The Power of Resets in Online Reinforcement Learning

Zakaria Mhammed
mhammedi@google.com

Dylan J. Foster
dylanfoster@microsoft.com

Alexander Rakhlin
rakhlin@mit.edu

Abstract

Simulators are a pervasive tool in reinforcement learning, but most existing algorithms cannot efficiently exploit simulator access—particularly in high-dimensional domains that require general function approximation. We explore the power of simulators through online reinforcement learning with *local simulator access* (or, local planning), an RL protocol where the agent is allowed to *reset* to previously observed states and follow their dynamics during training. We use local simulator access to unlock new statistical guarantees that were previously out of reach:

1. We show that MDPs with low *coverability* (Xie et al., 2023)—a general structural condition that subsumes Block MDPs and Low-Rank MDPs—can be learned in a sample-efficient fashion with *only $Q^\star$-realizability* (realizability of the optimal state-value function); existing online RL algorithms require significantly stronger representation conditions.
2. As a consequence, we show that the notorious *Exogenous Block MDP* problem (Efroni et al., 2022b) is tractable under local simulator access.

The results above are achieved through a computationally inefficient algorithm. We complement them with a more computationally efficient algorithm, RVFS (*Recursive Value Function Search*), which achieves provable sample complexity guarantees under a strengthened statistical assumption known as *pushforward coverability*. RVFS can be viewed as a principled, provable counterpart to a successful empirical paradigm that combines recursive search (e.g., MCTS) with value function approximation.

1 Introduction

Simulators are a widely used tool in reinforcement learning. Many of the most well-known benchmarks for reinforcement learning research make use of simulators (Atari (Bellemare et al., 2012), MuJoCo (Todorov et al., 2012), OpenAI Gym (Brockman et al., 2016), DeepMind Control Suite (Tassa et al., 2018)), and high-quality simulators are available for a wide range of real-world control tasks, including robotic control (Qassem et al., 2010; Akkaya et al., 2019), autonomous vehicles (Bojarski et al., 2016; Aradi, 2020), and game playing (Silver et al., 2016, 2018). Simulators also provide a useful abstraction for *planning* with a known or learned model, an important building block for many RL techniques (Schrittwieser et al., 2020). Yet, in spite of the ubiquity of simulators, almost all existing research into algorithm design—empirical and theoretical—has focused on the *online reinforcement learning* (where only trajectory-based feedback is available), and does not take advantage of the extra information available through the simulator. Relatively little is known about the full power of RL with simulator access, either in terms of algorithmic principles or fundamental limits.

We explore the power of simulators through online reinforcement learning with *local simulator access* (RLLS for short), also known as *local planning* (Weisz et al., 2021a; Li et al., 2021; Amortila et al., 2022; Weisz et al., 2022; Yin et al., 2022, 2023). Here, the agent learns by repeatedly executing policies and observing the resulting trajectories (as in online RL), but is allowed to *reset* to previously observed states and follow their dynamics during training.

Empirically, algorithms based on local simulators have received limited investigation, but with promising results. Notably, the Go-Explore algorithm (Ecoffet et al., 2019, 2021) uses local simulator access to achieve state-of-the-art performance for Montezuma's Revenge (a difficult Atari game that requires systematic exploration), beating the performance of the best agents trained with online RL (Badia et al., 2020; Guo et al., 2022) by a significant margin that has yet to be closed. The successful line of research on AlphaGo and successors (Silver et al., 2016, 2018; Schrittwieser et al., 2020) also uses local simulator access, albeit at test time in addition to training time.

These results suggest that developing improved algorithm design principles for RL with local simulator access could have significant practical implications, but current theoretical understanding of local simulators is limited. Recent work has shown that local simulator access has provable benefits for reinforcement learning with various types of linear function approximation (Weisz et al., 2021a; Li et al., 2021; Amortila et al., 2022; Yin et al., 2022; Weisz et al., 2022), but essentially nothing is known for RL problems in large state spaces that demand general, potential neural function approximation. This leads us to ask:

Can we develop algorithms for reinforcement learning with general function approximation that provably benefit from local simulator access?

From an algorithm design perspective, perhaps the greatest challenge in using local simulators to speed up learning is to understand which states are "informative" in the sense that we should prioritize revisiting them. Here, we are faced with a chicken-and-egg problem: to understand which states to prioritize, we must explore and gather information, but it is unclear how to do so efficiently unless we already have a way to understand which states are informative. It is natural to let function approximation (valued-based, model-based, or otherwise) guide us; to this end, recent research (Li et al., 2021; Yin et al., 2022; Weisz et al., 2022) on linearly-parameterized RL with local simulators makes use of *core-sets*: small, adaptively chosen sets of informative state-action pairs designed to cover the feature space and enable efficient value function learning. Core-sets facilitate sample complexity guarantees for linear models that are not possible without local simulator access (e.g., Li et al. (2021)). Yet, for general function classes—particularly rich models like neural networks that do not readily support extrapolation—defining a suitable notion of core-set is challenging. Consequently, existing techniques have yet to meaningfully leverage local simulator access beyond the linear regime.

1.1 Contributions

We show that local simulator access unlocks new guarantees for online reinforcement learning with general value function approximation—statistical and computational—that were previously out of reach.

Sample-efficient learning. We show that MDPs with low *coverability* (Xie et al., 2023)—a general structural condition that subsumes Block MDPs and Low-Rank MDPs—can be learned in a sample-efficient fashion with *only $Q^\star$-realizability* (that is, realizability for the optimal state-action value function). This is achieved through a new algorithm, SimGolf, that augments the principle of global optimism with local simulator access, and improves upon the best existing guarantees for the fully online RL setting, which require significantly stronger representation conditions. As a consequence, we show for the first time that the notoriously challenging *Exogenous Block MDP* (ExBMDP) problem (Efroni et al., 2022b,a) is tractable in its most general form under local simulator access.

Practical, computationally efficient learning. Our results above are achieved through a computationally inefficient algorithm. We complement them with a practical and computationally efficient algorithm, RVFS ("Recursive Value Function Search"), which achieves sample-efficient learning guarantees with general value function approximation under a strengthened, yet novel, statistical assumption known as *pushforward coverability* (Xie and Jiang, 2021; Amortila et al., 2024b). Assuming either i) realizability of the optimal state-value function $V^\star$ and a state-action gap or ii) realizability of V^π for all π, RVFS achieves polynomial sample complexity in a computationally efficient fashion, and leads to guarantees for a new class of Exogenous Block MDPs with *weakly correlated* exogenous noise. RVFS explores by building core-sets with a novel value function-guided scheme, and can be viewed as a principled counterpart to algorithms including MCTS and AlphaZero (Silver et al., 2016, 2018; Ecoffet et al., 2019, 2021; Yin et al., 2023), that combine recursive search with value function approximation. Compared to these approaches, RVFS is designed to provably address stochastic environments and distribution shift.

Paper organization. Section 2 formally introduces our problem setting, including the local simulator framework. Section 3 presents our main sample complexity guarantees, and Section 4 gives computationally efficient algorithms. We close with discussion in Section 5. All proofs are deferred to the appendix.

2 Setup: Reinforcement Learning with Local Simulator Access

We consider an episodic reinforcement learning setting. A Markov Decision Process (MDP) is a tuple $\mathcal{M} = (\mathcal{X}, \mathcal{A}, T, R, H)$, where $\mathcal{X}$ is a (large/potentially infinite) state space, $\mathcal{A}$ is the action space (we abbreviate $A = |\mathcal{A}|$), $H \in \mathbb{N}$ is the horizon, $R = \{R_h\}_{h=1}^H$ is the reward function (where $R_h : \mathcal{X} \times \mathcal{A} \to [0,1]$) and $T = \{T_h\}_{h=0}^H$ is the transition distribution (where $T_h : \mathcal{X} \times \mathcal{A} \to \Delta(\mathcal{X})$), with the convention that $T_0(\cdot \mid \varnothing)$ is the initial state distribution. A policy is a sequence of functions $\pi = \{\pi_h : \mathcal{X} \to \Delta(\mathcal{A})\}_{h=1}^H$; we use Π_{S} to denote the set of all such functions. When a policy is executed, it generates a trajectory $(\boldsymbol{x}_1, \boldsymbol{a}_1, \boldsymbol{r}_1), \dots, (\boldsymbol{x}_H, \boldsymbol{a}_H, \boldsymbol{r}_h)$ via the process $\boldsymbol{a}_h \sim \pi_h(\boldsymbol{x}_h), \boldsymbol{r}_h \sim R_h(\boldsymbol{x}_h, \boldsymbol{a}_h), \boldsymbol{x}_{h+1} \sim T_h(\cdot \mid \boldsymbol{x}_h, \boldsymbol{a}_h)$, initialized from $\boldsymbol{x}_1 \sim T_0(\cdot \mid \varnothing)$ (we use $\boldsymbol{x}_{H+1}$ to denote a terminal state with zero reward). We write $\mathbb{P}^\pi[\cdot]$ and $\mathbb{E}^\pi[\cdot]$ to denote the law and expectation under this process.

For a policy π, $J(\pi) \coloneqq \mathbb{E}^\pi\left[\sum_{h=1}^H \boldsymbol{r}_h\right]$ denotes expected reward, and the value functions are given by

$$V_h^\pi(x) \coloneqq \mathbb{E}^\pi\left[\sum_{h'=h}^H \boldsymbol{r}_{h'} \mid \boldsymbol{r}_h = x\right], \quad \text{and} \quad Q_h^\pi(x,a) \coloneqq \mathbb{E}^\pi\left[\sum_{h'=h}^H \boldsymbol{r}_{h'} \mid \boldsymbol{x}_h = x, \boldsymbol{a}_h = a\right].$$

We denote by $\pi^\star = \{\pi_h^\star\}_{h=1}^H$ the optimal deterministic policy that maximizes $Q^{\pi^\star}$ at all states, and write $Q^\star \coloneqq Q^{\pi^\star}$ and $V^\star \coloneqq V^{\pi^\star}$.

2.1 Online Reinforcement Learning with Local Simulator Access

In the standard online reinforcement learning framework, the learner repeatedly interacts with an (unknown) MDP by executing a policy and observing the resulting trajectory, with the goal of maximizing the total reward. Formally, for each episode $\tau \in [N_{\mathsf{episodes}}]$, the learner selects a policy $\pi^{(\tau)} = \{\pi_h^{(\tau)}\}_{h=1}^H$, executes it in the underlying MDP $\mathcal{M}^\star$ and observes the trajectory $\{(\boldsymbol{x}_h^{(\tau)}, \boldsymbol{a}_h^{(\tau)}, \boldsymbol{r}_h^{(\tau)})\}_{h=1}^H$. After all N_{episodes} episodes conclude, the learner produces a policy $\widehat{\pi} \in \Pi_{\mathsf{S}}$ with the goal of minimizing the risk given by $\mathbb{E}[J(\pi^\star) - J(\widehat{\pi})]$.

In online RL with local simulator access, or RLLS, (Weisz et al., 2021a; Li et al., 2021; Yin et al., 2022; Weisz et al., 2022; Yin et al., 2023), we augment the online RL protocol as follows:[1] At each episode $\tau \in [N]$, instead of starting from a random initial state $\boldsymbol{x}_1 \sim T_0(\cdot \mid \varnothing)$, the agent can *reset* the MDP to any layer $h \in [H]$ and any state $\boldsymbol{x}_h$ previously encountered, and proceed with a new episode starting from this point. As in the online RL protocol, the goal is to produce a policy $\widehat{\pi} \in \Pi_{\mathsf{S}}$ such that

$$\mathbb{E}[J(\pi^\star) - J(\widehat{\pi})] \le \varepsilon$$

with as few episodes of interaction as possible; our main results take $N_{\mathsf{episodes}} = \mathrm{poly}(C, \varepsilon^{-1})$ for a suitable problem parameter C.

Executable versus non-executable policies. We focus on learning policies that can be executed without access to a local simulator (in other words, the local simulator used at train time, but not test time). Some recent work using local simulators for RL with linear function approximation (Weisz et al., 2021a) considers a more permissive setting where the final policy π produced by the learner can be *non-executable*; our function approximation requirements can be slightly relaxed in this case.

Definition 2.1 (Non-executable policy). *We refer to a policy π for which computing $\pi(x) \in \Delta(\mathcal{A})$ for any $x \in \mathcal{X}$ requires n local simulator queries as a* non-executable policy *with sample complexity n.*

[1] We use the term "local simulator" instead of "local planning" because we find it to be slightly more descriptive.

Implications for planning. RL with local simulator access is a convenient abstraction for the problem of *planning*: Given a known (e.g., learned) model, compute an optimal policy. Planning with a learned model is an important task in theory (Foster et al., 2021; Liu et al., 2023) and practice (e.g., Schrittwieser et al. (2020)). Since the model is known, computing an optimal policy is a purely computational problem, not a statistical problem. Nonetheless, for planning problems in large state spaces, where enumerating over all states is undesirable, algorithms for online RL with local simulator access can be directly applied, using the model to simulate the environment the agent interacts with. Here, any computationally efficient algorithm in our framework immediately yields an efficient algorithm for planning. See Appendix A for more discussion.

2.2 Additional Notation

For any $m, n \in \mathbb{N}$, we denote by $[m \mathinner{..} n]$ the integer interval $\{m, \dots, n\}$. We also let $[n] \coloneqq [1 \mathinner{..} n]$. We refer to a scalar $c > 0$ as an *absolute constant* to indicate that it is independent of all problem parameters and use $\widetilde{O}(\cdot)$ to denote a bound up to factors polylogarithmic in parameters appearing in the expression. We define $\pi_{\mathsf{unif}} \in \Pi_{\mathsf{S}}$ as the random policy that selects actions in $\mathcal{A}$ uniformly at random at each layer. We define the occupancy measure for policy π via $d^{\pi}_h(x, a) \coloneqq \mathbb{P}^{\pi}[\boldsymbol{x}_h = x, \boldsymbol{a}_h = a]$. For functions $g : \mathcal{X} \times \mathcal{A} \to \mathbb{R}$ and $f : \mathcal{X} \to \mathbb{R}$, we define Bellman backup operators by $\mathcal{T}_h[g](x, a) = \mathbb{E}[\boldsymbol{r}_h + \max_{a' \in \mathcal{A}} g(\boldsymbol{x}_{h+1}, a') \mid \boldsymbol{x}_h = x, \boldsymbol{a}_h = a]$ and $\mathcal{P}_h[f](x, a) = \mathbb{E}[\boldsymbol{r}_h + f(\boldsymbol{x}_{h+1}) \mid \boldsymbol{x}_h = x, \boldsymbol{a}_h = a]$. For a stochastic policy $\pi \in \Pi_{\mathsf{S}}$, we will occasionally use the bold notation $\boldsymbol{\pi}_h(x)$ as shorthand for the random variable $\boldsymbol{a}_h \sim \pi_h(x) \in \Delta(\mathcal{A})$. For a function $f : \mathcal{A} \to \mathbb{R}$, we write $a' \in \arg\max_{a \in \mathcal{A}} f(a)$ to denote the action that maximizes f. If there are ties, we break them by picking the action with the smallest index; we assume without loss of generality that actions in $\mathcal{A}$ are index from $1, \dots, |\mathcal{A}|$.

3 New Sample-Efficient Learning Guarantees via Local Simulators

This section presents our most powerful results for RLLS. We present a new algorithm for learning with local simulator access, `SimGolf` (Section 3.1), and show that it enables sample-efficient RL for MDPs with low *coverability* (Xie et al., 2023) using only $Q^{\star}$-realizability (Section 3.2). We then give implications for the Exogenous Block MDP problem (Section 3.3).

Function approximation setup and coverability. To achieve sample complexity guarantees for online reinforcement learning that are suitable for large, high-dimensional state spaces, we appeal to *value function approximation*. We assume access to a function class $\mathcal{Q} \subset (\mathcal{X} \times \mathcal{A} \times [H] \to [0, H])$ that contains the optimal state-action value function $Q^{\star}$; we define $\mathcal{Q}_h = \{Q_h \mid Q \in \mathcal{Q}\}$.

Assumption 3.1 ($Q^{\star}$-realizability)**.** *For all $h \in [H]$, we have $Q^{\star}_h \in \mathcal{Q}_h$.*

$Q^{\star}$-realizability is widely viewed as a minimal representation condition for online RL (Wen and Van Roy, 2017; Du et al., 2020, 2019b; Lattimore et al., 2020; Weisz et al., 2021b; Wang et al., 2021). The class $\mathcal{Q}$ encodes the learner's prior knowledge about the MDP, and can be parameterized by rich function approximators like neural networks. We assume for simplicity of exposition that $\mathcal{Q}$ and Π are finite, and aim for sample complexity guarantees scaling with $\log|\mathcal{Q}|$ and $\log|\Pi|$; extending our results to infinite classes via standard uniform convergence arguments is straightforward.

Coverability. Beyond representation conditions like realizability, online reinforcement learning algorithms require *structural conditions* that limit the extent to which deliberately designed algorithms can be surprised by substantially new state distributions. We focus on a structural condition known as *coverability* (Xie et al., 2023), which is inspired by connections between online and offline reinforcement learning.

Assumption 3.2 (Coverability (Xie et al., 2023))**.** *The coverability coefficient $C_{\mathsf{cov}} > 0$ is given by*

$$C_{\mathsf{cov}} \coloneqq \max_{h \in [H]} \inf_{\mu_h \in \Delta(\mathcal{X} \times \mathcal{A})} \sup_{\pi \in \Pi_{\mathsf{S}}} \left\| \frac{d^{\pi}_h}{\mu_h} \right\|_{\infty}. \tag{1}$$

Coverability is an intrinsic strutural property of the underlying MDP. Examples of MDP families with low coverability include (Exogenous) Block MDPs, which have $C_{\mathsf{cov}} \le |\mathcal{S}||\mathcal{A}|$, where $\mathcal{S}$ is the *latent state space* (Xie et al., 2023), and Low-Rank MDPs, which have $C_{\mathsf{cov}} \le d|\mathcal{A}|$, where d is the feature dimension (Huang et al., 2023); importantly, these settings exhibit high-dimensional state spaces and require nonlinear function approximation. To the best of our knowledge, coverability is the only structural parameter that is known to be small for the Exogenous Block MDP problem, as other well-known parameters like Bellman rank and Bellman-Eluder dimension can be arbitrarily large (Xie et al., 2023). We emphasize that as in prior work (Xie et al., 2023; Amortila et al., 2024a), our algorithms require *no prior knowledge* of the distribution μ_h that minimizes Eq. (1).

3.1 Algorithm

Our main algorithm, `SimGolf`, is displayed in Algorithm 1. The algorithm is a variant of the `GOLF` method of Jin et al. (2021); Xie et al. (2023) with novel adaptations to exploit the availability of a local simulator. Like `GOLF`, `SimGolf` explores using the principle of *global optimisim*: At each iteration $t \in [N]$, it maintains a confidence set (or, version space) $\mathcal{Q}^{(t)} \subset \mathcal{Q}$ of candidate value functions with low squared Bellman error under the data collected so far, and chooses a new exploration policy $\pi^{(t)}$ by picking the most "optimistic" value function in this set. As the algorithm gathers more data, the confidence set shrinks, leaving only near-optimal policies.

The main novelty in `SimGolf` arises in the data collection strategy and design of confidence sets. Like `GOLF`, `SimGolf` algorithm constructs the confidence set $\mathcal{Q}^{(t)} \subset \mathcal{Q}$ such that all value functions $g \in \mathcal{Q}^{(t)}$ have small squared Bellman error:

$$\sum_{i<t} \mathbb{E}^{\pi^{(i)}}\Big[\big(g_h(\boldsymbol{x}_h, \boldsymbol{a}_h) - \mathcal{T}_h[g_{h+1}](\boldsymbol{x}_h, \boldsymbol{a}_h)\big)^2\Big] \lesssim \log|\mathcal{Q}| \quad \forall h \in [H]. \tag{2}$$

Due to the presence of the Bellman backup $\mathcal{T}_h[g_{h+1}]$ in Eq. (2), naively estimating squared Bellman error leads to the notorious *double sampling* problem. To avoid this, the approach taken with `GOLF` and related work (Zanette et al., 2020; Jin et al., 2021) is to adapt a certain de-biasing technique to remove double sampling bias, but this requires access to a value function class that satisfies *Bellman completeness*, a representation significantly more restrictive than realizability (e.g., Foster et al. (2022)).

The idea behind `SimGolf` is to use local simulator access to directly produce high-quality estimates for the Bellman backup function $\mathcal{T}_h[g_{h+1}]$ in Eq. (2). In particular, for a given state-action pair $(x,a) \in \mathcal{X} \times \mathcal{A}$, we can estimate the Bellman backup $\mathcal{T}_h[g_{h+1}](x,a)$ for all functions $g \in \mathcal{Q}$ simultaneously by collecting K next-state transitions $\widetilde{\boldsymbol{x}}_{h+1}^{(1)}, \dots, \widetilde{\boldsymbol{x}}_{h+1}^{(K)} \overset{\text{i.i.d.}}{\sim} T_h(\cdot \mid x,a)$ and K rewards $\widetilde{\boldsymbol{r}}_h^{(1)}, \dots, \widetilde{\boldsymbol{r}}_h^{(K)} \overset{\text{i.i.d.}}{\sim} R_h(x,a)$, then taking the empirical mean: $\mathcal{T}_h[g_{h+1}](x,a) \approx \frac{1}{K}\sum_{k=1}^{K}(\widetilde{\boldsymbol{r}}_h^{(k)} + \max_{a' \in \mathcal{A}} g_{h+1}(\widetilde{\boldsymbol{x}}_{h+1}^{(k)}, a'))$. Line 9 of `SimGolf` uses this technique to directly estimate the Bellman residual backup under a trajectory gathered with $\pi^{(t)}$, sidestepping the double sampling problem and removing the need for Bellman completeness. We suspect that the idea of performing estimation with respect to squared Bellman error directly using local simulator access may find broader use.

Remark 3.1 (Squared Bellman error versus average Bellman error)**.** *A general approach to removing the need for Bellman completeness is to estimate* average Bellman error *instead of squared Bellman error (e.g., Jiang et al. (2017)). However, Xie et al. (2023) show that this is insufficient to obtain sample complexity guarantees under coverability.*

3.2 Main Result

We now state the main guarantee for `SimGolf` and discuss some of its implications.

Theorem 3.1 (Main guarantee for `SimGolf`)**.** *Let $\varepsilon, \delta \in (0,1)$ be given and suppose Assumption 3.1 ($Q^\star$-realizability) and Assumption 3.2 (coverability) hold with $C_{\mathsf{cov}} > 0$. Then the policy $\widehat{\pi}$ produced by* `SimGolf`$(\mathcal{Q}, C_{\mathsf{cov}}, \varepsilon, \delta)$ *(Algorithm 1) has $J(\pi^\star) - \mathbb{E}[J(\widehat{\pi})] \le \varepsilon$ with probability at least $1-\delta$. The total sample*

Algorithm 1 `SimGolf`: Global Optimism via Local Simulator Access

```
1: input: Value function class $\mathcal{Q}$, coverability $C_{\mathsf{cov}} > 0$, suboptimality parameter $\varepsilon > 0$, and confidence parameter $\delta > 0$.
2: Set $N \leftarrow \widetilde{\Theta}(H^2 C_{\mathsf{cov}} \beta / \varepsilon^2)$, $\beta_{\mathsf{stat}} \leftarrow 16 \log(2HN|\mathcal{Q}|\delta^{-1})$, $\beta \leftarrow 2\beta_{\mathsf{stat}}$, and $K \leftarrow \frac{8N}{\beta_{\mathsf{stat}}}$.
3: initialize: $\mathcal{Q}^{(1)} \leftarrow \mathcal{Q}$.
4: for iteration $t = 1, 2, \ldots, N$ do
5:     Select $g^{(t)} = \arg\max_{g \in \mathcal{Q}^{(t)}} \sum_{s<t} \max_{a \in \mathcal{A}} g_1(\boldsymbol{x}_1^{(s)}, a)$.
6:     For each $h \in [H]$ and $x \in \mathcal{X}$, define $\pi_h^{(t)}(x) \in \arg\max_{a \in \mathcal{A}} g_h^{(t)}(x, a)$.
7:     Execute $\pi^{(t)}$ for an episode and observe $\boldsymbol{\tau}^{(t)} := (\boldsymbol{x}_1^{(t)}, \boldsymbol{a}_1^{(t)}), \ldots, (\boldsymbol{x}_H^{(t)}, \boldsymbol{a}_H^{(t)})$.
8:     for $h \in [H]$ do
9:         Draw $K$ independent samples $\boldsymbol{x}_{h+1}^{(t,k)} \sim T_h(\cdot \mid \boldsymbol{x}_h^{(t)}, \boldsymbol{a}_h^{(t)})$, $\boldsymbol{r}_h^{(t,k)} \sim R_h(\boldsymbol{x}_h^{(t)}, \boldsymbol{a}_h^{(t)})$.
10:    Compute confidence set:
           $$\mathcal{Q}^{(t+1)} \leftarrow \left\{ g \in \mathcal{Q} : \sum_{h \le t} \left( g_h(\boldsymbol{x}_h^{(t)}, \boldsymbol{a}_h^{(t)}) - \frac{1}{K} \sum_{k=1}^{K} \left( \boldsymbol{r}_h^{(t,k)} + \max_{a \in \mathcal{A}} g_{h+1}(\boldsymbol{x}_{h+1}^{(t,k)}, a) \right) \right)^2 \le \beta, \ \forall h \in [H] \right\}.$$
11: return: $\widehat{\pi} = \mathtt{unif}(\pi^{(1)}, \ldots, \pi^{(T)})$.
```

complexity in the RLLS framework is bounded by

$$\widetilde{O}\left(\frac{H^5 C_{\mathsf{cov}}^2 \log(|\mathcal{Q}|/\delta)}{\varepsilon^4} \right).$$

This result (whose proof is in Appendix D) shows that under only $Q^\star$-realizability and coverability, `SimGolf` learns an ε-optimal policy with polynomial sample complexity, significantly relaxing the representation assumptions (Bellman completeness, weight function realizability) required by prior algorithms for coverability (Xie et al., 2023; Amortila et al., 2024a). This is the first instance we are aware of where local simulator access unlocks sample complexity guarantees for reinforcement learning with *nonlinear* function approximation that were previously out of reach; perhaps the most important technical idea here is our approach to combining global optimism with local simulator access, in contrast to greedy layer-by-layer schemes used in prior work on local simulators (with the exception of Weisz et al. (2021a)). In particular, we suspect that the idea of performing estimation with respect to squared Bellman error directly using local simulator access may find broader use beyond coverability. Improving the polynomial dependence on problem parameters is an interesting question for future work.

A conjecture. By analogy to results in offline reinforcement learning, where $Q^\star$-realizability and concentrability (the offline counterpart to coverability) alone are known to be insufficient for sample-efficient learning (Chen and Jiang, 2019; Foster et al., 2022), we conjecture that $Q^\star$-realizability and coverability alone are not sufficient for polynomial sample complexity in vanilla online RL. If true, this would imply a new separation between online RL with and without local simulators.

Proof sketch. As described in Section 3.1, the main difference between `SimGolf` and `GOLF` lies in the construction of the confidence sets. The most important new step in the proof of Theorem 3.1 is to show that the local simulator-based confidence set construction in Line 10 is valid in the sense that the property Eq. (2) holds with high probability. From here, the sample complexity bound follows by adapting the change-of-measure argument based on coverability from Xie et al. (2023).

3.3 Implications for Exogenous Block MDPs

We now apply `SimGolf` and Theorem 3.1 to the *Exogenous Block MDP* (ExBMDP) problem (Efroni et al., 2022b,a; Lamb et al., 2023; Islam et al., 2023), a challenging rich-observation reinforcement learning setting in which the observed states $\boldsymbol{x}_h$ are high-dimensional, while the underlying dynamics of the system are low-dimensional, yet confounded by temporally correlated exogenous noise.

Formally, an Exogenous Block MDP $\mathcal{M} = (\mathcal{X}, \mathcal{S}, \Xi, \mathcal{A}, H, T, R, g)$ is defined by a *latent state space* and an *observation space*. We begin by describing the latent state space. Starting from an initial *endogenous state* $\boldsymbol{s}_1 \in \mathcal{S}$ and *exogenous state* $\boldsymbol{\xi}_1 \in \Xi$, the latent state $\boldsymbol{z}_h = (\boldsymbol{s}_h, \boldsymbol{\xi}_h)$ evolves for $h \in [H]$ via

$$\boldsymbol{s}_{h+1} \sim T_h^{\mathsf{endo}}(\cdot \mid \boldsymbol{s}_h, \boldsymbol{a}_h), \quad \text{and} \quad \boldsymbol{\xi}_{h+1} \sim T_h^{\mathsf{exo}}(\cdot \mid \xi_h),$$

where $\boldsymbol{a}_h \in \mathcal{A}$ is the agent's action at layer h; we adopt the convention that $\boldsymbol{s}_1 \sim T_0^{\mathsf{endo}}(\cdot \mid \varnothing)$ and $\boldsymbol{\xi}_1 \sim T_0^{\mathsf{exo}}(\cdot \mid \varnothing)$. Note that only the endogenous state is causally influenced by the action. The latent state is not observed; instead, at each step h, the agent receives an *observation* $\boldsymbol{x}_h \in \mathcal{X}$ generated via[2]

$$\boldsymbol{x}_h = g_h^{\mathsf{obs}}(\boldsymbol{s}_h, \boldsymbol{\xi}_h), \tag{3}$$

where $g_h^{\mathsf{obs}} : \mathcal{S} \times \Xi \to \mathcal{X}$ is the *emission function*. We assume the endogenous latent space $\mathcal{S}$ and action space $\mathcal{A}$ are finite, and define $S := |\mathcal{S}|$ and $A := |\mathcal{A}|$. However, the exogenous state space Ξ and observation space $\mathcal{X}$ may be arbitrarily large or infinite, with $|\Xi|, |\mathcal{X}| \gg |\mathcal{S}|$.[3]

The final property of the ExBMDP model is *decodability*, which asserts the existence of a *decoder* such that $\phi_\star : \mathcal{X} \to \mathcal{S}$ such that $\phi_\star(\boldsymbol{x}_h) = \boldsymbol{s}_h$ a.s. for all $h \in [H]$ under the process in Eq. (3). Informally, decodability ensures the existence of an (unknown to the learner) mapping that allows one to perfectly recover the endogenous latent state from observations. In addition to decodability, we assume the rewards in the ExBMDP are *endogenous*; that is, the reward distribution $R_h(\boldsymbol{x}_h, \boldsymbol{a}_h)$ only depends on the observations $(\boldsymbol{x}_h)$ through the corresponding latent states $(\phi^\star(\boldsymbol{x}_h) = \boldsymbol{s}_h)$. To enable sample-efficient learning, we assume access to a *decoder class* Φ that contains $\phi^\star$, as in prior work.

Assumption 3.3 (Decoder realizability)**.** *We have access to a decoder class Φ such that $\phi^\star \in \Phi$.*

Applying SimGolf and Theorem 3.1. To apply Theorem 3.1 to the ExBMDP problem, we need to verify that $Q^\star$-realizability and coverability hold. For coverability, Xie et al. (2023) show that ExBMDPs have $C_{\mathsf{cov}} \leq SA$ under decodability, in spite of the time-correlated exogenous noise process $(\boldsymbol{\xi}_h)$ and potentially infinite observation space $\mathcal{X}$ (interestingly, coverability is essentially the only useful structural property that ExBMDPs are known to satisfy, which is our primary motivation for studying it). Realizability is also a straightforward consequence of the decodability assumption.

Lemma 3.1 (Efroni et al. (2022a))**.** *For the ExBMDP setting, under Assumption 3.3, (i) the function class $\mathcal{Q}_h := \{x \mapsto g(\phi(x), a) : g \in [0, H]^{SA}, \phi \in \Phi\}$ satisfies Assumption 3.1, and (ii) the policy class $\Pi_h = \{x \mapsto \pi(\phi(x)) : \pi \in \mathcal{A}^S, \phi \in \Phi\}$ satisfies Assumption 4.3. Both classes have $\log|\mathcal{Q}_h| = \log|\Pi_h| = \widetilde{O}(S + \log|\Phi|)$.*[4]

This leads to the following corollary of Theorem 3.1.

Corollary 3.1 (SimGolf for ExBMDPs)**.** *Consider the ExBMDP setting. Suppose that Assumption 3.3 holds, and let $\mathcal{Q}$ be constructed as in Lemma 3.1 of Part I. Then for any $\varepsilon, \delta \in (0,1)$, the policy $\widehat{\pi} =$ SimGolf$(\mathcal{Q}, SA, \varepsilon, \delta)$ has $J(\pi^\star) - J(\widehat{\pi}) \leq \varepsilon$ with probability at least $1 - \delta$. The total sample complexity in the RLLS framework is bounded by*

$$\widetilde{O}\left(\frac{H^5 S^3 A^3 \log|\Phi|}{\varepsilon^4}\right).$$

This shows for the first time that general ExBMDPs are learnable with local simulator access. Prior to this work, online RL algorithms for ExBMDPs required either (i) deterministic latent dynamics (Efroni et al., 2022b), or (ii) factored emission structure (Efroni et al., 2022a). Xie et al. (2023) observed that ExBMDPs admit low coverability, but their algorithm requires Bellman completeness, which is not satisfied by ExBMDPs (see Islam et al. (2023)). See Appendix A for more discussion.

[2] A more standard formulation of the observation process (Efroni et al., 2022b,a; Lamb et al., 2023; Islam et al., 2023) assumes that observations are generated via $\boldsymbol{x}_h \sim q_h(\boldsymbol{s}_h, \boldsymbol{\xi}_h)$, where $q_h(\cdot, \cdot)$ is a conditional distribution with the decodability property. This is equivalent to Eq. (3) under mild measure-theoretic conditions, as randomness in the emission process can be included in the exogenous state without loss of generality, but the formulation in Eq. (3) makes our proofs slightly more compact.

[3] To simplify presentation, we assume that Ξ and $\mathcal{X}$ are countable; our results trivially extend to the case where the corresponding variables are continuous with an appropriate measure-theoretic treatment.

[4] Formally, this requires a standard covering number argument; we omit the details.

4 Computationally Efficient Learning with Local Simulators

Our result in Section 3 show that local simulator access facilitates sample-efficient learning in MDPs with low coverability, a challenging setting that was previously out of reach. However, our algorithm `SimGolf` is computationally-inefficient because it relies on global optimism, a drawback found in most prior work on RL with general function approximation (Jiang et al., 2017; Jin et al., 2021; Du et al., 2021). It remains an open question whether any form of global optimism can be implemented efficiently, and some variants have provable barriers to efficient implementation (Dann et al., 2018).

To address this drawback, in this section we present a new algorithm, `RVFS` (Recursive Value Function Search; Algorithm 3), which requires stronger versions of the coverability and realizability assumptions in Section 3, but is computationally efficient in the sense that it reduces to convex optimization over the state-value function class $\mathcal{V}$. `RVFS` makes use of a sophisticated recursive exploration scheme based on core-sets, sidestepping the need for global optimism, and can be applied to Exogenous Block MDPs with *weakly correlated* exogenous noise.

4.1 Function Approximation and Statistical Assumptions

To begin, we require the following strengthening of the coverability assumption in Assumption 3.2.

Assumption 4.1 (Pushforward coverability)**.** *The pushforward coverability coefficient* $C_{\mathsf{push}} > 0$ *is given by*

$$C_{\mathsf{push}} = \max_{h\in[H]} \inf_{\mu_h\in\Delta(\mathcal{X})} \sup_{(x_{h-1},a_{h-1},x_h)\in\mathcal{X}_{h-1}\times\mathcal{A}\times\mathcal{X}} \frac{T_{h-1}(x_h \mid x_{h-1},a_{h-1})}{\mu_h(x_h)}.$$

Pushforward coverability is inspired by the *pushforward concentrability* condition used in offline RL by Xie and Jiang (2021); Foster et al. (2022). Concrete examples include, (i) Block MDPs with latent space $\mathcal{S}$, which admit $C_{\mathsf{push}} \le |\mathcal{S}|$, (ii) Low-Rank MDPs in dimension d, which admit $C_{\mathsf{push}} \le d$ (Xie and Jiang, 2021), and (iii) Exogenous Block MDPs for which the exogenous noise process satisfies a *weak correlation condition* that we introduce in Section 4.4. Note that $C_{\mathsf{cov}} \le C_{\mathsf{push}}|\mathcal{A}|$, but the converse is not true in general.

Instead of state-action value function approximation as in `SimGolf`, in this section we make use of a state value function class $\mathcal{V} \subset (\mathcal{X}\times[H] \to [0,H])$, but require somewhat stronger representation conditions than in Section 3. We consider two complementary setups, which can be summarized briefly as follows:

- **Setup I:** Assumptions 4.2 and 4.3 ($V^\star/\pi^\star$-realizability) and Assumption 4.4 (Δ-gap) hold.
- **Setup II:** Assumption 4.5 (V^π-realizability) and Assumption 4.6 (π-realizability) hold.

We describe these assumptions in more detail below.

Function approximation setup I. First, instead of $Q^\star$-realizability, we consider the weaker $V^\star$-realizability (Jiang et al., 2017; Weisz et al., 2021a; Amortila et al., 2022).

Assumption 4.2 ($V^\star$-realizability)**.** *For all* $h \in [H]$*, we have* $V_h^\star \in \mathcal{V}_h$*.*

Under $V^\star$-realizability, our algorithm learns a near-optimal policy, but the policy is *non-executable* in the sense of Definition 2.1; this property is shared by prior work on local simulator access with value function realizability (Weisz et al., 2021a). To produce executable policies, we additionally require access to a policy class $\Pi \subset \Pi_{\mathsf{S}}$ containing $\pi^\star$; we define $\Pi_h = \{\pi_h \mid \pi \in \Pi\}$.

Assumption 4.3 ($\pi^\star$-realizability)**.** *The policy class* Π *contains the optimal policy* $\pi^\star$*.*

Note that $V^\star$-realizability (Assumption 4.2) and $\pi^\star$-realizability (Assumption 4.3) are both implied by $Q^\star$-realizability, and hence are weaker. However, we also assume the optimal Q-function admits constant suboptimality gap (this makes the representation conditions for **Setup I** incomparable to Assumption 3.1).

Assumption 4.4 (Δ-Gap)**.** *The optimal action* $\pi_h^\star(x)$ *is unique, and there exists* $\Delta > 0$ *such that for all* $h \in [H]$*,* $x \in \mathcal{X}$*, and* $a \in \mathcal{A} \setminus \{\pi_h^\star(x)\}$*,*

$$Q_h^\star(x, \pi_h^\star(x)) > Q_h^\star(x,a) + \Delta.$$

This condition has been used in a many prior works on computationally efficient RL with function approximation (Du et al., 2019b, 2020; Foster et al., 2020; Wang et al., 2021).

Function approximation setup II. As an alternative to assuming constant suboptimality gap, we also provide guarantees under the assumption that the value function class $\mathcal{V}$ satisfies *all-policy realizability* (Weisz et al., 2022; Yin et al., 2022; Weisz et al., 2023) in the sense that it contains the value functions V^π for all $\pi \in \Pi_{\mathsf{S}}$.

Assumption 4.5 (V^π-realizability)**.** *The value function class* $\mathcal{V} = \mathcal{V}_{1:H}$ *has* $V^\pi_h \in \mathcal{V}_h$ *for all* $\pi \in \Pi_{\mathsf{S}}$ *and* $h \in [H]$.

This assumption will be sufficient to learn a non-executable policy, but to learn executable policies we require an analogous strengthening of Assumption 4.5.

Assumption 4.6 (π-realizability)**.** *The policy class* Π *is such that for all* $\pi \in \Pi_{\mathsf{S}}$*, we have that* $x \mapsto \arg\max_{a\in\mathcal{A}} \mathcal{P}_h[V^\pi_{h+1}](\cdot, a) \in \Pi$.

This assumption, also known as *policy completeness* has been used by a number of prior works on computationally efficient RL (Bagnell et al., 2003; Misra et al., 2020). Assumptions 4.5 and 4.6 are both implied by the slightly simpler-to-state assumption of Q^π*-realizability* (Weisz et al., 2022; Yin et al., 2022; Weisz et al., 2023), which asserts access to a class $\mathcal{Q}$ that contains Q^π for all $\pi \in \Pi_{\mathsf{S}}$.

Remark 4.1. *Assumptions 4.5 and 4.6 can both be weakened to only require realizability for near-optimal policies; i.e.., we only need to assume that* $V^\pi_h \in \mathcal{V}$ *for all* $h \in [H]$ *and near-optimal policies* π *(instead of all policies). We will only use this weaker assumption in the analysis.*

4.2 Algorithm

For ease of exposition, we defer the full version of our algorithm, RVFS (Algorithm 3), to Appendix E and present a simplified version here (Algorithm 2). The algorithms are nearly identical, except that the simplified version assumes that certain quantities of interest (e.g., Bellman backups) can be computed exactly, while the full version (provably) approximates them from samples.

RVFS maintains a value function estimator $\widehat{V} = \widehat{V}_{1:H}$ that aims to approximate the optimal value function $V^\star_{1:H}$, as well as *core sets* $\mathcal{C}_1, \dots, \mathcal{C}_H$ of state-action pairs that are used to perform estimation and guide exploration. At a high level, RVFS alternates between (i) fitting the value function $\widehat{V}_h$ for a given layer $h \in [H]$ based on Monte-Carlo rollouts, and (ii) using the core-sets to test whether the current value function estimates $\widehat{V}_{h+1:H}$ remain accurate as the roll-in policy induced by $\widehat{V}_h$ changes.

In more detail, RVFS is based on recursion across the layers $h \in [H]$. When invoked for layer h with value function estimates $\widehat{V}_{h+1:H}$ and core-sets $\mathcal{C}_h, \dots, \mathcal{C}_H$, RVFS_h performs two steps:

1. For each state-action pair $(x_{h-1}, a_{h-1}) \in \mathcal{C}_h$,[6] the algorithm gathers N_{test} trajectories by rolling out from (x_{h-1}, a_{h-1}) with the greedy policy $\widehat{\pi}_\ell(x) \in \arg\max_{a\in\mathcal{A}} \mathcal{P}_\ell[\widehat{V}_{\ell+1}](x,a)$ that optimizes the estimated value function; in the full version of RVFS (see Algorithm 3), we estimate the bellman backup $\mathcal{P}_\ell[\widehat{V}_{\ell+1}](x,a)$ using the local simulator. For all states $x_{\ell-1} \in \{\boldsymbol{x}_h, \dots, \boldsymbol{x}_{H-1}\}$ encountered during this process, the algorithm checks whether $\left|\mathbb{E}\left[\widehat{V}_\ell(\boldsymbol{x}_\ell) - V^\star_\ell(\boldsymbol{x}_\ell) \mid \boldsymbol{x}_{\ell-1} = x_{\ell-1}, \boldsymbol{a}_{\ell-1} = a_{\ell-1}\right]\right| \lesssim \varepsilon$ for all $a_{\ell-1} \in \mathcal{A}$ using a test based on (implicitly maintained) confidence sets. If the test fails, this indicates that distribution shift has occurred, and the algorithm adds the pair $(x_{\ell-1}, a_{\ell-1})$ to the core-set $\mathcal{C}_\ell$ and recurses on layer ℓ via RVFS_ℓ.

2. If all tests above pass, this indicates that $\widehat{V}_{h+1}, \dots, \widehat{V}_H$ are accurate, and no distribution shift has occurred. In this case, the algorithm fits $\widehat{V}_h$ by collecting Monte-Carlo rollouts from all state-action pairs in the core-set $\mathcal{C}_h$ with $\widehat{\pi}_\ell(x) \in \arg\max_{a\in\mathcal{A}} \mathcal{P}_\ell[\widehat{V}_{\ell+1}](x,a)$ (cf. Line 16), and returns.

When the tests in Item 1 succeed for all layers $h \in [H]$, the algorithm terminates and returns the estimated value functions $\widehat{V}_{1:H}$; in this case, the greedy policy $\widehat{\pi}_\ell(x) \in \arg\max_{a\in\mathcal{A}} \mathcal{P}_\ell[\widehat{V}_{\ell+1}](x,a)$ is guaranteed to be near optimal. The full version of RVFS in Algorithm 3 uses local simulator access to estimate the Bellman backups $\mathcal{P}_h[\widehat{V}_{h+1}](x,a)$ for different state-action pairs (x,a) (see Eq. (15) of Algorithm 3). These backups are used

[6] Informally, $\mathcal{C}_h$ represents a collection of state-action pairs (x_{h-1}, a_{h-1}) at layer $h-1$ for which we want $\mathbb{E}\left[\left|\widehat{V}_h(\boldsymbol{x}_h) - V^\star_h(\boldsymbol{x}_h)\right| \mid \boldsymbol{x}_{h-1} = x_{h-1}, \boldsymbol{a}_{h-1} = a_{h-1}\right] \le \varepsilon$ for some small $\varepsilon > 0$.

