skill until completion. Thus, when such skills exhibit high variance in completion times, hierarchy-based learners fare worse than other learners which are able to replan based on the current state (e.g., in this case, choose another exit if a^* fails to take the agent to the current subgoal). Thus, (η, κ)-reliability serves to eliminate such MDPs, ensuring that the skills corresponding to reaching exits are reliable.

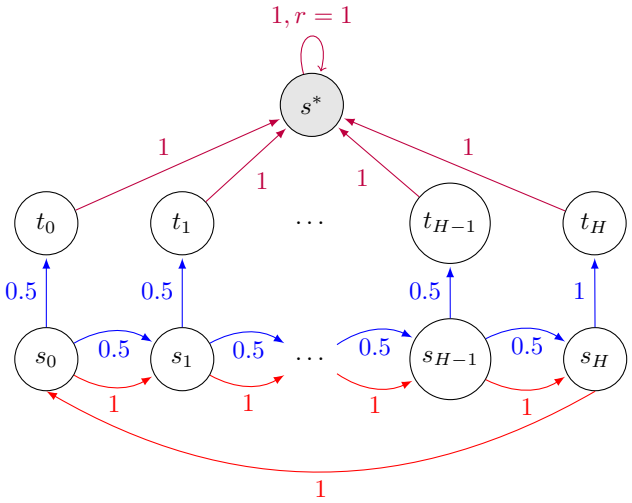

Figure 9: An MDP that does not satisfy low (η, κ)-unreliability, where a^* is in blue, and a_1 is in red (and purple for both actions). State shading represents state clusters, and rewards are 0 unless indicated otherwise.

B.5.2 γ-goal-reaching suboptimality

In this section, we show that even when a hierarchy-based learner has access to highly reliable skills as in the previous section, the learner may still incur high hierarchical sub-optimality. Consider the MDP in Figure 10, where we focus on a single room for simplicity. Furthermore, assume that there are two exits, one from $l_{H/2}$ and one from $r_{H/2}$. Note that a 0-suboptimal hierarchy oracle has highly reliable goal-reaching policies for reaching both of these exit states, requiring exactly $H/2$ timesteps with no stochasticity.

However, given the values assigned to $l_{H/2}$ and $r_{H/2}$, the optimal policy would opt to take the state t, which transitions to either state with probability at least $1/2$ in only two environment steps. Therefore, the optimal policy achieves an optimal value of $H - O(1)$. However, the optimal policy, in having to commit to exactly one of the exits, will achieve a value of $H/2$, and thus be $O(H)$-suboptimal despite having a perfect hierarchy oracle.

Hierarchy-based learners fail on the MDP in Figure 10 because an optimal policy for goal-reaching does not necessarily reach a goal as quickly as possible. Thus, γ-goal-reaching suboptimality is a regularity condition that ensures that this is indeed the case.

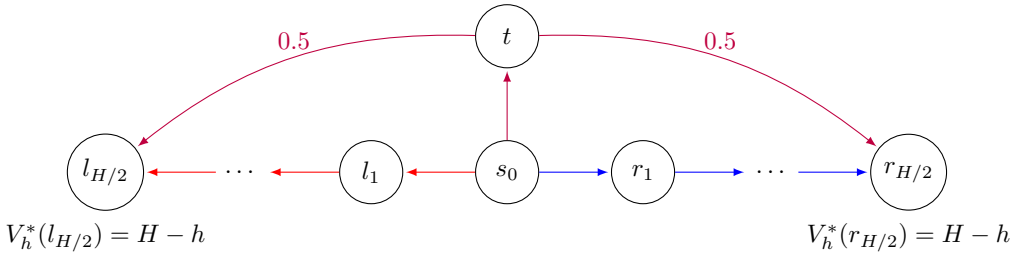

Figure 10: An MDP that does not satisfy low γ-goal-reaching suboptimality, with three actions indicated by red, blue, and purple, and exits l_h and r_h. The MDP satisfies $(\infty, 0)$-unreliability, yet nevertheless exhibits high hierarchical suboptimality.

C EXPERIMENTS

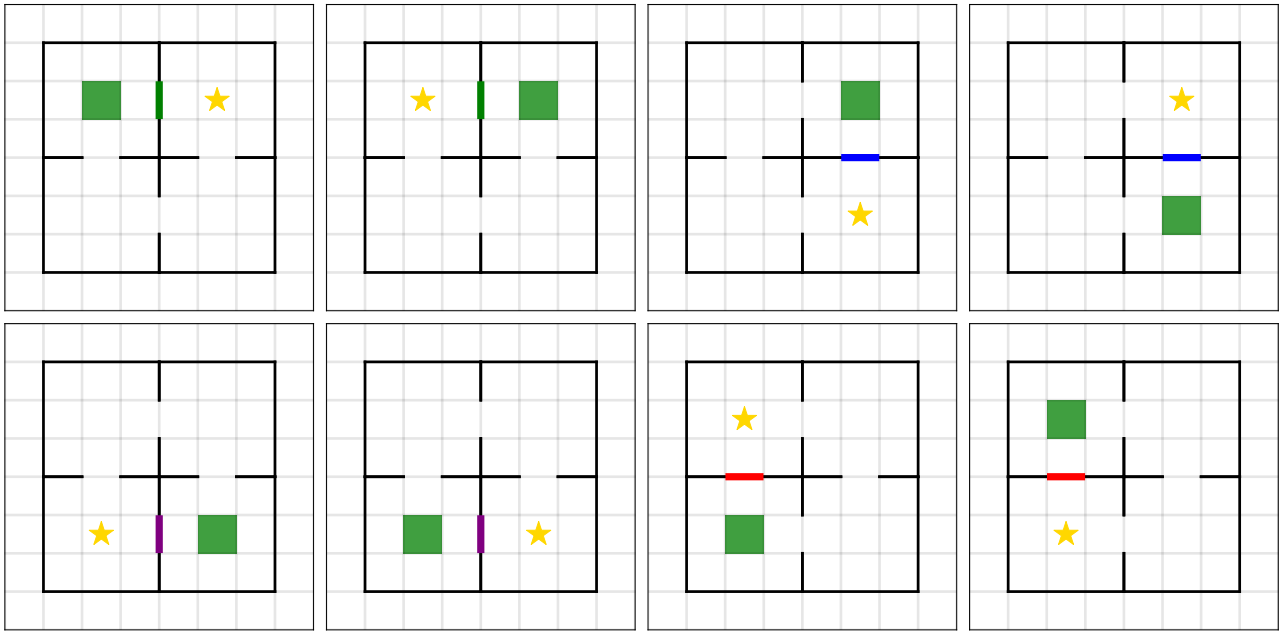

Figure 11: The tasks that were made available to the exit-learning algorithm.

We ran our proposed exit-detection algorithm on the tasks illustrated in Figure 11. Note that we used UCBVI instead of EULER in our implementation of the algorithm wherever it was used. Our algorithm was able to successfully detect all four gates with the parameters provided in Table 2 over 10 repeated trials, with no false positives in all trials.

Parameter	Value	Parameter	Value
N_{UCBVI}	10000	$N^{\text{ED}}_{\text{thresh}}$	50
$N^{\text{TS}}_{\text{thresh}}$	50	N_{ED}	50
N_{TS}	100	N_{EL}	100
$N^{\text{RF}}_{\text{UCBVI}}$	10000	$N^{\text{EL}}_{\text{UCBVI}}$	1000
N_{RF}	5000		

Table 2: Table of parameters.