Algorithm 2 RVFS_h: Recursive Value Function Search (Informal version of Algorithm 3)

```
1:  parameters: Value function class $\mathcal{V}$, suboptimality $\varepsilon \in (0,1)$, confidence $\delta \in (0,1)$.
2:    input:
        • Level $h \in \{0, \dots, H\}$.
        • Value function estimates $\widehat{V}_{h+1:H}$, confidence sets $\widehat{\mathcal{V}}_{h+1:H}$, state-action collections $\mathcal{C}_{h:H}$.
3:  Initialize parameters $M$, $N_{\mathsf{test}}$, $N_{\mathsf{reg}}$, $\varepsilon^2_{\mathsf{reg}}$, and $\beta$ (see Algorithm 3 for parameter settings).
    /* Test the fit for the estimated value functions $\widehat{V}_{h+1:H}$ at future layers. */
4:  for $(x_{h-1}, a_{h-1}) \in \mathcal{C}_h$ and $\ell = H, \dots, h+1$ do
5:      for $n = 1, \dots, N_{\mathsf{test}}$ do
6:          Draw $\boldsymbol{x}_h \sim T_{h-1}(\cdot \mid x_{h-1}, a_{h-1})$, then draw $\boldsymbol{x}_{\ell-1}$ by rolling out with $\widehat{\pi}_{h:H}$, where[5]
                $$\forall \tau \in [H], \quad \widehat{\pi}_\tau(\cdot) \in \arg\max_{a \in \mathcal{A}} \mathcal{P}_\tau[\widehat{V}_{\tau+1}](\cdot, a). \qquad (4)$$
7:          for $a_{\ell-1} \in \mathcal{A}$ do
                /* Test fit; if test fails, re-fit value functions $\widehat{V}_{h+1:\ell}$ up to layer $\ell$. */
8:              if $\sup_{f \in \widehat{\mathcal{V}}_\ell} |(\mathcal{P}_{\ell-1}[\widehat{V}_\ell] - \mathcal{P}_{\ell-1}[f_\ell])(\boldsymbol{x}_{\ell-1}, a_{\ell-1})| > \varepsilon + \varepsilon \cdot \beta$ then
9:                  $\mathcal{C}_\ell \leftarrow \mathcal{C}_\ell \cup \{(\boldsymbol{x}_{\ell-1}, a_{\ell-1})\}$.
10:                 for $\tau = \ell, \dots, h+1$ do
11:                     $(\widehat{V}_{\tau:H}, \widehat{\mathcal{V}}_{h:H}, \mathcal{C}_{\tau:H}) \leftarrow \mathsf{RVFS}_\tau(\widehat{V}_{\tau+1:H}, \widehat{\mathcal{V}}_{h+1:H}, \mathcal{C}_{\tau:H}; \mathcal{V}, \varepsilon, \delta)$.
12:                 go to line 4.
13: if $h = 0$ then return: $(\widehat{V}_{1:H}, \cdot, \cdot, \cdot, \cdot)$.
    /* Re-fit $\widehat{V}_h$ and build a new confidence set. */
14: for $(x_{h-1}, a_{h-1}) \in \mathcal{C}_h$ do          // $\mathbb{E}^{\widehat{\pi}_{h+1:H}}[\sum_{\ell=h}^H \boldsymbol{r}_\ell \mid \boldsymbol{x}_h, a]$ can be estimated using local simulator and roll-outs.
15:     Set $\mathcal{D}_h(x_{h-1}, a_{h-1}) \leftarrow \varnothing$. For $i = 1, \dots, N_{\mathsf{reg}}$, sample $\boldsymbol{x}_h \sim T_{h-1}(\cdot \mid x_{h-1}, a_{h-1})$ and update
        $\mathcal{D}_h(x_{h-1}, a_{h-1}) \leftarrow \mathcal{D}_h(x_{h-1}, a_{h-1}) \cup \{(\boldsymbol{x}_h, \mathbb{E}^{\widehat{\pi}_{h:H}}[\sum_{\ell=h}^H \boldsymbol{r}_\ell \mid \boldsymbol{x}_h])\}$.
16: Let $\widehat{V}_h := \arg\min_{f \in \widehat{\mathcal{V}}} \sum_{(x_{h-1}, a_{h-1}) \in \mathcal{C}_h} \sum_{(x_h, v_h) \in \mathcal{D}_h(x_{h-1}, a_{h-1})} (f(x_h) - v_h)^2$.
17: Compute value function confidence set:
        $$\widehat{\mathcal{V}}_h := \left\{ f \in \mathcal{V} \;\middle|\; \sum_{(x_{h-1}, a_{h-1}) \in \mathcal{C}_h} \frac{1}{N_{\mathsf{reg}}} \sum_{(x_h, -) \in \mathcal{D}_h(x_{h-1}, a_{h-1})} \left(\widehat{V}_h(x_h) - f(x_h)\right)^2 \le \varepsilon^2_{\mathsf{reg}} \right\}.$$
18: return $(\widehat{V}_{h:H}, \widehat{\mathcal{V}}_{h:H}, \mathcal{C}_{h:H})$.
```

to (i) compute actions of the greedy policy that maximizes $\widehat{V}_{1:H}$ via (e.g., Eq. (4)); (ii) generate trajectories by rolling out from state-action pairs in the core-sets (Line 6); and (iii) perform the test in Item 1 (Line 8).

RVFS is inspired by the DMQ algorithm of Du et al. (2019b); Wang et al. (2021), which was originally introduced in the context of online reinforcement learning with linearly realizable $Q^\star$. RVFS incorporates local simulator access (most critically, via core-set construction) to allow for more general *nonlinear* function approximation without restrictive statistical assumptions. Prior algorithms for RLLS have used core-sets of state-action pairs in a similar fashion (Li et al., 2021; Yin et al., 2022; Weisz et al., 2022), but in a way that is tailored to linear function approximation.

In what follows, we discuss various features of the algorithm in greater detail.

Bellman backup policies. Since RVFS works with state value functions instead of state-action value functions, we need a way to extract policies from the former. The most natural way to extract a policy from estimated value functions $\widehat{V}_{1:H} \in \mathcal{V}$ is as follows: for all $h \in [H]$, define $\widehat{\pi}_h(x) \in \arg\max_{a \in \mathcal{A}} \mathcal{P}_h[\widehat{V}_{h+1}](x, a)$. In reality, we do not have access to $\mathcal{P}_h[\widehat{V}_{h+1}](x, a)$ directly, so the full version of RVFS (Algorithm 3) estimates this quantity on the fly using the local simulator using the following scheme (Algorithm 5 in Appendix E): Given a state x, for each a, we sample K rewards $\boldsymbol{r}_h \sim R_h(x, a)$ and next-state transitions $\boldsymbol{x}_{h+1} \sim T_h(\cdot \mid x, a)$, then approximate $\mathcal{P}_h[\widehat{V}_{h+1}](x, a)$ by the empirical mean. We remark that the use of these

Bellman backup policies is actually crucial in the analysis for RVFS; even if we were to work with estimated state-action value functions $\widehat{Q}_{1:H}$ instead, our analysis would require executing the Bellman backup policies $\widehat{\pi}_h(x) \in \arg\max_{a\in\mathcal{A}} \mathcal{T}_h[\widehat{Q}_{h+1}](x,a)$ (instead of naively using $\widehat{\pi}_h(x) \in \arg\max_{a\in\mathcal{A}} \widehat{Q}_h(x,a)$).

Invoking the algorithm. The base invocation of RVFS takes the form

$$\widehat{V}_{1:H} \leftarrow \mathsf{RVFS}_0(\widehat{V}_{1:H} = \mathsf{arbitrary}, \widehat{\mathcal{V}}_{1:H} = \{\mathcal{V}_h\}_{h=1}^H, \mathcal{C}_{0:H} = \{\varnothing\}_{h=0}^H, ; \mathcal{V}, \varepsilon, \delta).$$

Whenever this call returns, the greedy policy induced by $\widehat{V}_{1:H}$ is guaranteed to be near-optimal. Naively, the approximate Bellman backup policy induced by $\widehat{V}_{1:H}$ (described above) is non-executable, and must be computed by invoking the local simulator. To provide an end-to-end guarantee to learn an executable policy, we give an outer-level algorithm, RVFS.bc (Algorithm 4, deferred to Appendix E for space), which invokes RVFS_0, then extracts an executable policy from $\widehat{V}_{1:H}$ using behavior cloning. Subsequent recursive calls to RVFS take the form

$$(\widehat{V}_{h:H}, \widehat{\mathcal{V}}_{h:H}, \mathcal{C}_{h:H}) \leftarrow \mathsf{RVFS}_h(\widehat{V}_{h+1:H}, \widehat{\mathcal{V}}_{h+1:H}, \mathcal{C}_{h:H}; \mathcal{V}, \varepsilon, \delta).$$

For such a call, the arguments are:

- $\widehat{V}_{h+1:H}$: Value function estimates for subsequent layers.
- $\widehat{\mathcal{V}}_{h+1:H}$: Value function confidence sets $\widehat{\mathcal{V}}_{h+1:H} \subset \mathcal{V}_{h+1:H}$, which are used in the test on Line 14 to quantify uncertainty on new state-action pairs and decide whether to expand the core-sets.
- $\mathcal{C}_{h:H}$: Core-sets for current and subsequent layers.

Importantly, the confidence sets $\widehat{\mathcal{V}}_{h+1:H}$ do not need to be explicitly maintained, and can be used implicitly whenever a *regression oracle* for the value function class is available (discussed below).

Oracle-efficiency. RVFS is *computationally efficient* in the sense that it reduces to convex optimization over the value function class $\mathcal{V}$. In particular, the only computationally intensive steps in the algorithm are (i) the regression step in Line 16, and (ii) the test in Line 8 involving the confidence set $\widehat{\mathcal{V}}_\ell$. For the latter, we do not explicitly need to maintain $\widehat{\mathcal{V}}_\ell$, as the optimization problem over this set in Line 8 (for the full version of RVFS in Algorithm 3) reduces to solving $\arg\max_{V\in\mathcal{V}}\{\pm\sum_{i=1}^n V(\widetilde{x}^{(i)}) \mid \sum_{i=1}^n (V(x^{(i)}) - y^{(i)})^2 \leq \beta^2\}$ for a dataset $\{(x^{(i)}, \widetilde{x}^{(i)}, y^{(i)})\}_{i=1}^n$. This is convex optimization problem in function space, and in particular can be implemented in a provably efficient fashion whenever $\mathcal{V}$ is linearly parameterized. We expect that the problem can also be reduced to a square loss regression by adapting the techniques in Krishnamurthy et al. (2017); Foster et al. (2018), but we do not pursue this here.

4.3 Main Result

We present the main guarantee for RVFS under the function approximation assumptions in Section 4.1.

Theorem 4.1 (Main guarantee for RVFS)**.** *Let $\varepsilon, \delta \in (0,1)$ be given, and suppose that Assumption 4.1 (pushforward coverability) holds with $C_{\mathsf{push}} > 0$. Further, suppose that one the following holds:*

- **Setup I:** *Assumptions 4.2 and 4.3 ($V^\star/\pi^\star$-realizability) and Assumption 4.4 (Δ-gap) hold, and $\varepsilon \leq 6H\cdot\Delta$.*
- **Setup II:** *Assumption 4.5 (V^π-realizability) and Assumption 4.6 (π-realizability) hold.*

Then, RVFS.bc$(\Pi, \mathcal{V}, \varepsilon, \delta)$ *(Algorithm 4) returns a policy $\widehat{\pi}_{1:H}$ such that $J(\pi^\star) - J(\widehat{\pi}_{1:H}) \leq 2\varepsilon$ with probability at least $1-\delta$, and has total sample complexity bounded by*

$$\widetilde{O}\left(C_{\mathsf{push}}^8 H^{23} A \cdot \varepsilon^{-13}\right).$$

Theorem 4.1 shows for the first time that sample- and computationally-efficient RL with local simulator access is possible under pushforward coverability. In particular, RVFS is the first computationally efficient algorithm for RL with local simulator access that supports nonlinear function approximation. The assumptions in Theorem 4.1, while stronger than those in Section 3, are not known to enable sample-efficient RL without simulator access. Nonetheless, understanding whether RVFS can be strengthened to support general coverability or weaker function approximation is an important open problem.

We mention in passing that RVFS can be slightly modified to recover other existing sample complexity guarantees for RL with linear function approximation and local simulator access (which do not require pushforward coverability), including linear-$Q^\star$ realizability with gap (Li et al., 2021) and Q^π-realizability (Yin et al., 2022); we leave a more general treatment for future work.

Connection to empirical algorithms. RVFS bears some similarity to Monte-Carlo Tree Search (MCTS) (Coulom, 2006; Kocsis and Szepesvári, 2006) and AlphaZero (Silver et al., 2018), which perform planning with local simulator. Informally, MCTS can be viewed as a form of breadth-first search over the state space (where each node represents a state at a given layer), and AlphaZero is a particular instantiation of a MCTS that leverages value function approximation (through a class $\mathcal{V}$ that aims to approximate $V^\star$) to accommodate stochastic environments and allow for generalization across states. Compared to RVFS, MCTS and AlphaZero perform exploration via simple bandit-style heuristics, and are not explicitly designed to handle *distribution shifts* that arise in settings where actions have long-term downstream effects. RVFS may be viewed as a provable counterpart that uses function approximation to address distribution shift in a principled fashion (in particular, through the use of confidence sets and the test in Line 14).[7]

RVFS also has some resemblance to the Go-Explore algorithm of Ecoffet et al. (2019, 2021) Like RVFS, Go-Explore makes use of core-sets of informative state-action pairs to guide exploration, and uses imitation learning to extract an executable policy after the exploration phase completes. However, Go-Explore uses an ad-hoc, domain specific approach to building the core set, and does not use function approximation in the exploration phase; such a strategy is unlikely to succeed in more challenging environments where the effective horizon is longer or less domain-specific information is available a-priori.

In light of these connections, we are optimistic that the techniques in RVFS—in particular, using value function approximation to guide systematic exploration—can help to inform the design of practical algorithms for learning and planning with local simulators.

4.4 Applying RVFS to Exogenous Block MDPs

We now apply RVFS to the Exogenous Block MDP (ExBMDP) model introduced in Section 3.3. ExBMDPs satisfy coverability (Assumption 3.2), but do not satisfy the pushforward coverability assumption (Assumption 4.1) required by RVFS in general. However, it turns out that ExBMDPs *do* satisfy pushforward coverability when the exogenous noise process is weakly correlated across time; we refer to this new statistical assumption as the *weak correlation condition.*

Assumption 4.7 (Weak correlation condition)**.** *For the underlying ExBMDP $\mathcal{M}$, there is a constant $C_{\mathsf{exo}} \geq 1$ such that for all $h \in [H-1]$ and $(\xi, \xi') \in \Xi_{h-1} \times \Xi_h$, we have*[8]

$$\mathbb{P}[\boldsymbol{\xi}_h = \xi, \boldsymbol{\xi}_{h+1} = \xi'] \leq C_{\mathsf{exo}} \cdot \mathbb{P}[\boldsymbol{\xi}_h = \xi] \cdot \mathbb{P}[\boldsymbol{\xi}_{h+1} = \xi'].$$

The weak correlation property asserts that the joint law for the exogenous noise variables $\boldsymbol{\xi}_h$ and $\boldsymbol{\xi}_{h+1}$ is at most a multiplicative factor $C_{\mathsf{exo}} \geq 1$ larger than the corresponding product distribution obtained by sampling $\boldsymbol{\xi}_h$ and $\boldsymbol{\xi}_{h+1}$ independently from their marginals. This setting strictly generalizes the (non-exogenous) Block MDP model (Krishnamurthy et al., 2016; Du et al., 2019a; Misra et al., 2019; Zhang et al., 2022; Mhammedi et al., 2023), by allowing for arbitrary stochastic dynamics for the endogenous state and an arbitrary emission process, but requires that temporal correlations in the exogenous noise decay over time.

[7] We note in passing that in the context of tree search, the pushforward coverability assumption (Assumption 4.1) may be viewed as the stochastic analogue of branching factor.

[8] Throughout this paper, when considering the law for the exogenous variables $\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_H$, we write $\mathbb{P}[\cdot]$ instead of $\mathbb{P}^\pi[\cdot]$ to emphasize that the law is independent of the agent's policy.

We show that under Assumption 4.7, pushforward coverability is satisfied with $C_{\mathsf{push}} \le C_{\mathsf{exo}} \cdot SA$ (Lemma J.3 in Appendix J.1). In addition, $V^\star$-realizability is implied by decoder realizability (Lemma 3.1). Thus, by applying Theorem 4.1 (**Setup I**), we conclude that RVFS efficiently learns a near-optimal policy for any weakly correlated ExBMDP for which the optimal value function has Δ-gap.

An improved algorithm for ExBMDPs: $\mathsf{RVFS}^{\mathsf{exo}}$. At first glance, removing the gap assumption for RVFS in ExBMDPs seems difficult: The V^π-realizability assumption required to invoke Theorem 4.1 (**Setup II**) is not satisfied by ExBMDPs, as decoder realizability only implies V^π realizability for *endogenous* policies π.[9] In spite of this, we now show that with a slight modification, RVFS can efficiently learn any weakly correlated ExBMDP under decoder realizability alone (without gap or V^π-realizability).

Our new variant of RVFS, $\mathsf{RVFS}^{\mathsf{exo}}$, is presented in Algorithm 6 (deferred to Appendix E for space). The algorithm is almost identical to RVFS (Algorithm 3), with the main difference being that we use an additional *randomized rounding* step to compute the policies $\widehat{\pi}_{1:H}$ from the learned value functions $\widehat{V}_{1:H}$. In particular, instead of directly defining the policies $\widehat{\pi}_{1:H}$ based on the bellman backups $\mathcal{P}_h[\widehat{V}_{h+1}]$ as in Eq. (15), $\mathsf{RVFS}^{\mathsf{exo}}$ targets a "rounded" version of the backup given by

$$\varepsilon \cdot \lceil \mathcal{P}_h[\widehat{V}_{h+1}](x,a)/\varepsilon + \boldsymbol{\zeta}_h \rceil, \tag{5}$$

where $\varepsilon \in (0,1)$ is a rounding parameter and $\boldsymbol{\zeta}_1, \dots, \boldsymbol{\zeta}_H$ are i.i.d. random variables sampled uniformly at random from the interval $[0, 1/2]$ (at the beginning of the algorithm's execution). Concretely, $\mathsf{RVFS}^{\mathsf{exo}}$ estimates the bellman backup $\mathcal{P}_h[\widehat{V}_{h+1}](x,a)$ in Eq. (5) using the local simulator (as in Eq. (15) of Algorithm 3), and defines its policies via

$$\widehat{\pi}_h(\cdot) \in \underset{a\in\mathcal{A}}{\arg\max} \lceil \mathcal{P}_h[\widehat{V}_{h+1}](\cdot,a)/\varepsilon + \boldsymbol{\zeta}_h \rceil. \tag{6}$$

This rounding scheme, which quantizes the Bellman backup into ε^{-1} bins with a random offset, is designed to emulate certain properties implied by the Δ-gap assumption (Assumption 4.4). Specifically, we show that with constant probability over the draw of $\boldsymbol{\zeta}_{1:H}$, the policy $\widehat{\pi}$ in (6) "snaps" on to an *endogenous* policy π. This means that for $\mathsf{RVFS}^{\mathsf{exo}}$ to succeed (with constant probability), it suffices to pass it a class $\mathcal{V}$ that realizes the value functions (V_h^π) for endogenous policies $\pi \in \Pi_{\mathsf{S}}$. Fortunately, such a function class can be constructed explicitly under decoder realizability (Assumption 3.3).

Lemma 4.1 (Efroni et al. (2022a)). *For the ExBMDP setting, under Assumption 3.3, the function class $\mathcal{V}_h \coloneqq \{x \mapsto f(\phi(x)) : f \in [0,H]^S, \phi \in \Phi\}$ is such that $V_h^\pi \in \mathcal{V}_h$ for all endogenous policies π. Furthermore, the policy class $\Pi_h \coloneqq \{\pi(\cdot) \in \arg\max_{a\in\mathcal{A}} f(\phi(\cdot),a) : f \in [0,H]^{S\times A}, \phi \in \Phi\}$ contains all endogenous policies.*

A small technical challenge with the scheme above is that it is only guaranteed to succeed with constant probability over the draw of the rounding parameters $\boldsymbol{\zeta}_1, \dots, \boldsymbol{\zeta}_H$. To address this, we provide an outer-level algorithm, $\mathsf{RVFS}^{\mathsf{exo}}.\mathsf{bc}$ (Algorithm 7, deferred to Appendix E for space), which performs confidence boosting by invoking $\mathsf{RVFS}^{\mathsf{exo}}$ multiple times independently, and extracts a high-quality executable policy using behavior cloning.

Main result. We now state the main guarantee for $\mathsf{RVFS}^{\mathsf{exo}}$ (the proof is in Appendix J).

Theorem 4.2 (Main guarantee of $\mathsf{RVFS}^{\mathsf{exo}}$ for EXBMDPs). *Consider the ExBMDP setting. Suppose the decoder class Φ satisfies Assumption 3.3, and that Assumption 4.7 holds with $C_{\mathsf{exo}} > 0$. Let $\varepsilon, \delta \in (0,1)$ be given, and let $\mathcal{V}_h$ and Π_h be as in Lemma 4.1. Then $\mathsf{RVFS}^{\mathsf{exo}}.\mathsf{bc}(\Pi, \mathcal{V}_{1:H}, \varepsilon, \zeta_{1:H}, \delta)$ (Algorithm 7) produces a policy $\widehat{\pi}_{1:H}$ such that $J(\pi^\star) - J(\widehat{\pi}_{1:H}) \le \varepsilon$, and has total sample complexity*

$$\widetilde{O}\left(C_{\mathsf{exo}}^8 S^8 H^{36} A^9 \cdot \varepsilon^{-26}\right).$$

This result shows for the first time that sample- and computationally-efficient learning is possible for ExBMDPs beyond deterministic or factored settings (Efroni et al., 2022b,a).

[9] We say that a policy π is *endogenous* if it does not depend on exogenous noise, in the sense that $\pi(\boldsymbol{x}_h)$ is a measurable function of $\phi^\star(\boldsymbol{x}_h)$.

We mention in passing that our use of randomized rounding to emulate certain consequences of the Δ-gap assumption leverages the fact that ExBMDPs have a finite number of (endogenous) latent states. It is unclear if this technique can be used when the (latent) state space is large or infinite.

5 Conclusion and Open Problems

Our results show that online RL with local simulator access can enable powerful sample complexity guarantees for learning with general value function approximation. Interesting open problems and directions for future research include:

- Is it true that value function realizability and coverability are *not sufficient* for sample-efficient learning in the online RL model? Combined with our results, this would imply a new separation between RLLS and online RL. More broadly, it would be interesting to develop a unified understanding for precisely when local simulator access can lead to statistical benefits over fully online RL, and to characterize the statistical complexity for this framework.
- The RVFS algorithm is computationally efficient, but requires stronger statistical assumptions than our inefficient algorithms. Closing this gap with an efficient algorithm is an important but challenging open problem.

In addition, we are excited to explore the empirical performance of exploration schemes inspired by RVFS.

Acknowledgements

Part of this work was done while ZM was at MIT. ZM and AR acknowledge support from the ONR through awards N00014-20-1-2336 and N00014-20-1-2394, and ARO through award W911NF-21-1-0328.

References

Alekh Agarwal, Daniel Hsu, Satyen Kale, John Langford, Lihong Li, and Robert Schapire. Taming the monster: A fast and simple algorithm for contextual bandits. In *International Conference on Machine Learning*, pages 1638–1646, 2014.

Ilge Akkaya, Marcin Andrychowicz, Maciek Chociej, Mateusz Litwin, Bob McGrew, Arthur Petron, Alex Paino, Matthias Plappert, Glenn Powell, Raphael Ribas, et al. Solving rubik's cube with a robot hand. *arXiv preprint arXiv:1910.07113*, 2019.

Philip Amortila, Nan Jiang, Dhruv Madeka, and Dean P Foster. A few expert queries suffices for sample-efficient rl with resets and linear value approximation. *Advances in Neural Information Processing Systems*, 35:29637–29648, 2022.

Philip Amortila, Dylan J Foster, Nan Jiang, Ayush Sekhari, and Tengyang Xie. Harnessing density ratios for online reinforcement learning. *International Conference on Learning Representations (ICLR)*, 2024a.

Philip Amortila, Dylan J Foster, and Akshay Krishnamurthy. Scalable online exploration via coverability. *arXiv preprint arXiv:2403.06571*, 2024b.

Szilárd Aradi. Survey of deep reinforcement learning for motion planning of autonomous vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 23(2):740–759, 2020.

Adrià Puigdomènech Badia, Bilal Piot, Steven Kapturowski, Pablo Sprechmann, Alex Vitvitskyi, Zhaohan Daniel Guo, and Charles Blundell. Agent57: Outperforming the atari human benchmark. In *International conference on machine learning*, pages 507–517. PMLR, 2020.

James Bagnell, Sham M Kakade, Jeff Schneider, and Andrew Ng. Policy search by dynamic programming. *Advances in neural information processing systems*, 16, 2003.

Marc Bellemare, Joel Veness, and Michael Bowling. Investigating contingency awareness using atari 2600 games. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 26, pages 864–871, 2012.

Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, et al. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*, 2016.

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

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

Rémi Coulom. Efficient selectivity and backup operators in monte-carlo tree search. In *International conference on computers and games*, pages 72–83. Springer, 2006.

Christoph Dann, Nan Jiang, Akshay Krishnamurthy, Alekh Agarwal, John Langford, and Robert E Schapire. On oracle-efficient PAC RL with rich observations. In *Advances in neural information processing systems*, pages 1422–1432, 2018.

Simon Du, Akshay Krishnamurthy, Nan Jiang, Alekh Agarwal, Miroslav Dudik, and John Langford. Provably efficient RL with rich observations via latent state decoding. In *International Conference on Machine Learning*, pages 1665–1674. PMLR, 2019a.

Simon S Du, Yuping Luo, Ruosong Wang, and Hanrui Zhang. Provably efficient Q-learning with function approximation via distribution shift error checking oracle. In *Advances in Neural Information Processing Systems*, pages 8060–8070, 2019b.

Simon S Du, Sham M Kakade, Ruosong Wang, and Lin F Yang. Is a good representation sufficient for sample efficient reinforcement learning? In *International Conference on Learning Representations*, 2020.

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

Adrien Ecoffet, Joost Huizinga, Joel Lehman, Kenneth O Stanley, and Jeff Clune. Go-explore: a new approach for hard-exploration problems. *arXiv preprint arXiv:1901.10995*, 2019.

Adrien Ecoffet, Joost Huizinga, Joel Lehman, Kenneth O Stanley, and Jeff Clune. First return, then explore. *Nature*, 590(7847):580–586, 2021.

Yonathan Efroni, Dylan J Foster, Dipendra Misra, Akshay Krishnamurthy, and John Langford. Sample-efficient reinforcement learning in the presence of exogenous information. In *Conference on Learning Theory*, pages 5062–5127. PMLR, 2022a.

Yonathan Efroni, Dipendra Misra, Akshay Krishnamurthy, Alekh Agarwal, and John Langford. Provably filtering exogenous distractors using multistep inverse dynamics. In *International Conference on Learning Representations*, 2022b.

Dylan J Foster, Alekh Agarwal, Miroslav Dudík, Haipeng Luo, and Robert E. Schapire. Practical contextual bandits with regression oracles. *International Conference on Machine Learning*, 2018.

Dylan J Foster, Alexander Rakhlin, David Simchi-Levi, and Yunzong Xu. Instance-dependent complexity of contextual bandits and reinforcement learning: A disagreement-based perspective. *Conference on Learning Theory (COLT)*, 2020.

Dylan J Foster, Sham M Kakade, Jian Qian, and Alexander Rakhlin. The statistical complexity of interactive decision making. *arXiv preprint arXiv:2112.13487*, 2021.

Dylan J Foster, Akshay Krishnamurthy, David Simchi-Levi, and Yunzong Xu. Offline reinforcement learning: Fundamental barriers for value function approximation. In *Conference on Learning Theory*, pages 3489–3489. PMLR, 2022.

Zhaohan Guo, Shantanu Thakoor, Miruna Pîslar, Bernardo Avila Pires, Florent Altché, Corentin Tallec, Alaa Saade, Daniele Calandriello, Jean-Bastien Grill, Yunhao Tang, et al. Byol-explore: Exploration by bootstrapped prediction. *Advances in neural information processing systems*, 35:31855–31870, 2022.

Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963.

Audrey Huang, Jinglin Chen, and Nan Jiang. Reinforcement learning in low-rank mdps with density features. *International Conference on Machine Learning (ICML)*, 2023.

Riashat Islam, Manan Tomar, Alex Lamb, Yonathan Efroni, Hongyu Zang, Aniket Didolkar, Dipendra Misra, Xin Li, Harm van Seijen, Remi Tachet des Combes, et al. Agent-controller representations: Principled offline rl with rich exogenous information. *International Conference on Machine Learning (ICML)*, 2023.

Nan Jiang, Akshay Krishnamurthy, Alekh Agarwal, John Langford, and Robert E Schapire. Contextual decision processes with low Bellman rank are PAC-learnable. In *International Conference on Machine Learning*, pages 1704–1713, 2017.

Chi Jin, Qinghua Liu, and Sobhan Miryoosefi. Bellman eluder dimension: New rich classes of RL problems, and sample-efficient algorithms. *Neural Information Processing Systems*, 2021.

Sham Machandranath Kakade. *On the sample complexity of reinforcement learning.* University of London, University College London (United Kingdom), 2003.

Michael Kearns and Satinder Singh. Finite-sample convergence rates for q-learning and indirect algorithms. *Advances in neural information processing systems*, 11, 1998.

Levente Kocsis and Csaba Szepesvári. Bandit based monte-carlo planning. In *European conference on machine learning*, pages 282–293. Springer, 2006.

Akshay Krishnamurthy, Alekh Agarwal, and John Langford. PAC reinforcement learning with rich observations. In *Advances in Neural Information Processing Systems*, pages 1840–1848, 2016.

Akshay Krishnamurthy, Alekh Agarwal, Tzu-Kuo Huang, Hal Daumé III, and John Langford. Active learning for cost-sensitive classification. In *International Conference on Machine Learning*, pages 1915–1924, 2017.

Alex Lamb, Riashat Islam, Yonathan Efroni, Aniket Rajiv Didolkar, Dipendra Misra, Dylan J Foster, Lekan P Molu, Rajan Chari, Akshay Krishnamurthy, and John Langford. Guaranteed discovery of control-endogenous latent states with multi-step inverse models. *Transactions on Machine Learning Research*, 2023.

Tor Lattimore, Csaba Szepesvari, and Gellert Weisz. Learning with good feature representations in bandits and in rl with a generative model. In *International Conference on Machine Learning*, pages 5662–5670. PMLR, 2020.

Gen Li, Yuxin Chen, Yuejie Chi, Yuantao Gu, and Yuting Wei. Sample-efficient reinforcement learning is feasible for linearly realizable mdps with limited revisiting. *Advances in Neural Information Processing Systems*, 34:16671–16685, 2021.

Qinghua Liu, Praneeth Netrapalli, Csaba Szepesvari, and Chi Jin. Optimistic mle: A generic model-based algorithm for partially observable sequential decision making. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, pages 363–376, 2023.

Zakaria Mhammedi, Dylan J Foster, and Alexander Rakhlin. Representation learning with multi-step inverse kinematics: An efficient and optimal approach to rich-observation rl. *International Conference on Machine Learning (ICML)*, 2023.

Dipendra Misra, Mikael Henaff, Akshay Krishnamurthy, and John Langford. Kinematic state abstraction and provably efficient rich-observation reinforcement learning. *arXiv preprint arXiv:1911.05815*, 2019.

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

Mohammed Abu Qassem, Iyad Abuhadrous, and Hatem Elaydi. Modeling and simulation of 5 dof educational robot arm. In *2010 2nd International Conference on Advanced Computer Control*, volume 5, pages 569–574. IEEE, 2010.

Stéphane Ross and Drew Bagnell. Efficient reductions for imitation learning. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 661–668. JMLR Workshop and Conference Proceedings, 2010.

Tim Salimans and Richard Chen. Learning montezuma's revenge from a single demonstration. *arXiv preprint arXiv:1812.03381*, 2018.

Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, et al. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588(7839):604–609, 2020.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pages 1889–1897. PMLR, 2015.

Aaron Sidford, Mengdi Wang, Xian Wu, Lin Yang, and Yinyu Ye. Near-optimal time and sample complexities for solving markov decision processes with a generative model. *Advances in Neural Information Processing Systems*, 31, 2018.

David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 362(6419):1140–1144, 2018.

Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, et al. Deepmind control suite. *arXiv preprint arXiv:1801.00690*, 2018.

Arash Tavakoli, Vitaly Levdik, Riashat Islam, Christopher M Smith, and Petar Kormushev. Exploring restart distributions. *arXiv preprint arXiv:1811.11298*, 2018.

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

Yuanhao Wang, Ruosong Wang, and Sham M Kakade. An exponential lower bound for linearly-realizable MDPs with constant suboptimality gap. *Neural Information Processing Systems (NeurIPS)*, 2021.

Gellert Weisz, Philip Amortila, Barnabás Janzer, Yasin Abbasi-Yadkori, Nan Jiang, and Csaba Szepesvári. On query-efficient planning in mdps under linear realizability of the optimal state-value function. In *Conference on Learning Theory*, pages 4355–4385. PMLR, 2021a.

Gellért Weisz, Philip Amortila, and Csaba Szepesvári. Exponential lower bounds for planning in MDPs with linearly-realizable optimal action-value functions. In *Algorithmic Learning Theory*, pages 1237–1264. PMLR, 2021b.

Gellért Weisz, András György, Tadashi Kozuno, and Csaba Szepesvári. Confident approximate policy iteration for efficient local planning in q^π-realizable mdps. *Advances in Neural Information Processing Systems*, 35: 25547–25559, 2022.

Gellért Weisz, András György, and Csaba Szepesvári. Online rl in linearly q^π-realizable mdps is as easy as in linear mdps if you learn what to ignore. *arXiv preprint arXiv:2310.07811*, 2023.

Zheng Wen and Benjamin Van Roy. Efficient reinforcement learning in deterministic systems with value function generalization. *Mathematics of Operations Research*, 42(3):762–782, 2017.

Tengyang Xie and Nan Jiang. Batch value-function approximation with only realizability. In *International Conference on Machine Learning*, pages 11404–11413. PMLR, 2021.

Tengyang Xie, Dylan J Foster, Yu Bai, Nan Jiang, and Sham M Kakade. The role of coverage in online reinforcement learning. In *The Eleventh International Conference on Learning Representations*, 2023.

Lin Yang and Mengdi Wang. Sample-optimal parametric Q-learning using linearly additive features. In *International Conference on Machine Learning*, pages 6995–7004. PMLR, 2019.

Dong Yin, Botao Hao, Yasin Abbasi-Yadkori, Nevena Lazić, and Csaba Szepesvári. Efficient local planning with linear function approximation. In *International Conference on Algorithmic Learning Theory*, pages 1165–1192. PMLR, 2022.

Dong Yin, Sridhar Thiagarajan, Nevena Lazic, Nived Rajaraman, Botao Hao, and Csaba Szepesvari. Sample efficient deep reinforcement learning via local planning. *arXiv preprint arXiv:2301.12579*, 2023.

Andrea Zanette, Alessandro Lazaric, Mykel Kochenderfer, and Emma Brunskill. Learning near optimal policies with low inherent bellman error. In *International Conference on Machine Learning*, pages 10978–10989. PMLR, 2020.

Xuezhou Zhang, Yuda Song, Masatoshi Uehara, Mengdi Wang, Alekh Agarwal, and Wen Sun. Efficient reinforcement learning in block mdps: A model-free representation learning approach. In *International Conference on Machine Learning*, pages 26517–26547. PMLR, 2022.

Contents of Appendix

A Additional Related Work

Local simulators: Theoretical research. RL with local simulators has received extensive interest in the context of linear function approximation. Most notably, Weisz et al. (2021a) show that reinforcement learning with linear $V^\star$ is tractable with local simulator access, and Li et al. (2021) show that RL with linear $Q^\star$ and a state-action gap is tractable; online RL is known to be intractable under the same assumptions (Weisz et al., 2021a; Wang et al., 2021). Amortila et al. (2022) show that the gap assumption can be removed if a small number of expert queries are available. Also of note are the works of Yin et al. (2022); Weisz et al. (2022), which give computationally efficient algorithms under linear Q^π-realizability for all π; this setting is known to be tractable in the online RL model (Weisz et al., 2023), but computationally efficient algorithms are currently only known for RLLS.

Global simulators—in which the agent can query arbitrary state-action pairs and observe next state transitions—have also received theoretical investigation, but like local simulators, results are largely restricted to tabular reinforcement learning and linear models (Kearns and Singh, 1998; Kakade, 2003; Sidford et al., 2018; Du et al., 2020; Yang and Wang, 2019; Lattimore et al., 2020).

Local simulators: Empirical research. The Go-Explore algorithm (Ecoffet et al., 2019, 2021) uses local simulator access to achieve state-of-the-art performance for the Atari games Montezuma's Revenge and Pitfall—both notoriously difficult games that require systematic exploration. To the best of our knowledge, the performance of Go-Explore on these tasks has yet to be matched by online reinforcement learning; the performing agents (Badia et al., 2020; Guo et al., 2022) are roughly a factor of four worse in terms of cumulative reward. Interestingly, like RVFS, Go-Explore makes use of core sets of informative state-action pairs to guide exploration. However, Go-Explore uses an ad-hoc, domain specific approach to designing the core set, and does not use function approximation to drive exploration.

Recent work of Yin et al. (2023) provides an empirical framework for online RL with local planning that can take advantage of deep neural function approximation, and is inspired by the theoretical works in Weisz et al. (2021a); Li et al. (2021); Yin et al. (2022); Weisz et al. (2022). This approach does not have provable guarantees, but achieves super-human performance at Montezuma's Revenge.

Other notable empirical works that incorporate local simulator access, as highlighted by Yin et al. (2023), include Schulman et al. (2015); Salimans and Chen (2018); Tavakoli et al. (2018).

Planning. RL with local simulator access is a convenient abstraction for the problem of *planning*: Given a known (e.g., learned) model, compute an optimal policy. Planning with a learned model is an important task in theory (Foster et al., 2021; Liu et al., 2023) and practice (e.g., MuZero (Schrittwieser et al., 2020)). Since the model is known, computing an optimal policy is a purely computational problem, not a statistical problem. Nonetheless, for planning problems in large state spaces, where enumerating over all states is undesirable, algorithms for online RL with local simulator access can be directly applied, treating the model as if it were the environment the agent is interacting with. Here, any computationally efficient RLLS algorithm immediately yields an efficient algorithm for planning.

Empirically, Monte-Carlo Tree Search (Coulom, 2006; Kocsis and Szepesvári, 2006) is a successful paradigm for planning, acting as a key component in AlphaGo (Silver et al., 2016) and AlphaZero (Silver et al., 2018).[10] Viewed as a planning algorithm, a potential advantage of RVFS is that it is well suited to stochastic environments, and provides a principled way to use estimated (neural) value function estimates to guide exploration.

Coverability. Xie et al. (2023) introduced coverability as a structural parameter for online reinforcement leanring, inspired by connections between online and offline RL. Existing guarantees for the online RL framework based on coverability require either Bellman completeness (Xie et al., 2023), model-based realizability (Amortila et al., 2024b), or weight function realizability (Amortila et al., 2024a,b)), and it is not currently known whether value function realizability is sufficient in this framework.

[10] Compare to our work, a small difference is that these works are not concerned with producing executable policies, c.f. Definition 2.1.

Exogenous Block MDPs. Our results in Section 3.3 (Corollary 3.1) show that general Exogenous Block MDPs are learnable with local simulator access. Prior work, on learning EXBMDPs in the online RL model requires additional assumptions:

- *Deterministic ExBMDP* (Efroni et al., 2022b). In this setting, the latent transition distribution T^{endo} is assumed to be deterministic. In this case, it suffices to learn *open-loop* policies (i.e., policies that play a deterministic sequence of actions). This avoids compounding errors due to learning imperfect decoders that depend on the exogenous noise, making this setting much less challenging than the general ExBMDP setting.
- *Factored ExMDP* (Efroni et al., 2022a). This is an ExBMDP setting with a restrictive structure in which the observation is a d-dimensional vector and the latent state is a k-dimensional subset of the observed coordinates. This structure prevents the setting from subsuming the basic (non-exogenous) Block MDP framework, and makes it possible to learn decoders that act only on the endogenous state, preventing compounding errors.
- *Bellman completeness*. Xie et al. (2023) observed that ExBMDPs admit low coverability, but their algorithm requires Bellman completeness, which is not satisfied by ExBMDPs (see Efroni et al. (2022b); Islam et al. (2023)).

B Helper Lemmas

This section of the appendix contains supporting lemmas used within the proofs of our main results.

B.1 Concentration and Probability

Lemma B.1. *Let $\delta \in (0,1)$ and $H \geq 1$ be given. If a sequence of events $\mathcal{E}_1, \ldots, \mathcal{E}_H$ satisfies $\mathbb{P}[\mathcal{E}_h \mid \mathcal{E}_1, \ldots, \mathcal{E}_{h-1}] \geq 1 - \delta/H$ for all $h \in [H]$, then*

$$\mathbb{P}[\mathcal{E}_{1:H}] \geq 1 - \delta.$$

Proof of Lemma B.1. By the chain rule, we have

$$\mathbb{P}[\mathcal{E}_{1:H}] = \prod_{h \in [H]} \mathbb{P}[\mathcal{E}_h \mid \mathcal{E}_1, \ldots, \mathcal{E}_{h-1}] \geq \prod_{h \in [H]} (1 - \delta/H) = (1 - \delta/H)^H \geq 1 - \delta.$$

□

We make use of the following version of Freedman's inequality, due to Agarwal et al. (2014, Lemma 9):

Lemma B.2. *Let $R > 0$ be given and let $\boldsymbol{w}_1, \ldots \boldsymbol{w}_n$ be a sequence of real-valued random variables adapted to filtration $\mathcal{H}_1, \cdots, \mathcal{H}_n$. Assume that for all $t \in [n]$, $\boldsymbol{w}_i \leq R$ and $\mathbb{E}[\boldsymbol{w}_i \mid \mathcal{H}_{i-1}] = 0$. Define $\boldsymbol{S}_n \coloneqq \sum_{t=1}^n \boldsymbol{w}_i$ and $V_n \coloneqq \sum_{t=1}^n \mathbb{E}[\boldsymbol{w}_i^2 \mid \mathcal{H}_{i-1}]$. Then, for any $\delta \in (0,1)$ and $\lambda \in [0, 1/R]$, with probability at least $1 - \delta$,*

$$\boldsymbol{S}_n \leq \lambda V_n + \log(1/\delta)/\lambda.$$

We will also use the following lemma, which is a standard consequence of Freedman's inequality.

Lemma B.3 (e.g., Foster et al. (2021)). *Let $(\boldsymbol{w}_t)_{t \leq T}$ be a sequence of random variables adapted to a filtration $(\mathcal{H}_t)_{t \leq T}$. If $0 \leq \boldsymbol{w}_t \leq R$ almost surely, then with probability at least $1 - \delta$,*

$$\sum_{t=1}^T \boldsymbol{w}_t \leq \frac{3}{2} \sum_{t=1}^T \mathbb{E}_{t-1}[\boldsymbol{w}_t] + 4R \log(2\delta^{-1}),$$

and

$$\sum_{t=1}^T \mathbb{E}_{t-1}[\boldsymbol{w}_t] \leq 2 \sum_{t=1}^T \boldsymbol{w}_t + 8R \log(2\delta^{-1}).$$

B.2 Regression

Using Lemmas B.2 and B.3, we obtain the following concentration lemma, which will be used to prove guarantees for square loss regression within our algorithms.

Lemma B.4. *Let $B > 0$ and $n \in \mathbb{N}$ be given, and let $\mathcal{Y}$ be an abstract set. Further, let $\mathcal{Q} \subseteq \{g : \mathcal{Y} \to [0, B]\}$ be a finite function class and $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n$ be a sequence of random variables in $\mathcal{Y}$ adapted to filtration a $\mathcal{H}_1, \cdots, \mathcal{H}_n$. Then, for any $\delta \in (0,1)$, with probability at least $1 - \delta$, we have*

$$\forall g \in \mathcal{Q}, \quad \frac{1}{2}\|g\|^2 - 2B^2 \log(2|\mathcal{Q}|/\delta) \leq \|g\|_n^2 \leq 2\|g\|^2 + 2B^2 \log(2|\mathcal{Q}|/\delta),$$

where $\|g\|^2 \coloneqq \sum_{i \in [n]} \mathbb{E}[g(\boldsymbol{y}_i)^2 \mid \mathcal{H}_{i-1}]$ and $\|g\|_n^2 \coloneqq \sum_{i=1}^n g(\boldsymbol{y}_i)^2$.

Proof of Lemma B.4. Fix $g \in \mathcal{Q}$. Applying Lemma B.2 with $\boldsymbol{w}_i = g(\boldsymbol{y}_i)^2 - \mathbb{E}[g(\boldsymbol{y}_i)^2 \mid \mathcal{H}_{i-1}]$, for all $i \in [n]$, and $(R, \lambda) = (B^2, 1/B^2)$, we get that with probability at least $1 - \delta/(2|\mathcal{Q}|)$:

$$\|g\|_n^2 - \|g\|^2 \leq \lambda B^2 \|g\|^2 + \log(2|\mathcal{Q}|/\delta)/\lambda.$$

By substituting $\lambda = B^{-2}$ and rearranging, we get

$$\|g\|_n^2 \leq 2\|g\|^2 + B^2 \log(2|\mathcal{Q}|/\delta). \tag{7}$$

Similarly, applying Lemma B.2 with $\boldsymbol{w}_i = \mathbb{E}[g(\boldsymbol{y}_i)^2 \mid \mathcal{H}_{i-1}] - g(\boldsymbol{y}_i)^2$, for all $i \in [n]$, and $(R, \lambda) = (B^2, 1/(2B^2))$, we get that with probability at least $1 - \delta/(2|\mathcal{Q}|)$:

$$\|g\|^2 - \|g\|_n^2 \leq \lambda B^2 \|g\|^2 + \log(2|\mathcal{Q}|/\delta)/\lambda.$$

By substituting $\lambda = 2^{-1}B^{-2}$ and rearranging, we get

$$\|g\|_n^2 \geq \frac{1}{2}\|g\|^2 - 2B^2 \log(2|\mathcal{Q}|/\delta).$$

Combining this with (7) and the union bound, we get the desired result. □

With this lemma, we now prove the following key result for square loss regression.

Lemma B.5 (Generic regression guarantee)**.** *Let $B > 0$ and $n \in \mathbb{N}$ be given and $\mathcal{Y}$ be an abstract set. Further, let $\mathcal{F} \subseteq \{f : \mathcal{Y} \to [0, B]\}$ be a finite function class, and suppose that there is a function $f_\star \in \mathcal{F}$ and a sequence of random variables $(\boldsymbol{y}_1, \boldsymbol{x}_1), \dots, (\boldsymbol{y}_n, \boldsymbol{x}_n) \in \mathcal{Y} \times \mathbb{R}$ such that for all $i \in [n]$:*

- $\boldsymbol{x}_i = f_\star(\boldsymbol{y}_i) + \boldsymbol{\varepsilon}_i + \boldsymbol{b}_i$*;*
- $|\boldsymbol{b}_i| \leq \xi$*;*
- $\boldsymbol{\varepsilon}_i \in [-B, B]$*; and*
- $\mathbb{E}[\boldsymbol{\varepsilon}_i \mid \mathfrak{F}_i] = 0$*, where* $\mathfrak{F}_i \coloneqq \sigma(\boldsymbol{y}_{1:i}, \boldsymbol{\varepsilon}_{1:i-1}, \boldsymbol{x}_{1:i-1}, \boldsymbol{b}_{1:i-1})$*.*

Then, for $\widehat{f} \in \arg\min_{f \in \mathcal{F}} \sum_{i=1}^n (f(\boldsymbol{y}_i) - \boldsymbol{x}_i)^2$ and any $\delta \in (0, 1)$, with probability at least $1 - \delta/2$,

$$\|\widehat{f} - f_\star\|_n^2 \leq 4B^2 \log(2|\mathcal{F}|/\delta) + 4B \sum_{i=1}^n |\boldsymbol{b}_i|,$$

where $\|\widehat{f} - f_\star\|_n^2 \coloneqq \sum_{i=1}^n (\widehat{f}(\boldsymbol{y}_i) - f^\star(\boldsymbol{y}_i))^2$.

Proof of Lemma B.5. Fix $\delta \in (0, 1)$ and let $\widehat{L}_n(f) \coloneqq \sum_{i=1}^n (f(\boldsymbol{y}_i) - \boldsymbol{x}_i)^2$, for $f \in \mathcal{F}$, and note that since $\widehat{f} \in \arg\min_{f \in \mathcal{F}} \widehat{L}_n(f)$, we have

$$0 \geq \widehat{L}_n(\widehat{f}) - \widehat{L}_n(f_\star) = \nabla \widehat{L}_n(f_\star)[\widehat{f} - f_\star] + \|\widehat{f} - f_\star\|_n^2,$$

where ∇ denotes directional derivative. Rearranging, we get that

$$\begin{aligned}
\|\widehat{f} - f_\star\|_n^2 &\leq -2\nabla \widehat{L}_n(f_\star)[\widehat{f} - f_\star] - \|\widehat{f} - f_\star\|_n^2, \\
&= 4\sum_{i=1}^n (\boldsymbol{x}_i - f_\star(\boldsymbol{y}_i))(\widehat{f}(\boldsymbol{y}_i) - f_\star(\boldsymbol{y}_i)) - \|\widehat{f} - f_\star\|_n^2, \\
&\leq 4\sum_{i=1}^n (\boldsymbol{\varepsilon}_i + \boldsymbol{b}_i)(\widehat{f}(\boldsymbol{y}_i) - f_\star(\boldsymbol{y}_i)) - \|\widehat{f} - f_\star\|_n^2, \\
&\leq \underbrace{4\sum_{i=1}^n \boldsymbol{\varepsilon}_i \cdot (\widehat{f}(\boldsymbol{y}_i) - f_\star(\boldsymbol{y}_i)) - \|\widehat{f} - f_\star\|_n^2}_{\text{I}} + \underbrace{4\sum_{i=1}^n \boldsymbol{b}_i \cdot (\widehat{f}(\boldsymbol{y}_i) - f_\star(\boldsymbol{y}_i))}_{\text{II}}.
\end{aligned} \tag{8}$$

Bounding Term I. To bound Term I, we apply Lemma B.2 with $\boldsymbol{w}_i = \boldsymbol{\varepsilon}_i \cdot (\widehat{f}(\boldsymbol{y}_i) - f_\star(\boldsymbol{y}_i))$, $R = B^2$, $\lambda = 1/(8B^2)$, and $\mathcal{H}_i = \mathfrak{F}_{i+1}^-$, and use

1. the union bound over $f \in \mathcal{F}$; and

2. the facts that $\mathbb{E}[\boldsymbol{y}_i \mid \mathfrak{F}_i^-] = \boldsymbol{y}_i$ and $\mathbb{E}[\boldsymbol{\varepsilon_i} \mid \mathfrak{F}_i^-] = 0$,

to get that with probability at least $1 - \delta/2$,

$$4\sum_{i=1}^{n} \boldsymbol{\varepsilon}_i \cdot (\widehat{f}(\boldsymbol{y}_i) - f_\star(\boldsymbol{y}_i)) \le \|\widehat{f} - f_\star\|_n^2 + 4B^2 \log(2|\mathcal{F}|/\delta).$$

By rearranging, we get that with probability at least $1 - \delta/2$,

$$\text{Term I} \le 4B^2 \log(2|\mathcal{F}|/\delta).$$

Bounding Term II. We now bound the second term in (8). For this, note that since $\|\widehat{f} - f_\star\|_\infty \le B$, we have

$$\text{Term II} \le 4B\sum_{i=1}^{n} |\boldsymbol{b}_i|.$$

This completes the proof. □

B.3 Reinforcement Learning

Lemma B.6 (Performance Difference Lemma (e.g., Kakade (2003))). *For any two policies $\widehat{\pi}, \pi \in \Pi_{\mathsf{S}}$ and $t \in [H]$, we have*

$$\mathbb{E}^\pi\left[V_t^\pi(\boldsymbol{x}_t) - V_t^{\widehat{\pi}}(\boldsymbol{x}_t)\right] = \mathbb{E}^\pi\left[\sum_{h=t}^{H} Q_h^{\widehat{\pi}}(\boldsymbol{x}_h, \boldsymbol{\pi}_h(\boldsymbol{x}_h)) - Q_h^{\widehat{\pi}}(\boldsymbol{x}_h, \widehat{\boldsymbol{\pi}}_h(\boldsymbol{x}_h))\right].$$

In particular, applying this for $t = 1$ gives

$$J(\pi) - J(\widehat{\pi}) = \mathbb{E}^\pi\left[\sum_{h=1}^{H} Q_t^{\widehat{\pi}}(\boldsymbol{x}_h, \boldsymbol{\pi}_h(\boldsymbol{x}_h)) - Q_h^{\widehat{\pi}}(\boldsymbol{x}_h, \widehat{\boldsymbol{\pi}}_h(\boldsymbol{x}_h))\right].$$

Lemma B.7 (Potential lemma (Xie et al., 2023)). *Fix $h \in [H]$. Suppose we have a sequence of functions $g^{(1)}, \dots, g^{(T)} \in [0, B]$ and policies $\pi^{(1)}, \dots, \pi^{(T)}$ such that*

$$\forall t \in [T], \quad \sum_{i<t} \mathbb{E}^{\pi^{(i)}}\left[(g^{(t)}(\boldsymbol{x}_h))^2\right] \le \beta^2$$

for some $\beta \ge 0$. Then under Assumption 4.1, we have

$$\sum_{t=1}^{T} \mathbb{E}^{\pi^{(t)}}[g^{(t)}(\boldsymbol{x}_h)] \le 2\sqrt{\beta^2 C_{\mathsf{push}} T \log(2T)} + 2BC_{\mathsf{push}},$$

and consequently

$$\min_{t\in[T]} \mathbb{E}^{\pi^{(t)}}[g^{(t)}(\boldsymbol{x}_h)] \le 2\sqrt{\frac{\beta^2 C_{\mathsf{push}} \log(2T)}{T}} + \frac{2BC_{\mathsf{push}}}{T}.$$

Proof of Lemma B.8. See proof of (Xie et al., 2023, Theorem 1). □

The following result is a variant of the coverability-based potential argument given in Xie et al. (2023).

Lemma B.8 (Pushforward coverability potential lemma). *Fix $h \in [H]$. Suppose we have a sequence of functions $g^{(1)}, \dots, g^{(T)} \in [0, B]$ and state-action pairs $(x^{(1)}, a^{(1)}), \dots, (x^{(T)}, a^{(T)})$ such that*

$$\forall t \in [T], \quad \sum_{i<t} \mathbb{E}\left[(g^{(t)}(\boldsymbol{x}_h))^2 \mid \boldsymbol{x}_{h-1} = x^{(i)}, \boldsymbol{a}_{h-1} = a^{(i)}\right] \le \beta^2$$

for some $\beta \geq 0$. Then under Assumption 4.1, we have

$$\sum_{t=1}^{T} \mathbb{E}[g^{(t)}(\boldsymbol{x}_h) \mid \boldsymbol{x}_{h-1} = x^{(t)}, \boldsymbol{a}_{h-1} = a^{(t)}] \leq 2\sqrt{\beta^2 C_{\mathsf{push}} T \log(2T)} + 2BC_{\mathsf{push}},$$

and consequently

$$\min_{t\in[T]} \mathbb{E}[g^{(t)}(\boldsymbol{x}_h) \mid \boldsymbol{x}_{h-1} = x^{(t)}, \boldsymbol{a}_{h-1} = a^{(t)}] \leq 2\sqrt{\frac{\beta^2 C_{\mathsf{push}} \log(2T)}{T}} + \frac{2BC_{\mathsf{push}}}{T}.$$

Proof of Lemma B.8. Define $d_h^{(t)}(x) \coloneqq \mathbb{P}[\boldsymbol{x}_h = x \mid \boldsymbol{x}_{h-1} = x^{(t)}, \boldsymbol{a}_{h-1} = a^{(t)}]$, and let $\widetilde{d}_h^{(t)} \coloneqq \sum_{i<t} d^{(i)}$. Let

$$\tau_h(x) \coloneqq \min\{t \mid \widetilde{d}_h^{(t)}(x) \geq C_{\mathsf{push}} \cdot \mu_h(x)\}.$$

We have

$$\begin{aligned}
&\sum_{t=1}^{T} \mathbb{E}[g^{(t)}(\boldsymbol{x}_h) \mid \boldsymbol{x}_{h-1} = x^{(t)}, \boldsymbol{a}_{h-1} = a^{(t)}] \\
&= \sum_{t=1}^{T}\sum_{x\in\mathcal{X}} d_h^{(t)}(x) g^{(t)}(x) \\
&\leq \sum_{t=1}^{T}\sum_{x\in\mathcal{X}} d_h^{(t)}(x) g^{(t)}(x)\mathbb{I}\{t \geq \tau_h(x)\} + B\sum_{t=1}^{T}\sum_{x\in\mathcal{X}} d_h^{(t)}(x)\mathbb{I}\{t < \tau_h(x)\}.
\end{aligned}$$

From the definition of pushforward coverability, we can bound

$$\sum_{t=1}^{T}\sum_{x\in\mathcal{X}} d_h^{(t)}(x)\mathbb{I}\{t < \tau_h(x)\} = \sum_{x\in\mathcal{X}} \widetilde{d}_h^{(\tau_h(x))}(x) \leq 2C_{\mathsf{push}} \sum_{x\in\mathcal{X}} \mu_h(x) = 2C_{\mathsf{push}}.$$

For the other term, we bound

$$\begin{aligned}
&\sum_{t=1}^{T}\sum_{x\in\mathcal{X}} d_h^{(t)}(x) g^{(t)}(x)\mathbb{I}\{t \geq \tau_h(x)\} \\
&\leq \left(\sum_{t=1}^{T}\sum_{x\in\mathcal{X}} \frac{(d_h^{(t)}(x))^2}{\widetilde{d}_h^{(t)}(x)}\mathbb{I}\{t \geq \tau_h(x)\}\right)^{1/2} \cdot \left(\sum_{t=1}^{T}\sum_{x\in\mathcal{X}} \widetilde{d}_h^{(t)}(x)(g_h^{(t)}(x))^2\right)^{1/2} \\
&= \left(\sum_{t=1}^{T}\sum_{x\in\mathcal{X}} \frac{(d_h^{(t)}(x))^2}{\widetilde{d}_h^{(t)}(x)}\mathbb{I}\{t \geq \tau_h(x)\}\right)^{1/2} \cdot \left(\sum_{t=1}^{T}\sum_{i<t} \mathbb{E}\big[(g^{(t)}(\boldsymbol{x}_h))^2 \mid \boldsymbol{x}_{h-1} = x^{(i)}, \boldsymbol{a}_{h-1} = a^{(i)}\big]\right)^{1/2} \\
&\leq \left(\sum_{t=1}^{T}\sum_{x\in\mathcal{X}} \frac{(d_h^{(t)}(x))^2}{\widetilde{d}_h^{(t)}(x)}\mathbb{I}\{t \geq \tau_h(x)\}\right)^{1/2} \cdot \sqrt{\beta^2 T}.
\end{aligned}$$

Finally, we have

$$\begin{aligned}
\sum_{t=1}^{T}\sum_{x\in\mathcal{X}} \frac{(d_h^{(t)}(x))^2}{\widetilde{d}_h^{(t)}(x)}\mathbb{I}\{t \geq \tau_h(x)\} &\leq 2\sum_{t=1}^{T}\sum_{x\in\mathcal{X}} \frac{(d_h^{(t)}(x))^2}{\widetilde{d}_h^{(t)}(x) + C_{\mathsf{push}}\mu_h(x)} \\
&\leq 2C_{\mathsf{push}}\sum_{t=1}^{T}\sum_{x\in\mathcal{X}} \mu_h(x)\frac{d_h^{(t)}(x)}{\widetilde{d}_h^{(t)}(x) + C_{\mathsf{push}}\mu_h(x)} \\
&= 2C_{\mathsf{push}}\sum_{x\in\mathcal{X}} \mu_h(x)\sum_{t=1}^{T}\frac{d_h^{(t)}(x)}{\widetilde{d}_h^{(t)}(x) + C_{\mathsf{push}}\mu_h(x)} \\
&= 4C_{\mathsf{push}}\log(T+1),
\end{aligned}$$

where the last line uses Lemma 4 of Xie et al. (2023). □

Part I

Proofs for **SimGolf** (Section 3)

As described in Section 3.1, the main difference between SimGolf and GOLF lies in the construction of the confidence sets. The most important new step in the proof of Theorem 3.1 is to show that the local simulator-based confidence set construction in Line 10 is valid in the sense that the property Eq. (2) holds with high probability. From here, the sample complexity bound follows by adapting the change-of-measure argument based on coverability from Xie et al. (2023).

To this end, this part of the appendix is organized as follows. We first state and prove technical lemmas concerning realizability (Lemma 3.1) and the confidence set construction (Lemma C.1 and Lemma C.2) in Appendix C. Then, in Appendix D, we prove Theorem 3.1 as a consequence.

C Preliminary Lemmas for Proof of Theorem 3.1

For this section, we define

$$\ell_h^{(t)}(g) \coloneqq \left(g_h(\boldsymbol{x}_h^{(t)}, \boldsymbol{a}_h^{(t)}) - \frac{1}{K}\sum_{k=1}^{K}\left(\boldsymbol{r}_h^{(t,k)} + \max_{a\in\mathcal{A}} g_{h+1}(\boldsymbol{x}_{h+1}^{(t,k)}, a)\right)\right)^2$$

and

$$\bar{\ell}_h^{(t)}(g) \coloneqq \mathbb{E}^{\pi^{(t)}}\Big[(g_h(\boldsymbol{x}_h, \boldsymbol{a}_h) - \mathcal{T}_h[g_{h+1}](\boldsymbol{x}_h, \boldsymbol{a}_h))^2\Big],$$

where $(\boldsymbol{x}_h^{(t)}, \boldsymbol{a}_h^{(t)}, \boldsymbol{r}_h^{(t,k)}, \boldsymbol{x}_{h+1}^{(t,k)})$ are as in Algorithm 1.

Lemma C.1. *With probability at least $1-\delta$, for all $h \in [H]$, $t \in [N]$, and $g \in \mathcal{Q}$,*

$$\sum_{i<t} \ell_h^{(i)}(g) \le 3\sum_{i<t}\mathbb{E}^{\pi^{(i)}}\Big[(g_h(\boldsymbol{x}_h, \boldsymbol{a}_h) - \mathcal{T}_h[g_{h+1}](\boldsymbol{x}_h, \boldsymbol{a}_h))^2\Big] + \frac{8N}{K} + 16\log(2HN|\mathcal{Q}|\delta^{-1}),$$

and

$$\sum_{i<t}\mathbb{E}^{\pi^{(i)}}\Big[(g_h(\boldsymbol{x}_h, \boldsymbol{a}_h) - \mathcal{T}_h[g_{h+1}](\boldsymbol{x}_h, \boldsymbol{a}_h))^2\Big] \le 4\sum_{i<t}\ell_h^{(i)}(g) + \frac{8N}{K} + 64\log(2HN|\mathcal{Q}|\delta^{-1}).$$

Proof of Lemma C.1. Let $t \in [N]$ and $h \in [H]$ be fixed. Let us denote $\boldsymbol{z}_h^{(t)} = \big\{(\boldsymbol{r}_h^{(t,k)}, \boldsymbol{x}_{h+1}^{(t,k)})\big\}_{k\in[K]}$. Define a filtration

$$\mathcal{H}^{(t)} = \sigma(\boldsymbol{\tau}^{(1)}, \boldsymbol{z}_1^{(1)}, \ldots, \boldsymbol{z}_H^{(1)}, \ldots, \boldsymbol{\tau}^{(t)}, \boldsymbol{z}_1^{(t)}, \ldots, \boldsymbol{z}_H^{(t)}),$$

where $\boldsymbol{\tau}^{(i)}$ is the trajectory generated in the ith iteration of Algorithm 1 (see Line 7). Fix $g \in \mathcal{Q}$. Observe that $\ell_h^{(i)}(g) \in [0,4]$, so Lemma B.3 ensures that with probability at least $1-\delta$,

$$\sum_{i<t}\ell_h^{(i)}(g) \le \frac{3}{2}\sum_{i<t}\mathbb{E}\big[\ell_h^{(i)}(g) \mid \mathcal{H}^{(i-1)}\big] + 16\log(2\delta^{-1}),$$

and

$$\sum_{i<t}\mathbb{E}\big[\ell_h^{(i)}(g) \mid \mathcal{H}^{(i-1)}\big] \le 2\sum_{i<t}\ell_h^{(i)}(g) + 32\log(2\delta^{-1}). \tag{9}$$

By the AM-GM inequality, for all $i < t$, we can bound

$$\begin{aligned}
&\mathbb{E}\big[\ell_h^{(i)}(g) \mid \mathcal{H}^{(i-1)}\big] \\
&= \mathbb{E}\left[\left(g_h(\boldsymbol{x}_h^{(i)}, \boldsymbol{a}_h^{(i)}) - \frac{1}{K}\sum_{k=1}^{K}\left(\boldsymbol{r}_h^{(i,k)} + \max_{a\in\mathcal{A}} g_{h+1}(\boldsymbol{x}_{h+1}^{(i,k)}, a)\right)\right)^2 \mid \mathcal{H}^{(i-1)}\right] \\
&\leq 2\,\mathbb{E}\Big[\big(g_h(\boldsymbol{x}_h^{(i)}, \boldsymbol{a}_h^{(i)}) - \mathcal{T}_h[g_{h+1}](\boldsymbol{x}_h^{(i)}, \boldsymbol{a}_h^{(i)})\big)^2 \mid \mathcal{H}^{(i-1)}\Big] \\
&\quad + 2\,\mathbb{E}\left[\left(\mathcal{T}_h[g_{h+1}](\boldsymbol{x}_h^{(i)}, \boldsymbol{a}_h^{(i)}) - \frac{1}{K}\sum_{k=1}^{K}\left(\boldsymbol{r}_h^{(i,k)} + \max_{a\in\mathcal{A}} g_{h+1}(\boldsymbol{x}_{h+1}^{(i,k)}, a)\right)\right)^2 \mid \mathcal{H}^{(i-1)}\right].
\end{aligned}$$

and

$$\begin{aligned}
&\mathbb{E}\big[\ell_h^{(i)}(g) \mid \mathcal{H}^{(i-1)}\big] \\
&\geq \frac{1}{2}\,\mathbb{E}\Big[\big(g_h(\boldsymbol{x}_h^{(i)}, \boldsymbol{a}_h^{(i)}) - \mathcal{T}_h[g_{h+1}](\boldsymbol{x}_h^{(i)}, \boldsymbol{a}_h^{(i)})\big)^2 \mid \mathcal{H}^{(i-1)}\Big] \\
&\quad - \mathbb{E}\left[\left(\mathcal{T}_h[g_{h+1}](\boldsymbol{x}_h^{(i)}, \boldsymbol{a}_h^{(i)}) - \frac{1}{K}\sum_{k=1}^{K}\left(\boldsymbol{r}_h^{(i,k)} + \max_{a\in\mathcal{A}} g_{h+1}(\boldsymbol{x}_{h+1}^{(i,k)}, a)\right)\right)^2 \mid \mathcal{H}^{(i-1)}\right].
\end{aligned}$$

We have

$$\mathbb{E}\Big[\big(g_h(\boldsymbol{x}_h^{(i)}, \boldsymbol{a}_h^{(i)}) - \mathcal{T}_h[g_{h+1}](\boldsymbol{x}_h^{(i)}, \boldsymbol{a}_h^{(i)})\big)^2 \mid \mathcal{H}^{(i-1)}\Big] = \mathbb{E}^{\pi^{(i)}}\Big[\big(g_h(\boldsymbol{x}_h, \boldsymbol{a}_h) - \mathcal{T}_h[g_{h+1}](\boldsymbol{x}_h, \boldsymbol{a}_h)\big)^2\Big]$$

and

$$\begin{aligned}
&\mathbb{E}\left[\left(\mathcal{T}_h[g_{h+1}](\boldsymbol{x}_h^{(i)}, \boldsymbol{a}_h^{(i)}) - \frac{1}{K}\sum_{k=1}^{K}\left(\boldsymbol{r}_h^{(i,k)} + \max_{a\in\mathcal{A}} g_{h+1}(\boldsymbol{x}_{h+1}^{(i,k)}, a)\right)\right)^2 \mid \mathcal{H}^{(i-1)}\right] \\
&= \mathbb{E}\left[\mathbb{E}\left[\left(\mathcal{T}_h[g_{h+1}](\boldsymbol{x}_h, \boldsymbol{a}_h) - \frac{1}{K}\sum_{k=1}^{K}\left(\boldsymbol{r}_h^{(i,k)} + \max_{a\in\mathcal{A}} g_{h+1}(\boldsymbol{x}_{h+1}^{(i,k)}, a)\right)\right)^2 \mid \boldsymbol{x}_h = \boldsymbol{x}_h^{(i)}, \boldsymbol{a}_h = \boldsymbol{a}_h^{(i)}\right] \mid \mathcal{H}^{(i-1)}\right].
\end{aligned}$$

Since

$$\mathbb{E}\left[\boldsymbol{r}_h^{(i,k)} + \max_{a\in\mathcal{A}} g_{h+1}(\boldsymbol{x}_{h+1}^{(i,k)}, a) \mid \boldsymbol{x}_h = \boldsymbol{x}_h^{(i)}, \boldsymbol{a}_h = \boldsymbol{a}_h^{(i)}\right] = \mathcal{T}_h[g_{h+1}](\boldsymbol{x}_h^{(i)}, \boldsymbol{a}_h^{(i)})$$

and $\big\{(\boldsymbol{r}_h^{(i,k)}, \boldsymbol{x}_{h+1}^{(i,k)})\big\}_{k\in[K]}$ are i.i.d. conditioned on $(\boldsymbol{x}_h, \boldsymbol{a}_h) = (\boldsymbol{x}_h^{(i)}, \boldsymbol{a}_h^{(i)})$, we have,

$$\begin{aligned}
&\mathbb{E}\left[\left(\mathcal{T}_h[g_{h+1}](\boldsymbol{x}_h, \boldsymbol{a}_h) - \frac{1}{K}\sum_{k=1}^{K}\left(\boldsymbol{r}_h^{(i,k)} + \max_{a\in\mathcal{A}} g_{h+1}(\boldsymbol{x}_{h+1}^{(i,k)}, a)\right)\right)^2 \mid \boldsymbol{x}_h = \boldsymbol{x}_h^{(i)}, \boldsymbol{a}_h = \boldsymbol{a}_h^{(i)}\right] \\
&= \frac{1}{K}\,\mathbb{E}\left[\left(\mathcal{T}_h[g_{h+1}](\boldsymbol{x}_h, \boldsymbol{a}_h) - \left(\boldsymbol{r}_h^{(i,k)} + \max_{a\in\mathcal{A}} g_{h+1}(\boldsymbol{x}_h^{(i,k)}, a)\right)\right)^2 \mid \boldsymbol{x}_h = \boldsymbol{x}_h^{(i)}, \boldsymbol{a}_h = \boldsymbol{a}_h^{(i)}\right] \\
&\leq \frac{4}{K},
\end{aligned}$$

so that

$$\mathbb{E}\left[\left(\mathcal{T}_h[g_{h+1}](\boldsymbol{x}_h^{(i)}, \boldsymbol{a}_h^{(i)}) - \frac{1}{K}\sum_{k=1}^{K}\left(\boldsymbol{r}_h^{(i,k)} + \max_{a\in\mathcal{A}} g_{h+1}(\boldsymbol{x}_{h+1}^{(i,k)}, a)\right)\right)^2 \mid \mathcal{H}^{(i-1)}\right] \leq \frac{4}{K}.$$

Combining these bounds with (9) and rearranging thus gives

$$\sum_{i<t} \ell_h^{(i)}(g) \leq 3\sum_{i<t} \mathbb{E}^{\pi^{(i)}}\Big[\big(g_h(\boldsymbol{x}_h, \boldsymbol{a}_h) - \mathcal{T}_h[g_{h+1}](\boldsymbol{x}_h, \boldsymbol{a}_h)\big)^2\Big] + \frac{8N}{K} + 16\log(2\delta^{-1}),$$

and

$$\sum_{i<t} \mathbb{E}^{\pi^{(i)}}\Big[\big(g_h(\boldsymbol{x}_h, \boldsymbol{a}_h) - \mathcal{T}_h[g_{h+1}](\boldsymbol{x}_h, \boldsymbol{a}_h)\big)^2\Big] \leq 4\sum_{i<t} \ell_h^{(i)}(g) + \frac{8N}{K} + 64\log(2\delta^{-1}).$$

Taking a union bound yields the result. □

Lemma C.2. *Define* $\beta_{\mathsf{stat}} = 16\log(2HN|\mathcal{Q}|\delta^{-1})$. *Suppose we set* $K \geq \frac{8N}{\beta_{\mathsf{stat}}}$ *and* $\beta \geq 2\beta_{\mathsf{stat}}$. *Then with probability at least* $1-\delta$, *for all* $t \in [N]$ *and* $h \in \mathcal{H}$*:*

- $Q^\star \in \mathcal{Q}^{(t)}$.
- *All* $g \in \mathcal{Q}^{(t)}$ *satisfy*

$$\sum_{i<t} \mathbb{E}^{\pi^{(i)}}\Big[\big(g_h(\boldsymbol{x}_h, \boldsymbol{a}_h) - \mathcal{T}_h[g_{h+1}](\boldsymbol{x}_h, \boldsymbol{a}_h)\big)^2\Big] \leq 9\beta.$$

Proof of Lemma C.2. Condition on the event in Lemma C.1. For any fixed $t \in [N]$ and $h \in [H]$, we have that

$$\begin{aligned}\sum_{i<t} \ell_h^{(i)}(Q^\star) &\leq 3\sum_{i<t} \mathbb{E}^{\pi^{(i)}}\Big[\big(Q_h^\star(\boldsymbol{x}_h, \boldsymbol{a}_h) - \mathcal{T}_h[Q_{h+1}^\star](\boldsymbol{x}_h, \boldsymbol{a}_h)\big)^2\Big] + \frac{8N}{K} + 16\log(2HN|\mathcal{Q}|\delta^{-1}) \\ &\leq \frac{8N}{K} + 16\log(2HN|\mathcal{Q}|\delta^{-1}) \leq 2\beta_{\mathsf{stat}},\end{aligned}$$

where the first inequality uses that $Q_h^\star = \mathcal{T}_h[Q_{h+1}^\star]$ and the second inequality uses our choice for K. It follows that $Q^\star \in \mathcal{Q}^{(t)}$ as long as $\beta \geq 2\beta_{\mathsf{stat}}$.

To prove the second claim, we note that for all $g \in \mathcal{Q}^{(t)}$, by construction,

$$\begin{aligned}\sum_{i<t} \mathbb{E}^{\pi^{(i)}}\Big[\big(g_h(\boldsymbol{x}_h) - \mathcal{T}_h[g_{h+1}](\boldsymbol{x}_h, \boldsymbol{a}_h)\big)^2\Big] &\leq 4\sum_{i<t} \ell_h^{(i)}(g) + \frac{8N}{K} + 64\log(2HN|\mathcal{Q}|\delta^{-1}) \\ &\leq 4\sum_{i<t} \ell_h^{(i)}(g) + 5\beta_{\mathsf{stat}} \leq 9\beta.\end{aligned}$$

□

D Proof of Theorem 3.1

Proof of Theorem 3.1. From Lemma C.2, the parameter setting in the theorem statement ensures that with probability at least $1-\delta$, for all $t \in [2\,..\,N]$, $Q^\star \in \mathcal{Q}^{(t)}$, and all $g \in \mathcal{Q}^{(t)}$ satisfy

$$\sum_{i<t} \mathbb{E}^{\pi^{(i)}}\Big[\big(g_h(\boldsymbol{x}_h) - \mathcal{T}_h[g_{h+1}](\boldsymbol{x}_h, \boldsymbol{a}_h)\big)^2\Big] \leq 9\beta. \tag{10}$$

for all h. Let us condition on this event going forward. First, note that since $Q^\star \in \mathcal{Q}^{(t)}$ for all $t \in [2\,..\,N]$, we have that

$$J(\pi^\star) \leq \mathbb{E}\Big[\max_{a\in\mathcal{A}} Q_1^\star(\boldsymbol{x}_1, a)\Big] \leq \sup_{g\in\mathcal{Q}^{(t)}} \mathbb{E}\Big[\max_{a\in\mathcal{A}} g_1(\boldsymbol{x}_1, a)\Big]. \tag{11}$$

On the other hand, we have $g^{(t)} \in \arg\max_{g\in\mathcal{Q}^{(t)}} \sum_{s<t} \max_{a\in\mathcal{A}} g_1(\boldsymbol{x}_1^{(s)}, a)$, and so since $\boldsymbol{x}_1^{(1)}, \boldsymbol{x}_1^{(2)}, \ldots$ are i.i.d. and any $g \in \mathcal{Q}^{(t)}$ take values in $[0, H]$, we have that by Hoeffding's inequality, there is an event $\mathcal{E}$ of probability at least $1-\delta$ under which

$$\forall t \in [2\,..\,N], \forall g \in \mathcal{Q}, \quad \left|\mathbb{E}\Big[\max_{a\in\mathcal{A}} g_1(\boldsymbol{x}_1, a)\Big] - \frac{1}{t-1}\sum_{s<t} \max_{a\in\mathcal{A}} g_1(\boldsymbol{x}_1^{(s)}, a)\right| \leq \sqrt{(t-1)^{-1}\log(2N|\mathcal{Q}|/\delta)}. \tag{12}$$

This implies that under $\mathcal{E}$, we have

$$\begin{aligned}\forall t \in [2\,..\,N], \quad \sup_{g\in\mathcal{Q}^{(t)}} \mathbb{E}\left[\max_{a\in\mathcal{A}} g_1(\boldsymbol{x}_1, a)\right] &\leq \sup_{g\in\mathcal{Q}^{(t)}} \frac{1}{t-1}\sum_{s<t}\max_{a\in\mathcal{A}} g_1(\boldsymbol{x}_1^{(s)}, a) + \sqrt{(t-1)^{-1}\log(2N|\mathcal{Q}|/\delta)}, \\ &= \frac{1}{t-1}\sum_{s<t}\max_{a\in\mathcal{A}} g_1^{(t)}(\boldsymbol{x}_1^{(s)}, a) + \sqrt{(t-1)^{-1}\log(2N|\mathcal{Q}|/\delta)}, \\ &\leq \mathbb{E}\left[\max_{a\in\mathcal{A}} g_1^{(t)}(\boldsymbol{x}_1, a)\right] + 2\sqrt{(t-1)^{-1}\log(2N|\mathcal{Q}|/\delta)}, \end{aligned} \tag{13}$$

where in the last inequality we have used (12) with $f = g^{(t)}$. Thus, summing (13) for $t = 2, \dots N$ and using (11) gives that under $\mathcal{E}$:

$$\sum_{t=2}^{N} J(\pi^\star) \leq \sum_{t=2}^{N} \mathbb{E}[g_1^{(t)}(\boldsymbol{x}_1, \boldsymbol{a}_1)] + 4\sqrt{N\log(2N|\mathcal{Q}|/\delta)},$$

and so since $J(\pi^\star) \leq H$,

$$\sum_{t=1}^{N} J(\pi^\star) \leq \sum_{t=1}^{N} \mathbb{E}[g_1^{(t)}(\boldsymbol{x}_1, \boldsymbol{a}_1)] + 4\sqrt{N\log(2N|\mathcal{Q}|/\delta)} + H. \tag{14}$$

On the other hand, using that $g_{H+1}^{(t)} \equiv 0$, we get

$$\begin{aligned} &\sum_{t=1}^{N}(\mathbb{E}\left[g_1^{(t)}(\boldsymbol{x}_1, \boldsymbol{a}_1)\right] - J(\pi^{(t)})) \\ &\leq \sum_{t=1}^{N}\sum_{h=1}^{H} \mathbb{E}^{\pi^{(t)}}\left[g_h^{(t)}(\boldsymbol{x}_h, \boldsymbol{a}_h) - \boldsymbol{r}_h - \max_{a\in\mathcal{A}} g_{h+1}^{(t)}(\boldsymbol{x}_{h+1}, a)\right], \\ &= \sum_{t=1}^{N}\sum_{h=1}^{H} \mathbb{E}^{\pi^{(t)}}\left[g_h^{(t)}(\boldsymbol{x}_h, \boldsymbol{a}_h) - \mathbb{E}\left[\boldsymbol{r}_h + \max_{a\in\mathcal{A}} g_{h+1}^{(t)}(\boldsymbol{x}_{h+1}, a) \mid \boldsymbol{x}_h, \boldsymbol{a}_h\right]\right], \quad \text{(law of total expectation)} \\ &= \sum_{t=1}^{N}\sum_{h=1}^{H} \mathbb{E}^{\pi^{(t)}}\left[g_h^{(t)}(\boldsymbol{x}_h, \boldsymbol{a}_h) - \mathcal{T}_h[g_{h+1}^{(t)}](\boldsymbol{x}_h, \boldsymbol{a}_h)\right]. \end{aligned}$$

and so, by the potential lemma (Lemma B.7) and (10), we have

$$\leq 6H\sqrt{C_{\mathsf{cov}}\beta N\log(2N)} + 2H^2 C_{\mathsf{cov}}.$$

Combining this with (14), we obtain that with probability at least $1 - 2\delta$,

$$\sum_{t=1}^{N}(J(\pi^\star) - J(\pi^{(t)})) \leq 6H\sqrt{C_{\mathsf{cov}}\beta N\log(2N)} + 4\sqrt{N\log(2N|\mathcal{Q}|/\delta)} + 3H^2 C_{\mathsf{cov}}.$$

It follows that if $N = \widetilde{O}(H^2 C_{\mathsf{cov}}\beta/\varepsilon^2)$, then the policy

$$\widehat{\pi} \in \texttt{unif}(\pi^{(1)}, \dots, \pi^{(N)})$$

returned by `SimGolf` satisfies, with probability at least $1 - \delta$:

$$J(\pi^\star) - \mathbb{E}[J(\widehat{\pi})] \leq \varepsilon.$$

Sample complexity. We now bound the number of episodes. Note that that within an iteration t of `SimGolf`, the local simulator is called KH times to update the confidence set, where $K \leq N/\log(2HN|\mathcal{Q}|/\delta))$. Consequently, the total sample complexity is bounded by

$$HNK \leq \widetilde{O}(H^5 C_{\mathsf{cov}}^2 \log(|\mathcal{Q}|/\delta)/\varepsilon^4).$$

□

Part II

Proofs for RVFS (Section 4)

E Full Version of RVFS

Algorithm 3 displays the full version of RVFS. Algorithm 4 contains an "outer-level" wrapper for RVFS, RVFS.bc, which invokes RVFS and extracts an executable policy with imitation learning, and Algorithm 5 contains the subroutine used within Algorithm 3 to approximate Bellman backups for value functions using local simulator access. Additionally, we display the variant of RVFS for Exogenous Block MDPs, described in Section 4.4, in Algorithms 6 and 7. Before diving into the proof, we first describe how the full version of the algorithm differs from the informal version presented in the main body in greater detail.

Differences between full version (Algorithm 3) and informal version (Algorithm 2) of RVFS. The main difference between Algorithm 2 and its full version in Algorithm 3 is that in the former we simply assume access to quantities involving conditional expectations such as:

- The bellman backups $\mathcal{P}_h[\widehat{V}_{h+1}]$, which are required to evaluate the actions of RVFS's policies (see (4)), and to perform the tests in Line 8; and
- The value functions $\mathbb{E}^{\widehat{\pi}_{h+1:H}}[\sum_{\ell=h}^{H} \boldsymbol{r}_\tau \mid \boldsymbol{x}_h = x, \boldsymbol{a}_h = a]$ in Line 15, which are needed in the regression problem in Line 16.

These quantities are not available to the algorithm directly, but they can be estimated using the local simulator. This is reflected in the full version of RVFS in Algorithm 3.

Extracting policies from value functions. Let us briefly comment in more detail on how Algorithm 3 extracts the policy $\pi^{(t)}$ from the optimistic value function $f^{(t)} \in \mathcal{V}$ at iteration t. From the Bellman equation, the ideal choice would be to set $\pi_h^{(t)}(x) = \arg\max_{a\in\mathcal{A}} \mathcal{P}_h[f_{h+1}^{(t)}](x,a)$, but this requires knowledge of the transition distribution. Instead, given parameters $\varepsilon, \delta \in (0,1)$, SimGolf invokes Algorithm 5 via $\pi_h^{(t)}(x) \in \arg\max_{a\in\mathcal{A}} \widehat{\boldsymbol{\mathcal{P}}}_{h,\varepsilon,\delta}[f_{h+1}^{(t)}](x,a)$. The operator $\widehat{\boldsymbol{\mathcal{P}}}_{h,\varepsilon,\delta}[f]$ (Algorithm 5), when given input $(x,a) \in \mathcal{X}\times\mathcal{A}$ and $f_{h+1} : \mathcal{X} \to \mathbb{R}$, uses the local simulator to generate $N_{\mathtt{sim}} \geq 1$ next states $\boldsymbol{x}_{h+1}^{(1)}, \dots, \boldsymbol{x}_{h+1}^{(N_{\mathtt{sim}})} \overset{\text{i.i.d.}}{\sim} T_h(\cdot \mid x,a)$ to estimate the bellman back-up $\mathcal{P}_h[f_{h+1}]$ via $\frac{1}{N_{\mathtt{sim}}} \sum_{i=1}^{N_{\mathtt{sim}}} (\boldsymbol{r}_h^{(i)} + f_{h+1}(\boldsymbol{x}_{h+1}^{(i)}))$, where $\boldsymbol{r}_h^{(1)}, \dots, \boldsymbol{r}_h^{(N_{\mathtt{sim}})} \overset{\text{i.i.d.}}{\sim} R_h(x,a)$. The number of samples $N_{\mathtt{sim}}$ in Algorithm 5 is set as a function of (ε, δ) such that with probability at least $1-\delta$, $|\widehat{\boldsymbol{\mathcal{P}}}_{h,\varepsilon,\delta}[f_{h+1}](x,a) - \mathcal{P}_h[f_{h+1}](x,a)| \leq \varepsilon$.

Invoking the algorithm. The base invocation of RVFS takes the form

$$\widehat{V}_{1:H} \leftarrow \mathsf{RVFS}_0(\widehat{V}_{1:H} = \mathsf{arbitrary}, \widehat{\mathcal{V}}_{1:H} = \{\mathcal{V}\}_{h=1}^{H}, \mathcal{C}_{0:H} = \{\varnothing\}_{h=0}^{H}, \mathcal{B}_{0:H} = \{\varnothing\}_{h=0}^{H}, t_{1:H} = \{1\}_{h=1}^{H}; \cdots).$$

Whenever this call returns, the greedy policy induced by $\widehat{V}_{1:H}$ is guaranteed to be near-optimal. Naively, the policy induced by $\widehat{V}_{1:H}$ is non-executable, and must be computed by invoking the local simulator through Line 14. To provide an end-to-end guarantee to learn an executable policy, the outer-level algorithm, RVFS.bc (Algorithm 4, invokes RVFS_0, then extracts an executable policy from $\widehat{V}_{1:H}$ using imitation learning.

Subsequent recursive calls take the form

$$(\widehat{V}_{h:H}, \widehat{\mathcal{V}}_{h:H}, \mathcal{C}_{h:H}, \mathcal{B}_{h:H}, t_{h:H}) \leftarrow \mathsf{RVFS}_h(\widehat{V}_{h+1:H}, \widehat{\mathcal{V}}_{h+1:H}, \mathcal{C}_{h:H}, \mathcal{B}_{h:H}, t_{h:H}; \mathcal{V}, \varepsilon, \delta).$$

For such a call, the arguments above are:

- $\widehat{V}_{h+1:H}$: Value function estimates for subsequent layers.
- $\widehat{\mathcal{V}}_{h+1:H}$: Value function confidence sets $\widehat{\mathcal{V}}_{h+1:H} \subset \mathcal{V}_{h+1:H}$, which are used in the test on Line 14 to quantify uncertainty on new state-action pairs and decide whether to expand the core-sets.

- $\mathcal{C}_{h:H}$: Core-sets for current and subsequent layers.
- $\mathcal{B}_{h:H}$: Buffers of tuples $(x_{h-1}, a_{h-1}, \widehat{V}_h, \widehat{\mathcal{V}}_h, t_h)$, which record relevant features of the algorithm's state whenever the test on Line 14 fails and a recursive call is performed.
- $t_{h:H}$: Counters that track the number of times Algorithm 5 is called in the test on Line 14, which facilitate tuning of confidence parameters.

Importantly, the confidence sets $\widehat{\mathcal{V}}_{h+1:H}$ do not need to be explicitly maintained, and can be invoked implicitly whenever a *regression oracle* for the value function class is available (cf. discussion in Section 4). Likewise, the buffers $\mathcal{B}_{h:H}$ are only used in our analysis, and do not need to be explicitly maintained.

E.1 RVFS Pseudocode

Algorithm 3 RVFS_h: Recursive Value Function Search

```
1:  parameters: Value function class $\mathcal{V}$, suboptimality $\varepsilon \in (0,1)$, confidence $\delta \in (0,1)$.
2:    input:
        • Level $h \in \{0, \dots, H\}$.
        • Value function estimates $\widehat{V}_{h+1:H}$, confidence sets $\widehat{\mathcal{V}}_{h+1:H}$, state-action collections $\mathcal{C}_{h:H}$, buffers $\mathcal{B}_{h:H}$,
          and counters $t_{h:H}$.
    /* Initialize parameters. */
3:  Set $M \leftarrow \lceil 8\varepsilon^{-1} C_{\mathsf{push}} H \rceil$.
4:  Set $N_{\mathsf{test}} \leftarrow 2^8 M^2 H \varepsilon^{-1} \log(8M^6 H^8 \varepsilon^{-2} \delta^{-1})$, $N_{\mathsf{reg}} \leftarrow 2^8 M^2 \varepsilon^{-1} \log(8|\mathcal{V}|^2 H M^2 \delta^{-1})$,
5:  Set $N_{\mathsf{est}}(k) \leftarrow 2N_{\mathsf{reg}}^2 \log(8AN_{\mathsf{reg}} H k^3/\delta)$ and $\delta' \leftarrow \delta/(8M^7 N_{\mathsf{test}}^2 H^8 |\mathcal{V}|)$.
6:  Set $\varepsilon_{\mathsf{reg}}^2 \leftarrow \frac{9MH^2 \log(8M^2 H |\mathcal{V}|^2/\delta)}{N_{\mathsf{reg}}} + \frac{34MH^3 \log(8M^6 N_{\mathsf{test}}^2 H^8/\delta)}{N_{\mathsf{test}}}$.
7:  Set $\beta(t) \leftarrow \sqrt{2\log_{1/\delta'}(8AM|\mathcal{V}|t^2/\delta)}$.
    /* Test the fit for the estimated value functions $\widehat{V}_{h+1:H}$ at future layers. */
8:  for $(x_{h-1}, a_{h-1}) \in \mathcal{C}_h$ do
9:      for layer $\ell = H, \dots, h+1$ do
10:         for $n = 1, \dots, N_{\mathsf{test}}$ do
11:             Draw $\boldsymbol{x}_h \sim T_{h-1}(\cdot \mid x_{h-1}, a_{h-1})$, then draw $\boldsymbol{x}_{\ell-1}$ by rolling out with $\widehat{\pi}_{h:H}$, where[11]

                    $\forall \tau \in [H], \quad \widehat{\boldsymbol{\pi}}_\tau(\cdot) \in \arg\max_{a \in \mathcal{A}} \widehat{\boldsymbol{\mathcal{P}}}_{\tau,\varepsilon,\delta'}[\widehat{V}_{\tau+1}](\cdot, a), \quad \text{with} \quad \widehat{V}_{H+1} \equiv 0.$    (15)

12:             for $a_{\ell-1} \in \mathcal{A}$ do
                    /* Number of times $\widehat{\mathcal{P}}_{\ell-1,\varepsilon,\delta'}$ (Algorithm 5) is called in the test on Line 14. */
13:                 Update $t_\ell \leftarrow t_\ell + 1$.
                    /* Test fit; if test fails, re-fit value functions $\widehat{V}_{h+1:\ell}$ up to layer $\ell$. */
14:                 if $\sup_{f \in \widehat{\mathcal{V}}_\ell} |(\widehat{\boldsymbol{\mathcal{P}}}_{\ell-1,\varepsilon,\delta'}[\widehat{V}_\ell] - \widehat{\boldsymbol{\mathcal{P}}}_{\ell-1,\varepsilon,\delta'}[f_\ell])(\boldsymbol{x}_{\ell-1}, a_{\ell-1})| > \varepsilon + \varepsilon \cdot \beta(t_\ell)$ then
15:                     $\mathcal{C}_\ell \leftarrow \mathcal{C}_\ell \cup \{(\boldsymbol{x}_{\ell-1}, a_{\ell-1})\}$ and $\mathcal{B}_\ell \leftarrow \mathcal{B}_\ell \cup \{(\boldsymbol{x}_{\ell-1}, a_{\ell-1}, \widehat{V}_\ell, \widehat{\mathcal{V}}_\ell, t_\ell)\}$.
16:                     for $\tau = \ell, \dots, h+1$ do
17:                         $(\widehat{V}_{\tau:H}, \widehat{\mathcal{V}}_{\tau:H}, \mathcal{C}_{\tau:H}, \mathcal{B}_{\tau:H}, t_{\tau:H}) \leftarrow \mathsf{RVFS}_\tau(\widehat{V}_{\tau+1:H}, \widehat{\mathcal{V}}_{\tau+1:H}, \mathcal{C}_{\tau:H}, \mathcal{B}_{\tau:H}, t_{\tau:H}; \mathcal{V}, \varepsilon, \delta)$.
18:                     go to line 8.
19: if $h = 0$ then return: $(\widehat{V}_{1:H}, \cdot, \cdot, \cdot, \cdot)$.
    /* Re-fit $\widehat{V}_h$ and build a new confidence set. */
20: for $(x_{h-1}, a_{h-1}) \in \mathcal{C}_h$ do
21:     Set $\mathcal{D}_h(x_{h-1}, a_{h-1}) \leftarrow \varnothing$.
22:     for $i = 1, \dots, N_{\mathsf{reg}}$ do
23:         Sample $\boldsymbol{x}_h \sim T_{h-1}(\cdot \mid x_{h-1}, a_{h-1})$.
24:         Let $\widehat{V}_h(\boldsymbol{x}_h)$ be a Monte-Carlo estimate for $\mathbb{E}^{\widehat{\pi}_{h:H}}[\sum_{\ell=h}^H \boldsymbol{r}_\ell \mid \boldsymbol{x}_h]$ computed by collecting $N_{\mathsf{est}}(|\mathcal{C}_h|)$
            trajectories starting from $\boldsymbol{x}_h$ and rolling out with $\widehat{\pi}_{h:H}$.
25:         Update $\mathcal{D}_h(x_{h-1}, a_{h-1}) \leftarrow \mathcal{D}_h(x_{h-1}, a_{h-1}) \cup \{(\boldsymbol{x}_h, \widehat{V}_h(\boldsymbol{x}_h))\}$.
26: Let $\widehat{V}_h := \arg\min_{f \in \mathcal{V}} \sum_{(x_{h-1}, a_{h-1}) \in \mathcal{C}_h} \sum_{(x_h, v_h) \in \mathcal{D}_h(x_{h-1}, a_{h-1})} (f(x_h) - v_h)^2$.
27: Compute value function confidence set

        $\widehat{\mathcal{V}}_h := \left\{ f \in \mathcal{V} \;\middle|\; \sum_{(x_{h-1}, a_{h-1}) \in \mathcal{C}_h} \frac{1}{N_{\mathsf{reg}}} \sum_{(x_h, -) \in \mathcal{D}_h(x_{h-1}, a_{h-1})} \left(\widehat{V}_h(x_h) - f(x_h)\right)^2 \le \varepsilon_{\mathsf{reg}}^2 \right\}.$    (16)

28: return $(\widehat{V}_{h:H}, \widehat{\mathcal{V}}_{h:H}, \mathcal{C}_{h:H}, \mathcal{B}_{h:H}, t_{h:H})$.
```

Algorithm 4 `RVFS.bc`: Learn an executable policy with `RVFS` via behavior cloning.

```
input: Value function class $\mathcal{V}$, policy class $\Pi$, suboptimality $\varepsilon \in (0,1)$, confidence $\delta \in (0,1)$.
/* Set parameters for RVFS. */
Set $\varepsilon_{\mathsf{RVFS}} \leftarrow \varepsilon H^{-1}/48$.
Set $\widehat{V}_{1:H} \leftarrow \mathsf{arbitrary}$, $\widehat{\mathcal{V}}_{1:H} \leftarrow \mathcal{V}$, $\mathcal{C}_{0:H} \leftarrow \varnothing$, $\mathcal{B}_{0:H} \leftarrow \varnothing$, and $t_i \leftarrow 0$, for all $i \in [0\,..\,H]$.
/* Set parameters for BehaviorCloning. */
Set $M \leftarrow \lceil 8\varepsilon_{\mathsf{RVFS}}^{-1} C_{\mathsf{push}} H \rceil$ and $N_{\mathsf{test}} \leftarrow 2^8 M^2 H \varepsilon_{\mathsf{RVFS}}^{-1} \log(80 M^6 H^8 \varepsilon_{\mathsf{RVFS}}^{-2} \delta^{-1})$.
$N_{\mathsf{reg}} \leftarrow 2^8 M^2 \varepsilon_{\mathsf{RVFS}}^{-1} \log(80|\mathcal{V}|^2 H M^2 \delta^{-1})$ and $\delta' = \frac{\delta}{40 M^7 N^2 H^8 |\mathcal{V}|}$.
/* Get value functions from RVFS */
$(\widehat{V}_{1:H}, \cdot, \cdot, \cdot, \cdot) \leftarrow \mathsf{RVFS}_0(\widehat{V}_{1:H}, \widehat{\mathcal{V}}_{1:H}, \mathcal{C}_{0:H}, \mathcal{B}_{0:H}, t_{0:H}; \mathcal{V}, N_{\mathsf{reg}}, N_{\mathsf{test}}, \varepsilon_{\mathsf{RVFS}}, \delta/10)$.
/* Extract executable policy via BehaviorCloning algorithm for imitation learning. */
Define $\widehat{\boldsymbol{\pi}}_h^{\mathsf{RVFS}}(\cdot) \in \arg\max_{a \in \mathcal{A}} \widehat{\boldsymbol{\mathcal{P}}}_{h, \varepsilon_{\mathsf{RVFS}}, \delta'}[\widehat{V}_{h+1}](\cdot, a)$.
Compute $\widehat{\pi}_{1:H} \leftarrow \mathtt{BehaviorCloning}(\Pi, \varepsilon, \widehat{\pi}_{1:H}^{\mathsf{RVFS}}, \delta/2)$
return: $\widehat{\pi}_{1:H}$.
```

Algorithm 5 $\widehat{\boldsymbol{\mathcal{P}}}_{h,\varepsilon,\delta}[f]$: Estimate conditional expectation $\mathbb{E}[\boldsymbol{r}_h + f(\boldsymbol{x}_{h+1}) \mid \boldsymbol{x}_h = \cdot, \boldsymbol{a}_h = \cdot]$.

```
parameters: Layer $h$, suboptimality $\varepsilon \in (0,1)$, confidence $\delta \in (0,1)$, target function $f$.
input: $(x, a) \in \mathcal{X} \times \mathcal{A}$.
Set $N_{\mathsf{sim}} := 2\log(1/\delta)/\varepsilon^2$.
Set $\mathcal{D} \leftarrow \varnothing$
for $i = 1, \dots, N_{\mathsf{sim}}$ do
    Sample $\boldsymbol{r}_h \sim R_h(x, a)$ and $\boldsymbol{x}_{h+1} \sim T_h(\cdot \mid x, a)$.        // Uses local simulator access.
    Update $\mathcal{D} \leftarrow \mathcal{D} \cup \{(\boldsymbol{r}_h, \boldsymbol{x}_{h+1})\}$.
return: $N_{\mathsf{sim}}^{-1} \cdot \sum_{(r,x) \in \mathcal{D}} (r + f(x))$.
```

E.2 $\mathsf{RVFS}^{\mathsf{exo}}$ Pseudocode

Algorithm 6 $\mathsf{RVFS}^{\mathsf{exo}}_h$: Recursive Value Function Search for Exogenous Block MDPs

```
1:  parameters: Value function class $\mathcal{V}$, suboptimality $\varepsilon \in (0,1)$, seeds $\zeta_{1:H} \in (0,1)$, confidence $\delta \in (0,1)$.
2:  input: Level $h \in [0\,..\,H]$, value function estimates $\widehat{V}_{h+1:H}$, confidence sets $\widehat{\mathcal{V}}_{h+1:H}$, state-action
    collections $\mathcal{C}_{h:H}$, and buffers $\mathcal{B}_{h:H}$, and counters $t_{h:H}$.
    /* Initialize parameters. */
3:  Set $M \leftarrow \lceil 8\varepsilon^{-2} C_{\mathsf{exo}} SAH \rceil$.
4:  Set $N_{\mathsf{test}} \leftarrow 2^8 M^2 H \varepsilon^{-2} \log(8M^6 H^8 \varepsilon^{-2} \delta^{-1})$, $N_{\mathsf{reg}} \leftarrow 2^8 M^2 \varepsilon^{-2} \log(8|\mathcal{V}| H M^2 \delta^{-1})$.
5:  Set $N_{\mathsf{est}}(k) \leftarrow 2N_{\mathsf{reg}}^2 \log(8AN_{\mathsf{reg}} H k^3/\delta)$ and $\delta' \leftarrow \delta/(4M^7 N_{\mathsf{test}}^2 H^8 |\mathcal{V}|)$.
6:  Set $\varepsilon_{\mathsf{reg}}^2 \leftarrow \frac{9MH^2 \log(8M^2 H|\mathcal{V}|/\delta)}{N_{\mathsf{reg}}} + \frac{34MH^3 \log(8M^6 N_{\mathsf{test}}^2 H^8/\delta)}{N_{\mathsf{test}}}$.
7:  Set $\beta(t) \leftarrow \sqrt{\log_{1/\delta'}(8MA|\mathcal{V}|t^2/\delta)}$.
    /* Test the fit for the estimated value functions $\widehat{V}_{h+1:H}$ at future layers. */
8:  for $(x_{h-1}, a_{h-1}) \in \mathcal{C}_h$ do
9:      for layer $\ell = H, \dots, h+1$ do
10:         for $n = 1, \dots, N_{\mathsf{test}}$ do
11:             Draw $\boldsymbol{x}_h \sim T_{h-1}(\cdot \mid x_{h-1}, a_{h-1})$, then draw $\boldsymbol{x}_{\ell-1}$ by rolling out with $\widehat{\pi}_{h+1:H}$, where
                    $$\forall \tau \in [H], \quad \widehat{\pi}_\tau(\cdot) \in \operatorname*{arg\,max}_{a \in \mathcal{A}} \left[ \widehat{\boldsymbol{\mathcal{P}}}_{\tau,\varepsilon^2,\delta'}[\widehat{V}_{\tau+1}](\cdot, a) \cdot \varepsilon^{-1} + \zeta_\tau \right], \quad \text{with} \quad \widehat{V}_{H+1} \equiv 0.$$
12:             for $a_{\ell-1} \in \mathcal{A}$ do
13:                 Update $t_\ell \leftarrow t_\ell + 1$.
                    /* Test fit; if test fails, re-fit value functions $\widehat{V}_{h+1:\ell}$ up to layer $\ell$. */
14:                 if $\sup_{f \in \widehat{\mathcal{V}}_\ell} |(\widehat{\boldsymbol{\mathcal{P}}}_{\ell-1,\varepsilon^2,\delta'}[\widehat{V}_\ell] - \widehat{\boldsymbol{\mathcal{P}}}_{\ell-1,\varepsilon^2,\delta'}[f_\ell])(\boldsymbol{x}_{\ell-1}, a_{\ell-1})| > \varepsilon^2 + \varepsilon^2 \cdot \beta(t_\ell)$ then
15:                     $\mathcal{C}_\ell \leftarrow \mathcal{C}_\ell \cup \{(\boldsymbol{x}_{\ell-1}, a_{\ell-1})\}$ and $\mathcal{B}_\ell \leftarrow \mathcal{B}_\ell \cup \{(\boldsymbol{x}_{\ell-1}, a_{\ell-1}, \widehat{V}_\ell, \widehat{\mathcal{V}}_\ell, t_\ell)\}$.
16:                     for $\tau = \ell, \dots, h+1$ do
17:                         $(\widehat{V}_{\tau:H}, \widehat{\mathcal{V}}_{\tau:H}, \mathcal{C}_{\tau:H}, \mathcal{B}_{\tau:H}, t_{\tau:H}) \leftarrow \mathsf{RVFS}^{\mathsf{exo}}_\tau(\widehat{V}_{\tau+1:H}, \widehat{\mathcal{V}}_{\tau+1:H}, \mathcal{C}_{\tau:H}, \mathcal{B}_{\tau:H}, t_{\tau:H}; \mathcal{V}, \varepsilon, \zeta_{1:H}, \delta)$.
18:                     go to line 8.
19: if $h = 0$ then return $(\widehat{V}_{1:H}, \cdot, \cdot, \cdot, \cdot)$.
    /* Re-fit $\widehat{V}_h$ and build a new confidence set. */
20: for $(x_{h-1}, a_{h-1}) \in \mathcal{C}_h$ do
21:     Set $\mathcal{D}_h(x_{h-1}, a_{h-1}) \leftarrow \varnothing$.
22:     for $i = 1, \dots, N_{\mathsf{reg}}$ do
23:         Sample $\boldsymbol{x}_h \sim T_{h-1}(\cdot \mid x_{h-1}, a_{h-1})$.
24:         For each $a \in \mathcal{A}$, let $\widehat{V}_h(\boldsymbol{x}_h)$ be a Monte-Carlo estimate for $\mathbb{E}^{\widehat{\pi}_{h:H}}[\sum_{\ell=h}^H \boldsymbol{r}_\ell \mid \boldsymbol{x}_h]$ computed by collecting
            $N_{\mathsf{est}}(|\mathcal{C}_h|)$ trajectories starting from $\boldsymbol{x}_h$ and rolling out with $\widehat{\pi}_{h:H}$.
25:         Update $\mathcal{D}(x_{h-1}, a_{h-1}) \leftarrow \mathcal{D}(x_{h-1}, a_{h-1}) \cup \{(\boldsymbol{x}_h, \widehat{V}_h(\boldsymbol{x}_h))\}$.
26: Let $\widehat{V}_h \coloneqq \operatorname*{arg\,min}_{f \in \mathcal{V}_h} \sum_{(x_{h-1}, a_{h-1}) \in \mathcal{C}_h} \sum_{(x_h, v_h) \in \mathcal{D}_h(x_{h-1}, a_{h-1})} (f(x_h) - v_h)^2$.
27: Compute value function confidence set
        $$\widehat{\mathcal{V}}_h \coloneqq \left\{ f \in \mathcal{V}_h \;\middle|\; \sum_{(x_{h-1}, a_{h-1}) \in \mathcal{C}_h} \frac{1}{N_{\mathsf{reg}}} \sum_{(x_h, -) \in \mathcal{D}_h(x_{h-1}, a_{h-1})} \left( \widehat{V}_h(x_h) - f(x_h) \right)^2 \le \varepsilon_{\mathsf{reg}}^2 \right\}. \tag{17}$$
28: return $(\widehat{V}_{h:H}, \widehat{\mathcal{V}}_{h:H}, \mathcal{C}_{h:H}, \mathcal{B}_{h:H}, t_{h:H})$.
```

Algorithm 7 $\mathsf{RVFS}^{\mathsf{exo}}$.bc: Learn an executable policy with $\mathsf{RVFS}^{\mathsf{exo}}$ via imitation learning.

```
input: Decoder class $\Phi$, suboptimality $\varepsilon \in (0,1)$, confidence $\delta \in (0,1)$.
/* Set parameters for RVFS and define the value function and policy classes. */
Set $\varepsilon_{\mathsf{RVFS}} \leftarrow \varepsilon H^{-1}/48$.
Set $\mathcal{V} = \mathcal{V}_{1:H}$, where $\mathcal{V}_h = \{x \mapsto f(\phi(x)) : f \in [0,H]^S, \phi \in \Phi\}$, $\forall h \in [H]$.
Set $\Pi = \Pi_{1:H}$, where $\Pi_h = \left\{\pi(\cdot) \in \arg\max_{a\in\mathcal{A}} f(\phi(\cdot),a) : f \in [0,H]^{S\times A}, \phi \in \Phi\right\}$, $\forall h \in [H]$.
/* Set parameters for BehaviorCloning. */
Set $N_{\mathsf{bc}} \leftarrow 8H^2 \log(4H|\Pi|/\delta)/\varepsilon$, $N_{\mathsf{boost}} \leftarrow \log(1/\delta)/\log(24SAH\varepsilon)$, $N_{\mathsf{eval}} \leftarrow 16^2\varepsilon^{-2}\log(2N_{\mathsf{boost}}/\delta)$.
Set $M \leftarrow \lceil 8\varepsilon^{-1}_{\mathsf{RVFS}} SAC_{\mathsf{cov}} H\rceil$, $N_{\mathsf{test}} \leftarrow 2^8 M^2 H \varepsilon^{-1}_{\mathsf{RVFS}} \log(80M^6H^8N_{\mathsf{boost}}\varepsilon^{-2}_{\mathsf{RVFS}}\delta^{-1})$, and $\delta' = \frac{\delta}{40M^7N^2H^8|\mathcal{V}|N_{\mathsf{boost}}}$.
Set $N_{\mathsf{reg}} \leftarrow 2^8 M^2 \varepsilon^{-1}_{\mathsf{RVFS}} \log(80|\Phi|^2 HM^2 N_{\mathsf{boost}}\delta^{-1})$.
Set $\widehat{V}_{1:H} \leftarrow \mathsf{arbitrary}$, $\widehat{\mathcal{V}}_{1:H} \leftarrow \mathcal{V}$, $\mathcal{C}_{0:H} \leftarrow \varnothing$, $\mathcal{B}_{0:H} \leftarrow \varnothing$, $i_{\mathsf{opt}} = 1$, and $J_{\max} = 0$.
/* Repeatedly invoke RVFS^exo and extract policy with BehaviorCloning to boost confidence. */
for $i = 1, \dots, N_{\mathsf{boost}}$ do
    /* Invoke RVFS^exo. */
    $(\widehat{V}^{(i)}_{1:H}, \cdot, \cdot, \cdot, \cdot) \leftarrow \mathsf{RVFS}^{\mathsf{exo}}_0(\widehat{V}_{1:H}, \widehat{\mathcal{V}}_{1:H}, \mathcal{C}_{0:H}, \mathcal{B}_{0:H}; \mathcal{V}, N_{\mathsf{reg}}, N_{\mathsf{test}}, \varepsilon_{\mathsf{RVFS}}, \delta/(10N_{\mathsf{boost}}))$.
    /* Imitation learning with BehaviorCloning. */
    Define $\widehat{\boldsymbol{\pi}}^{\mathsf{RVFS}}_h(\cdot) \in \arg\max_{a\in\mathcal{A}} \widehat{\boldsymbol{\mathcal{P}}}_{h,\varepsilon_{\mathsf{RVFS}},\delta'}[\widehat{V}^{(i)}_{h+1}](\cdot,a)$.
    Compute $\widehat{\pi}^{(i)}_{1:H} \leftarrow \mathtt{BehaviorCloning}(\Pi, \varepsilon, \widehat{\pi}^{\mathsf{RVFS}}_{1:H}, \delta/(2N_{\mathsf{boost}}))$.
    /* Evaluate current policy. */
    $v = 0$.
    for $= 1, \dots, N_{\mathsf{eval}}$ do
        Sample trajectory $(\boldsymbol{x}_1, \boldsymbol{a}_1, \boldsymbol{r}_1, \dots, \boldsymbol{x}_H, \boldsymbol{a}_H, \boldsymbol{r}_H)$ by executing $\widehat{\pi}^{(i)}_{1:H}$.
        Set $v \leftarrow v + \sum_{h=1}^{H} \boldsymbol{r}_h$.
    Set $\widehat{J}(\widehat{\pi}^{(i)}_{1:H}) \leftarrow v/N_{\mathsf{eval}}$.
    if $\widehat{J}(\widehat{\pi}^{(i)}_{1:H}) > J_{\max}$ then
        Set $i_{\mathsf{opt}} = i$.
        Set $J_{\max} = \widehat{J}(\widehat{\pi}^{(i)}_{1:H})$.
return: $\widehat{\pi}_{1:H} = \widehat{\pi}^{(i_{\mathsf{opt}})}_{1:H}$.
```

F Organization

This remainder of Part II of the appendix contains the proofs for the main results concerning the `RVFS` algorithm (Theorem 4.1 and Theorem 4.2).

- First, in Appendix G we give a brief overview of the analysis and introduce a restricted set of *benchmark policies* which will be used throughout the proofs for Theorem 4.1 and Theorem 4.2. The benchmark policy class is central to the regret decomposition for `RVFS`, and facilitates an analysis that does not require optimism.
- In Appendix H, we prove Theorem 4.1 under **Setup II** (V^{π}-realizability). This constitutes the main technical development for Theorem 4.1. The central technical results proven here are Theorem H.1, Theorem H.2 and which generalize Theorem 4.1.
- In Appendix I, we prove Theorem 4.1 under **Setup I** ($V^{\star}$-realizability), as a straightforward consequence of the tools developed in Appendix H (Theorem H.1 and Theorem H.2).
- Finally, in Appendix J, we prove Theorem 4.2 (analysis of $\texttt{RVFS}^{\texttt{exo}}$ for the ExBMDP problem). This analysis has a similar structure to the proof of Theorem 4.1 under **Setup II** in Appendix H, and builds on the same analysis techniques, but requires specialized arguments due to extra technical challenges in the ExBMDP setting.
- Appendix L gives a self-contained presentation of the `BehaviorCloning` algorithm for imitation learning, which is used within `RVFS.bc` and $\texttt{RVFS}^{\texttt{exo}}$`.bc`.

G Overview of Analysis and Preliminaries

In this section, we present some notation, technical tools, and preliminary results we require for the analysis of `RVFS` in the settings we described in Section 4. We start by defining a set of restricted *benchmark policies* used in the regret decomposition for `RVFS`.

G.1 Benchmark Policy Class and Randomized Policies

As described above, central to our analysis is a set of $O(\varepsilon)$-suboptimal policies against which we benchmark the policies returned `RVFS`, which emulate certain consequences of the Δ-gap assumption (Assumption 4.4). Before introducing this concept formally, we first define the notion of a *randomized policy.*

Induced stochastic policies. Given an arbitrary collection of independent random variables $\widetilde{\boldsymbol{Q}} = (\widetilde{\boldsymbol{Q}}_h(x,a))_{(h,x,a)\in[H]\times\mathcal{X}\times\mathcal{A}}$, we say that a policy π is *induced* by $\widetilde{\boldsymbol{Q}}$ if π satisfies

$$\forall h \in [H], \forall x \in \mathcal{X}, \quad \boldsymbol{\pi}_h(x) \in \underset{a'\in\mathcal{A}}{\arg\max}\, \widetilde{\boldsymbol{Q}}_h(x,a'), \tag{18}$$

where we use the bold notation $\boldsymbol{\pi}_h(x)$ as shorthand for the random variable $\boldsymbol{a}_h \sim \pi_h(x) \in \Delta(\mathcal{A})$; in other words, for each $x \in \mathcal{X}$, $\pi_h(x) \in \Delta(\mathcal{A})$ is the distribution induced by sampling $\mathbf{Q}_h(x,\cdot)$ and playing the action $\boldsymbol{\pi}_h(x) \in \arg\max_{a'\in\mathcal{A}} \widetilde{\boldsymbol{Q}}_h(x,a')$. If there are ties in (18), we break them by picking the action with the smallest index; we assume without loss of generality that actions in $\mathcal{A}$ are index from $1,\dots,|\mathcal{A}|$.

Benchmark policy class. We now define the benchmark policy class as follows.

Definition G.1 (Benchmark policy class)**.** *For $\varepsilon \in (0,1)$, let $\Pi_\varepsilon \subseteq \Pi_{\mathsf{S}}$ be the set of stochastic policies such that $\pi \in \Pi_\varepsilon$ if and only if there exists a collection of* independent *random variables $\widetilde{\boldsymbol{Q}} = (\widetilde{\boldsymbol{Q}}_h(x,a))_{(h,x,a)\in[H]\times\mathcal{X}\times\mathcal{A}}$ in $[0,H]$ such that:*

- *π is induced by $\widetilde{\boldsymbol{Q}}$ (i.e. Eq. (18) is satisfied); and*
- *For all $(h,x,a) \in [H]\times\mathcal{X}\times\mathcal{A}$, we have $|(\widetilde{\boldsymbol{Q}}_h - Q^{\pi}_h)(x,a)| \le \varepsilon$, almost surely under the draw of $\widetilde{\boldsymbol{Q}}$.*

Intuitively, the set Π_ε contains the set of all (stochastic) policies corresponding to (randomized) state-action value functions that are point-wise $O(\varepsilon)$ close to $Q^\star$. We formalize this claim in the next lemma.

Lemma G.1 (Suboptimality of benchmark policies)**.** *Let $\varepsilon \in (0,1)$ be given. Let $\tilde{\pi} \in \Pi_\varepsilon$ be a stochastic policy induced by a collection of (independent) random state-action value functions $(\widetilde{\boldsymbol{Q}}_h(x,a))_{(h,x,a)\in[H]\times\mathcal{X}\times\mathcal{A}} \subset [0,H]$ such that for all $h \in [H]$ and all $(x,a) \in \mathcal{X}\times\mathcal{A}$:*

$$|\widetilde{\boldsymbol{Q}}_h(x,a) - Q_h^{\tilde{\pi}}(x,a)| \le \varepsilon \quad \textit{almost surely, and} \quad \tilde{\boldsymbol{\pi}}_h(x) \in \arg\max_{a'\in\mathcal{A}} \widetilde{\boldsymbol{Q}}_h(x,a').$$

Then, for all $h \in [H]$,

$$\forall x \in \mathcal{X}, \quad V_h^\star(x) \le V_h^{\tilde{\pi}}(x) + 3H\varepsilon. \tag{19}$$

Proof of Lemma G.1. Using backward induction over $\ell = H,\dots,1$, we start by showing that all ℓ:

$$\forall (x,a) \in \mathcal{X}\times\mathcal{A}, \quad Q_\ell^\star(x,a) \le \widetilde{\boldsymbol{Q}}_\ell(x,a) + 2\varepsilon\cdot(H-\ell+1). \tag{20}$$

almost surely. We then instantiate this with $\ell = 1$ and use the fact that $\|\widetilde{\boldsymbol{Q}}_h - Q_h^{\tilde{\pi}}\|_\infty \le \varepsilon$ to get the desired result.

Base case $[\ell = H]$**.** By definition of the state-action value function, we have, for all $\pi \in \Pi_{\mathsf{S}}$, $Q_H^\star \equiv Q_H^\pi$. Thus, since $\sup_{(x,a)\in\mathcal{X}\times\mathcal{A}} |(\widetilde{\boldsymbol{Q}}_H - Q_H^{\tilde{\pi}})(x,a)| \le \varepsilon$ almost surely (by definition of $\widetilde{\boldsymbol{Q}}_{1:H}$), we get that

$$\forall (x,a) \in \mathcal{X}\times\mathcal{A}, \quad |\widetilde{\boldsymbol{Q}}_H(x,a) - Q_H^\star(x,a)| \le \varepsilon,$$

almost surely. This implies (19) for $\ell = H$.

General case $[\ell < h]$**.** Fix $h \in [H-1]$ and suppose that (20) holds for all $\ell \in [h+1,\dots,H]$ almost surely. We show that it holds for $\ell = h$ almost surely. Fix $(x,a) \in \mathcal{X}\times\mathcal{A}$. We have

$$\begin{aligned}
Q_h^\star(x,a) - \widetilde{\boldsymbol{Q}}_h(x,a) &= \mathcal{T}_h[Q_{h+1}^\star](x,a) - \mathcal{T}_h[\widetilde{\boldsymbol{Q}}_{h+1}](x,a) + \mathcal{T}_h[\widetilde{\boldsymbol{Q}}_{h+1}](x,a) - \widetilde{\boldsymbol{Q}}_h(x,a), \\
&\le 2\varepsilon\cdot(H-h) + \mathcal{T}_h[\widetilde{\boldsymbol{Q}}_{h+1}](x,a) - \widetilde{\boldsymbol{Q}}_h(x,a),
\end{aligned} \tag{21}$$

almost surely, where the last step follows by the induction hypothesis. We now bound $\mathcal{T}_h[\widetilde{\boldsymbol{Q}}_{h+1}](x,a) - \widetilde{\boldsymbol{Q}}_h(x,a)$. We have, almost surely, that

$$\begin{aligned}
\mathcal{T}_h[\widetilde{\boldsymbol{Q}}_{h+1}](x,a) - \widetilde{\boldsymbol{Q}}_h(x,a) &= \mathcal{T}_h[\widetilde{\boldsymbol{Q}}_{h+1}](x,a) - \mathcal{P}_h[V_{h+1}^{\tilde{\pi}}](x,a) + \mathcal{P}_h[V_{h+1}^{\tilde{\pi}}](x,a) - \widetilde{\boldsymbol{Q}}_h(x,a), \\
&= \mathcal{T}_h[\widetilde{\boldsymbol{Q}}_{h+1}](x,a) - \mathcal{P}_h[V_{h+1}^{\tilde{\pi}}](x,a) + Q_h^{\tilde{\pi}}(x,a) - \widetilde{\boldsymbol{Q}}_h(x,a), \\
&= \mathcal{T}_h[\widetilde{\boldsymbol{Q}}_{h+1}](x,a) - \mathcal{P}_h[V_{h+1}^{\tilde{\pi}}](x,a) + \varepsilon, \quad \text{(by the assumption on } \widetilde{\boldsymbol{Q}}_h\text{)} \\
&= \mathbb{E}\left[\max_{a'\in\mathcal{A}} \widetilde{\boldsymbol{Q}}_{h+1}(\boldsymbol{x}_{h+1},a') - Q_{h+1}^{\tilde{\pi}}(\boldsymbol{x}_{h+1},\tilde{\boldsymbol{\pi}}_{h+1}(\boldsymbol{x}_{h+1})) \mid \boldsymbol{x}_h = x, \boldsymbol{a}_h = a\right] + \varepsilon, \\
&= \mathbb{E}\left[\widetilde{\boldsymbol{Q}}_{h+1}(\boldsymbol{x}_{h+1},\tilde{\boldsymbol{\pi}}_{h+1}(\boldsymbol{x}_{h+1})) - Q_{h+1}^{\tilde{\pi}}(\boldsymbol{x}_{h+1},\tilde{\boldsymbol{\pi}}_{h+1}(\boldsymbol{x}_{h+1})) \mid \boldsymbol{x}_h = x, \boldsymbol{a}_h = a\right] + \varepsilon, \\
&\le 2\varepsilon,
\end{aligned} \tag{22}$$

where the penultimate inequality follows by the definition of $\tilde{\boldsymbol{\pi}}_{h+1}$, and the last inequality follows by the fact that $\|\widetilde{\boldsymbol{Q}}_{h+1} - Q_{h+1}^{\tilde{\pi}}\|_\infty \le \varepsilon$ almost surely, by assumption. Plugging (22) into (21) completes the induction, and so we have that

$$\forall (x,a) \in \mathcal{X}\times\mathcal{A}, \quad Q_1^\star(x,a) \le \widetilde{\boldsymbol{Q}}_1(x,a) \le 2H\varepsilon.$$

In particular, taking the max over a on both sides and using the definition of $\tilde{\pi}$, we get that

$$\forall x \in \mathcal{X}, \quad V_1^\star(x) \le \widetilde{\boldsymbol{Q}}_1(x,\tilde{\boldsymbol{\pi}}_1(x)) \le 2H\varepsilon,$$

almost surely. Combining this with the fact that $\widetilde{\boldsymbol{Q}}_1(x, \tilde{\boldsymbol{\pi}}_1(x)) \leq Q_1^{\tilde{\pi}}(x, \tilde{\boldsymbol{\pi}}_1(x)) + \varepsilon$, almost surely (since $\|\widetilde{\boldsymbol{Q}}_1 - Q_1^{\tilde{\pi}}\|_\infty \leq \varepsilon$ almost surely by assumption) implies that

$$V_1^\star(x) \leq Q_1^{\pi}(x, \tilde{\boldsymbol{\pi}}_1(x)) + 2H\varepsilon + \varepsilon.$$

Taking expectation over $\tilde{\boldsymbol{\pi}}_1(x)$ and bounding $2H\varepsilon + \varepsilon$ by $3H\varepsilon$ leads to the desired result.

□

G.2 Additional Preliminaries

The following lemma gives a guarantee for the Bellman backup approximation algorithm $\widehat{\boldsymbol{\mathcal{P}}}$ (Algorithm 5) that is tailored to the analysis of RVFS.

Lemma G.2. *Let $\varepsilon, \delta, \delta' \in (0,1)$, $B > 0$, and $h \in [H]$, be given and let $\mathcal{V}$ be a finite function class. For any sequence $(x_i)_{i\geq 1} \subset \mathcal{X}$ of state action pairs, the outputs $(\widehat{\boldsymbol{\mathcal{P}}}_{h,\varepsilon,\delta'}[f](x_i, a))_{i\geq 1, a\in\mathcal{A}}$ of Algorithm 5 satisfy, with probability at least $1-\delta$,*

$$\forall i \geq 1, \forall f \in \mathcal{V}, \forall a \in \mathcal{A}, \quad |\widehat{\boldsymbol{\mathcal{P}}}_{h,\varepsilon,\delta'}[f](x_i, a) - \mathcal{P}_h[f](x_i, a)| \leq \varepsilon \cdot \sqrt{2\log_{1/\delta'}(2Ai^2|\mathcal{V}|/\delta)}.$$

Proof of Lemma G.2. By Hoeffding's inequality (Hoeffding, 1963) and the union bound over $a \in \mathcal{A}$ and $f \in \mathcal{V}$, we have that for any $i \geq 1$, with probability at least $1 - \delta/(2i^2)$,

$$\forall f \in \mathcal{V}, \forall a \in \mathcal{A}, \quad |\widehat{\boldsymbol{\mathcal{P}}}_{h,\varepsilon,\delta'}[f](x_i, a) - \mathcal{P}_h[f](x_i, a)| \leq \varepsilon \cdot \sqrt{2\log_{1/\delta'}(2Ai^2|\mathcal{V}|/\delta)}.$$

The desired result follows by the union bound over $i \geq 1$ and the fact that $\sum_{i\geq 1} 1/i^2 = \pi^2/6 \leq 2$. □

H Guarantee under V^π-Realizability (Proof of Theorem 4.1, Setup II)

In this section, we prove Theorem 4.1 under **Setup II**. First, in Appendix H.1 we state a number of supporting technical lemmas, then use them to prove a more general version of Theorem 4.1, Theorem H.2, which holds under a weaker realizability assumption (informally, V^π-realizability only for *near-optimal* policies π); Theorem 4.1 follows as an immediate consequence. The remainder of the section (Appendix H.2 through to Appendix H.6) contains the proofs for the intermediate results.

H.1 Analysis: Proof of Theorem 4.1 (Setup II)

We analyze RVFS in the setting of Theorem 4.1 (**Setup II**), where we have a function class $\mathcal{V}$ satisfying V^π-realizability (Assumption 4.5). We will actually show that the conclusion of Theorem 4.1 holds under a weaker function approximation setup we refer to as *relaxed V^π-realizability*: instead of requiring V^π-realizability for all $\pi \in \Pi_{\mathsf{S}}$, we only require it for policies π in the set of near-optimal policies corresponding to the *benchmark policy class* $\Pi_{\varepsilon_{\mathsf{real}}}$ for some $\varepsilon_{\mathsf{real}} > 0$ (Π_ε is defined in Appendix G).

Assumption H.1 (Relaxed V^π-realizability)**.** *For $\varepsilon_{\mathsf{real}} > 0$ and all $\pi \in \Pi_{\varepsilon_{\mathsf{real}}}$ and $h \in [H]$, we have $V_h^\pi(x) \in \mathcal{V} \subseteq \{f : \mathcal{X} \to [0, H]\}$.*

We will analyze RVFS under Assumption H.1 and Assumption 4.1. However, it turns out that all of the main results for RVFS can be derived under this assumption: As we will see in Appendix I in the sequel, when the Δ-gap assumption (Assumption 4.4) is satisfied, then $\Pi_{\varepsilon_{\mathsf{real}}} = \{\pi^\star\}$ for all $\varepsilon_{\mathsf{real}} < \Delta$, allowing us to prove Theorem 4.1 under **Setup I** as a special case of relaxed V^π-realizability. Our analysis for the ExBMDP setting in Appendix J requires more work, but uses that for ExBMDPs, Assumption H.1 is satisfied for a subset of $\Pi_{\varepsilon_{\mathsf{real}}}$ corresponding to endogenous policies.

We begin with our analysis under **Setup II** by bounding the number of times the test in Line 14 fails. Since the sizes of the core sets $\mathcal{C}_{1:H}$ in RVFS are directly related to the number of test failures, the next result, which bounds $|\mathcal{C}_h|$ for $h \in [H]$, allows us to show that RVFS terminates in polynomial iterations with high probability. The proof is in Appendix H.2.

Lemma H.1 (Bounding the number of test failures)**.** *Let $\delta, \varepsilon \in (0,1)$ be given, and suppose that Assumption 4.1 (pushforward coverability) holds with parameter $C_{\mathsf{push}} > 0$. Further, let $f \in \mathcal{V}$, be given, where $\mathcal{V}$ is an arbitrary function class. Then there is an event $\mathcal{E}$ of probability at least $1-\delta$ under which any call $\mathsf{RVFS}_0(f, \mathcal{V}_{1:H}, \varnothing, \varnothing, 0; \mathcal{V}, \varepsilon, \delta)$ (Algorithm 3) terminates, and throughout the execution of RVFS_0, we have*

$$\forall h \in [H], \quad |\mathcal{C}_h| \le \lceil 8\varepsilon^{-1} C_{\mathsf{push}} H \rceil. \tag{23}$$

In particular, Lemma H.1 ensures that with high probability, every call to RVFS_h (made recursively via the call to RVFS_0) terminates in polynomial iterations. When RVFS_h terminates, all the tests in Line 14 must have passed for all $\ell > h$. Using this and a standard concentration argument, we get the following guarantee for the estimated value functions and confidence sets returned by each call to RVFS_h. The proof is in Appendix H.3.

Lemma H.2 (Consequence of passing the tests)**.** *Let $h \in [0\,..\,H]$ and $\varepsilon, \delta \in (0,1)$ be given and consider a call to RVFS_0 in the setting of Lemma H.1. Further, let $\mathcal{E}$ be the event of Lemma H.1. There exists an event $\mathcal{E}'_h$ of probability at least $1-\delta/H$ such that under $\mathcal{E} \cap \mathcal{E}'_h$, if a call to RVFS_h within the execution of RVFS_0 terminates and returns $(\widehat{V}_{h:H}, \widehat{\mathcal{V}}_{h:H}, \mathcal{C}_{h:H}, \mathcal{B}_{h:H}, t_{h:H})$, then for any $(x_{h-1}, a_{h-1}) \in \mathcal{C}_h$ and $\ell \in [h+1\,..\,H+1]$:*

$$\mathbb{P}^{\widehat{\pi}}\left[\sup_{f\in\widehat{\mathcal{V}}_\ell}\max_{a\in\mathcal{A}}\left|\mathcal{P}_{\ell-1}[\widehat{V}_\ell]-\mathcal{P}_{\ell-1}[f_\ell](\boldsymbol{x}_{\ell-1},a)\right|>3\varepsilon \mid \boldsymbol{x}_{h-1}=x_{h-1},\boldsymbol{a}_{h-1}=a_{h-1}\right]\le\frac{4\log(8M^6N_{\mathsf{test}}^2H^8/\delta)}{N_{\mathsf{test}}}, \tag{24}$$

where $(\widehat{\pi}_\tau)_{\tau\ge h}$ is the stochastic policy induced by $\widehat{V}_{h:H}$ and M and N_{test} are defined as in Algorithm 3.

We now give a guarantee for the estimated value functions $\widehat{V}_{1:H}$ computed within RVFS in Line 26 (the proof is in Appendix H.4).

Lemma H.3 (Value function regression guarantee)**.** *Let $h \in [0\,..\,H]$ and $\delta, \varepsilon' \in (0,1)$ be given, and consider a call to RVFS_0 in the setting of Lemma H.1. Further, let $\Pi' \subseteq \Pi_{\mathsf{S}}$ be a finite policy class such that the class $\mathcal{V}$ realizes the value functions V^π for $\pi \in \Pi'$ (i.e. $\mathcal{V}$ satisfies Assumption H.1 with $\Pi_{\varepsilon_{\mathsf{real}}}$ replaced by Π'). Then, there is an event $\mathcal{E}''_h$ of probability at least $1-\delta/H$ under which for all $k \ge 1$, if*

1. *RVFS_h gets called for the kth time during the execution of RVFS_0; and*
2. *this kth call terminates and returns $(\widehat{V}_{h:H}, \widehat{\mathcal{V}}_{h:H}, \mathcal{C}_{h:H}, \mathcal{B}_{h:H}, t_{h:H})$,*

then if $(\widehat{\pi}_\tau)_{\tau\ge h}$ is the policy induced by $\widehat{V}_{h:H}$ and N_{reg} is set as in Algorithm 3, we have that for all $\pi \in \Pi'$,

$$\begin{aligned}&\sum_{(x_{h-1},a_{h-1})\in\mathcal{C}_h}\frac{1}{N_{\mathsf{reg}}}\sum_{(x_h,-)\in\mathcal{D}_h(x_{h-1},a_{h-1})}\left(\widehat{V}_h(x_h)-V_h^\pi(x_h)\right)^2\\&\le\frac{9kH^2\log(8k^2H|\Pi'||\mathcal{V}|/\delta)}{N_{\mathsf{reg}}}+8H^2\sum_{(x_{h-1},a_{h-1})\in\mathcal{C}_h}\sum_{\tau=h}^{H}\mathbb{E}^{\widehat{\pi}}\left[D_{\mathrm{tv}}(\widehat{\pi}_\tau(\boldsymbol{x}_\tau),\pi_\tau(\boldsymbol{x}_\tau))\mid\boldsymbol{x}_{h-1}=x_{h-1},\boldsymbol{a}_{h-1}=a_{h-1}\right],\end{aligned}$$

where the datasets $\{\mathcal{D}_h(x,a):(x,a)\in\mathcal{C}_h\}$ are as in the definition of $\widehat{\mathcal{V}}_h$ in (16).

Next, we use this result to show that the confidence sets $\widehat{\mathcal{V}}_{1:H}$ returned by RVFS_0 are "valid" in the sense that they contain a value function ($V_h^{\tilde{\pi}}$) corresponding to a near-optimal stochastic policy $\tilde{\pi}$ in the benchmark class $\Pi_{4\varepsilon}$. In the sequel, we use this fact to substitute $V_\ell^{\tilde{\pi}}$ for f_ℓ in Eq. (24) and bound the suboptimality of the learned policy $\widehat{\pi}$.

Lemma H.4 (Confidence sets)**.** *Let $\varepsilon, \delta \in (0,1)$ be given and suppose that Assumption 4.1 (pushforward coverability) holds with parameter $C_{\mathsf{push}} > 0$. Let $f \in \mathcal{V}$ be arbitrary, and suppose that $\mathcal{V}$ satisfies Assumption H.1 with $\varepsilon_{\mathsf{real}} = 4\varepsilon$. Then, there is an event $\mathcal{E}'''$ of probability at least $1-3\delta$ under which a call to $\mathsf{RVFS}_0(f, \mathcal{V}, \varnothing, \varnothing, 0; \mathcal{V}, \varepsilon, \delta)$ (Algorithm 3) terminates and returns tuple $(\widehat{V}_{1:H}, \widehat{\mathcal{V}}_{1:H}, \mathcal{C}_{1:H}, \mathcal{B}_{1:H}, t_{1:H})$ such that*

$$\forall h \in [H], \quad V_h^{\tilde{\pi}} \in \widehat{\mathcal{V}}_h,$$

where $\tilde{\pi}_{1:H} \in \Pi_{\mathsf{S}}$ is the stochastic policy defined recursively via

$$\forall x \in \mathcal{X}, \quad \tilde{\boldsymbol{\pi}}_\tau(x) \in \arg\max_{a\in\mathcal{A}} \begin{cases} \widehat{\boldsymbol{Q}}_\tau(x,a), & \text{if } \|\widehat{\boldsymbol{Q}}_\tau(x,\cdot) - \mathcal{P}_\tau[V^{\tilde{\pi}}_{\tau+1}](x,\cdot)\|_\infty \le 4\varepsilon, \\ \mathcal{P}_\tau[V^{\tilde{\pi}}_{\tau+1}](x,a), & \text{otherwise,} \end{cases} \quad \text{for } \tau = H,\dots,1, \tag{25}$$

where $\widehat{\boldsymbol{Q}}_\tau(x,a) \coloneqq \widehat{\boldsymbol{\mathcal{P}}}_{\tau,\varepsilon,\delta'}[\widehat{V}_{\tau+1}](x,a)$ is a realization of the stochastic output of the $\widehat{\boldsymbol{\mathcal{P}}}$ operator in Algorithm 5 given input (x,a), and δ' is as in Algorithm 3. Furtherore, we have $\tilde{\pi} \in \Pi_{4\varepsilon}$.

The proof of the lemma is in Appendix H.5.

Equipped with the preceding lemmas, we now state the main technical result of this section, Theorem H.1, a generalization of Theorem 4.1 which holds under relaxed V^{π}-realizability (Assumption H.1). The proof is in Appendix H.6.

Theorem H.1 (Guarantee for RVFS under relaxed V^{π}-realizability)**.** *Let $\delta, \varepsilon \in (0,1)$ be given, and suppose that Assumption 4.1 (pushforward coverability) holds with parameter $C_{\mathsf{push}} > 0$. Let $f \in \mathcal{V}$ be arbitrary, and assume that $\mathcal{V}$ that satisfies Assumption H.1 with $\varepsilon_{\mathsf{real}} = 4\varepsilon$. Then, with probability at least $1 - 5\delta$, $\mathsf{RVFS}_0(f, \mathcal{V}, \varnothing, \varnothing, 0; \mathcal{V}, \varepsilon, \delta)$ (Algorithm 3) terminates and returns value functions $\widehat{V}_{1:H}$ that satisfy*

$$\forall h \in [H], \quad \mathbb{E}^{\widehat{\pi}}[D_{\mathrm{tv}}(\widehat{\pi}_h(\boldsymbol{x}_h), \tilde{\pi}_h(\boldsymbol{x}_h))] \le \frac{\varepsilon}{4H^3 C_{\mathsf{push}}},$$

where $\widehat{\boldsymbol{\pi}}_h(x) \in \arg\max_{a\in\mathcal{A}} \widehat{\boldsymbol{\mathcal{P}}}_{h,\varepsilon,\delta'}[\widehat{V}_{h+1}](x,a)$ for all $h \in [H]$, with $\tilde{\pi} \in \Pi_{4\varepsilon}$ defined as in Lemma H.4 and δ' defined as in Algorithm 3. Furthermore, the number of episodes is bounded by

$$\widetilde{O}(C_{\mathsf{push}}^8 H^{10} A \cdot \varepsilon^{-13}).$$

Next, we state a guarantee for the outer-level algorithm, $\mathsf{RVFS.bc}$, under relaxed V^{π}-realizability. Recall that $\mathsf{RVFS.bc}$ invokes RVFS_0, then extracts an executable policy by applying the $\mathsf{BehaviorCloning}$ algorithm (see Appendix L), with the "expert" policy set to be the output of RVFS.

Theorem H.2 (Main guarantee of $\mathsf{RVFS.bc}$)**.** *Let $\delta, \varepsilon \in (0,1)$ be given, and define $\varepsilon_{\mathsf{RVFS}} = \varepsilon H^{-1}/48$. Suppose that*

- *Assumption 4.1 (pushforward coverability) holds with parameter $C_{\mathsf{push}} > 0$;*
- *the function class $\mathcal{V}$ satisfies Assumption H.1 with $\varepsilon_{\mathsf{real}} = 1$ (i.e. all π-realizability); and*
- *the policy class Π satisfies Assumption 4.3.*

Then, with probability at least $1 - \delta$, $\widehat{\pi}_{1:H} = \mathsf{RVFS.bc}(\Pi, \mathcal{V}, \varepsilon, \delta)$ (Algorithm 4) satisfies

$$J(\pi^\star) - J(\widehat{\pi}_{1:H}) \le \varepsilon. \tag{26}$$

Furthermore, the total number sample complexity in the RLLS framework is bounded by

$$\widetilde{O}\left(C_{\mathsf{push}}^8 H^{23} A \epsilon^{-13}\right).$$

The proof is in Appendix H.7. Note that Theorem H.2 is a restatement of Theorem 4.1 in **Setup II** (restated for convenience). As a result, Theorem 4.1 is an immediate corollary.

Proof of Theorem 4.1. The result follows from Theorem H.2, since Assumption 4.5 is stronger than Assumption H.1. □

H.2 Proof of Lemma H.1 (Number of Test Failures)

Proof of Lemma H.1. Fix $h \in [H]$. We note that the size of $\mathcal{C}_h$ corresponds to the number of times the test in Line 14 fails for $\ell = h$ throughout the execution of $\mathsf{RVFS}_0(f, \mathcal{V}, \varnothing, \varnothing; \mathcal{V}, \varepsilon, \delta)$.

Let $M \coloneqq \lceil 8\varepsilon^{-1} C_{\mathsf{push}} H \rceil$ denote the desired upper bound on $|\mathcal{C}_h|$. Suppose that the test in Line 14 fails at least twice for $\ell = h$ (if the test fails at most twice, then $|\mathcal{C}_h| \le 2$ and so (23) holds for $\ell = h$ trivially), and let

$$(x_{h-1}^{(1)}, a_{h-1}^{(1)}, \widehat{V}_h^{(1)}, \widehat{\mathcal{V}}_h^{(1)}, t_h^{(1)}), (x_{h-1}^{(2)}, a_{h-1}^{(2)}, \widehat{V}_h^{(2)}, \widehat{\mathcal{V}}_h^{(2)}, t_h^{(2)}), \dots$$

denote the elements of the set $\mathcal{B}_h$ in the order at which they are added to the latter in Line 15 of Algorithm 3. Note that $|\mathcal{B}_h| = |\mathcal{C}_h|$. Note also that $t_h^{(i)}$ represents the number of times the subroutine $\widehat{\boldsymbol{\mathcal{P}}}_{h-1,\varepsilon,\delta'}$ has been called in the test of Line 14 throughout the execution of RVFS_0 and up to the time the test failed for $(x_{h-1}^{(i)}, a_{h-1}^{(i)})$. We will use this fact in a concentration argument in the sequel.

By definition of $(\widehat{\mathcal{V}}_h^{(i)})$ and Lemma B.4 (Freedman's inequality) instantiated with

- $\mathcal{Q} = \{\widehat{V}_h^{(i)} - f_h : f \in \widehat{\mathcal{V}}_h^{(i)}\}$;
- $\boldsymbol{y}_h = \boldsymbol{x}_h$;
- $B = H$; and
- $n = N_{\mathsf{reg}} \cdot i$;

and the union bound over $i \in [M \wedge |\mathcal{C}_h|]$, we get that there is an event $\mathcal{E}_h$ of probability at least $1 - \delta/(2H)$ under which

$$\begin{aligned}\forall i \in [M \wedge |\mathcal{C}_h|], \forall f \in \widehat{\mathcal{V}}_h^{(i)}, \quad & \sum_{j<i} \mathbb{E}\Big[(\widehat{V}_h^{(i)}(\boldsymbol{x}_h) - f_h(\boldsymbol{x}_h))^2 \mid \boldsymbol{x}_{h-1} = x_{h-1}^{(j)}, \boldsymbol{a}_{h-1} = a_{h-1}^{(j)}\Big] \\ & \le \tilde{\varepsilon}_{\mathsf{reg}}^2 \coloneqq 2\varepsilon_{\mathsf{reg}}^2 + \frac{4H^2 \log(4MH|\mathcal{V}|/\delta)}{N_{\mathsf{reg}}}. \end{aligned} \tag{27}$$

Now, define $f_h^{(i)} \in \arg\max_{f \in \widehat{\mathcal{V}}_h^{(i)}} \left|\mathbb{E}\left[\widehat{V}_h^{(i)}(\boldsymbol{x}_h) - f_h(\boldsymbol{x}_h) \mid \boldsymbol{x}_{h-1} = x_{h-1}^{(i)}, \boldsymbol{a}_{h-1} = a_{h-1}^{(i)}\right]\right|$. From (27), we have that under $\mathcal{E}_h$:

$$\forall i \in [M \wedge |\mathcal{C}_h|], \quad \sum_{j<i} \mathbb{E}\Big[(\widehat{V}_h^{(i)}(\boldsymbol{x}_h) - f_h^{(i)}(\boldsymbol{x}_h))^2 \mid \boldsymbol{x}_{h-1} = x_{h-1}^{(j)}, \boldsymbol{a}_{h-1} = a_{h-1}^{(j)}\Big] \le \tilde{\varepsilon}_{\mathsf{reg}}^2. \tag{28}$$

We now use this to bound the number of times the test in Line 14 fails for $\ell = h$. Suppose for the sake of contradiction that the test fails at least N times for some $N \ge M$ (i.e. $|\mathcal{C}_h| = N \ge M$). Conditioned on $\mathcal{E}_h$, we have by Lemma B.8 and Eq. (28),

$$\begin{aligned} & \min_{i \in [M]} \sup_{f \in \widehat{\mathcal{V}}_h^{(i)}} \left|\mathbb{E}\left[\widehat{V}_h^{(i)}(\boldsymbol{x}_h) - f_h(\boldsymbol{x}_h) \mid \boldsymbol{x}_{h-1} = x_{h-1}^{(i)}, \boldsymbol{a}_{h-1} = a_{h-1}^{(i)}\right]\right| \\ & = \min_{i \in [M]} \left|\mathbb{E}\left[\widehat{V}_h^{(i)}(\boldsymbol{x}_h) - f_h^{(i)}(\boldsymbol{x}_h) \mid \boldsymbol{x}_{h-1} = x_{h-1}^{(i)}, \boldsymbol{a}_{h-1} = a_{h-1}^{(i)}\right]\right|, \\ & \le 2\left(\frac{C_{\mathsf{push}}}{M^2} M \tilde{\varepsilon}_{\mathsf{reg}}^2 \log(2M)\right)^{1/2} + \frac{2C_{\mathsf{push}} H}{M}. \end{aligned}$$

Now, substituting the expression of $\tilde{\varepsilon}_{\mathsf{reg}}^2$ in (27) and using the definition of $\varepsilon_{\mathsf{reg}}^2$ in Line 6 of Algorithm 3, we get

$$\begin{aligned} & = 2\left(\frac{C_{\mathsf{push}}}{M} \cdot \left(2\varepsilon_{\mathsf{reg}}^2 + \frac{4MH^2 \log(4MH|\mathcal{V}|/\delta)}{N_{\mathsf{reg}}}\right)\right)^{1/2} + \frac{2C_{\mathsf{push}} H}{M}, \\ & \le 2\left(\frac{C_{\mathsf{push}}}{M} \cdot \left(\frac{22MH^2 \log(8M^2 H|\mathcal{V}|^2/\delta)}{N_{\mathsf{reg}}} + \frac{68MH^3 \log(8M^6 N_{\mathsf{test}}^2 H^8/\delta)}{N_{\mathsf{test}}}\right)\right)^{1/2} + \frac{2C_{\mathsf{push}} H}{M}, \\ & \le \varepsilon, \end{aligned} \tag{29}$$

where the last inequality uses that $M = \lceil 8\varepsilon^{-1} C_{\mathsf{push}} H \rceil$ and

$$N_{\mathsf{reg}} = 2^8 M^2 \varepsilon^{-1} \log(8|\mathcal{V}|^2 H M^2 \delta^{-1}) \quad \text{and} \quad N_{\mathsf{test}} = 2^8 M^2 H \varepsilon^{-1} \log(8M^6 H^8 \varepsilon^{-2} \delta^{-1});$$

see Line 5 of Algorithm 3.

On the other hand, by Lemma G.2, there is an event $\mathcal{E}'_h$ of probability at least $1-\delta/(2MH)$ under which for all $f \in \mathcal{V}$, all $i \in [M]$, and δ' as in Algorithm 3:

$$\begin{aligned}
&\left|\widehat{\boldsymbol{\mathcal{P}}}_{h-1,\varepsilon,\delta'}[\widehat{V}_h^{(i)}](x_{h-1}^{(i)}, a_{h-1}^{(i)}) - \widehat{\boldsymbol{\mathcal{P}}}_{h-1,\varepsilon,\delta'}[f_h](x_{h-1}^{(i)}, a_{h-1}^{(i)})\right| \\
&\leq \left|\mathcal{P}_{h-1}[\widehat{V}_h^{(i)} - f_h](x_{h-1}^{(i)}, a_{h-1}^{(i)})\right| + \varepsilon \cdot \sqrt{2\log_{1/\delta'}(4MAH|\mathcal{V}|(t_h^{(i)})^2/\delta)}, \\
&= \left|\mathcal{P}_{h-1}[\widehat{V}_h^{(i)} - f_h](x_{h-1}^{(i)}, a_{h-1}^{(i)})\right| + \varepsilon \cdot \beta(t_h^{(i)}),
\end{aligned} \tag{30}$$

where $\beta(t_h^{(i)})$ is as in Algorithm 3. Thus, under $\mathcal{E}'_h$, the test in Line 14 fails for $\ell = h$ at least M times only if

$$\begin{aligned}
\forall i \in [M], \quad \varepsilon &< \sup_{f \in \widehat{\mathcal{V}}_h^{(i)}} \left|(\widehat{\boldsymbol{\mathcal{P}}}_{h-1,\varepsilon,\delta'}[\widehat{V}_h^{(i)}] - \widehat{\boldsymbol{\mathcal{P}}}_{h-1,\varepsilon,\delta'}[f_h])(x_{h-1}^{(i)}, a_{h-1}^{(i)})\right| \\
&\quad - \varepsilon \cdot \beta(t_h^{(i)}), \\
&\leq \sup_{f \in \widehat{\mathcal{V}}_h^{(i)}} \left|\mathbb{E}\left[\widehat{V}_h^{(i)}(\boldsymbol{x}_h) - f_h(\boldsymbol{x}_h) \mid \boldsymbol{x}_{h-1} = x_{h-1}^{(i)}, \boldsymbol{a}_{h-1} = a_{h-1}^{(i)}\right]\right| \quad \text{(by (30))}, \\
&< \sup_{f \in \widehat{\mathcal{V}}_h^{(i)}} \left|\mathbb{E}\left[\widehat{V}_h^{(i)}(\boldsymbol{x}_h) - f_h(\boldsymbol{x}_h) \mid \boldsymbol{x}_{h-1} = x_{h-1}^{(i)}, \boldsymbol{a}_{h-1} = a_{h-1}^{(i)}\right]\right|.
\end{aligned}$$

Unless $N < M$, this is a contradiction to Eq. (29). We conclude that under the event $\mathcal{E}_h \cap \mathcal{E}'_h$, the test in Line 14 fails at most $N < M = \lceil 8\varepsilon^{-1} C_{\mathsf{push}} H \rceil$ times for $\ell = h$, and so under $\mathcal{E}_1 \cap \mathcal{E}'_1 \cap \cdots \cap \mathcal{E}_H \cap \mathcal{E}_H$, we have

$$\forall h \in [H], \quad |\mathcal{C}_h| \leq \lceil 8\varepsilon^{-1} C_{\mathsf{push}} H \rceil.$$

By the union bound, we have $\mathbb{P}[\mathcal{E}_1 \cap \mathcal{E}'_1 \cap \cdots \cap \mathcal{E}_H \cap \mathcal{E}'_H] \geq 1 - \delta$, which completes the proof. □

H.3 Proof of Lemma H.2 (Consequence of Passing the Tests)

Proof of Lemma H.2. Let $h \in [H]$ be given. Fix $\ell \in [h+1\,..\,H]$ and let $\boldsymbol{x}_{\ell-1}^{(1)}, \boldsymbol{x}_{\ell-1}^{(2)}, \dots$ denote the sequence of states used in the tests of Line 14 throughout the execution of RVFS_0; we assume that the sequence is *ordered* in the sense that if $i < j$, then $\boldsymbol{x}_{\ell-1}^{(i)}$ is used in the test of Line 14 before $\boldsymbol{x}_{\ell-1}^{(j)}$. Let $\boldsymbol{T}_\ell \in \mathbb{N} \cup \{+\infty\}$ be the random variable representing the total number of times the operator $\widehat{\boldsymbol{\mathcal{P}}}_{\ell-1,\varepsilon,\delta'}$ is invoked in Line 14 throughout the execution of RVFS_0 ($\boldsymbol{T}_\ell$ is also the random length of the sequence $\boldsymbol{x}_{\ell-1}^{(1)}, \boldsymbol{x}_{\ell-1}^{(2)}, \dots$; if RVFS_0 terminates, then $\boldsymbol{T}_\ell$ is finite. The first step of the proof will be to show that under the event $\mathcal{E}$ of Lemma H.1, $\boldsymbol{T}_\ell$ is no larger than $M^3 N_{\mathsf{test}} H^3$ at any point during the execution of RVFS_0. This will help us establish key concentration results, leading to the desired inequality (24).

Bounding $\boldsymbol{T}_\ell$ under $\mathcal{E}$. First, note that under the event $\mathcal{E}$ of Lemma H.1, we have that for any $\tau \in [H]$,

$$|\mathcal{C}_\tau| \leq M := \lceil 8\varepsilon^{-1} C_{\mathsf{push}} H \rceil, \tag{31}$$

and so RVFS_τ gets called at most M times throughout the execution of RVFS_0. For the rest of this paragraph, we condition on $\mathcal{E}$ and fix $\tau \in [0\,..\,H]$. Within any given call to RVFS_τ (throughout the execution of RVFS_0), the operator $\widehat{\boldsymbol{\mathcal{P}}}_{\ell-1,\varepsilon,\delta'}$ is invoked at most

$$\underbrace{|\mathcal{C}_\tau| N_{\mathsf{test}} H}_{\text{Due to the for-loops in Line 8, Line 9, \& Line 10}} \qquad \times \qquad \underbrace{HM}_{\text{Number of times } \mathsf{RVFS}_\tau \text{ returns to Line 8 (see below)}} \qquad \leq M^2 N_{\mathsf{test}} H^2$$

times. This is because the for-loop in Line 8 of RVFS_τ resumes whenever a test in Line 14 fails for one of the layers $\tau+1, \dots, H$ (see Line 18) once the recursive calls return, and the total number of test failures across all these layers is bounded by HM (by (31)). Now, since RVFS_τ gets called at most M times throughout the execution of RVFS_0 (as argued in the prequel), the total number of times the operator $\widehat{\boldsymbol{\mathcal{P}}}_{\ell-1,\varepsilon,\delta'}$ is invoked in Line 14 within RVFS_τ is at most

$$M^3 N_{\mathsf{test}} H^2.$$

Finally, the total number of times the operator $\widehat{\boldsymbol{\mathcal{P}}}_{\ell-1,\varepsilon,\delta'}$ is called in Line 14 throughout the execution of RVFS_0 is at most H times larger (accounting for the contributions from RVFS_τ for all $\tau \in [H]$); that is, it is at most $M^3 N_{\mathsf{test}} H^3$. We conclude that the random variable $\boldsymbol{T}_\ell$ satisfies

$$\boldsymbol{T}_\ell \leq M^3 N_{\mathsf{test}} H^3 \tag{32}$$

under $\mathcal{E}$.

Specifying $\mathcal{E}'_h$. In this paragraph, we no longer condition on $\mathcal{E}$. We will specify the event $\mathcal{E}'_h$ in the lemma statement. Let δ' be defined as in Algorithm 3. By Lemma G.2, we have that there is an event $\mathcal{E}'_{h,\ell}$ of probability at least $1 - \delta/(2H^2)$ under which:

$$\begin{aligned}
\forall i \in [\boldsymbol{T}_\ell], \forall a_{\ell-1} \in \mathcal{A} \,:\, & \sup_{f \in \widehat{\mathcal{V}}_\ell} |(\widehat{\boldsymbol{\mathcal{P}}}_{\ell-1,\varepsilon,\delta'}[\widehat{V}_\ell] - \widehat{\boldsymbol{\mathcal{P}}}_{\ell-1,\varepsilon,\delta'}[f_\ell])(\boldsymbol{x}^{(i)}_{\ell-1}, a_{\ell-1})| - \varepsilon - \varepsilon \cdot \beta(\boldsymbol{T}_\ell) \\
&= \sup_{f \in \widehat{\mathcal{V}}_\ell} |(\widehat{\boldsymbol{\mathcal{P}}}_{\ell-1,\varepsilon,\delta'}[\widehat{V}_\ell] - \widehat{\boldsymbol{\mathcal{P}}}_{\ell-1,\varepsilon,\delta'}[f_\ell])(\boldsymbol{x}^{(i)}_{\ell-1}, a_{\ell-1})| - \varepsilon - \varepsilon \cdot \sqrt{2 \log_{1/\delta'}(8AH^2 M |\mathcal{V}| \boldsymbol{T}_\ell^2/\delta)}, \\
&\geq \sup_{f \in \widehat{\mathcal{V}}_\ell} |(\mathcal{P}_{\ell-1}[\widehat{V}_\ell] - \mathcal{P}_{\ell-1}[f_\ell])(\boldsymbol{x}^{(i)}_{\ell-1}, a_{\ell-1})| - \varepsilon - \varepsilon \cdot \sqrt{2 \log_{1/\delta'}(8AH^2 M |\mathcal{V}| \boldsymbol{T}_\ell^2/\delta)} \\
&\quad - \varepsilon \cdot \sqrt{2 \log_{1/\delta'}(4AH^2 M |\mathcal{V}| i^2/\delta)}, \quad \text{(Lemma G.2)} \\
&\geq \sup_{f \in \widehat{\mathcal{V}}_\ell} |(\mathcal{P}_{\ell-1}[\widehat{V}_\ell] - \mathcal{P}_{\ell-1}[f_\ell])(\boldsymbol{x}^{(i)}_{\ell-1}, a_{\ell-1})| - \varepsilon - 2\varepsilon \cdot \sqrt{2 \log_{1/\delta'}(8AH^2 M |\mathcal{V}| \boldsymbol{T}_\ell^2/\delta)}.
\end{aligned} \tag{33}$$

On the other hand, for $k \in [\boldsymbol{T}_\ell - N_{\mathsf{test}} + 1]$, we have by Lemma B.4 (Freedman's inequality) instantiated with

- $n = N_{\mathsf{test}}$ and $\boldsymbol{y}_i = \mathbb{I}\left\{\sup_{f \in \widehat{\mathcal{V}}_\ell} \max_{a \in \mathcal{A}} \left|(\mathcal{P}_{\ell-1}[\widehat{V}_\ell] - \mathcal{P}_{\ell-1})[f_\ell](\boldsymbol{x}^{(k+i)}_{\ell-1}, a)\right| > 3\varepsilon\right\}$, for all $i \in [N_{\mathsf{test}}]$;
- $\mathcal{Q} = \{\mathrm{id}\}$;
- $B = 1$; and
- $\lambda = 1$;

that there is an event $\mathcal{E}''_{h,\ell,k}$ of probability at least $1 - \delta/(4k^2 H^2)$ under which

$$\begin{aligned}
&\sum_{0 \leq i < N_{\mathsf{test}}} \mathbb{P}\left[\sup_{f \in \widehat{\mathcal{V}}_\ell} \max_{a \in \mathcal{A}} \left|(\mathcal{P}_{\ell-1}[\widehat{V}_\ell] - \mathcal{P}_{\ell-1}[f_\ell])(\boldsymbol{x}^{(k+i)}_{\ell-1}, a)\right| > 3\varepsilon\right] \\
&\leq 4 \log(8H^2 \boldsymbol{T}_\ell^2/\delta) + \sum_{0 \leq i < N_{\mathsf{test}}} \mathbb{I}\left\{\sup_{f \in \widehat{\mathcal{V}}_\ell, a \in \mathcal{A}} |(\mathcal{P}_{\ell-1}[\widehat{V}_\ell] - \mathcal{P}_{\ell-1}[f_\ell])(\boldsymbol{x}^{(k+i)}_{\ell-1}, a)| > 3\epsilon\right\}.
\end{aligned}$$

Now, let $\mathcal{E}''_{h,\ell} \coloneqq \bigcap_{k \in [\boldsymbol{T}_\ell - N_{\mathsf{test}} + 1]} \mathcal{E}''_{h,\ell,k}$. By the union bound and the fact that $\sum_{k \geq 1} 1/k^2 = \pi^2/6 \leq 2$, we have that $\mathbb{P}[\mathcal{E}''_{h,\ell}] \geq 1 - \delta/(2H^2)$. Furthermore, under $\mathcal{E}''_{h,\ell}$, we have

$$\begin{aligned}
&\forall k \in [\boldsymbol{T}_\ell - N_{\mathsf{test}} + 1], \\
&\sum_{0 \leq i < N_{\mathsf{test}}} \mathbb{P}\left[\sup_{f \in \widehat{\mathcal{V}}_\ell} \max_{a \in \mathcal{A}} \left|(\mathcal{P}_{\ell-1}[\widehat{V}_\ell] - \mathcal{P}_{\ell-1}[f_\ell])(\boldsymbol{x}^{(k+i)}_{\ell-1}, a)\right| > 3\varepsilon\right] \\
&\leq 4 \log(8H^2 \boldsymbol{T}_\ell^2/\delta) + \sum_{0 \leq i < N_{\mathsf{test}}} \mathbb{I}\left\{\sup_{f \in \widehat{\mathcal{V}}_\ell, a \in \mathcal{A}} |(\mathcal{P}_{\ell-1}[\widehat{V}_\ell] - \mathcal{P}_{\ell-1}[f_\ell])(\boldsymbol{x}^{(k+i)}_{\ell-1}, a)| > 3\epsilon\right\}.
\end{aligned} \tag{34}$$

We define $\mathcal{E}'_h \coloneqq \mathcal{E}'_{h,1} \cap \mathcal{E}''_{h,1} \cap \cdots \cap \mathcal{E}'_{h,H} \cap \mathcal{E}''_{h,H}$. Note that by the union bound, we have $\mathbb{P}[\mathcal{E}'_h] \geq 1 - \frac{\delta}{H}$ as desired.

Termination of RVFS_h under $\mathcal{E} \cap \mathcal{E}'_h$. We now show that under $\mathcal{E} \cap \mathcal{E}'_h$, if RVFS_h terminates, its output satisfies (24). For the rest of the proof, we condition on $\mathcal{E} \cap \mathcal{E}'_h$. Suppose that RVFS_h terminates and returns $(\widehat{V}_{h:H}, \widehat{\mathcal{V}}_{h:H}, \mathcal{C}_{h:H}, \mathcal{B}_{h:H}, t_{h:H})$. In this case, the value function $\widehat{V}_\ell$ must have passed the tests in Line 14 for all $(x_{h-1}, a_{h-1}) \in \mathcal{C}_h$, $n \in N_{\mathsf{test}}$, and $a_{\ell-1} \in \mathcal{A}$. Fix $(x_{h-1}, a_{h-1}) \in \mathcal{C}_h$ and let $k \in [\boldsymbol{T}_\ell - N_{\mathsf{test}} \cdot A + 1]$ be such that $(\boldsymbol{x}^{k+j}_{\ell-1})_{j \in [0..N_{\mathsf{test}}-1]}$ represents a subsequence of states that pass the tests in Line 14 at layer ℓ for (x_{h-1}, a_{h-1}) within the call to RVFS_h. The fact that the sequence $(\boldsymbol{x}^{(i)}_{\ell-1})_{i \geq 1}$ is ordered (see definition in the first paragraph of this proof) and that $(\boldsymbol{x}^{k+j}_{\ell-1})_{j \in [0..N_{\mathsf{test}}-1]}$ pass the tests imply that

1. The states $(\boldsymbol{x}^{(k+i)}_{\ell-1})_{i \in [0..N_{\mathsf{test}}-1]}$ at layer $\ell - 1$ are i.i.d., and are obtained by rolling out with $\widehat{\pi}_{h:H}$ starting from (x_{h-1}, a_{h-1}); and
2. The test in Line 14 succeeds for all $(\boldsymbol{x}^{(k+j)}_{\ell-1})_{j \in [0..N_{\mathsf{test}}-1]}$; that is

$$\begin{aligned}
\forall j \in [0 .. N_{\mathsf{test}} - 1], \forall a_{\ell-1} \in \mathcal{A}, \quad & \sup_{f \in \widehat{\mathcal{V}}_\ell} |(\widehat{\boldsymbol{\mathcal{P}}}_{\ell-1,\varepsilon,\delta'}[\widehat{V}_\ell] - \widehat{\boldsymbol{\mathcal{P}}}_{\ell-1,\varepsilon,\delta'}[f_\ell])(\boldsymbol{x}^{(k+j)}_{\ell-1}, a_{\ell-1})| \\
& \leq \varepsilon + \varepsilon \cdot \beta(k+j), \\
& \leq \varepsilon + \varepsilon \cdot \sqrt{2 \log_{1/\delta'}(8AM|\mathcal{V}|(k+j)^2/\delta)}, \\
& \leq \varepsilon + \varepsilon \cdot \sqrt{2 \log_{1/\delta'}(8AM|\mathcal{V}|\boldsymbol{T}_\ell^2/\delta)}.
\end{aligned}$$

This implies that

$$\begin{aligned}
& \forall i \in [0 .. N_{\mathsf{test}} - 1], \forall a_{\ell-1} \in \mathcal{A} \ : \\
& \sup_{f \in \widehat{\mathcal{V}}_\ell} |\mathcal{P}_{\ell-1}[\widehat{V}_\ell - f_\ell](\boldsymbol{x}^{(k+i)}_{\ell-1}, a_{\ell-1})| - 3\varepsilon \\
& \leq \sup_{f \in \widehat{\mathcal{V}}_\ell} |\mathcal{P}_{\ell-1}[\widehat{V}_\ell - f_\ell](\boldsymbol{x}^{(k+i)}_{\ell-1}, a_{\ell-1})| - \varepsilon - 2\varepsilon \cdot \sqrt{2 \log_{1/\delta'}(4AMH^2|\mathcal{V}|\boldsymbol{T}_\ell^2 \delta)}, && (35) \\
& \overset{(33)}{\leq} \sup_{f \in \widehat{\mathcal{V}}_\ell} |(\widehat{\boldsymbol{\mathcal{P}}}_{\ell-1,\varepsilon,\delta'}[\widehat{V}_\ell] - \widehat{\boldsymbol{\mathcal{P}}}_{\ell-1,\varepsilon,\delta'}[f_\ell])(\boldsymbol{x}^{(k+i)}_{\ell-1}, a_{\ell-1})| - \varepsilon - \varepsilon \cdot \sqrt{2 \log_{1/\delta'}(4AMH^2|\mathcal{V}|\boldsymbol{T}_\ell^2/\delta)}, \\
& \leq 0. \quad \text{(by Item 2)} && (36)
\end{aligned}$$

where (35) follows by (32) and the choice of δ' in Algorithm 3.

Now, by Item 1, we have that $\boldsymbol{x}^{(k+i)}_{\ell-1}$ has probability law $\mathbb{P}^{\widehat{\pi}_{h:H}}[\cdot \mid \boldsymbol{x}_{h-1} = x_{h-1}, \boldsymbol{a}_{h-1} = a_{h-1}]$ for all $i \in [0 .. N_{\mathsf{test}} - 1]$, and so by (34), we have:

$$\begin{aligned}
& \mathbb{P}^{\widehat{\pi}}\left[\sup_{f \in \widehat{\mathcal{V}}_\ell} \max_{a \in \mathcal{A}} \left|(\mathcal{P}_{\ell-1}[\widehat{V}_\ell] - \mathcal{P}_{\ell-1}[f_\ell])(\boldsymbol{x}_{\ell-1}, a)\right| > 3\varepsilon \mid \boldsymbol{x}_{h-1} = x_{h-1}, \boldsymbol{a}_{h-1} = a_{h-1} \right] \\
& \leq \frac{4 \log(8H^2\boldsymbol{T}_\ell^2/\delta)}{N_{\mathsf{test}}} + \frac{1}{N_{\mathsf{test}}} \sum_{0 \leq i < N_{\mathsf{test}}} \mathbb{I}\left\{ \sup_{f \in \widehat{\mathcal{V}}_\ell, a \in \mathcal{A}} |\mathcal{P}_{\ell-1}[\widehat{V}_\ell] - \mathcal{P}_{\ell-1}[f_\ell])(\boldsymbol{x}^{(k+i)}_{\ell-1}, a)| > 3\epsilon \right\}, \\
& \leq \frac{4 \log(8M^6 N_{\mathsf{test}}^2 H^8/\delta)}{N_{\mathsf{test}}} \quad \text{(using (32) and the fact that all the tests pass, i.e. (36))}.
\end{aligned}$$

Concluding. We have established that under $\mathcal{E} \cap \mathcal{E}'_h$, we have for all $\ell \in [h+1 .. H]$ and all $(x_{h-1}, a_{h-1}) \in \mathcal{C}_h$:

$$\mathbb{P}^{\widehat{\pi}}\left[\sup_{f \in \widehat{\mathcal{V}}_\ell} \max_{a \in \mathcal{A}} \left|(\mathcal{P}_{\ell-1}[\widehat{V}_\ell] - \mathcal{P}_{\ell-1}[f_\ell])(\boldsymbol{x}_{\ell-1}, a)\right| > 3\varepsilon \mid \boldsymbol{x}_{h-1} = x_{h-1}, \boldsymbol{a}_{h-1} = a_{h-1} \right] \leq \frac{4 \log(8M^6 N_{\mathsf{test}}^2 H^8/\delta)}{N_{\mathsf{test}}},$$

Furthermore, we have $\mathbb{P}[\mathcal{E}'_h] \geq 1 - \delta/H$. This completes the proof. □

H.4 Proof of Lemma H.3 (Value Function Regression Guarantee)

Proof of Lemma H.3. Fix $\pi \in \Pi' \subseteq \Pi_{\mathsf{S}}$ and $k \geq 1$, and consider the kth call to RVFS_h as per the lemma statement, and let $\mathcal{S}_k$ be the state of RVFS_0 during the k^{th} call to RVFS_h and immediately before Line 20, i.e. immediately before gathering data for the regression step in RVFS_h.

Relating the regression targets to $V_h^{\widehat{\pi}}$. Observe that $\widehat{V}_h$ is the least-squares solution of the objective in Line 26, where the targets are empirical estimates of $V_h^{\widehat{\pi}}$. In particular, if we let $\{\mathcal{D}_h(x,a) : (x,a) \in \mathcal{C}_h\}$ be the datasets in the definition of $\widehat{\mathcal{V}}_h$ in (16), then for any $(x_{h-1}, a_{h-1}) \in \mathcal{C}_h$ and $(x_h, v_h) \in \mathcal{D}_h(x_{h-1}, a_{h-1})$, the target v_h satisfies

$$v_h = \widehat{V}_h(x_h),$$

where $\widehat{V}_h(x_h)$ is an empirical estimate of $V_h^{\widehat{\pi}}(x_h)$ obtained by sampling $N_{\mathsf{est}}(|\mathcal{C}_h|) = N_{\mathsf{est}}(k)$ episodes (for $N_{\mathsf{est}}(\cdot)$ defined as in Algorithm 3) by rolling out with $\widehat{\pi}$ after starting from x_h and playing action a at layer h; note that $|\mathcal{C}_h| = k$ because we are considering the kth call to RVFS_h. Thus, by Hoeffding's inequality and the union bound over $(x_{h-1}, a_{h-1}) \in \mathcal{C}_h$ and $(x_h, -) \in \mathcal{D}_h(x_{h-1}, a_{h-1})$, there is an event $\mathcal{E}''_{h,k}(\mathcal{S}_k)$ of probability at least $1 - \delta/(8k^2 H)$ under which

$$\begin{aligned}
&\forall (x_{h-1}, a_{h-1}) \in \mathcal{C}_h, \forall (x_h, -) \in \mathcal{D}_h(x_{h-1}, a_{h-1}) : \\
&|V_h^{\widehat{\pi}}(x_h) - \widehat{V}_h(x_h)| \leq H\sqrt{\frac{2\log(8|\mathcal{C}_h| N_{\mathsf{reg}} H k^2/\delta)}{N_{\mathsf{est}}(k)}} \leq H\sqrt{\frac{2\log(8 N_{\mathsf{reg}} H k^3/\delta)}{N_{\mathsf{est}}(k)}},
\end{aligned} \tag{37}$$

where N_{reg} is as in Line 5, and the last inequality follows by $|\mathcal{C}_h| \leq k$ since we are considering the kth call to RVFS_h. Thus, under $\mathcal{E}''_{h,k}$, we have

$$\begin{aligned}
&\forall (x_{h-1}, a_{h-1}) \in \mathcal{C}_h, \forall (x_h, v_h) \in \mathcal{D}_h(x_{h-1}, a_{h-1}) : \\
&|V_h^{\widehat{\pi}}(x_h) - v_h| = \left|V_h^{\widehat{\pi}}(x_h) - \widehat{V}_h(x_h)\right| \leq H\sqrt{\frac{2\log(8 N_{\mathsf{reg}} H k^3/\delta)}{N_{\mathsf{est}}(k)}} \leq \frac{H}{N_{\mathsf{reg}}},
\end{aligned} \tag{38}$$

where the second-to-last inequality is by (37) and the last inequality follows by the choice of N_{est} in Algorithm 3.

Bounding the discrepancy $V_h^{\widehat{\pi}} - V_h^{\pi}$. On the other hand, by the performance difference lemma, the value function $V_h^{\widehat{\pi}}$ satisfies:

$$\begin{aligned}
\forall x \in \mathcal{X}, \quad |V_h^{\widehat{\pi}}(x) - V_h^{\pi}(x)| &\leq \sum_{\tau=h}^{H} \mathbb{E}^{\widehat{\pi}}[|Q_\tau^{\pi}(\boldsymbol{x}_\tau, \boldsymbol{\pi}_\tau(\boldsymbol{x}_\tau)) - Q_\tau^{\pi}(\boldsymbol{x}_\tau, \widehat{\boldsymbol{\pi}}_\tau(\boldsymbol{x}_\tau))| \mid \boldsymbol{x}_h = x], \\
&\leq H \sum_{\tau=h}^{H} \mathbb{E}^{\widehat{\pi}}[D_{\text{tv}}(\widehat{\pi}_\tau(\boldsymbol{x}_\tau), \pi_\tau(\boldsymbol{x}_\tau)) \mid \boldsymbol{x}_h = x].
\end{aligned} \tag{39}$$

Now, let $(x_{h-1}^{(1)}, a_{h-1}^{(1)}), (x_{h-1}^{(2)}, a_{h-1}^{(2)}), \dots$ denote the elements of $\mathcal{C}_h$ in the order in which they are added to the latter in Line 15. By Lemma B.2 (Freedman's inequality) instantiated with

- $n = N_{\mathsf{reg}} \cdot k$.
- $\boldsymbol{w}_i = |V_h^{\pi}(\boldsymbol{x}_h^{(i)}) - V_h^{\widehat{\pi}}(\boldsymbol{x}_h^{(i)})| - \mathbb{E}[|V_h^{\pi}(\boldsymbol{x}_h) - V_h^{\widehat{\pi}}(\boldsymbol{x}_h)| \mid \boldsymbol{x}_{h-1} = x_{h-1}^{(j)}, \boldsymbol{a}_{h-1} = a_{h-1}^{(j)}]$, for all $i \in [n]$ and $j = \lfloor i/N_{\mathsf{reg}} \rfloor + 1$, where $\boldsymbol{x}_h^{(N_{\mathsf{reg}} \cdot j)}, \dots, \boldsymbol{x}_h^{(N_{\mathsf{reg}} \cdot j + N_{\mathsf{reg}} - 1)} \overset{\text{i.i.d.}}{\sim} T_h(\cdot \mid \boldsymbol{x}_{h-1} = x_{h-1}^{(j)}, \boldsymbol{a}_{h-1} = a_{h-1}^{(j)})$;
- $\mathcal{H}_i = \sigma(\boldsymbol{x}_h^{(1)}, \dots \boldsymbol{x}_h^{(i-1)})$, for all $i \in [n]$;
- $R = H$; and
- $\lambda = 1/H$;

we get that there is an event $\widetilde{\mathcal{E}}''_{h,k,\pi}(\mathcal{S}_k)$ of probability at least $1 - \delta/(8k^2 H|\Pi'|)$ under which:

$$
\begin{aligned}
&\sum_{(x_{h-1},a_{h-1})\in\mathcal{D}_h} \sum_{(x_h,-)\in\mathcal{D}_h(x_{h-1},a_{h-1})} |V_h^{\pi}(x_h) - V_h^{\widehat{\pi}}(x_h)| \\
&= 2N_{\mathsf{reg}} \sum_{(x_{h-1},a_{h-1})\in\mathcal{D}_h} \mathbb{E}\left[|V_h^{\pi}(\boldsymbol{x}_h) - V_h^{\widehat{\pi}}(\boldsymbol{x}_h)| \mid \boldsymbol{x}_{h-1} = x_{h-1}, \boldsymbol{a}_{h-1} = a_{h-1}\right] + H\log(8k^2|\Pi'|H/\delta), \\
&\leq 2HN_{\mathsf{reg}} \sum_{(x_{h-1},a_{h-1})\in\mathcal{D}_h} \sum_{\tau=h}^{H} \mathbb{E}^{\widehat{\pi}}\left[D_{\mathrm{tv}}(\widehat{\pi}_\tau(\boldsymbol{x}_\tau), \pi_\tau(\boldsymbol{x}_\tau)) \mid \boldsymbol{x}_{h-1} = x_{h-1}, \boldsymbol{a}_{h-1} = a_{h-1}\right] + H\log(8k^2|\Pi'|H/\delta), \quad (40)
\end{aligned}
$$

where the last inequality follows by (39) and the law of total expectation.

Regression guarantee. Since $\pi \in \Pi' \subseteq \Pi_{\mathsf{S}}$ and Assumption H.1 holds, Lemma B.5 (regression guarantee) instantiated with

- $f_\star(x) = V_h^{\pi}(x)$;
- $B = H$;
- $\boldsymbol{b}_i = \boldsymbol{v}_h - V_h^{\pi}(\boldsymbol{x}_h)$ (where $\boldsymbol{v}_h \coloneqq \max_{a\in\mathcal{A}} \widehat{Q}_h(\boldsymbol{x}_h, a)$); and
- $\xi = H$;

implies that there is an event $\breve{\mathcal{E}}''_{h,k,\pi}(\mathcal{S}_k)$ of probability at least $1 - \delta/(4k^2 H|\Pi'|)$ under which we have:

$$
\begin{aligned}
&\sum_{(x_{h-1},a_{h-1})\in\mathcal{C}_h} \frac{1}{N_{\mathsf{reg}}} \sum_{(x_h,-)\in\mathcal{D}_h(x_{h-1},a_{h-1})} \left(\widehat{V}_h(x_h) - V_h^{\pi}(x_h)\right)^2 \\
&\leq \frac{4kH^2\log(4k^2H|\Pi'||\mathcal{V}|/\delta)}{N_{\mathsf{reg}}} + \frac{4H}{N_{\mathsf{reg}}} \sum_{(x_{h-1},a_{h-1})\in\mathcal{C}_h} \sum_{(x_h,v_h)\in\mathcal{D}_h(x_{h-1},a_{h-1})} |V_h^{\pi}(x_h) - v_h|, \\
&\leq \frac{4kH^2\log(4k^2H|\Pi'||\mathcal{V}|/\delta)}{N_{\mathsf{reg}}} + \frac{4H}{N_{\mathsf{reg}}} \sum_{(x_{h-1},a_{h-1})\in\mathcal{C}_h} \sum_{(x_h,v_h)\in\mathcal{D}_h(x_{h-1},a_{h-1})} |V_h^{\widehat{\pi}}(x_h) - v_h| \\
&\quad + \frac{4H}{N_{\mathsf{reg}}} \sum_{(x_{h-1},a_{h-1})\in\mathcal{C}_h} \sum_{(x_h,v_h)\in\mathcal{D}_h(x_{h-1},a_{h-1})} |V_h^{\pi}(x_h) - V_h^{\widehat{\pi}}(x_h)|, \quad (41)
\end{aligned}
$$

where the last step follows by the triangle inequality. Thus, by plugging (40) and (38) into (41), we get that under $\mathcal{E}''_{h,k}(\mathcal{S}_k) \cap \widetilde{\mathcal{E}}''_{h,k,\pi}(\mathcal{S}_k) \cap \breve{\mathcal{E}}''_{h,k,\pi}(\mathcal{S}_k)$:

$$
\begin{aligned}
&\sum_{(x_{h-1},a_{h-1})\in\mathcal{C}_h} \frac{1}{N_{\mathsf{reg}}} \sum_{(x_h,-)\in\mathcal{D}_h(x_{h-1},a_{h-1})} \left(\widehat{V}_h(x_h) - V_h^{\pi}(x_h)\right)^2 \\
&\leq \frac{9kH^2\log(8k^2H|\Pi'||\mathcal{V}|/\delta)}{N_{\mathsf{reg}}} + 8H^2 \sum_{(x_{h-1},a_{h-1})\in\mathcal{C}_h} \sum_{\tau=h}^{H} \mathbb{E}^{\widehat{\pi}}\left[D_{\mathrm{tv}}(\widehat{\pi}_\tau(\boldsymbol{x}_\tau), \pi_\tau(\boldsymbol{x}_\tau)) \mid \boldsymbol{x}_{h-1} = x_{h-1}, \boldsymbol{a}_{h-1} = a_{h-1}\right]. \\
&\qquad (42)
\end{aligned}
$$

Applying the union bound to conclude. Let $\boldsymbol{\mathcal{S}}_k$ be the random state of RVFS_0 during the k^{th} call to RVFS_h and immediately before Line 20, i.e. immediately before gathering data for the regression step in RVFS_h. Further, let $\boldsymbol{\mathcal{S}}_k^+$ be the random state of RVFS_0 during the k^{th} call to RVFS_h and immediately before Line 26, i.e. immediately before the regression step in RVFS_h. If RVFS_0 terminates before the k^{th} call to RVFS_h, we use the convention that $\boldsymbol{\mathcal{S}}_k = \boldsymbol{\mathcal{S}}_k^+ = \mathfrak{t}$, where $\mathfrak{t}$ denotes a terminal state, and define $\mathcal{E}''_{h,k}(\mathfrak{t}) = \widetilde{\mathcal{E}}''_{h,k,\pi}(\mathfrak{t}) = \breve{\mathcal{E}}''_{h,k,\pi}(\mathfrak{t}) = \{\mathfrak{t}\}$. Further, we define

$$
\mathcal{E}''_h \coloneqq \left\{ \prod_{k\in\mathbb{N},\pi\in\Pi'} \mathbb{I}\{\boldsymbol{\mathcal{S}}_k^+ \in \mathcal{E}''_{h,k}(\boldsymbol{\mathcal{S}}_k) \cap \widetilde{\mathcal{E}}''_{h,k,\pi}(\boldsymbol{\mathcal{S}}_k) \cap \breve{\mathcal{E}}''_{h,k,\pi}(\boldsymbol{\mathcal{S}}_k)\} = 1 \right\}.
$$

Note that by the argument in the sequel and the union bound, we have that

$$\forall k \geq 1, \forall \mathcal{S}_k, \quad \mathbb{P}[\boldsymbol{\mathcal{S}}_k^+ \in \mathcal{E}''_{h,k}(\mathcal{S}_k) \cap \widetilde{\mathcal{E}}''_{h,k,\pi}(\mathcal{S}_k) \cap \breve{\mathcal{E}}''_{h,k,\pi}(\mathcal{S}_k)] \geq 1 - \frac{\delta}{2k^2 H}, \tag{43}$$

where $\mathcal{S}_k$ denotes the state of RVFS_0 during the k^{th} call to RVFS_h and immediately before Line 20. By letting $\boldsymbol{\mathcal{S}}'_1, \boldsymbol{\mathcal{S}}'_2, \dots$ denote an identical, independent copy of the sequence $\boldsymbol{\mathcal{S}}_1, \boldsymbol{\mathcal{S}}_2, \dots$, we have by the chain rule:

$$\begin{aligned}
\mathbb{P}[\mathcal{E}''_h] &= \mathbb{E}_{\boldsymbol{\mathcal{S}}'_1, \boldsymbol{\mathcal{S}}'_2, \dots}\left[\prod_{k\geq 1} \mathbb{P}[\boldsymbol{\mathcal{S}}_k^+ \in \mathcal{E}''_{h,k}(\boldsymbol{\mathcal{S}}_k) \cap \widetilde{\mathcal{E}}''_{h,k,\pi}(\boldsymbol{\mathcal{S}}_k) \cap \breve{\mathcal{E}}''_{h,k,\pi}(\boldsymbol{\mathcal{S}}_k) \mid \boldsymbol{\mathcal{S}}_k = \boldsymbol{\mathcal{S}}'_k]\right], \\
&\geq \prod_{k\geq 1}\left(1 - \frac{\delta}{2k^2 H}\right), \quad \text{(by (43))} \\
&\geq 1 - \frac{\delta}{H},
\end{aligned} \tag{44}$$

where the last inequality follows from the fact that for any sequence $x_1, x_2, \dots \in (0,1)$, we have $\prod_{k\geq 1}(1 - x_k) \geq 1 - \sum_{k\geq 1} x_k$. Combining (44) with (42) implies that $\mathcal{E}''_h$ gives the desired result. □

H.5 Proof of Lemma H.4 (Guarantee for Confidence Sets)

To prove Lemma H.4, we need one additional result pertaining to the order in which the instances $(\mathsf{RVFS}_h)_{h\in[H]}$ are called.

Lemma H.5. *Let $h \in [0\,..\,H]$ be given, and consider the setting of Lemma H.4. Further, consider a call to $\mathsf{RVFS}_0(f, \mathcal{V}, \varnothing, \varnothing; \mathcal{V}, \varepsilon, \delta)$ that terminates, and let $h \in [H]$ be any layer such that RVFS_h is called during the execution of RVFS_0. Then, after the last call to RVFS_h terminates, no instance of RVFS in $(\mathsf{RVFS}_\tau)_{\tau > h}$ is called before RVFS_0 terminates.*

Proof of Lemma H.5. Suppose there is an instance of RVFS in $(\mathsf{RVFS}_\tau)_{\tau>h}$ that is called after the last call to RVFS_h terminates. Let $\tau > h$ be the lowest layer where RVFS_τ is called after the last call to RVFS_h terminates. Let $\mathsf{RVFS}_\tau^{\mathsf{last}}$ denote the corresponding instance of RVFS_τ. Further, let $\ell < \tau$ be such that RVFS_ℓ is the parent instance of $\mathsf{RVFS}_\tau^{\mathsf{last}}$ (i.e. the instance that called $\mathsf{RVFS}_\tau^{\mathsf{last}}$). Note that we cannot have $\ell = h$ as this would imply that an instance of RVFS_h terminates after $\mathsf{RVFS}_\tau^{\mathsf{last}}$, and we have assumed that $\mathsf{RVFS}_\tau^{\mathsf{last}}$ terminates after the last call RVFS_h. It is also not possible to have $\ell > h$ as this would imply that τ is not the lowest layer where RVFS_τ is called after the last call to RVFS_h terminates. Thus, we must have that $\ell < h$. Now, the for-loop in Line 16 ensures that that there is an instance of RVFS_h that is called after $\mathsf{RVFS}_\tau^{\mathsf{last}}$ terminates and before RVFS_ℓ does. This contradicts the assumption that $\mathsf{RVFS}_\tau^{\mathsf{last}}$ is called after the last call to RVFS_h. □

Proof of Lemma H.4. We start by showing that $\tilde{\pi} \in \Pi_{4\varepsilon}$ by constructing the corresponding collection of random state-action value functions $(\widetilde{\boldsymbol{Q}}_h(x,a))_{(h,x,a)\in[H]\times\mathcal{X}\times\mathcal{A}} \subset [0, H]$ in the definition of $\Pi_{4\varepsilon}$. In particular, for $(h, x, a) \in [H] \times \mathcal{X} \times \mathcal{A}$, we define

$$\widetilde{\boldsymbol{Q}}_h(x,a) = \begin{cases} \widehat{\boldsymbol{Q}}_h(x,a), & \text{if } \|\widehat{\boldsymbol{Q}}_h(x,\cdot) - \mathcal{P}_h[V^{\tilde{\pi}}_{h+1}](x,\cdot)\|_\infty \leq 4\varepsilon, \\ \mathcal{P}_h[V^{\tilde{\pi}}_{h+1}](x,a), & \text{otherwise,} \end{cases} \quad \text{for } h = H, \dots, 1.$$

where $\widehat{\boldsymbol{Q}}_\tau(x,a) \coloneqq \widehat{\boldsymbol{\mathcal{P}}}_{\tau,\varepsilon,\delta'}[\widehat{V}_{\tau+1}](x,a)$. Note that $\widetilde{\boldsymbol{Q}}_h(x,a)$ only depends on the randomness of $\widehat{\boldsymbol{\mathcal{P}}}_{h,\varepsilon,\delta'}[\widehat{V}_{h+1}](x,a)$, and so $(\widetilde{\boldsymbol{Q}}_h(x,a))_{(h,x,a)\in[H]\times\mathcal{X}\times\mathcal{A}}$ are independent random variables. Furthermore, since $\mathcal{P}_h[V^{\tilde{\pi}}_{h+1}] \equiv Q^{\tilde{\pi}}_h$, we have that

$$\forall (x,a) \in \mathcal{X} \times \mathcal{A}, \quad \|\widetilde{\boldsymbol{Q}}_h(x,a) - Q^{\tilde{\pi}}_h(x,a)\| \leq 4\varepsilon.$$

Finally, since $\tilde{\boldsymbol{\pi}}_h(\cdot) \in \arg\max_{a\in\mathcal{A}} \widetilde{\boldsymbol{Q}}_h(\cdot, a)$, we have that $\tilde{\pi} \in \Pi_{4\varepsilon}$.

We show $V_h^{\tilde{\pi}} \in \widehat{\mathcal{V}}_h$. We prove that $V_h^{\tilde{\pi}} \in \widehat{\mathcal{V}}_h$, for all $h \in [H]$, under the event $\mathcal{E}''' \coloneqq \mathcal{E} \cap \mathcal{E}_1' \cap \mathcal{E}_1'' \cap \cdots \cap \mathcal{E}_H' \cap \mathcal{E}_H''$, where $\mathcal{E}$, $(\mathcal{E}_h')$, and $(\mathcal{E}_h'')$ are the events defined in Lemma H.1, Lemma H.2, and Lemma H.3, respectively. Throughout, we condition on $\mathcal{E}'''$. First, note that by Lemma H.1, RVFS_0 terminates. Let $(\widehat{V}_{1:H}, \widehat{\mathcal{V}}_{1:H}, \mathcal{C}_{1:H}, \mathcal{B}_{1:H}, t_{1:H})$ be the tuple it returns.

We will show via backwards induction over $\ell = H+1, \dots, 1$, that

$$V_\ell^{\tilde{\pi}} \in \widehat{\mathcal{V}}_\ell, \tag{45}$$

where $\tilde{\pi}_{1:H}$ is the stochastic policy defined recursively via

$$\tilde{\boldsymbol{\pi}}_\tau(x) \in \operatorname*{arg\,max}_{a \in \mathcal{A}} \begin{cases} \widehat{\boldsymbol{Q}}_\tau(x,a) \coloneqq \widehat{\boldsymbol{\mathcal{P}}}_{\tau,\varepsilon,\delta'}[\widehat{V}_{\tau+1}](x,a), & \text{if } \|\widehat{\boldsymbol{Q}}_\tau(x,\cdot) - \mathcal{P}_\tau[V_{\tau+1}^{\tilde{\pi}}](x,\cdot)\|_\infty \le 4\varepsilon, \\ \mathcal{P}_\tau[V_{\tau+1}^{\tilde{\pi}}](x,a), & \text{otherwise,} \end{cases} \quad \text{for } \tau = H, \dots, 1,$$

where $\widehat{\boldsymbol{Q}}_\tau(x,a) \coloneqq \widehat{\boldsymbol{\mathcal{P}}}_{\tau,\varepsilon,\delta'}[\widehat{V}_{\tau+1}](x,a)$.

Base case $[\ell = H+1]$. This holds trivially because $V_{H+1}^{\pi} \equiv 0$ for any $\pi \in \Pi_{\mathsf{S}}$ by convention.

General case $[\ell \le H]$. Fix $h \in [H]$ and suppose that (45) holds for all $\ell \in [h+1 \,..\, H+1]$. We show as a consequence that (45) holds for $\ell = h$. First, note that if RVFS_h is never called during the execution of RVFS_0, then $\widehat{\mathcal{V}}_h = \mathcal{V}$, and so (45) trivially holds for $\ell = h$ under Assumption H.1 with $\varepsilon_{\mathsf{real}} = 4\varepsilon$.

Now, suppose that RVFS_h is called at least once, and let $(\widehat{V}_{h:H}^{+}, \widehat{\mathcal{V}}_{h:H}^{+}, \mathcal{C}_{h:H}^{+}, \mathcal{B}_{h:H}^{+}, t_{h:H}^{+})$ be the output of the last call to RVFS_h during the execution of RVFS_0. We claim that

$$(\widehat{V}_{h:H}^{+}, \widehat{\mathcal{V}}_{h:H}^{+}, \mathcal{C}_{h:H}^{+}) = (\widehat{V}_{h:H}, \widehat{\mathcal{V}}_{h:H}, \mathcal{C}_{h:H}). \tag{46}$$

To see this, first note that the for-loop in Line 16 ensures that no instance of $(\mathsf{RVFS}_\tau)_{\tau > h}$ can be called after the last call to RVFS_h (by Lemma H.5). Thus, the estimated value functions, confidence sets, and core sets for layers $h+1, \dots, H$ remain unchanged after the last call to RVFS_h; that is, (46) holds. Thus, by Lemma H.2, and since we are conditioning on $\mathcal{E}_{h+1:H}'$, we have that for all $(x_{h-1}, a_{h-1}) \in \mathcal{C}_h$ and $\ell \in [h+1 \,..\, H+1]$:

$$\mathbb{P}^{\widehat{\pi}}\left[\sup_{f \in \widehat{\mathcal{V}}_\ell} \max_{a \in \mathcal{A}} \left|(\mathcal{P}_{\ell-1}[\widehat{V}_\ell] - \mathcal{P}_{\ell-1}[f_\ell])(\boldsymbol{x}_{\ell-1}, a)\right| > 3\varepsilon \mid \boldsymbol{x}_{h-1} = x_{h-1}, \boldsymbol{a}_{h-1} = a_{h-1}\right] \le \frac{4\log(8M^6 N_{\mathsf{test}}^2 H^8/\delta)}{N_{\mathsf{test}}}, \tag{47}$$

where $M = \lceil 8\varepsilon^{-1} C_{\mathsf{push}} H \rceil$. Now, by the induction hypothesis, we have $V_\ell^{\tilde{\pi}} \in \widehat{\mathcal{V}}_\ell$, and so substituting $V_\ell^{\tilde{\pi}}$ for f_ℓ in (47), we get that for all $(x_{h-1}, a_{h-1}) \in \mathcal{C}_h$ and $\ell \in [h+1 \,..\, H+1]$:

$$\mathbb{P}^{\widehat{\pi}}\left[\max_{a \in \mathcal{A}} \left|(\mathcal{P}_{\ell-1}[\widehat{V}_\ell] - \mathcal{P}_{\ell-1}[V_\ell^{\tilde{\pi}}])(\boldsymbol{x}_{\ell-1}, a)\right| > 3\varepsilon \mid \boldsymbol{x}_{h-1} = x_{h-1}, \boldsymbol{a}_{h-1} = a_{h-1}\right] \le \frac{4\log(8M^6 N_{\mathsf{test}}^2 H^8/\delta)}{N_{\mathsf{test}}}.$$

Therefore, by Lemma K.1 (instantiated with $\mu[\cdot] = \mathbb{P}^{\widehat{\pi}}[\cdot \mid \boldsymbol{x}_{h-1} = x_{h-1}, \boldsymbol{a}_{h-1} = a_{h-1}]$, $\tau = \ell - 1$, and $V_{\tau+1} = V_\ell^{\tilde{\pi}}$), we have that $(x_{h-1}, a_{h-1}) \in \mathcal{C}_h$ and $\ell \in [h+1 \,..\, H+1]$:

$$\mathbb{E}^{\widehat{\pi}}[D_{\mathrm{tv}}(\widehat{\pi}_{\ell-1}(\boldsymbol{x}_{\ell-1}), \tilde{\pi}_{\ell-1}(\boldsymbol{x}_{\ell-1})) \mid \boldsymbol{x}_{h-1} = x_{h-1}, \boldsymbol{a}_{h-1} = a_{h-1}] \le \frac{4\log(8M^6 N_{\mathsf{test}}^2 H^8/\delta)}{N_{\mathsf{test}}} + \delta', \tag{48}$$

where δ' is as in Algorithm 3.

Applying the regression guarantee to conclude the induction. Note that $\tilde{\pi} \in \Pi'$, where $\Pi' \subset \Pi_{\mathsf{S}}$ is the set of stochastic policies such that $\pi \in \Pi'$ if and only if there exists $V_{1:H} \in \mathcal{V}$ such that π is defined recursively as

$$\boldsymbol{\pi}_\tau(x) \in \operatorname*{arg\,max}_{a \in \mathcal{A}} \begin{cases} \boldsymbol{Q}_\tau(x,a) \coloneqq \widehat{\boldsymbol{\mathcal{P}}}_{\tau,\varepsilon,\delta'}[V_{\tau+1}](x,a), & \text{if } \|\boldsymbol{Q}_\tau(x,\cdot) - \mathcal{P}_\tau[V_{\tau+1}^{\pi}](x,\cdot)\|_\infty \le 4\varepsilon, \\ \mathcal{P}_\tau[V_{\tau+1}^{\pi}](x,a), & \text{otherwise,} \end{cases}$$

for $\tau = H, \dots, 1$, where $\boldsymbol{Q}_\tau(x,a) \coloneqq \widehat{\boldsymbol{\mathcal{P}}}_{\tau,\varepsilon,\delta'}[V_{\tau+1}](x,a)$. The policy class Π' is finite and $|\Pi'| \le |\mathcal{V}|$. Furthermore, we have $\Pi' \subseteq \Pi_{4\varepsilon}$ as shown at the beginning of this proof. Therefore, if we let $\{\mathcal{D}_h(x,a) : (x,a) \in \mathcal{C}_h\}$ be the datasets in the definition of $\widehat{\mathcal{V}}_h$ in (16), we have by Lemma H.3 (under Assumption H.1 with $\varepsilon_{\mathsf{real}} = 4\varepsilon$) and the conditioning on $\mathcal{E}''_{h+1:H}$ and $\mathcal{E}$:

$$\begin{aligned}
&\sum_{(x_{h-1},a_{h-1})\in\mathcal{C}_h} \frac{1}{N_{\mathsf{reg}}} \sum_{(x_h,-)\in\mathcal{D}_h(x_{h-1},a_{h-1})} \left(\widehat{V}_h(x_h) - V_h^{\tilde{\pi}}(x_h)\right)^2 \\
&\le \frac{9|\mathcal{C}_h|H^2\log(8|\mathcal{C}_h|^2H|\mathcal{V}|^2/\delta)}{N_{\mathsf{reg}}} + 8H^2 \sum_{(x_{h-1},a_{h-1})\in\mathcal{C}_h} \sum_{\tau=h}^{H} \mathbb{E}^{\widehat{\pi}}\left[D_{\mathrm{tv}}(\widehat{\pi}_\tau(\boldsymbol{x}_\tau), \tilde{\pi}_\tau(\boldsymbol{x}_\tau)) \mid \boldsymbol{x}_{h-1} = x_{h-1}, \boldsymbol{a}_{h-1}a_{h-1}\right], \\
&\le \frac{9MH^2\log(8M^2H|\mathcal{V}|^2/\delta)}{N_{\mathsf{reg}}} + 8H^2 \sum_{(x_{h-1},a_{h-1})\in\mathcal{C}_h} \sum_{\tau=h}^{H} \mathbb{E}^{\widehat{\pi}}\left[D_{\mathrm{tv}}(\widehat{\pi}_\tau(\boldsymbol{x}_\tau), \tilde{\pi}_\tau(\boldsymbol{x}_\tau)) \mid \boldsymbol{x}_{h-1} = x_{h-1}, \boldsymbol{a}_{h-1} = a_{h-1}\right],
\end{aligned} \tag{49}$$

where the last inequality follows by the fact that $|\mathcal{C}_h| \le M$ under $\mathcal{E}$. Combining (49) with (48), implies that

$$\begin{aligned}
&\sum_{(x_{h-1},a_{h-1})\in\mathcal{C}_h} \frac{1}{N_{\mathsf{reg}}} \sum_{(x_h,-)\in\mathcal{D}_h(x_{h-1},a_{h-1})} \left(\widehat{V}_h(x_h) - V_h^{\tilde{\pi}}(x_h)\right)^2 \\
&\le \frac{9MH^2\log(8M^2H|\mathcal{V}|^2/\delta)}{N_{\mathsf{reg}}} + 8MH^3 \cdot \frac{4\log(8M^6N_{\mathsf{test}}^2H^8/\delta)}{N_{\mathsf{test}}} + 8MH^3\delta', \\
&= \frac{9MH^2\log(8M^2H|\mathcal{V}|^2/\delta)}{N_{\mathsf{reg}}} + 8MH^3 \cdot \frac{4\log(8M^6N_{\mathsf{test}}^2H^8/\delta)}{N_{\mathsf{test}}} + 8MH^3 \frac{\delta}{4M^7N_{\mathsf{test}}^2H^8|\mathcal{V}|}, \\
&\le \varepsilon_{\mathsf{reg}}^2,
\end{aligned} \tag{50}$$

where the last inequality follows by the fact that $\delta \in (0,1)$ and the definition of $\varepsilon^2_{\mathsf{reg}}$ in Algorithm 3. By the definition of $\widehat{\mathcal{V}}_h$ in (16), (50) implies that $V_h^{\tilde{\pi}} \in \widehat{\mathcal{V}}_h$, which completes the induction. □

H.6 Proof of Theorem H.1 (Main Guarantee of RVFS)

Proof of Theorem H.1. We condition on the event $\widetilde{\mathcal{E}} \coloneqq \mathcal{E} \cap \mathcal{E}''' \cap \mathcal{E}'_1 \cap \dots \cap \mathcal{E}'_H$, where $\mathcal{E}$, $\mathcal{E}'''$, and $(\mathcal{E}'_h)$ are the events in Lemma H.1, Lemma H.4, and Lemma H.2, respectively. Note that by the union bound, we have $\mathbb{P}[\widetilde{\mathcal{E}}] \ge 1 - 5\delta$. By Lemma H.2, we have that

$$\forall h \in [H], \quad \mathbb{P}^{\widehat{\pi}}\left[\sup_{f\in\widehat{\mathcal{V}}_h} \max_{a\in\mathcal{A}} \left|(\mathcal{P}_{h-1}[\widehat{V}_h] - \mathcal{P}_{h-1}[f_h])(\boldsymbol{x}_{h-1}, a)\right| > 3\varepsilon\right] \le \frac{4\log(8M^6N_{\mathsf{test}}^2H^8/\delta)}{N_{\mathsf{test}}}, \tag{51}$$

where $M = \lceil 8\varepsilon^{-1}C_{\mathsf{push}}H \rceil$ and $N_{\mathsf{test}} = 2^8M^2H\varepsilon^{-1}\log(8M^6H^8\varepsilon^{-2}\delta^{-1})$. On the other hand, by Lemma H.4, we have

$$\forall h \in [H], \quad V_h^{\tilde{\pi}} \in \widehat{\mathcal{V}}_h.$$

Thus, substituting $V_h^{\tilde{\pi}}$ for f_h in (51) we get that for all $h \in [H+1]$.

$$\mathbb{P}^{\widehat{\pi}}\left[\max_{a\in\mathcal{A}} \left|(\mathcal{P}_{h-1}[\widehat{V}_h] - \mathcal{P}_{h-1}[V_h^{\tilde{\pi}}])(\boldsymbol{x}_{h-1}, a)\right| > 3\varepsilon\right] \le \frac{4\log(8M^6N_{\mathsf{test}}^2H^8/\delta)}{N_{\mathsf{test}}}.$$

This together with Lemma K.1, instantiated with $\mu[\cdot] = \mathbb{P}^{\widehat{\pi}}[\cdot]$; $\tau = h-1$; $V_{\tau+1} = V_h^{\tilde{\pi}}$; and $\delta = \delta'$ (with δ' as in Algorithm 3), translates to:

$$\begin{aligned}
\forall h \in [H], \quad \mathbb{E}^{\widehat{\pi}}[D_{\mathrm{tv}}(\widehat{\pi}_h(\boldsymbol{x}_h), \tilde{\pi}_h(\boldsymbol{x}_h))] &\le \frac{4\log(8M^6N_{\mathsf{test}}^2H^8/\delta)}{N_{\mathsf{test}}} + \delta', \\
&= \frac{4\log(8M^6N_{\mathsf{test}}^2H^8/\delta)}{N_{\mathsf{test}}} + \frac{\delta}{4M^7N_{\mathsf{test}}^2H^8|\mathcal{V}|}, \\
&\le \frac{\varepsilon}{4H^3C_{\mathsf{push}}},
\end{aligned}$$

where the last step follows from the fact that $N_{\mathtt{test}} = 2^8 M^2 H \varepsilon^{-1} \log(8 M^6 H^8 \varepsilon^{-2} \delta^{-1})$ (with M as in Line 3).

Bounding the sample complexity. We now bound the number of episodes used by Algorithm 3 under the event $\widetilde{\mathcal{E}}$. First, we fix $h \in [H]$, and focus on the number of episodes used within a to call RVFS_h, excluding any episodes used by any subsequent calls to RVFS_τ for $\tau > h$. We start by counting the number of episodes used to test the fit of the estimated value functions $\widehat{V}_{h+1:H}$. Starting from Line 8, there are for-loops over $(x_{h-1}, a_{h-1}) \in \mathcal{C}_h$, $\ell = H, \dots, h+1$, and $n \in [N_{\mathtt{test}}]$ to collect partial episodes using the learned policy $\widehat{\pi}$ in Algorithm 3, where $N_{\mathtt{test}} = 2^8 M^2 H \varepsilon^{-1} \log(8 M^6 H^8 \varepsilon^{-2} \delta^{-1})$ and $M = \lceil 8 \varepsilon^{-1} C_{\mathsf{push}} H \rceil$. Note that executing $\widehat{\pi}$ requires the local simulator and uses $N_{\mathtt{sim}} = 2 \log(4 M^7 N_{\mathtt{test}}^2 H^2 |\mathcal{V}| / \delta) / \varepsilon^2$ local simulator queries to output an action at each layer (since Algorithm 3 calls Algorithm 5 with confidence level $\delta' = \delta / (8 M^7 N_{\mathtt{test}}^2 H^8 |\mathcal{V}|)$). Also, note that whenever a test fails in Line 14 and the recursive RVFS calls return, the for-loop in Line 8 resumes. We also know (by Lemma H.1) that the number of times the test fails in Line 14 is at most M. Thus, the number of times the for-loop in Line 8 resumes is bounded by HM; here, H accounts for possible test failures across all layers $\tau \in [h+1 \,..\, H]$. Thus, the total sample complexity required to generate episodes between lines Line 8 and Line 11 is bounded by

$$\#\text{ episodes for roll-outs} \le \underbrace{MH}_{\#\text{ of times Line 8 resumes}} \cdot \underbrace{MH^2 N_{\mathtt{test}} N_{\mathtt{sim}}}_{\text{Sample complexity in case of no test failures}}. \tag{52}$$

Note that the test in Line 14 also uses episodes because it calls the operator $\widehat{\boldsymbol{\mathcal{P}}}$ for every $a \in \mathcal{A}$. Thus, the number of episodes used for the test in Line 14 is bounded by

$$\#\text{ episodes for the tests} \le \underbrace{MH}_{\#\text{ of times Line 8 resumes}} \cdot \underbrace{MHAN_{\mathtt{test}} N_{\mathtt{sim}}}_{\text{Sample complexity for Line 14}}. \tag{53}$$

We now count the number of episodes used to re-fit the value function in Line 16 and onwards. Note that starting from Line 16, there are for-loops over $(x_{h-1}, a_{h-1}) \in \mathcal{C}_h$ and $i \in [N_{\mathtt{reg}}]$ to generate $A \cdot N_{\mathtt{est}}(|\mathcal{C}_h|) \le A \cdot N_{\mathtt{est}}(M)$ partial episodes using $\widehat{\pi}$, where $N_{\mathtt{est}}(k) = 2 N_{\mathtt{reg}}^2 \log(8 A N_{\mathtt{reg}} H k^3 / \delta)$ is defined as in Algorithm 3. Since $\widehat{\pi}$ uses the local simulator and requires $N_{\mathtt{est}}$ samples (see Algorithm 5) to output an action at each layer, the number of episodes used to refit the value function is bounded by

$$\#\text{ episodes for } V\text{-refitting} \le M N_{\mathtt{reg}} A N_{\mathtt{est}}(M) H N_{\mathtt{sim}}. \tag{54}$$

Therefore, by (52), (53), and (54), the number of episodes used within a single call to RVFS_h (not accounting for episodes used by recursive calls to RVFS_τ, for $\tau > h$) is bounded by

$$\#\text{ episodes used locally within } \mathsf{RVFS}_h \le M^2 H (H + A) N_{\mathtt{test}} N_{\mathtt{sim}} + M N_{\mathtt{reg}} A N_{\mathtt{est}}(M) H N_{\mathtt{sim}}. \tag{55}$$

Finally, by Lemma H.1, RVFS_h may be called at most M times throughout the execution of RVFS_0. Using this together with (55) and accounting for the number of episodes from all layers $h \in [H]$, we get that the total number of episodes is bounded by

$$M^3 H^2 (H + A) N_{\mathtt{test}} N_{\mathtt{sim}} + M^2 H^2 N_{\mathtt{reg}} A N_{\mathtt{est}}(M) N_{\mathtt{sim}}.$$

Substituting the expressions of M, $N_{\mathtt{test}}$, $N_{\mathtt{est}}$, $N_{\mathtt{sim}}$, and $N_{\mathtt{reg}}$ from Algorithm 3 and Algorithm 5, we obtain the desired number of episodes, which concludes the proof. □

H.7 Proof of Theorem H.2 (Guarantee of $\mathsf{RVFS.bc}$)

Proof of Theorem H.2. Let $\widehat{V}_{1:H}$ be the value function estimates produced by RVFS_0 within Algorithm 4, and let $\widehat{\boldsymbol{\pi}}_h^{\mathsf{RVFS}}(\cdot) \in \arg\max_{a \in \mathcal{A}} \widehat{\boldsymbol{\mathcal{P}}}_{h, \varepsilon_{\mathsf{RVFS}}, \delta'}[\widehat{V}_{h+1}](\cdot, a)$, for all $h \in [H]$ with $\widehat{V}_{H+1} \equiv 0$ with $\varepsilon_{\mathsf{RVFS}}$ and δ' as in Algorithm 4. Further, let and let $\widetilde{\pi}_{1:H} \in \Pi_{\mathsf{S}}$ be the stochastic policy defined recursively via

$$\forall x \in \mathcal{X}, \quad \widetilde{\boldsymbol{\pi}}_\tau(x) \in \arg\max_{a \in \mathcal{A}} \begin{cases} \widehat{\boldsymbol{Q}}_\tau(x, a), & \text{if } \|\widehat{\boldsymbol{Q}}_\tau(x, \cdot) - \mathcal{P}_\tau[V_{\tau+1}^{\widetilde{\pi}}](x, \cdot)\|_\infty \le 4 \varepsilon_{\mathsf{RVFS}}, \\ \mathcal{P}_\tau[V_{\tau+1}^{\widetilde{\pi}}](x, a), & \text{otherwise,} \end{cases} \tag{56}$$

for $\tau = H, \dots, 1$, where $\widehat{\boldsymbol{Q}}_\tau(x,a) \coloneqq \widehat{\boldsymbol{\mathcal{P}}}_{\tau,\varepsilon_{\mathsf{RVFS}},\delta'}[\widehat{V}_{\tau+1}](x,a)$. By Theorem H.1, there is an event $\widetilde{\mathcal{E}}$ of probability at least $1-\delta/2$ under which:

$$\tilde{\pi} \in \Pi_{4\varepsilon_{\mathsf{RVFS}}}, \tag{57}$$

and

$$\forall h \in [H], \quad \mathbb{E}^{\widehat{\pi}^{\mathsf{RVFS}}}\left[D_{\mathrm{tv}}(\widehat{\pi}_h^{\mathsf{RVFS}}(\boldsymbol{x}_h), \tilde{\pi}_h(\boldsymbol{x}_h))\right] \le \frac{\varepsilon_{\mathsf{RVFS}}}{4H^3 C_{\mathsf{push}}} \le \frac{\varepsilon}{4H^2}, \tag{58}$$

where the last inequality follows by the choice of $\varepsilon_{\mathsf{RVFS}}$ in Algorithm 4.

For the rest of the proof, we condition on $\widetilde{\mathcal{E}}$. By (57), (61), and Proposition L.1 instantiated with $\varepsilon_{\mathtt{mis}} = 0$ (due to all π-realizability), we have that there is an event $\widetilde{\mathcal{E}}'$ of probability at least $1-\delta/2$ under which the policy $\widehat{\pi}_{1:H}$ produced by `BehaviorCloning` ensures that

$$J(\widehat{\pi}_{1:H}^{\mathsf{RVFS}}) - J(\widehat{\pi}_{1:H}) \le \frac{\varepsilon}{2}. \tag{59}$$

Now, by Lemma B.6 (the performance difference lemma), we have for $\tilde{\pi}$ as in (56):

$$\begin{aligned} J(\tilde{\pi}) - J(\widehat{\pi}_{1:H}^{\mathsf{RVFS}}) &= \sum_{h=1}^{H} \mathbb{E}^{\widehat{\pi}^{\mathsf{RVFS}}}[Q_h^{\tilde{\pi}}(\boldsymbol{x}_h, \tilde{\boldsymbol{\pi}}_h(\boldsymbol{x}_h)) - Q_h^{\tilde{\pi}}(\boldsymbol{x}_h, \widehat{\boldsymbol{\pi}}_h^{\mathsf{RVFS}}(\boldsymbol{x}_h))], \\ &\le H \sum_{h=1}^{H} \mathbb{E}^{\widehat{\pi}^{\mathsf{RVFS}}}[D_{\mathrm{tv}}(\widehat{\pi}_h^{\mathsf{RVFS}}(\boldsymbol{x}_h), \tilde{\pi}_h(\boldsymbol{x}_h))], \end{aligned}$$

and so by (61), we have

$$J(\tilde{\pi}) - J(\widehat{\pi}_{1:H}^{\mathsf{RVFS}}) \le \varepsilon/4. \tag{60}$$

Finally, since $\tilde{\pi} \in \Pi_{4\varepsilon_{\mathsf{RVFS}}}$ (see (57)), we have by Lemma G.1,

$$J(\pi^\star) - J(\tilde{\pi}) \le 12H\varepsilon_{\mathsf{RVFS}} \le \varepsilon/4,$$

where the last inequality follows by the choice $\varepsilon_{\mathsf{RVFS}}$ in Algorithm 4. Combining this with (59) and (60), we conclude that under $\widetilde{\mathcal{E}} \cap \widetilde{\mathcal{E}}'$:

$$J(\pi^\star) - J(\widehat{\pi}_{1:H}) \le \varepsilon.$$

By the union bound, we have $\mathbb{P}[\widetilde{\mathcal{E}} \cap \widetilde{\mathcal{E}}'] \ge 1-\delta$, and so the desired suboptimality guarantee in (26) holds with probability at least $1-\delta$.

Bounding the sample complexity. The sample complexity is dominated by the call to RVFS_0 within `RVFS.bc` (Algorithm 4). Since `RVFS.bc` calls RVFS_0 with $\varepsilon = \varepsilon_{\mathsf{RVFS}} = \varepsilon H^{-1}/48$, we conclude from Theorem H.1 that the total sample complexity is bounded by

$$\widetilde{O}\left(C_{\mathsf{push}}^8 H^{23} A \cdot \varepsilon^{-13}\right).$$

□

I Guarantee under $V^\star$-Realizability (Proof of Theorem 4.1, Setup I)

In this section, we prove Theorem 4.1 under **Setup I** ($V^\star/\pi^\star$-realizability (Assumptions 4.2 and 4.3) and Δ-gap (Assumption 4.4)). We prove this result as a consequence of the more general results (Theorem H.2) in Appendix H by appealing to the relaxed V^π-realizability condition in Assumption H.1.

I.1 Analysis: Proof of Theorem 4.1 (Setup I)

We begin by showing that Assumption 4.2 and Assumption 4.4 together imply that Assumption H.1 holds for any $\varepsilon_{\mathsf{real}} \le \Delta/2$; we prove this by showing that the benchmark policy class $\Pi_{\varepsilon'}$ (Appendix G.1) reduces to $\{\pi^\star\}$ when $\varepsilon' \le \Delta/2$.

Lemma I.1. *Assume that $\mathcal{V}$ satisfies Assumption 4.2 ($V^\star$-realizability), and that Assumption 4.4 (gap) holds with $\Delta > 0$. Then, for all $\varepsilon' \le \Delta/2$, we have $\Pi_{\varepsilon'} = \{\pi^\star\}$ and $\mathcal{V}$ satisfies Assumption H.1 with $\varepsilon_{\mathsf{real}} = \varepsilon'$.*

Informally Lemma I.1, whose proof is in Appendix I.2, states that under Assumption 4.2, Assumption 4.4 with $\Delta > 0$, and Assumption 4.1 (pushforward coverability) with $C_{\mathsf{push}} > 0$, we are essentially in the setting of Theorem H.1 (guarantee of RVFS under relaxed V^π-realizability), as long as we choose $\varepsilon_{\mathsf{real}} \le \Delta/2$. With this, we now state and prove a central guarantee for RVFS under $V^\star$-realizability with a gap.

Lemma I.2 (Intermediate guarantee for RVFS under **Setup I**)**.** *Let $\delta \in (0,1)$ be given, and suppose that:*

- *Assumption 4.1 (pushforward coverability) holds with parameter $C_{\mathsf{push}} > 0$;*
- *Assumption 4.4 (gap) holds with parameter $\Delta > 0$;*
- *The function class $\mathcal{V}$ satisfies Assumption 4.2 ($V^\star$-realizability).*

Then, for any $f \in \mathcal{V}$ and $\varepsilon \in (0, \Delta/8)$, with probability at least $1-\delta$, $\mathsf{RVFS}_0(f, \mathcal{V}, \varnothing, \varnothing; \mathcal{V}, \varepsilon, \delta)$ (Algorithm 3) terminates and returns value functions $\widehat{V}_{1:H}$ that satisfy

$$\forall h \in [H], \quad \mathbb{P}^{\widehat{\pi}}[\widehat{\boldsymbol{\pi}}_h(\boldsymbol{x}_h) \neq \pi^\star_h(\boldsymbol{x}_h)] \le \frac{\varepsilon}{4C_{\mathsf{push}}H^3},$$

where $\widehat{\boldsymbol{\pi}}_h(x) \in \arg\max_{a\in\mathcal{A}} \widehat{\boldsymbol{\mathcal{P}}}_{\tau,\varepsilon,\delta'}[\widehat{V}_{h+1}](x,a)$, for all $h \in [H]$, with δ' is defined as in Algorithm 3.

Proof of Lemma I.2. From Lemma I.1, we have that $\Pi_{4\varepsilon} = \{\pi^\star\}$, and so Theorem H.1 implies that with probability at least $1-\delta$,

$$\forall h \in [H], \quad \frac{\varepsilon}{4C_{\mathsf{push}}H^3} \ge \mathbb{E}^{\widehat{\pi}}[D_{\mathrm{tv}}(\widehat{\pi}_h(\boldsymbol{x}_h), \pi^\star_h(\boldsymbol{x}_h))] = \mathbb{P}^{\widehat{\pi}}[\widehat{\boldsymbol{\pi}}_h(\boldsymbol{x}_h) \neq \pi^\star_h(\boldsymbol{x}_h)],$$

where the equality follows by the fact that $\pi^\star$ is deterministic. □

From here, Theorem 4.1 follows swiftly as a consequence.

Proof of Theorem 4.1 (Setup I). Let $\widehat{V}_{1:H}$ be the value function estimates produced by RVFS_0 within Algorithm 4, and let $\widehat{\boldsymbol{\pi}}^{\mathsf{RVFS}}_h(\cdot) \in \arg\max_{a\in\mathcal{A}} \widehat{\boldsymbol{\mathcal{P}}}_{h,\varepsilon_{\mathsf{RVFS}},\delta'}[\widehat{V}_{h+1}](\cdot, a)$, for all $h \in [H]$ with $\widehat{V}_{H+1} \equiv 0$ with $\varepsilon_{\mathsf{RVFS}}$ and δ' as in Algorithm 4. By Lemma I.2, there is an event $\widetilde{\mathcal{E}}$ of probability at least $1-\delta/2$ under which:

$$\mathbb{P}^{\widehat{\pi}}[\widehat{\boldsymbol{\pi}}_h(\boldsymbol{x}_h) \neq \pi^\star_h(\boldsymbol{x}_h)] \le \frac{\varepsilon_{\mathsf{RVFS}}}{4H^3C_{\mathsf{push}}} \le \frac{\varepsilon}{4H^2}, \tag{61}$$

where the last inequality follows by the choice of $\varepsilon_{\mathsf{RVFS}}$ in Algorithm 4.

For the rest of the proof, we condition on $\widetilde{\mathcal{E}}$. By (61) and Assumption 4.3 ($\pi^\star$-realizability), the policy $\widehat{\pi}^{\mathsf{RVFS}}_{1:H}$ returned by RVFS_0 satisfies the condition in Proposition L.1 with $\varepsilon_{\mathtt{mis}} = \varepsilon/(4C_{\mathsf{push}}H^3)$. Thus, by Proposition L.1, there is an event $\widetilde{\mathcal{E}}'$ of probability at least $1-\delta/2$ under which the policies $\widehat{\pi}_{1:H}$ produced by $\mathsf{RVFS.bc}$ satisfy

$$J(\widehat{\pi}^{\mathsf{RVFS}}_{1:H}) - J(\widehat{\pi}_{1:H}) \le \frac{\varepsilon}{H} + \frac{\varepsilon}{2} \le \frac{3\varepsilon}{2}.$$

We now condition on $\widetilde{\mathcal{E}} \cap \widetilde{\mathcal{E}}'$. By Lemma B.6 (performance difference lemma), we have

$$\begin{aligned} J(\pi^\star) - J(\widehat{\pi}^{\mathsf{RVFS}}_{1:H}) &= \sum_{h=1}^{H} \mathbb{E}^{\widehat{\pi}^{\mathsf{RVFS}}}[Q^{\pi^\star}_h(\boldsymbol{x}_h, \pi^\star_h(\boldsymbol{x}_h)) - Q^{\pi^\star}_h(\boldsymbol{x}_h, \widehat{\boldsymbol{\pi}}^{\mathsf{RVFS}}_h(\boldsymbol{x}_h))], \\ &\le H \sum_{h=1}^{H} \mathbb{P}^{\widehat{\pi}}[\widehat{\boldsymbol{\pi}}_h(\boldsymbol{x}_h) \neq \pi^\star_h(\boldsymbol{x}_h)], \\ &\le \varepsilon/(4H), \end{aligned}$$

where the last inequality follows by (61).

Finally, by the union bound, we have $\mathbb{P}[\widetilde{\mathcal{E}} \cap \widetilde{\mathcal{E}}'] \geq 1 - \delta$, and so the desired suboptimality guarantee in (26) holds with probability at least $1 - \delta$.

Bounding the sample complexity. The sample complexity is dominated by the call to RVFS_0 within $\mathtt{RVFS.bc}$ (Algorithm 4). Since $\mathtt{RVFS.bc}$ calls RVFS_0 with $\varepsilon = \varepsilon_{\mathsf{RVFS}} = \varepsilon H^{-1}/48$, we conclude from Theorem H.1 that the total number of episodes is bounded by

$$\widetilde{O}\left(C_{\mathsf{push}}^{8} H^{23} A \cdot \varepsilon^{-13}\right).$$

□

I.2 Proof of Lemma I.1 (Relaxed V^{π}-Realizability under Gap)

Proof of Lemma I.1. Fix $\varepsilon' \in (0,1)$ and $\pi \in \Pi_{\varepsilon'}$. Let $(\widetilde{\boldsymbol{Q}}_h(x,a))_{(h,x,a)\in[H]\times\mathcal{X}\times\mathcal{A}}$ be independent random variables such that

$$\forall h \in [H], \quad \boldsymbol{\pi}_h(\cdot) \in \arg\max_{a'\in\mathcal{A}} \widetilde{\boldsymbol{Q}}_h(\cdot, a') \quad \text{and} \quad \|\widetilde{\boldsymbol{Q}}_h - Q_h^{\pi}\|_{\infty} \leq \varepsilon', \text{ almost surely.}$$

Such a collection of random state-action value functions $(\widetilde{\boldsymbol{Q}}_h(x,a))_{(h,x,a)\in[H]\times\mathcal{X}\times\mathcal{A}}$ is guaranteed to exist for *pi* by the definition of $\Pi_{\varepsilon'}$. We will show via backward induction over $\ell = H+1,\dots,1$ that

$$\forall x \in \mathcal{X}_{\ell}, \quad \boldsymbol{\pi}_{\ell}(x) = \pi_{\ell}^{\star}(x) \tag{62}$$

almost surely, with the convention that $\pi_{H+1} \equiv \pi_{H+1}^{\star} \equiv \pi_{\mathtt{unif}}$. This convention makes the base case, $\ell = H+1$, hold trivially.

Now, we consider the general case. Fix $h \in [H]$ and suppose that (62) holds for all $\ell \in [h+1 \,..\, H+1]$. We will show that it holds for $\ell = h$.

Thanks to the induction hypothesis, we have for all $x \in \mathcal{X}_{h+1}$ and $a \in \mathcal{A}$:

$$V_{h+1}^{\pi}(x) = V_{h+1}^{\star}(x),$$

and so

$$Q_h^{\pi} \equiv \mathcal{P}_h[V_{h+1}^{\pi}] \equiv \mathcal{P}_h[V_{h+1}^{\star}] = V_h^{\star}. \tag{63}$$

Fix $x \in \mathcal{X}$. We will show that $\pi_h(x) = \pi_h^{\star}(x)$ almost surely. Note that thanks to (63), the fact that $\|\widetilde{\boldsymbol{Q}}_h - \mathcal{T}_h[Q_{h+1}^{\pi}]\|_{\infty} \leq \varepsilon'$ almost surely, implies that

$$\|\widetilde{\boldsymbol{Q}}_h - Q_h^{\star}\|_{\infty} \leq \varepsilon',$$

almost surely. Using this, we have, almost surely

$$\begin{aligned}
Q_h^{\star}(x, \boldsymbol{\pi}_h(x)) &\geq \widetilde{\boldsymbol{Q}}_h(x, \boldsymbol{\pi}_h(x)) - \varepsilon', \\
&\geq \widetilde{\boldsymbol{Q}}_h(x, \pi_h^{\star}(x)) - \varepsilon', \\
&\geq Q_h^{\star}(x, \pi_h^{\star}(x)) - 2\varepsilon' = Q_h^{\star}(x, \pi_h^{\star}(x)) - \Delta.
\end{aligned} \tag{64}$$

On the other hand, if $\boldsymbol{\pi}_h(x) \neq \pi_h^{\star}(x)$, then

$$Q_h^{\star}(x, \boldsymbol{\pi}_h(x)) < Q_h^{\star}(x, \pi_h^{\star}(x)) - \Delta,$$

which would contradict (64). Thus, $\boldsymbol{\pi}_h(x) = \pi_h^{\star}(x)$, which concludes the induction and shows that $\pi \equiv \pi^{\star}$. We conclude that $\Pi_{\varepsilon'} = \{\pi^{\star}\}$. □

J Guarantee for Weakly Correlated ExBMDPs (Proof of Theorem 4.2)

In this section, we prove Theorem 4.2, the main guarantee for $\texttt{RVFS}^{\texttt{exo}}$. First, in Appendix J.1 we state a number of supporting technical lemmas and use them to prove Theorem 4.2. The remainder of the section (Appendix J.2 through Appendix J.6) contains the proofs for the intermediate results.

J.1 Analysis: Proof of Theorem 4.2

Recall that the the V^{π}-realizability assumption required by $\texttt{RVFS}$ for Theorem 4.1 is not satisfied in ExBMDPs, as the value functions for policies that act on the exogenous noise variables cannot be realized as a function of the true decoder $\phi^{\star}$. In $\texttt{RVFS}^{\texttt{exo}}$, we address this issue by applying the randomized rounding technique in Line 11 to the learned value functions. The crux of the analysis will be to show that for an appropriate choice of the rounding parameters $\zeta_{1:H}$, the policies produced by $\texttt{RVFS}^{\texttt{exo}}$ are endogenous in the sense that we can write $\pi(x) = \pi(\phi^{\star}(x))$ for all $x \in \mathcal{X}$. This will allow us to leverage the decoder realizability (Assumption 3.3), which implies that the function class $\mathcal{V} = \mathcal{V}_{1:H}$ given by

$$\mathcal{V}_h := \{x \mapsto f(\phi(x)) : f \in [0,H]^S, \phi \in \Phi\}, \tag{65}$$

satisfies V^{π}-realizability for all endogenous policies π.

In what follows, we first motivate the randomized rounding approach in $\texttt{RVFS}^{\texttt{exo}}$ in detail and prove that it succeeds, then use this result to proceed with an analysis similar to that of Theorem 4.1 (**Setup II**), re-using many of the technical tools developed for Theorem 4.1.

J.1.1 Randomized Rounding for Endogeneity

Naively, to ensure that the policies we execute are endogenous, it would seem that we require knowledge of the true decoder $\phi^{\star}$. Alas, knowing $\phi^{\star}$ trivializes the ExBMDP problem by reducing it to the tabular setting. To avoid requiring knowledge of $\phi^{\star}$, we apply a *randomized rounding* to the policies learned by $\texttt{RVFS}^{\texttt{exo}}$ to ensure their endogeneity.

Let $\varepsilon > 0$ be fixed going forward. Recall that compared to $\texttt{RVFS}$, $\texttt{RVFS}^{\texttt{exo}}$ (Algorithm 6) takes an additional input $\zeta_{1:H} \subset (0, 1/2)$ and executes the following coarsened policies:

$$\widehat{\boldsymbol{\pi}}_h(\cdot) \in \operatorname*{arg\,max}_{a \in \mathcal{A}} \lceil \widehat{\boldsymbol{\mathcal{P}}}_{h,\varepsilon,\delta}[\widehat{V}_{h+1}](\cdot, a)/\varepsilon + \zeta_h \rceil.$$

The *rounding parameters* $\zeta_{1:H}$, which can be thought of as an offset, are chosen randomly; this will be elucidated in the sequel.

Following a similar analysis to Appendix H (**Setup II**), we can associate a near-optimal benchmark policy $\tilde{\pi} \in \Pi_{2\varepsilon}$ with $\widehat{\pi}$ in order to emulate certain properties of the Δ-gap assumption. In particular, generalizing the construction in Eq. (25), we define a near-optimal benchmark policy $\tilde{\pi}$ recursively via:

$$\forall x \in \mathcal{X},\ \tilde{\boldsymbol{\pi}}_\tau(x; \zeta_{1:H}, \varepsilon, \delta) \in \operatorname*{arg\,max}_{a \in \mathcal{A}} \begin{cases} \lceil \widehat{\boldsymbol{Q}}_\tau(x,a)/\varepsilon + \zeta_\tau \rceil, & \text{if } \|\widehat{\boldsymbol{Q}}_\tau(x,\cdot) - \mathcal{P}_\tau[V^{\tilde{\pi}}_{\tau+1}](x,\cdot)\|_\infty \le 4\varepsilon^2, \\ \lceil \mathcal{P}_\tau[V^{\tilde{\pi}}_{\tau+1}](x,a)/\varepsilon + \zeta_\tau \rceil, & \text{otherwise}, \end{cases} \tag{66}$$

for $\tau = H, \dots, 1$, where $\widehat{\boldsymbol{Q}}_\tau(\cdot, a) := \widehat{\boldsymbol{\mathcal{P}}}_{\tau,\varepsilon,\delta}[\widehat{V}_{\tau+1}](\cdot, a)$.

Naively, to use the benchmark policy $\tilde{\pi}$ within the analysis based on relaxed V^{π}-realizability (Assumption H.1) in Appendix H , we would require the function class $\mathcal{V}$ to realize $(V^{\tilde{\pi}}_h)$. However, as argued earlier, this is not feasible unless $\tilde{\pi}$ is an endogenous policy. Fortunately, it turns out that if $\zeta_{1:H}$ (the additional input to $\texttt{RVFS}^{\texttt{exo}}$) are drawn randomly from uniform distribution over $[0, 1/2]$, then with constant probability, $\tilde{\pi}$ is indeed endogenous. What's more, under such an event, and for all possible choices of $(\widehat{V}_h)$ in (66) uniformly, $\tilde{\pi}$ "snaps" onto the stochastic endogenous policy $\bar{\pi}(\cdot; \zeta_{1:H}, \varepsilon)$ defined recursively as follows:

$$\bar{\pi}_h(\cdot; \zeta_{1:H}, \varepsilon) \in \operatorname*{arg\,max}_{a \in \mathcal{A}} \lceil \mathcal{P}_h[V^{\bar{\pi}}_{h+1}](x,a)/\varepsilon + \zeta_h \rceil, \tag{67}$$

for $h = H, \ldots, 1$. Informally, this happens because, as long as $\zeta_{1:H} \subset (0, 1/2)$ avoid certain pathological locations in $(0, 1/2)$, the coarsened state-action value functions $\varepsilon \cdot \lceil \mathcal{P}_\tau[V^{\bar{\pi}}_{\tau+1}](x, a)/\varepsilon + \zeta_h \rceil$ defining $\bar{\pi}$ exhibit a "gap" of order $\Theta(\varepsilon^2)$ separating optimal actions from the rest. This "snapping" behavior is analogous to what happens in **Setup I** with $V^\star$-realizability and Δ-gap, where Π_ε reduces to $\{\pi^\star\}$ for all $\varepsilon < \Delta/2$ (see Lemma I.1). We formalize these claims in the next two lemmas. We start by showing that $\bar{\pi}$ is endogenous and that $\bar{\pi} \in \Pi_{2\varepsilon}$. The proof is in Appendix J.2.

Lemma J.1 (Endogenous Benchmark policies)**.** *For any $\delta \in (0, 1)$, $\varepsilon \in (0, 1/2)$, and $\zeta_{1:H} \subset (0, 1/2)$, the stochastic policy $\bar{\pi}(\cdot; \zeta_{1:H}, \varepsilon)$ defined in Eq. (67) is endogenous, and we have $\bar{\pi}(\cdot; \zeta_{1:H}, \varepsilon) \in \Pi_{2\varepsilon}$.*

Next, we show that $\tilde{\pi}$ "snaps" onto $\bar{\pi}$ for the certain choices of $\zeta_{1:H}$. The proof is in Appendix J.3.

Lemma J.2 (Snapping probability)**.** *Let $\delta \in (0, 1)$, $\varepsilon \in (0, 1/2)$ be given, and $\mathbb{P}^\zeta$ denote the probability law of $\zeta_1, \ldots, \zeta_H \sim \texttt{unif}([0, 1/2])$. Then, there is an event $\mathcal{E}_{\mathsf{rand}}$ of probability at least $1 - 24SAH\varepsilon$ under $\boldsymbol{\zeta}_{1:H} \sim \mathbb{P}^\zeta$ such that for all $\widetilde{V} \in (\mathcal{X} \times [H] \to [0, H])$ simultaneously,*

$$\forall h \in [H], \quad \tilde{\boldsymbol{\pi}}_h(\cdot; \widetilde{V}, \boldsymbol{\zeta}_{1:H}, \varepsilon, \delta) = \bar{\pi}_h(\cdot; \boldsymbol{\zeta}_{1:H}, \varepsilon),$$

where $\tilde{\boldsymbol{\pi}}_h(\cdot; \widetilde{V}, \boldsymbol{\zeta}_{1:H}, \varepsilon, \delta)$ is defined as in (66) with $\widehat{V} = \widetilde{V}$, and $\bar{\pi}$ is defined as in (67).

The lemma together, with Lemma J.1, implies that with constant probability under $\mathbb{P}^\zeta$, the benchmark policies $(\tilde{\pi}_h)$ used in the analysis of $\mathsf{RVFS}^{\mathsf{exo}}$ are endogenous and satisfy $\tilde{\pi} \in \Pi_{2\varepsilon}$.

J.1.2 Pushforward Coverability

In order to proceed with the analysis strategy in Appendix H, we need to verify that pushforward coverability is satisfied for ExBMDPs under the weak correlation assumption. We do so in the next lemma; see Appendix J.4 for a proof.

Lemma J.3 (Pushforward coverability)**.** *A weakly correlated ExBMDP with constant C_{exo} (see Assumption 4.7) satisfies C_{push}-pushforward coverability (Assumption 4.1) with $C_{\mathsf{push}} = C_{\mathsf{exo}} \cdot SA$, where $S \in \mathbb{N}$ is the number of endogenous states.*

Equipped with the preceding lemmas, we proceed with an analysis similar to the approach for Theorem 4.1 (**Setup II**) in Appendix H. In what follows, we state a number of technical lemmas that apply the relevant results from Appendix H to the ExBMDP setting we consider here.

J.1.3 Bounding the Number of Test Failures

Since the size of the core sets $\mathcal{C}_{1:H}$ in $\mathsf{RVFS}^{\mathsf{exo}}$ is directly proportional to the number of test failures, the next result, which bounds $|\mathcal{C}_h|$ for all $h \in [H]$, allows us to show that $\mathsf{RVFS}^{\mathsf{exo}}$ (Algorithm 6) terminates in a polynomial number of iterations.

Lemma J.4 (Bounding the number of test failures)**.** *Let $\delta, \varepsilon \in (0, 1)$ and $\zeta_{1:H} \in [0, 1/2]$ be given, and suppose that Assumption 4.7 (weak correlation) holds with $C_{\mathsf{exo}} > 0$. Let $f \in \mathcal{V}$, be given, where $\mathcal{V}$ is an arbitrary function class. Then, there is an event $\mathcal{E}$ of probability at least $1 - \delta$ under which the call to $\mathsf{RVFS}^{\mathsf{exo}}_0(f, \mathcal{V}^H, \varnothing, \varnothing, 0; \mathcal{V}, \varepsilon, \zeta_{1:H}, \delta)$ (Algorithm 6) terminates, and throughout the execution of $\mathsf{RVFS}^{\mathsf{exo}}_0$, we have*

$$|\mathcal{C}_h| \leq \lceil 8\varepsilon^{-2} C_{\mathsf{exo}} SAH \rceil.$$

Proof of Lemma J.4. The results follows from Lemma J.3 and Lemma H.1. □

J.1.4 Value Function Regression Guarantee

We next give a guarantee for the estimated value functions $\widehat{V}_{1:H}$ computed within $\mathsf{RVFS}^{\mathsf{exo}}$ in Line 26 of Algorithm 6.

Lemma J.5 (Value function regression guarantee)**.** *Let $h \in [0 \,..\, H]$, $\delta, \varepsilon \in (0,1)$, and $\zeta_{1:H} \in [0,1/2]$ be given, and consider a call to $\mathsf{RVFS}_0^{\mathsf{exo}}$ in the setting of Lemma J.4. Further, let $\mathcal{V}$ be defined as in Eq. (65), and assume that Φ satisfies Assumption 3.3. Then, for any* endogenous *policy π in Π_{S}, there is an event $\mathcal{E}''_h$ of probability at least $1-\delta/H$ under which for any $k \geq 1$, if*

- *$\mathsf{RVFS}_h^{\mathsf{exo}}$ gets called for the kth time during the execution of $\mathsf{RVFS}_0^{\mathsf{exo}}$; and*
- *this kth call terminates and returns $(\widehat{V}_{h:H}, \widehat{\mathcal{V}}_{h:H}, \mathcal{C}_{h:H}, \mathcal{B}_{h:H}, t_{h:H})$,*

then if $(\widehat{\pi}_\tau)_{\tau \geq h}$ is the policy induced by $\widehat{V}_{h:H}$ and N_{reg} is set as in Algorithm 6, we have

$$\sum_{(x_{h-1},a_{h-1})\in\mathcal{C}_h} \frac{1}{N_{\mathsf{reg}}} \sum_{(x_h,-)\in\mathcal{D}_h(x_{h-1},a_{h-1})} \left(\widehat{V}_h(x_h) - V_h^\pi(x_h)\right)^2$$
$$\leq \frac{9kH^2 \log(8k^2H|\mathcal{V}|/\delta)}{N_{\mathsf{reg}}} + 8H^2 \sum_{(x_{h-1},a_{h-1})\in\mathcal{C}_h} \sum_{\tau=h}^{H} \mathbb{E}^{\widehat{\pi}}\left[D_{\mathrm{tv}}(\widehat{\pi}_\tau(\boldsymbol{x}_\tau), \pi_\tau(\boldsymbol{x}_\tau)) \mid \boldsymbol{x}_{h-1} = x_{h-1}, \boldsymbol{a}_{h-1} = a_{h-1}\right],$$

where the datasets $\{\mathcal{D}_h(x,a) : (x,a) \in \mathcal{C}_h\}$ are as in the definition of $\widehat{\mathcal{V}}_h$ in (17).

Proof of Lemma J.5. Since Φ satisfies Assumption 3.3, the function class $\mathcal{V} = \mathcal{V}_{1:H}$ satisfies V^π-realizability for all endogenous policies π (see Lemma 3.1). Thus, the proof of Lemma J.5 follows from that of Lemma H.3 (see Appendix H.4). □

J.1.5 Confidence Sets

We now state a version of the confidence set validity lemma (Lemma H.4) that supports the ExBMDP setting.

Lemma J.6 (Confidence sets)**.** *Let $\varepsilon \in (0,1/2)$ and $\zeta_{1:H} \subset [0,1/2]$ be given, and suppose that*

- *Assumption 4.7 holds with $C_{\mathsf{exo}} > 0$;*
- *The decoder class Φ satisfies Assumption 3.3;*
- *$\zeta_{1:H} \in \mathcal{E}_{\mathsf{rand}}$, where $\mathcal{E}_{\mathsf{rand}}$ is the event in Lemma J.2.*

Let $f \in \mathcal{V}$ be arbitrary. There is an event $\mathcal{E}'''$ of probability at least $1-3\delta$ under which a call to $\mathsf{RVFS}_0^{\mathsf{exo}}(f, \mathcal{V}, \varnothing, \varnothing, 0; \mathcal{V}, \varepsilon, \zeta_{1:H}, \delta)$ terminates and returns tuple $(\widehat{V}_{1:H}, \widehat{\mathcal{V}}_{1:H}, \mathcal{C}_{1:H}, \mathcal{B}_{1:H}, t_{1:H})$ such that

$$\forall h \in [H], \quad V_h^{\bar{\pi}} \in \widehat{\mathcal{V}}_h,$$

where $\bar{\pi}_{1:H}$ is the policy defined recursively via

$$\bar{\pi}_\tau(x) \in \operatorname*{arg\,max}_{a\in\mathcal{A}} \left[\mathcal{P}_\tau[V_{\tau+1}^{\bar{\pi}}](x,a)/\varepsilon + \zeta_h\right], \quad \text{for } \tau = H, \ldots, 1. \tag{68}$$

While the proof of this lemma is very similar to that of Lemma H.4, we need a dedicated treatment to handle the rounding in $\mathsf{RVFS}^{\mathsf{exo}}$. The fully proof of Lemma J.8 is in Appendix H.5.

J.1.6 Main Guarantee for $\mathsf{RVFS}^{\mathsf{exo}}$

We now state the central technical guarantee for $\mathsf{RVFS}^{\mathsf{exo}}$, Lemma J.7, which shows that the base invocation of the algorithm returns a set of value functions $\widehat{V}_{1:H}$ that induce a near-optimal policy $\widehat{\pi}$, as long as the randomized rounding parameters $\zeta_{1:H}$ satisfy $\zeta_{1:H} \in \mathcal{E}_{\mathsf{rand}}$, where $\mathcal{E}_{\mathsf{rand}}$ is the success event in Lemma J.2. The proof of the theorem is in Appendix J.6.

Lemma J.7 (Main guarantee for $\mathsf{RVFS}^{\mathsf{exo}}$)**.** *Let $\delta, \varepsilon \in (0,1)$ and $\zeta_{1:H} \subset [0, \frac{1}{2}]$ be given, and suppose that*

- *Assumption 4.7 holds with $C_{\mathsf{exo}} > 0$;*
- *The decoder class Φ satisfies Assumption 3.3;*

- $\zeta_{1:H} \in \mathcal{E}_{\mathsf{rand}}$, *where* $\mathcal{E}_{\mathsf{rand}}$ *is the event in Lemma J.2.*

Then, for any $f \in \mathcal{V}$*, with probability at least* $1-5\delta$*, a call to* $\mathsf{RVFS}^{\mathsf{exo}}_0(f, \mathcal{V}^H, \varnothing, \varnothing, 0; \mathcal{V}, \varepsilon, \zeta_{1:H}, \delta)$ *(Algorithm 6) terminates and returns value functions* $\widehat{V}_{1:H}$ *such that*

$$\forall h \in [H], \quad \mathbb{P}^{\widehat{\pi}}[\widehat{\boldsymbol{\pi}}_h(\boldsymbol{x}_h) \neq \bar{\pi}_h(\boldsymbol{x}_h)] \leq \frac{\varepsilon^2}{4H^3 SAC_{\mathsf{exo}}},$$

where $\widehat{\boldsymbol{\pi}}_h(x) \in \arg\max_{a \in \mathcal{A}} \lceil \widehat{\boldsymbol{\mathcal{P}}}_{h,\varepsilon,\delta'}[\widehat{V}_{h+1}](x,a)/\varepsilon + \zeta_h \rceil$*, for all* $h \in [H]$*,* $\bar{\pi}$ *is defined as in Eq. (67), and* δ' *is defined as in Algorithm 6. Furthermore, the number of episodes used by* $\mathsf{RVFS}^{\mathsf{exo}}_0$ *is bounded by*

$$\widetilde{O}\left(C_{\mathsf{exo}}^8 S^8 H^{10} A^9 \cdot \varepsilon^{-26}\right).$$

J.1.7 Concluding: Main Guarantee for $\mathsf{RVFS}^{\mathsf{exo}}.\mathsf{bc}$

To conclude, we prove Theorem 4.2, which shows that $\mathsf{RVFS}^{\mathsf{exo}}.\mathsf{bc}$ succeeds with high probability. Recall that $\mathsf{RVFS}^{\mathsf{exo}}.\mathsf{bc}$ (i) invokes $\mathsf{RVFS}^{\mathsf{exo}}$ multiple times for random samples $\zeta_{1:H}$ to ensure that the success event for Lemma J.7 occurs for at least one invocation, and (ii) extracts an executable policy using behavior cloning. Regarding the former point, note that the probability of the success event of Lemma J.2 can be boosted by sampling i.i.d. $\boldsymbol{\zeta}^{(1)}_{1:H}, \dots, \boldsymbol{\zeta}^{(n)}_{1:H} \sim \mathbb{P}^{\zeta}$ inputs to $\mathsf{RVFS}^{\mathsf{exo}}$ for $n \geq 1$; as long as n is polynomially large, with high probability at least one of the inputs $\boldsymbol{\zeta}^{(1)}_{1:H}, \dots, \boldsymbol{\zeta}^{(n)}_{1:H}$ will satisfy the conclusion of Lemma J.2. Thus, it suffices to pick the policy with the highest value function among the different calls to $\mathsf{RVFS}^{\mathsf{exo}}$. Using this, we prove Theorem 4.2.

Proof of Theorem 4.2. Recall that Algorithm 7 picks the final policy $\widehat{\pi}^{(i_{\mathsf{opt}})}_{1:H}$ based on empirical value function estimates. In particular, for every $i \in [N_{\mathsf{boost}}]$ (with N_{boost} as in Algorithm 7), the estimate $\widehat{J}(\widehat{\pi}^{(i)}_{1:H})$ for $J(\widehat{\pi}^{(i)}_{1:H})$ is computed using N_{eval} episodes. Thus, by Hoeffding's inequality and the union bound, we have that there is an event $\breve{\mathcal{E}}$ of probability at least $1-\delta/4$ under which

$$\forall i \in [N_{\mathsf{boost}}], \quad |J(\widehat{\pi}^{(i)}_{1:H}) - \widehat{J}(\widehat{\pi}^{(i)}_{1:H})| \leq \sqrt{2\log(2N_{\mathsf{boost}}/\delta)/N_{\mathsf{eval}}}.$$

Therefore, by definition of i_{opt} in Algorithm 7, we have that under $\breve{\mathcal{E}}$:

$$\begin{aligned}\forall i \in [N_{\mathsf{boost}}], \quad J(\widehat{\pi}^{(i)}_{1:H}) &\leq J(\widehat{\pi}^{(i_{\mathsf{opt}})}_{1:H}) + \sqrt{2\log(2N_{\mathsf{boost}}/\delta)/N_{\mathsf{eval}}}, \\ &\leq J(\widehat{\pi}^{(i_{\mathsf{opt}})}_{1:H}) + \varepsilon/8, \end{aligned} \tag{69}$$

where the last inequality follows by the choice of N_{eval} in Algorithm 7. On the other hand, by Lemma J.2, there is an event $\mathcal{E}^{\mathsf{success}}$ of $\mathbb{P}^{\zeta}$-probability at least

$$1-(24SAH\varepsilon)^{N_{\mathsf{boost}}} \geq 1-\delta/4 \quad \text{(by the choice of } N_{\mathsf{boost}} \text{ in Algorithm 7)}$$

under which there exists $j \in [N_{\mathsf{boost}}]$ such that $\boldsymbol{\zeta}^{(j)}_{1:H} \in \mathcal{E}_{\mathsf{rand}}$, where $\mathcal{E}_{\mathsf{rand}}$ is defined as in Lemma J.2. In what follows, we condition on the event $\mathcal{E}^{\mathsf{success}}$ and let $j \in [N_{\mathsf{boost}}]$ be such that $\boldsymbol{\zeta}^{(j)}_{1:H} \in \mathcal{E}_{\mathsf{rand}}$. Further, we use $\widehat{\pi}^{\mathsf{RVFS}}_{1:H}$ to denote the policy returned by the instance of $\mathsf{RVFS}^{\mathsf{exo}}$ that is used by $\mathsf{RVFS}^{\mathsf{exo}}.\mathsf{bc}$ to learn $\widehat{\pi}^{(j)}_{1:H}$.

By Proposition L.1 (instantiated with $\varepsilon_{\mathsf{mis}} = 0$), there is an event $\widetilde{\mathcal{E}}'$ of probability at least $1-\delta/4$ under which the policy $\widehat{\pi}^{(j)}$ produced by `BehaviorCloning` satisfies

$$J(\widehat{\pi}^{\mathsf{RVFS}}_{1:H}) - J(\widehat{\pi}^{(j)}_{1:H}) \leq \frac{\varepsilon}{2}. \tag{70}$$

By Lemma J.7 and the fact that $\boldsymbol{\zeta}^{(j)}_{1:H} \in \mathcal{E}_{\mathsf{rand}}$, there is an event $\widetilde{\mathcal{E}}$ of probability at least $1-\delta/2$ under which:

$$\forall h \in [H], \quad \forall h \in [H], \quad \mathbb{P}^{\widehat{\pi}}[\widehat{\boldsymbol{\pi}}_h(\boldsymbol{x}_h) \neq \bar{\pi}_h(\boldsymbol{x}_h)] \leq \frac{\varepsilon^2_{\mathsf{RVFS}}}{4H^3 SAC_{\mathsf{exo}}} \leq \frac{\varepsilon}{4H^2}, \tag{71}$$

where $\bar{\pi}(\cdot) := \bar{\pi}(\cdot; \boldsymbol{\zeta}^{(j)}_{1:H}, \varepsilon_{\mathsf{RVFS}})$ which is defined in (67).

Moving forward, we condition on $\breve{\mathcal{E}} \cap \widetilde{\mathcal{E}} \cap \widetilde{\mathcal{E}}'$. By Lemma B.6 (the performance difference lemma), we have

$$\begin{aligned}
J(\bar{\pi}) - J(\widehat{\pi}^{\mathsf{RVFS}}_{1:H}) &= \sum_{h=1}^{H} \mathbb{E}^{\widehat{\pi}^{\mathsf{RVFS}}}[Q^{\bar{\pi}}_h(\boldsymbol{x}_h, \bar{\pi}_h(\boldsymbol{x}_h)) - Q^{\bar{\pi}}_h(\boldsymbol{x}_h, \widehat{\boldsymbol{\pi}}^{\mathsf{RVFS}}_h(\boldsymbol{x}_h))], \\
&\leq H \sum_{h=1}^{H} \mathbb{P}^{\widehat{\pi}}[\widehat{\boldsymbol{\pi}}_h(\boldsymbol{x}_h) \neq \bar{\pi}_h(\boldsymbol{x}_h)], \\
&\leq \varepsilon/4, \qquad (72)
\end{aligned}$$

where the last inequality follows by (71). Now, by Lemma J.1, we have $\bar{\pi} \in \Pi_{2\varepsilon_{\mathsf{RVFS}}}$, and so by Lemma G.1,

$$J(\pi^\star) - J(\bar{\pi}) \leq 6H\varepsilon_{\mathsf{RVFS}} \leq \varepsilon/8, \qquad (73)$$

where the last inequality follows by the choice of $\varepsilon_{\mathsf{RVFS}}$ in Algorithm 7. Combining (73) with (69), (70), and (72), we get that

$$J(\pi^\star) - J(\widehat{\pi}_{1:H}) = J(\pi^\star) - J(\widehat{\pi}^{(i_{\mathrm{opt}})}_{1:H}) \leq \varepsilon.$$

Finally, by the union bound, we have $\mathbb{P}[\breve{\mathcal{E}} \cap \widetilde{\mathcal{E}} \cap \widetilde{\mathcal{E}}'] \geq 1 - \delta$, and so the desired suboptimality guarantee holds with probability at least $1 - \delta$.

Bounding the sample complexity. The sample complexity is dominated by the calls to $\mathsf{RVFS}_0^{\mathsf{exo}}$ within $\mathsf{RVFS}^{\mathsf{exo}}.\mathsf{bc}$ (Algorithm 7). Since $\mathsf{RVFS}^{\mathsf{exo}}.\mathsf{bc}$ calls $\mathsf{RVFS}_0^{\mathsf{exo}}$ with suboptimality parameter $\varepsilon_{\mathsf{RVFS}} = \varepsilon H^{-1}/48$, we get by Lemma J.7 that the total sample complexity is bounded by

$$\widetilde{O}\left(C_{\mathsf{exo}}^8 S^8 H^{36} A^9 \cdot \varepsilon^{-26}\right).$$

□

J.2 Proof of Lemma J.1 (Endogenous Benchmark Policies)

Proof of Lemma J.1. Fix $\delta \in (0,1)$, $\varepsilon \in (0, 1/2)$, and $\zeta'_{1:H} \subset [0, 1/2]$. We show via backward induction over $\ell = H+1, \dots, 1$ that $\bar{\pi}_\tau(\cdot; \zeta'_{1:H}, \varepsilon)$ is endogenous for all $\tau \in [\ell \,..\, H+1]$, with the convention that $\bar{\pi}_{H+1} = \pi_{\mathtt{unif}}$. The base case holds trivially by convention.

Fix $h \in [H]$ and suppose that the induction hypothesis holds for all $\ell \in [h+1 \,..\, H+1]$. We show that it holds for $\ell = h$. First, by the induction hypothesis, $\bar{\pi}_\ell(\cdot; \zeta'_{1:H}, \varepsilon)$ is endogenous for all $\ell \in [h+1 \,..\, H]$. Thus, there exists a function $f_{h+1} : \mathcal{S} \to [0, H-h]$ such that

$$V^{\bar{\pi}}_{h+1}(x') = f_{h+1}(\phi^\star(x')), \quad \forall x' \in \mathcal{X}.$$

Therefore, we have for all $(x, a) \in \mathcal{X} \times \mathcal{A}$:

$$\begin{aligned}
\mathcal{P}_h[V^{\bar{\pi}}_{h+1}](x,a) &= r_h(x,a) + \mathbb{E}[f_{h+1}(\phi^\star(\boldsymbol{x}_{h+1})) \mid \boldsymbol{x}_h = x, \boldsymbol{a}_h = a], \\
&= r_h(x,a) + \mathbb{E}[f_{h+1}(\boldsymbol{s}_{h+1}) \mid \boldsymbol{x}_h = x, \boldsymbol{a}_h = a], \\
&= r_h(x,a) + \mathbb{E}[f_{h+1}(\boldsymbol{s}_{h+1}) \mid \boldsymbol{s}_h = \phi^\star(x), \boldsymbol{a}_h = a], \qquad (74)
\end{aligned}$$

where the last equality follows by the ExBMDP transition structure. Eq. (74) together with the fact that the rewards are endogenous (by assumption) implies that there exists $g_h : \mathcal{S} \times \mathcal{A} \to [0, H-h+1]$ such that

$$\forall (x,a) \in \mathcal{X} \times \mathcal{A}, \quad \mathcal{P}_h[V^{\bar{\pi}}_{h+1}](x,a) = g_h(\phi^\star(x), a),$$

which in turn implies that $x \mapsto \lceil \mathcal{P}_h[V^{\bar{\pi}}_{h+1}](x,a)/\varepsilon + \zeta'_h \rceil$ is only a function of x through $\phi^\star(x)$ for all $a \in \mathcal{A}$. Thus, $\bar{\pi}_h$ is an endogenous policy and the induction is completed.

For the second claim, observe that for the functions $\widetilde{Q}_1, \cdots, \widetilde{Q}_H \in [0, H]^{\mathcal{X} \times \mathcal{A}}$ defined as

$$\forall h \in [H], \forall (x,a) \in \mathcal{X} \times \mathcal{A}, \quad \widetilde{Q}_h(x,a) = \varepsilon \cdot \lceil \mathcal{P}_h[V^{\bar{\pi}}_{h+1}](x,a)/\varepsilon + \zeta'_h \rceil,$$

we have

$$\forall h \in [H], \quad \bar{\pi}_h(\cdot;\zeta'_{1:H},\varepsilon) \in \arg\max_{a\in\mathcal{A}} \widetilde{Q}_h(\cdot,a) \quad \text{and} \quad \|\widetilde{Q}_h - Q_h^{\bar{\pi}}\|_\infty \le 2\varepsilon,$$

which implies that $\bar{\pi}(\cdot;\zeta'_{1:H},\varepsilon) \in \Pi_{2\varepsilon}$. □

J.3 Proof of Lemma J.2 (Snapping Probability)

Proof of Lemma J.2. Fix $\varepsilon \in (0,1)$ and $\delta \in (0,1/2)$. For $\tau \le \ell \in [H]$, let $\mathbb{P}^{\zeta}_{\tau:\ell}$ denote the probability law of $\boldsymbol{\zeta}_\tau,\dots,\boldsymbol{\zeta}_\ell$. We also use the shorthand $\mathbb{P}^{\zeta}_\tau$ for $\mathbb{P}^{\zeta}_{\tau:\tau}$, for all $\tau \in [H]$. We show via backward induction over $\ell = H+1,\dots,1$ that there exists an event $\mathcal{E}_\ell$ of $\mathbb{P}^{\zeta}_{\ell:H}$-probability at least $1 - 24SA(H-\ell+1)\varepsilon$ under which for all $\widetilde{V} \in (\mathcal{X}\times[H] \to [0,H])$:

$$\forall \tau \in [\ell\,..\,H], \quad \tilde{\boldsymbol{\pi}}_\tau(\cdot;\widetilde{V},\boldsymbol{\zeta}_{1:H},\varepsilon,\delta) = \bar{\pi}_\tau(\cdot;\boldsymbol{\zeta}_{1:H},\varepsilon),$$

with the convention that $\bar{\pi}_{H+1} \equiv \tilde{\pi}_{H+1} \equiv \pi_{\mathsf{unif}}$. We then set $\mathcal{E}_{\mathsf{rand}} = \mathcal{E}_1$.

The base case follows trivially by convention.

We now proceed with the inductive step. Fix $h \in [H]$ and suppose that the induction hypothesis holds for all $\ell \in [h+1\,..\,H]$. We show that it holds for $\ell = h$. Throughout, we condition on $\mathcal{E}_{h+1}$. By definition of $\mathcal{E}_{h+1}$, we have for all $\widetilde{V} \in (\mathcal{X}\times[H] \to [0,H])$:

$$\forall \ell \in [h+1\,..\,H], \quad \bar{\pi}_\ell(\cdot;\boldsymbol{\zeta}_{1:H},\varepsilon) = \tilde{\boldsymbol{\pi}}_\ell(\cdot;\widetilde{V},\boldsymbol{\zeta}_{1:H},\varepsilon,\delta).$$

This implies that for all $\widetilde{V} \in (\mathcal{X}\times[H] \to [0,H])$:

$$\forall x \in \mathcal{X}, \quad \tilde{\boldsymbol{\pi}}_h(x;\widetilde{V},\boldsymbol{\zeta}_{1:H},\varepsilon,\delta) \in \arg\max_{a\in\mathcal{A}} \begin{cases} \lceil \widehat{\boldsymbol{Q}}_h(x,a)/\varepsilon + \boldsymbol{\zeta}_h \rceil, & \text{if } \|\widehat{\boldsymbol{Q}}_h(x,\cdot) - \mathcal{P}_h[V^{\bar{\pi}}_{h+1}](x,\cdot)\|_\infty \le 4\varepsilon^2, \\ \lceil \mathcal{P}_h[V^{\bar{\pi}}_{h+1}](x,a)/\varepsilon + \boldsymbol{\zeta}_h \rceil, & \text{otherwise,} \end{cases}$$

by the definition of $\tilde{\pi}_h$ in (66), where $\widehat{\boldsymbol{Q}}_h(\cdot,a) \coloneqq \widehat{\boldsymbol{\mathcal{P}}}_{h,\varepsilon,\delta}[\widetilde{V}_{h+1}](\cdot,a)$. From this, we see that to prove $\tilde{\boldsymbol{\pi}}_h(\cdot;\widetilde{V},\boldsymbol{\zeta}_{1:H},\varepsilon,\delta) = \bar{\pi}_h(\cdot;\boldsymbol{\zeta}_{1:H},\varepsilon)$ for all $\widetilde{V}$, it suffices to show that for all $x \in \mathcal{X}$ and $\widetilde{V}$,

$$\arg\max_{a\in\mathcal{A}} \lceil \widehat{\boldsymbol{Q}}_h(x,a)/\varepsilon + \boldsymbol{\zeta}_h \rceil = \arg\max_{a\in\mathcal{A}} \lceil \mathcal{P}_h[V^{\bar{\pi}}_{h+1}](x,a)/\varepsilon + \boldsymbol{\zeta}_h \rceil, \;\; \text{whenever} \;\; |\widehat{\boldsymbol{Q}}_h(x,a) - \mathcal{P}_h[V^{\bar{\pi}}_{h+1}](x,a)| \le 4\varepsilon^2.$$

Observe that a sufficient condition for this to hold is that

$$\forall x \in \mathcal{X}, \forall a \in \mathcal{A}, \forall \delta \in [-4\varepsilon^2, 4\varepsilon^2], \quad \left\lceil (\mathcal{P}_h[V^{\bar{\pi}}_{h+1}](x,a) + \delta)\cdot\varepsilon^{-1} + \boldsymbol{\zeta}_h \right\rceil = \left\lceil \mathcal{P}_h[V^{\bar{\pi}}_{h+1}](x,a)\cdot\varepsilon^{-1} + \boldsymbol{\zeta}_h \right\rceil, \tag{75}$$

where δ represents all the possible values that the difference $\widehat{\boldsymbol{Q}}_h(x,a) - \mathcal{P}_h[V^{\bar{\pi}}_{h+1}](x,a)$ is allowed to take. By Lemma J.1, we know that $\bar{\pi}$ is endogenous, and so there exists a function $g_h : \mathcal{S}\times\mathcal{A} \to [0, H-h+1]$ such that

$$\forall x \in \mathcal{X}, a \in \mathcal{A}, \quad \mathcal{P}_h[V^{\bar{\pi}}_{h+1}](x,a) = g_h(\phi^\star(x),a).$$

Toward proving Eq. (75), observe that for any $(s,a) \in \mathcal{S} \in \mathcal{A}$, if $\boldsymbol{\zeta}_h$ is such that

$$\begin{aligned} & g_h(s,a)/\varepsilon + \boldsymbol{\zeta}_h + 4\varepsilon \le \lceil g_h(s,a)/\varepsilon + \boldsymbol{\zeta}_h \rceil, \\ \text{and} \qquad & g_h(s,a)/\varepsilon + \boldsymbol{\zeta}_h - 4\varepsilon > \lceil g_h(s,a)/\varepsilon + \boldsymbol{\zeta}_h \rceil - 1, \end{aligned} \tag{76}$$

then, for all $\delta \in [-4\varepsilon^2, 4\varepsilon^2]$ and all $x \in \mathcal{X}$ such that $\phi^\star(x) = s$, we have

$$\lceil (\mathcal{P}_h[V^{\bar{\pi}}_{h+1}](x,a) + \delta)/\varepsilon + \boldsymbol{\zeta}_h \rceil = \lceil (g_h(s,a)+\delta)/\varepsilon + \boldsymbol{\zeta}_h \rceil = \lceil g_h(s,a)/\varepsilon + \boldsymbol{\zeta}_h \rceil = \lceil \mathcal{P}_h[V^{\bar{\pi}}_{h+1}](x,a)/\varepsilon + \boldsymbol{\zeta}_h \rceil.$$

Therefore, if we let $\mathcal{E}_h(s,a)$ denote the event in (76), then under $\bigcap_{(s,a)\in\mathcal{S}\times\mathcal{A}} \mathcal{E}_h(s,a)$, the desired condition in (75) holds. At this point, setting $\mathcal{E}_h = (\bigcap_{(s,a)\in\mathcal{S}\times\mathcal{A}} \mathcal{E}_h(s,a)) \cap \mathcal{E}_{h+1}$ would complete the induction as long as

$\mathbb{P}^{\zeta}_{h:H}[\mathcal{E}_h] \geq 1 - 24SA(H-h+1)\varepsilon$. We now show that this is indeed the case by bounding the probability of the event $\bigcap_{(s,a)\in\mathcal{S}\times\mathcal{A}} \mathcal{E}_h(s,a)$. By the union bound, we have

$$\mathbb{P}^{\zeta}_{h:H}\left[\bigcap_{(s,a)\in\mathcal{S}\times\mathcal{A}} \mathcal{E}_h(s,a) \mid \mathcal{E}_{h+1}\right] \geq 1 - \sum_{(s,a)\in\mathcal{S}\times\mathcal{A}} \mathbb{P}^{\zeta}_{h:H}\left[\mathcal{E}_h(s,a)^c \mid \mathcal{E}_{h+1}\right], \tag{77}$$

where $\mathcal{E}_h(s,a)^c$ denotes the complement of $\mathcal{E}_h(s,a)$. We now bound the probability

$$\mathbb{P}^{\zeta}_{h:H}\left[\mathcal{E}_h(s,a)^c \mid \mathcal{E}_{h+1}\right].$$

Fix $(s,a)\in\mathcal{S}\times\mathcal{A}$. We have that $\boldsymbol{\zeta}_h \in \mathcal{E}(s,a)^c$ if and only if

$$\begin{aligned} & g_h(s,a)/\varepsilon + \boldsymbol{\zeta}_h + 4\varepsilon > \lceil g_h(s,a)/\varepsilon + \boldsymbol{\zeta}_h \rceil, \\ \text{or} \quad & g_h(s,a)/\varepsilon + \boldsymbol{\zeta}_h - 4\varepsilon \leq \lceil g_h(s,a)/\varepsilon + \boldsymbol{\zeta}_h \rceil - 1. \end{aligned} \tag{78}$$

Now, since $\boldsymbol{\zeta}_h \in [0,1/2]$, Lemma K.3 (instantiated with $(x,\zeta,\nu) = (g_h(s,a)/\varepsilon, \boldsymbol{\zeta}_h, 4\varepsilon)$) implies that (78) holds only if

$$\lceil g_h(s,a)/\varepsilon \rceil - 4\varepsilon \leq g_h(s,a)/\varepsilon + \boldsymbol{\zeta}_h \leq \lceil g_h(s,a)/\varepsilon \rceil + 4\varepsilon \quad \text{or} \quad 0 \leq \boldsymbol{\zeta}_h \leq 4\varepsilon. \tag{79}$$

Further, note that since $\boldsymbol{\zeta}_h$ is uniformly distributed over $[0,1/2]$, the $\mathbb{P}^{\zeta}_h$-probability of the event in (79) is at most the sum of the lengths of the intervals

$$[\lceil g_h(s,a)/\varepsilon \rceil - g_h(s,a)/\varepsilon - 4\varepsilon, \lceil g_h(s,a)/\varepsilon \rceil - g_h(s,a)/\varepsilon + 4\varepsilon] \quad \text{and} \quad [0, 4\varepsilon],$$

multiplied by 2, which is equal to 24ε. Therefore, we have

$$\begin{aligned} & \mathbb{P}^{\zeta}_{h:H}\left[\mathcal{E}_h(s,a)^c \mid \mathcal{E}_{h+1}\right] \\ & \leq \mathbb{P}^{\zeta}_h\left[\lceil g_h(s,a)/\varepsilon \rceil - 4\varepsilon \leq g_h(s,a)/\varepsilon + \boldsymbol{\zeta}_h \leq \lceil g_h(s,a)/\varepsilon \rceil + 4\varepsilon \ \text{ or } \ 0 \leq \boldsymbol{\zeta}_h \leq 4\varepsilon\right] \leq 24\varepsilon. \end{aligned}$$

Combining this with (77), we obtain

$$\mathbb{P}^{\zeta}_{h:H}\left[\bigcap_{(s,a)\in\mathcal{S}\times\mathcal{A}} \mathcal{E}_h(s,a) \mid \mathcal{E}_{h+1}\right] \geq 1 - \sum_{(s,a)\in\mathcal{S}\times\mathcal{A}} \mathbb{P}^{\zeta}_{h:H}\left[\mathcal{E}_h(s,a)^c \mid \mathcal{E}_{h+1}\right] \geq 1 - 24SA\varepsilon.$$

Thus, by setting $\mathcal{E}_h = (\bigcap_{(s,a)\in\mathcal{S}\times\mathcal{A}} \mathcal{E}_h(s,a)) \cap \mathcal{E}_{h+1}$, we get that

$$\begin{aligned} \mathbb{P}^{\zeta}_{h:H}[\mathcal{E}_h] \geq \mathbb{P}^{\zeta}_{h+1:H}[\mathcal{E}_{h+1}] \cdot \mathbb{P}^{\zeta}_{h:H}[\mathcal{E}_h \mid \mathcal{E}_{h+1}] & \geq (1 - 24SA(H-h)\varepsilon)(1 - 24SA\varepsilon), \\ & \geq 1 - 24SA(H-h+1)\varepsilon, \end{aligned}$$

which completes the induction. □

J.4 Proof of Lemma J.3 (Coverability in Weakly Correlated ExBMDP)

Proof of Lemma J.3. Fix $h \in [2\,..\,H]$ and define the measure μ as

$$\mu(x) \coloneqq \sum_{\xi'\in\Xi} q(x' \mid (\phi^\star_h(x'), \xi')) \cdot \mathbb{P}[\boldsymbol{\xi}_h = \xi'] \cdot \mathbb{P}[\boldsymbol{s}_h = \phi^\star_h(x') \mid \boldsymbol{s}_{h-1} = \phi^\star_{h-1}(x), \boldsymbol{a}_{h-1} = a],$$

for all $h \in [H]$ and $x \in \mathcal{X}$. We show that μ satisfies Assumption 4.1 with $C_{\mathsf{push}} = C_{\mathsf{exo}} \cdot SA$. First, note that μ is indeed a probability measure over $\mathcal{X}$. Fix $(x,a,x') \in \mathcal{X}\times\mathcal{A}\times\mathcal{X}$. We have

$$\begin{aligned} & \mathbb{P}[\boldsymbol{x}_h = x' \mid \boldsymbol{x}_{h-1} = x, \boldsymbol{a}_{h-1} = a] \\ & = \frac{\mathbb{P}[\boldsymbol{x}_h = x', \boldsymbol{x}_{h-1} = x \mid \boldsymbol{a}_{h-1} = a]}{\mathbb{P}[\boldsymbol{x}_{h-1} = x \mid \boldsymbol{a}_{h-1} = a]}, \\ & = \frac{\mathbb{P}[\boldsymbol{x}_h = x', \boldsymbol{x}_{h-1} = x \mid \boldsymbol{a}_{h-1} = a]}{\mathbb{P}[\boldsymbol{x}_{h-1} = x]}, \\ & = \frac{\sum_{\xi,\xi'\in\Xi} q(x' \mid (\phi^\star_h(x'), \xi')) \cdot q(x \mid (\phi^\star_{h-1}(x), \xi)) \cdot \mathbb{P}[\boldsymbol{s}_h = \phi^\star_h(x'), \boldsymbol{\xi}_h = \xi', \boldsymbol{s}_{h-1} = \phi^\star_{h-1}(x), \boldsymbol{\xi}_{h-1} = \xi \mid \boldsymbol{a}_{h-1} = a]}{\sum_{\xi\in\Xi} q(x \mid (\phi^\star_{h-1}(x), \xi)) \cdot \mathbb{P}[\boldsymbol{s}_{h-1} = \phi^\star_{h-1}(x), \boldsymbol{\xi}_{h-1} = \xi]}, \end{aligned}$$

and so by the ExBMDP structure:

$$= \frac{\sum_{\xi,\xi'\in\Xi} q(x' \mid (\phi^\star_h(x'),\xi')) \cdot q(x \mid (\phi^\star_{h-1}(x),\xi)) \cdot \mathbb{P}[\boldsymbol{\xi}_h = \xi', \boldsymbol{\xi}_{h-1} = \xi] \cdot \mathbb{P}[\boldsymbol{s}_h = \phi^\star_h(x') \mid \boldsymbol{s}_{h-1} = \phi^\star_{h-1}(x), \boldsymbol{a}_{h-1} = a]}{\sum_{\xi\in\Xi} q(x \mid (\phi^\star_h(x),\xi)) \cdot \mathbb{P}[\boldsymbol{\xi}_{h-1} = \xi]},$$

and by Assumption 4.7

$$\begin{aligned}
&\le C_{\mathsf{exo}} \frac{\sum_{\xi,\xi'\in\Xi} q(x' \mid (\phi^\star_h(x'),\xi')) \cdot q(x \mid (\phi^\star_{h-1}(x),\xi)) \cdot \mathbb{P}[\boldsymbol{\xi}_h = \xi'] \cdot \mathbb{P}[\boldsymbol{\xi}_{h-1} = \xi] \cdot \mathbb{P}[\boldsymbol{s}_h = \phi^\star_h(x') \mid \boldsymbol{s}_{h-1} = \phi^\star_{h-1}(x), \boldsymbol{a}_{h-1} = a]}{\sum_{\xi\in\Xi} q(x \mid (\phi^\star_h(x),\xi)) \cdot \mathbb{P}[\boldsymbol{\xi}_{h-1} = \xi]}, \\
&= C_{\mathsf{exo}} \sum_{\xi'\in\Xi} q(x' \mid (\phi^\star_h(x'),\xi')) \cdot \mathbb{P}[\boldsymbol{\xi}_h = \xi'] \cdot \mathbb{P}[\boldsymbol{s}_h = \phi^\star_h(x') \mid \boldsymbol{s}_{h-1} = \phi^\star_{h-1}(x), \boldsymbol{a}_{h-1} = a], \\
&= C_{\mathsf{exo}} SA \cdot \mu(x'),
\end{aligned}$$

This completes the proof. □

J.5 Proof of Lemma J.6 (Confidence Sets)

To prove Lemma J.6, we need the following consequence of tests in Line 14 passing for all $\ell \in [h+1\,..\,H]$.

Lemma J.8 (Consequence of passed tests)**.** *Let $h \in [0\,..\,H]$, $\varepsilon > 0$, and $\zeta_{1:H} \in [0, 1/2]$ be given and consider a call to $\mathsf{RVFS}^{\mathsf{exo}}_0$ (Algorithm 6) in the setting of Lemma J.4. Further, let $\mathcal{E}$ be the event of Lemma J.4. There exists an event $\mathcal{E}'_h$ of probability at least $1 - \delta/H$ such that under $\mathcal{E} \cap \mathcal{E}'_h$, if a call to $\mathsf{RVFS}^{\mathsf{exo}}_h$ during the execution of $\mathsf{RVFS}^{\mathsf{exo}}_0$ terminates and returns $(\widehat{V}_{h:H}, \widehat{\mathcal{V}}_{h:H}, \mathcal{C}_{h:H}, \mathcal{B}_{h:H}, t_{h:H})$, then for any $(x_{h-1}, a_{h-1}) \in \mathcal{C}_h$ and $\ell \in [h+1\,..\,H+1]$:*

$$\mathbb{P}^{\widehat{\pi}}\left[\sup_{f\in\widehat{\mathcal{V}}_\ell} \max_{a\in\mathcal{A}} \left|(\mathcal{P}_{\ell-1}[\widehat{V}_\ell] - \mathcal{P}_{\ell-1}[f_\ell])(\boldsymbol{x}_{\ell-1}, a)\right| > 3\varepsilon^2 \mid \boldsymbol{x}_{h-1} = x_{h-1}, \boldsymbol{a}_{h-1} = a_{h-1}\right] \le \frac{4\log(8M^6 N^2_{\mathsf{test}} H^8/\delta)}{N_{\mathsf{test}}},$$

where $(\widehat{\pi}_\tau)_{\tau\ge h} \subset \Pi_{\mathsf{S}}$, M, and N_{test} are as in $\mathsf{RVFS}^{\mathsf{exo}}_h$ (Algorithm 6).

Proof of Lemma J.8. This is just a restatement of Lemma H.2, and the proof is exactly the same as the latter. □

We will also use Lemma H.5; even though this result is stated in section for the V^π-realizable setting, it is also applicable to the ExBMDP variant of RVFS as it merely says something about the order in which the $(\mathsf{RVFS}^{\mathsf{exo}}_h)$ instances are called. With this, we now prove Lemma J.6.

Proof of Lemma J.6. The proof is very similar to that of Lemma H.2, with differences to account for the "coarsening" of the learned and benchmark policies.

We prove the desired result for $\mathcal{E}''' \coloneqq \mathcal{E} \cap \mathcal{E}'_1 \cap \mathcal{E}''_1 \cap \cdots \cap \mathcal{E}'_H \cap \mathcal{E}''_H$, where $\mathcal{E}$, $(\mathcal{E}'_h)$, and $(\mathcal{E}''_h)$ are the events in Lemma J.4, Lemma J.8, and Lemma J.5, respectively. Throughout, we condition on $\mathcal{E}'''$. First, note that by Lemma J.4, $\mathsf{RVFS}^{\mathsf{exo}}_0$ terminates. Let $(\widehat{V}_{1:H}, \widehat{\mathcal{V}}_{1:H}, \mathcal{C}_{1:H}, \mathcal{B}_{1:H}, t_{1:H})$ be its returned tuple.

We show via backward induction over $\ell = H+1, \dots, 1$, that

$$V^{\tilde{\pi}}_\ell \in \widehat{\mathcal{V}}_\ell, \tag{80}$$

where $\tilde{\pi}_{1:H}$ is the stochastic policy defined recursively via

$$\forall x \in \mathcal{X},\ \tilde{\boldsymbol{\pi}}_\tau(x; \zeta_{1:H}, \varepsilon, \delta) \in \operatorname*{arg\,max}_{a\in\mathcal{A}} \begin{cases} \lceil \widehat{\boldsymbol{Q}}_\tau(x,a)/\varepsilon + \zeta_\tau \rceil, & \text{if } \|\widehat{\boldsymbol{Q}}_\tau(x,\cdot) - \mathcal{P}_\tau[V^{\tilde{\pi}}_{\tau+1}](x,\cdot)\|_\infty \le 4\varepsilon^2, \\ \lceil \mathcal{P}_\tau[V^{\tilde{\pi}}_{\tau+1}](x,a)/\varepsilon + \zeta_\tau \rceil, & \text{otherwise,} \end{cases}$$

for $\tau = H, \dots, 1$, where $\widehat{\boldsymbol{Q}}_\tau(\cdot, a) \coloneqq \widehat{\boldsymbol{\mathcal{P}}}_{\tau,\varepsilon,\delta}[\widehat{V}_{\tau+1}](\cdot, a)$. Note that since $\zeta_{1:H} \in \mathcal{E}_{\mathsf{rand}}$ (for $\mathcal{E}_{\mathsf{rand}}$ is defined in Lemma J.2), we have $\tilde{\pi} \equiv \bar{\pi}$, where $\bar{\pi}$ is as in (68). Thus, instantiating the induction hypothesis with $\ell = h$ and using the definition of the confidence sets $(\widehat{\mathcal{V}}_\ell)$ in (17) together with $V^{\tilde{\pi}}_h \equiv V^{\bar{\pi}}_h$ (since $\tilde{\pi} \equiv \bar{\pi}$) implies the desired result.

Base case $[\ell = H+1]$. Holds trivially since $V^{\pi}_{H+1} \equiv 0$ for any $\pi \in \Pi_{\mathsf{S}}$ by convention.

General case $[\ell \leq H]$. Fix $h \in [H]$ and suppose that (80) holds for all $\ell \in [h+1 \mathinner{..} H+1]$. We show that this remains true for $\ell = h$. First, note that if $\mathsf{RVFS}^{\mathsf{exo}}_h$ is never called during the execution of $\mathsf{RVFS}^{\mathsf{exo}}_0$, then $\widehat{\mathcal{V}}_h = \mathcal{V}_h$, and so (80) holds for $\ell = h$, since $\tilde{\pi} = \bar{\pi}$ is endogenous under $\zeta_{1:H} \in \mathcal{E}_{\mathsf{rand}}$, where $\mathcal{E}_{\mathsf{rand}}$ is the event in Lemma J.2.

Now, suppose that $\mathsf{RVFS}^{\mathsf{exo}}_h$ is called at least once, and let $(\widehat{V}^{+}_{h:H}, \widehat{\mathcal{V}}^{+}_{h:H}, \mathcal{C}^{+}_{h:H}, \mathcal{B}^{+}_{h:H}, t^{+}_{h:H})$ be the output of the last call to $\mathsf{RVFS}^{\mathsf{exo}}_h$ throughout the execution of $\mathsf{RVFS}^{\mathsf{exo}}_0$. Next, we show that

$$(\widehat{V}^{+}_{h:H}, \widehat{\mathcal{V}}^{+}_{h:H}, \mathcal{C}^{+}_{h:H}) = (\widehat{V}_{h:H}, \widehat{\mathcal{V}}_{h:H}, \mathcal{C}_{h:H}). \tag{81}$$

The for-loop in Line 16 ensures that no instance of $(\mathsf{RVFS}^{\mathsf{exo}}_{\tau})_{\tau > h}$ can be called after the last call to $\mathsf{RVFS}^{\mathsf{exo}}_h$ (see Lemma H.5). Thus, the estimated value functions, confidence sets, and core sets for layers $h+1, \dots, H$ remain unchanged after the last call to $\mathsf{RVFS}^{\mathsf{exo}}_h$; that is, (81) holds. Thus, by Lemma J.8, and since we are conditioning on $\mathcal{E}'_{h+1:H}$, we have that for all $(x_{h-1}, a_{h-1}) \in \mathcal{C}_h$ and $\ell \in [h+1 \mathinner{..} H+1]$:

$$\mathbb{P}^{\widehat{\pi}}\left[\sup_{f \in \widehat{\mathcal{V}}_{\ell}} \max_{a \in \mathcal{A}} \left|(\mathcal{P}_{\ell-1}[\widehat{V}_{\ell}] - \mathcal{P}_{\ell-1}[f_{\ell}])(\boldsymbol{x}_{\ell-1}, a)\right| > 3\varepsilon^2 \mid \boldsymbol{x}_{h-1} = x_{h-1}, \boldsymbol{a}_{h-1} = a_{h-1}\right] \leq \frac{4\log(8M^6 N^2_{\mathsf{test}} H^8/\delta)}{N_{\mathsf{test}}}. \tag{82}$$

Now, by the induction hypothesis, we have $V^{\tilde{\pi}}_{\ell} \in \widehat{\mathcal{V}}_{\ell}$, and so substituting $V^{\tilde{\pi}}_{\ell}$ for f_{ℓ} in (82), we get that for all $(x_{h-1}, a_{h-1}) \in \mathcal{C}_h$ and $\ell \in [h+1 \mathinner{..} H+1]$:

$$\mathbb{P}^{\widehat{\pi}}\left[\max_{a \in \mathcal{A}} \left|(\mathcal{P}_{\ell-1}[\widehat{V}_{\ell}] - \mathcal{P}_{\ell-1}[V^{\tilde{\pi}}_{\ell}])(\boldsymbol{x}_{\ell-1}, a)\right| > 3\varepsilon^2 \mid \boldsymbol{x}_{h-1} = x_{h-1}, \boldsymbol{a}_{h-1} = a_{h-1}\right] \leq \frac{4\log(8M^6 N^2_{\mathsf{test}} H^8/\delta)}{N_{\mathsf{test}}}.$$

Therefore, by Lemma K.2 (instantiated with $\mu[\cdot] = \mathbb{P}^{\widehat{\pi}}[\cdot \mid \boldsymbol{x}_{h-1} = x_{h-1}, \boldsymbol{a}_{h-1} = a_{h-1}]$, $\tau = \ell - 1$, $\varepsilon' = \varepsilon^2$, and $V_{\tau+1} = V^{\tilde{\pi}}_{\ell}$), we have that for all $(x_{h-1}, a_{h-1}) \in \mathcal{C}_h$ and $\ell \in [h+1 \mathinner{..} H+1]$:

$$\mathbb{E}^{\widehat{\pi}}[D_{\mathrm{tv}}(\widehat{\pi}_{\ell-1}(\boldsymbol{x}_{\ell-1}), \tilde{\pi}_{\ell-1}(\boldsymbol{x}_{\ell-1})) \mid \boldsymbol{x}_{h-1} = x_{h-1}, \boldsymbol{a}_{h-1} = a_{h-1}] \leq \frac{4\log(8M^6 N^2_{\mathsf{test}} H^8/\delta)}{N_{\mathsf{test}}} + \delta'. \tag{83}$$

Now, since $\bar{\pi}(\cdot; \zeta_{1:H}, \varepsilon)$ is endogenous and $\tilde{\pi} \equiv \bar{\pi}$ (thanks to $\zeta_{1:H} \in \mathcal{E}_{\mathsf{rand}}$), Lemma J.5 (applied with $\pi = \bar{\pi}$) and the conditioning on $\mathcal{E}''_{h+1:H}$ imply that:

$$\begin{aligned}
&\sum_{(x_{h-1}, a_{h-1}) \in \mathcal{C}_h} \frac{1}{N_{\mathsf{reg}}} \sum_{(x_h, -) \in \mathcal{D}_h(x_{h-1}, a_{h-1})} \left(\widehat{V}_h(x_h) - V^{\tilde{\pi}}_h(x_h)\right)^2 \\
&\leq \frac{9kH^2 \log(8k^2 H|\mathcal{V}|/\delta)}{N_{\mathsf{reg}}} + 8H^2 \sum_{(x_{h-1}, a_{h-1}) \in \mathcal{C}_h} \sum_{\tau=h}^{H} \mathbb{E}^{\widehat{\pi}}\left[D_{\mathrm{tv}}(\widehat{\pi}_{\tau}(\boldsymbol{x}_{\tau}), \tilde{\pi}_{\tau}(\boldsymbol{x}_{\tau})) \mid \boldsymbol{x}_{h-1} = x_{h-1}, \boldsymbol{a}_{h-1} = a_{h-1}\right],
\end{aligned} \tag{84}$$

where the datasets $\{\mathcal{D}_h(x, a) : (x, a) \in \mathcal{C}_h\}$ are as in the definition of $\widehat{\mathcal{V}}_h$ in (17). Combining (84) with (83), we conclude that

$$\begin{aligned}
&\sum_{(x_{h-1}, a_{h-1}) \in \mathcal{C}_h} \frac{1}{N_{\mathsf{reg}}} \sum_{(x_h, -) \in \mathcal{D}_h(x_{h-1}, a_{h-1})} \left(\widehat{V}_h(x_h) - V^{\tilde{\pi}}_h(x_h)\right)^2 \\
&\leq \frac{9MH^2 \log(8M^2 H|\mathcal{V}|/\delta)}{N_{\mathsf{reg}}} + 8MH^3 \cdot \frac{4\log(8M^6 N^2_{\mathsf{test}} H^8/\delta)}{N_{\mathsf{test}}} + 8MH^3 \delta', \\
&= \frac{9MH^2 \log(8M^2 H|\mathcal{V}|/\delta)}{N_{\mathsf{reg}}} + 8MH^3 \cdot \frac{4\log(8M^6 N^2_{\mathsf{test}} H^8/\delta)}{N_{\mathsf{test}}} + 8MH^3 \frac{\delta}{4M^7 N^2_{\mathsf{test}} H^8 |\mathcal{V}|}, \\
&\leq \varepsilon^2_{\mathsf{reg}},
\end{aligned} \tag{85}$$

where we have used that $|\mathcal{C}_h| \leq M$. By the definition of $\widehat{\mathcal{V}}_h$ in (17), (85) implies that $V^{\tilde{\pi}}_h \in \widehat{\mathcal{V}}_h$, which completes the induction. □

J.6 Proof of Lemma J.7 (Main Guarantee of $\mathsf{RVFS}^{\mathsf{exo}}$)

Proof of Lemma J.7. We condition on the event $\widetilde{\mathcal{E}} \coloneqq \mathcal{E} \cap \mathcal{E}''' \cap \mathcal{E}'_1 \cap \dots \cap \mathcal{E}'_H$, where $\mathcal{E}$, $\mathcal{E}'''$, and $(\mathcal{E}'_h)$ are the events in Lemma J.4, Lemma J.6, and Lemma J.8, respectively. Note that by the union bound, we have $\mathbb{P}[\widetilde{\mathcal{E}}] \geq 1 - 5\delta$. By Lemma J.6, we have that

$$\forall h \in [H], \quad \mathbb{P}^{\widehat{\pi}}\left[\sup_{f \in \widehat{\mathcal{V}}_h} \max_{a \in \mathcal{A}} \left|(\mathcal{P}_{h-1}[\widehat{V}_h] - \mathcal{P}_{h-1}[f_h])(\boldsymbol{x}_{h-1}, a)\right| > 3\varepsilon^2\right] \leq \frac{4\log(8M^6 N_{\mathsf{test}}^2 H^8/\delta)}{N_{\mathsf{test}}}, \tag{86}$$

where $M = \lceil 8\varepsilon^{-2} C_{\mathsf{exo}} SAH \rceil$ and $N_{\mathsf{test}} = 2^8 M^2 H \varepsilon^{-2} \log(8M^6 H^8 \varepsilon^{-2} \delta^{-1})$. On the other hand, by Lemma J.6, we have

$$\forall h \in [H], \quad V_h^{\tilde{\pi}} \in \widehat{\mathcal{V}}_h.$$

Thus, substituting $V_h^{\tilde{\pi}}$ for f_h in (86) we get that for all $h \in [H+1]$.

$$\mathbb{P}^{\widehat{\pi}}\left[\max_{a \in \mathcal{A}} \left|(\mathcal{P}_{h-1}[\widehat{V}_h] - \mathcal{P}_{h-1}[V_h^{\tilde{\pi}}])(\boldsymbol{x}_{h-1}, a)\right| > 3\varepsilon^2\right] \leq \frac{4\log(8M^6 N_{\mathsf{test}}^2 H^8/\delta)}{N_{\mathsf{test}}}.$$

This together with Lemma K.2, instantiated with $\mu[\cdot] = \mathbb{P}^{\widehat{\pi}}[\cdot]$; $\tau = h-1$; $V_{\tau+1} = V_h^{\tilde{\pi}}$; and $\delta = \delta'$ (with δ' as in Algorithm 3), translates to:

$$\begin{aligned}
\forall h \in [H], \quad \mathbb{E}^{\widehat{\pi}}[D_{\mathrm{tv}}(\tilde{\pi}_h(\boldsymbol{x}_h), \widehat{\pi}_h(\boldsymbol{x}_h))] &\leq \frac{4\log(8M^6 N_{\mathsf{test}}^2 H^8/\delta)}{N_{\mathsf{test}}} + \delta', \\
&= \frac{4\log(8M^6 N_{\mathsf{test}}^2 H^8/\delta)}{N_{\mathsf{test}}} + \frac{\delta}{4M^7 N_{\mathsf{test}}^2 H^8 |\mathcal{V}|}, \\
&\leq \frac{\varepsilon^2}{4H^3 SAC_{\mathsf{exo}}},
\end{aligned} \tag{87}$$

where the last step follows from the fact that $N_{\mathsf{test}} = 2^8 M^2 H \varepsilon^{-2} \log(8M^6 H^8 \varepsilon^{-2} \delta^{-1})$ (with M as in Line 3). Now, since $\zeta_{1:H} \in \mathcal{E}_{\mathsf{rand}}$ (by assumption), Lemma J.2 implies that $\tilde{\pi} \equiv \bar{\pi}$, where the latter is the deterministic policy defined in (67). Thus, by (87), we have

$$\forall h \in [H], \quad \mathbb{P}^{\widehat{\pi}}[\bar{\pi}_h(\boldsymbol{x}_h) \neq \boldsymbol{\widehat{\pi}}_h(\boldsymbol{x}_h)] = \mathbb{E}^{\widehat{\pi}}[D_{\mathrm{tv}}(\tilde{\pi}_h(\boldsymbol{x}_h), \widehat{\pi}_h(\boldsymbol{x}_h))] \leq \frac{\varepsilon^2}{4H^3 SAC_{\mathsf{exo}}},$$

where the first equality follows by the fact that $\mathbb{P}[\bar{\pi}_h(x) \neq \boldsymbol{\widehat{\pi}}_h(x)] = D_{\mathrm{tv}}(\tilde{\pi}_h(x), \widehat{\pi}_h(x))$, for all $x \in \mathcal{X}$, since $\bar{\pi}_h$ is deterministic.

Bounding the sample complexity. We now bound the number of episodes used by Algorithm 6 under $\widetilde{\mathcal{E}}$. First, we fix $h \in [H]$, and focus on the number of episodes used within a to call $\mathsf{RVFS}_h^{\mathsf{exo}}$; excluding any episodes used by any recursive calls to $\mathsf{RVFS}_\tau^{\mathsf{exo}}$ for $\tau > h$. We start by counting the number of episodes used to test the fit of the estimated value functions $\widehat{V}_{h+1:H}$. Starting from Line 8, there are for-loops over $(x_{h-1}, a_{h-1}) \in \mathcal{C}_h$, $\ell = H, \dots, h+1$, and $n \in [N_{\mathsf{test}}]$ to collected partial episodes using the learned policy $\widehat{\pi}$ in Algorithm 6, where $N_{\mathsf{test}} = 2^8 M^2 H \varepsilon^{-2} \log(8M^6 H^8 \varepsilon^{-2} \delta^{-1})$ and $M = \lceil 8\varepsilon^{-2} C_{\mathsf{exo}} SAH \rceil$. Note that $\widehat{\pi}$ uses the local simulator and requires $N_{\mathsf{sim}} = 2\log(4M^7 N_{\mathsf{test}}^2 H^2 |\mathcal{V}|/\delta)/\varepsilon^2$ samples to output an action at each layer (since Algorithm 6 calls Algorithm 5 with confidence level $\delta' = \delta/(8M^7 N_{\mathsf{test}}^2 H^8 |\mathcal{V}|)$). Also, note that whenever a test fails in Line 14, the for-loop in Line 8 resumes. We also know (by Lemma J.4) that the number of times the test fails in Line 14 is at most M. Thus, the number of times the for-loop in Line 8 resumes is bounded by HM; here, H accounts for test failures for all layers $\tau \in [h+1\,..\,H]$. Thus, the number of episodes required to between lines Line 8 and Line 11 is bounded by

$$\#\text{ episodes for roll-outs} \leq \underbrace{MH}_{\#\text{ of times Line 8 resumes}} \cdot \underbrace{MH^2 N_{\mathsf{test}} N_{\mathsf{sim}}}_{\text{Number of episodes in case of no test failures}}. \tag{88}$$

Note that the test in Line 14 also uses local simulator access because it calls the operator $\widehat{\boldsymbol{\mathcal{P}}}$ for every $a \in \mathcal{A}$. Thus, the number of episodes used for the test in Line 14 is bounded by

$$\#\text{ episodes for the tests} \le \underbrace{MH}_{\#\text{ of times Line 8 resumes}} \cdot \underbrace{MHAN_{\mathtt{test}}N_{\mathtt{sim}}}_{\text{Number of episodes used in Line 14}}. \tag{89}$$

We now count the number of episodes used to re-fit the value function; Line 16 onwards. Note that starting from Line 16, there are for-loops over $(x_{h-1}, a_{h-1}) \in \mathcal{C}_h$ and $i \in [N_{\mathtt{reg}}]$ to generate $A \cdot N_{\mathtt{est}}(|\mathcal{C}_h|) \le A \cdot N_{\mathtt{est}}(M)$ partial episodes using $\widehat{\pi}$, where $N_{\mathtt{est}}(k) = 2N_{\mathtt{reg}}^2 \log(8AN_{\mathtt{reg}}Hk^3/\delta)$ is as in Algorithm 6. And, since $\widehat{\pi}$ uses local simulator access and requires $N_{\mathtt{est}}$ samples (see Algorithm 5) to output an action at each layer, the number of episodes used to refit the value function is bounded by

$$\#\text{ episodes for } V\text{-refitting} \le MN_{\mathtt{reg}}AN_{\mathtt{est}}(M)HN_{\mathtt{sim}}. \tag{90}$$

Therefore, by (88), (89), and (90), the number of episodes used within a single call to $\mathsf{RVFS}_h^{\mathsf{exo}}$ (not accounting for episodes used by recursive calls to $\mathsf{RVFS}_\tau^{\mathsf{exo}}$, for $\tau > h$) is bounded by

$$\#\text{ episodes used locally within } \mathsf{RVFS}_h^{\mathsf{exo}} \le M^2H(H+A)N_{\mathtt{test}}N_{\mathtt{sim}} + MN_{\mathtt{reg}}AN_{\mathtt{est}}(M)HN_{\mathtt{sim}}. \tag{91}$$

Finally, by Lemma J.4, $\mathsf{RVFS}_h^{\mathsf{exo}}$ may be called at most M times throughout the execution of $\mathsf{RVFS}_0^{\mathsf{exo}}$. Using this together with (91) and accounting for the number of episodes from all layers $h \in [H]$, we get that the total number of episodes is bounded by

$$M^3H^2(H+A)N_{\mathtt{test}}N_{\mathtt{sim}} + M^2H^2N_{\mathtt{reg}}AN_{\mathtt{est}}(M)N_{\mathtt{sim}}.$$

Substituting the expressions of M, $N_{\mathtt{test}}$, $N_{\mathtt{est}}$, $N_{\mathtt{sim}}$, and $N_{\mathtt{reg}}$ from Algorithm 6 and Algorithm 5, we obtain the desired number of episodes, which concludes the proof. □

K Additional Technical Lemmas

Lemma K.1. *Let $\tau \in [H]$ and $\varepsilon, \delta, \nu \in (0,1)$ be given. Consider two value functions $V_{\tau+1}, \widehat{V}_{\tau+1} \in [0,H]$ and a measure $\mu \in \Delta(\mathcal{X})$ such that*

$$\mathbb{P}_{\boldsymbol{x}_\tau \sim \mu}\left[\mathbb{I}\left\{\max_{a \in \mathcal{A}} \left|(\mathcal{P}_\tau[\widehat{V}_{\tau+1}] - \mathcal{P}_\tau[V_{\tau+1}])(\boldsymbol{x}_\tau, a)\right| > 3\varepsilon\right\}\right] \le \nu. \tag{92}$$

Further, for $x \in \mathcal{X}$, let $\widehat{\boldsymbol{\pi}}_\tau(x) \in \arg\max_{a \in \mathcal{A}} \widehat{\boldsymbol{Q}}_\tau(x,a) \coloneqq \widehat{\boldsymbol{\mathcal{P}}}_{\tau,\varepsilon,\delta}[\widehat{V}_{\tau+1}](x,a)$ and inductively define a randomized policy $\tilde{\pi}$ via

$$\tilde{\boldsymbol{\pi}}_\tau(x) \in \arg\max_{a \in \mathcal{A}} \begin{cases} \widehat{\boldsymbol{Q}}_\tau(x,a), & \text{if } \|\widehat{\boldsymbol{Q}}_\tau(x,\cdot) - \mathcal{P}_\tau[V_{\tau+1}](x,\cdot)\|_\infty \le 4\varepsilon, \\ \mathcal{P}_\tau[V_{\tau+1}](x,a), & \text{otherwise.} \end{cases}$$

Then, we have

$$\mathbb{E}_{\boldsymbol{x}_\tau \sim \mu}[D_{\mathrm{tv}}(\widehat{\pi}_\tau(\boldsymbol{x}_\tau), \tilde{\pi}_\tau(\boldsymbol{x}_\tau))] \le \nu + \delta.$$

Proof of Lemma K.1. In this proof, we let $\mathbb{P}_\mu$ denote the probability law of $\boldsymbol{x}_\tau$ and $\mathbb{P}_{\mathcal{P}}$ denote the probability law of $\widehat{\boldsymbol{\mathcal{P}}}_{\tau,\varepsilon,\delta}$. Denote by $\mathcal{E}$ the $\mathbb{P}_\mu$-measurable event that $\max_{a \in \mathcal{A}} \left|(\mathcal{P}_\tau[\widehat{V}_{\tau+1}] - \mathcal{P}_\tau[V_{\tau+1}])(\boldsymbol{x}_\tau, a)\right| \le 3\varepsilon$. Fix $x \in \mathcal{E}$, and let $\mathcal{E}_x$ be the $\mathbb{P}_{\mathcal{P}}$-measurable event that $\max_{a \in \mathcal{A}} |\widehat{\boldsymbol{\mathcal{P}}}_{\tau,\varepsilon,\delta}[\widehat{V}_{\tau+1}](x,a) - \mathcal{P}_\tau[V_{\tau+1}](x,a)| \le 4\varepsilon$. From the definition of $\tilde{\pi}_\tau$, we have that

$$\begin{aligned}
&D_{\mathrm{tv}}(\widehat{\pi}_\tau(x), \tilde{\pi}_\tau(x))\\
&= \frac{1}{2}\sum_{a\in\mathcal{A}} \left|\mathbb{P}_{\mathcal{P}}\left[\widehat{\boldsymbol{\pi}}_\tau(x) = a\right] - \mathbb{P}_{\mathcal{P}}\left[\tilde{\boldsymbol{\pi}}_\tau(x) = a\right]\right|,\\
&= \frac{1}{2}\sum_{a\in\mathcal{A}} \left|\mathbb{P}_{\mathcal{P}}[\mathcal{E}_x]\mathbb{P}_{\mathcal{P}}\left[\widehat{\boldsymbol{\pi}}_\tau(x) = a \mid \mathcal{E}_x\right] + \mathbb{P}_{\mathcal{P}}[\mathcal{E}_x^c]\mathbb{P}_{\mathcal{P}}\left[\widehat{\boldsymbol{\pi}}_\tau(x) = a \mid \mathcal{E}_x^c\right] - \mathbb{P}_{\mathcal{P}}[\mathcal{E}_x]\mathbb{P}_{\mathcal{P}}\left[\tilde{\boldsymbol{\pi}}_\tau(x) = a \mid \mathcal{E}_x\right] - \mathbb{P}_{\mathcal{P}}[\mathcal{E}_x^c]\mathbb{P}_{\mathcal{P}}\left[\tilde{\boldsymbol{\pi}}_\tau(x) = a \mid \mathcal{E}_x^c\right]\right|,\\
&\le \frac{1}{2}\sum_{a\in\mathcal{A}} \mathbb{P}_{\mathcal{P}}[\mathcal{E}_x]\cdot\left|\mathbb{P}_{\mathcal{P}}\left[\widehat{\boldsymbol{\pi}}_\tau(x) = a \mid \mathcal{E}_x\right] - \mathbb{P}_{\mathcal{P}}\left[\tilde{\boldsymbol{\pi}}_\tau(x) = a \mid \mathcal{E}_x\right]\right|\\
&\quad + \sum_{a\in\mathcal{A}} \mathbb{P}_{\mathcal{P}}[\mathcal{E}_x^c]\cdot\left|\mathbb{P}_{\mathcal{P}}\left[\widehat{\boldsymbol{\pi}}_\tau(x) = a \mid \mathcal{E}_x^c\right] - \mathbb{P}_{\mathcal{P}}\left[\tilde{\boldsymbol{\pi}}_\tau(x) = a \mid \mathcal{E}_x^c\right]\right|, \quad \text{(Jensen's inequality)}
\end{aligned}$$

and since $\mathbb{P}_{\mathcal{P}}\left[\widehat{\boldsymbol{\pi}}_\tau(x) = a \mid \mathcal{E}_x\right] = \mathbb{P}_{\mathcal{P}}\left[\tilde{\boldsymbol{\pi}}_\tau(x) = a \mid \mathcal{E}_x\right]$ $\forall a \in \mathcal{A}$, we have that

$$\begin{aligned}
&= \frac{1}{2}\sum_{a\in\mathcal{A}} \mathbb{P}_{\mathcal{P}}[\mathcal{E}_x^c]\cdot\left|\mathbb{P}_{\mathcal{P}}\left[\widehat{\boldsymbol{\pi}}_\tau(x) = a \mid \mathcal{E}_x^c\right] - \mathbb{P}_{\mathcal{P}}\left[\tilde{\boldsymbol{\pi}}_\tau(x) = a \mid \mathcal{E}_x^c\right]\right|,,\\
&\le \mathbb{P}_{\mathcal{P}}\left[\mathcal{E}_x^c\right],\\
&= \mathbb{P}_{\mathcal{P}}\left[\max_{a\in\mathcal{A}}|\widehat{\boldsymbol{\mathcal{P}}}_{\tau,\varepsilon,\delta}[\widehat{V}_{\tau+1}](x,a) - \mathcal{P}_\tau[V_{\tau+1}](x,a)| > 4\varepsilon\right],\\
&\le \mathbb{P}_{\mathcal{P}}\left[\max_{a\in\mathcal{A}}|\widehat{\boldsymbol{\mathcal{P}}}_{\tau,\varepsilon,\delta}[\widehat{V}_{\tau+1}](x,a) - \mathcal{P}_\tau[\widehat{V}_{\tau+1}](x,a)| > \varepsilon\right], \quad \text{(see below)} \qquad (93)\\
&\le \delta, \qquad (94)
\end{aligned}$$

where (93) follows from $x \in \mathcal{E}$ and the last inequality follows from Lemma G.2. Therefore, we have

$$\begin{aligned}
\mathbb{E}_\mu[D_{\mathrm{tv}}(\widehat{\pi}_\tau(\boldsymbol{x}_\tau), \tilde{\pi}_\tau(\boldsymbol{x}_\tau))] &\le \mathbb{P}_\mu[\mathcal{E}]\cdot\mathbb{E}_\mu[D_{\mathrm{tv}}(\widehat{\pi}_\tau(\boldsymbol{x}_\tau), \tilde{\pi}_\tau(\boldsymbol{x}_\tau)) \mid \mathcal{E}] + \mathbb{P}_\mu[\mathcal{E}^c],\\
&\le \delta + \nu,
\end{aligned}$$

where the first inequality follows by the fact that the total variation distance is bounded by 1, and the last inequality follows by (92) and (94). □

Lemma K.2. *Let $\tau \in [H]$ and $\varepsilon', \delta, \nu \in (0,1)$, and $\zeta_{1:H} \in [0, 1/2]$ be given. Further, consider two value functions $V_{\tau+1}, \widehat{V}_{\tau+1} \in [0, H]$ and measure $\mu \in \Delta(\mathcal{X})$ such that*

$$\mathbb{P}_{\boldsymbol{x}_\tau\sim\mu}\left[\mathbb{I}\left\{\max_{a\in\mathcal{A}}\left|(\mathcal{P}_\tau[\widehat{V}_{\tau+1}] - \mathcal{P}_\tau[V_{\tau+1}])(\boldsymbol{x}_\tau, a)\right| > 3\varepsilon'\right\}\right] \le \nu. \qquad (95)$$

Further, for $x \in \mathcal{X}_\tau$, let $\widehat{\boldsymbol{\pi}}_\tau(x) \in \arg\max_{a\in\mathcal{A}}\lceil \widehat{\boldsymbol{Q}}_\tau(x,a)/\varepsilon' + \zeta_\tau\rceil$, where $\widehat{\boldsymbol{Q}}_\tau(x,a) \coloneqq \widehat{\boldsymbol{\mathcal{P}}}_{\tau,\varepsilon',\delta}[\widehat{V}_{\tau+1}](x,a)$, and inductively define

$$\tilde{\boldsymbol{\pi}}_\tau(x) \in \arg\max_{a\in\mathcal{A}}\begin{cases} \lceil \widehat{\boldsymbol{Q}}_\tau(x,a)/\varepsilon' + \zeta_\tau\rceil, & \text{if } \|\widehat{\boldsymbol{Q}}_\tau(x,\cdot) - \mathcal{P}_\tau[V_{\tau+1}](x,\cdot)\|_\infty \le 4\varepsilon',\\ \lceil \mathcal{P}_\tau[V_{\tau+1}](x,a)/\varepsilon' + \zeta_\tau\rceil, & \text{otherwise.}\end{cases}$$

Then, we have

$$\mathbb{E}_{\boldsymbol{x}_\tau\sim\mu}[D_{\mathrm{tv}}(\widehat{\pi}_\tau(\boldsymbol{x}_\tau), \tilde{\pi}_\tau(\boldsymbol{x}_\tau))] \le \nu + \delta.$$

Proof of Lemma K.2. In this proof, we let $\mathbb{P}_\mu$ denote the probability law of $\boldsymbol{x}_\tau$ and $\mathbb{P}_{\mathcal{P}}$ denote the probability law of $\widehat{\boldsymbol{\mathcal{P}}}_{\tau,\varepsilon,\delta}$. Denote by $\mathcal{E}$ be the $\mathbb{P}_\mu$-measurable event that $\max_{a\in\mathcal{A}}\left|(\mathcal{P}_\tau[\widehat{V}_{\tau+1}] - \mathcal{P}_\tau[V_{\tau+1}])(\boldsymbol{x}_\tau, a)\right| \le 3\varepsilon'$. Fix $x \in \mathcal{E}$, and let $\mathcal{E}_x$ be the $\mathbb{P}_{\mathcal{P}}$-measurable event that $\max_{a\in\mathcal{A}}|\widehat{\boldsymbol{\mathcal{P}}}_{\tau,\varepsilon',\delta}[\widehat{V}_{\tau+1}](x,a) - \mathcal{P}_\tau[V_{\tau+1}](x,a)| \le 4\varepsilon'$. From the definition of $\tilde{\pi}_\tau$, we have that

$$\begin{aligned}
&D_{\mathrm{tv}}(\widehat{\pi}_\tau(x), \tilde{\pi}_\tau(x)) \\
&= \frac{1}{2}\sum_{a\in\mathcal{A}} \left|\mathbb{P}_{\mathcal{P}}\left[\widehat{\boldsymbol{\pi}}_\tau(x) = a\right] - \mathbb{P}_{\mathcal{P}}\left[\tilde{\boldsymbol{\pi}}_\tau(x) = a\right]\right|, \\
&= \frac{1}{2}\sum_{a\in\mathcal{A}} \left|\mathbb{P}_{\mathcal{P}}[\mathcal{E}_x]\mathbb{P}_{\mathcal{P}}\left[\widehat{\boldsymbol{\pi}}_\tau(x) = a \mid \mathcal{E}_x\right] + \mathbb{P}_{\mathcal{P}}[\mathcal{E}_x^c]\mathbb{P}_{\mathcal{P}}\left[\widehat{\boldsymbol{\pi}}_\tau(x) = a \mid \mathcal{E}_x^c\right] - \mathbb{P}_{\mathcal{P}}[\mathcal{E}_x]\mathbb{P}_{\mathcal{P}}\left[\tilde{\boldsymbol{\pi}}_\tau(x) = a \mid \mathcal{E}_x\right] - \mathbb{P}_{\mathcal{P}}[\mathcal{E}_x^c]\mathbb{P}_{\mathcal{P}}\left[\tilde{\boldsymbol{\pi}}_\tau(x) = a \mid \mathcal{E}_x^c\right]\right|, \\
&\le \frac{1}{2}\sum_{a\in\mathcal{A}} \mathbb{P}_{\mathcal{P}}[\mathcal{E}_x]\cdot\left|\mathbb{P}_{\mathcal{P}}\left[\widehat{\boldsymbol{\pi}}_\tau(x) = a \mid \mathcal{E}_x\right] - \mathbb{P}_{\mathcal{P}}\left[\tilde{\boldsymbol{\pi}}_\tau(x) = a \mid \mathcal{E}_x\right]\right| \\
&\quad + \sum_{a\in\mathcal{A}} \mathbb{P}_{\mathcal{P}}[\mathcal{E}_x^c]\cdot\left|\mathbb{P}_{\mathcal{P}}\left[\widehat{\boldsymbol{\pi}}_\tau(x) = a \mid \mathcal{E}_x^c\right] - \mathbb{P}_{\mathcal{P}}\left[\tilde{\boldsymbol{\pi}}_\tau(x) = a \mid \mathcal{E}_x^c\right]\right|, \quad \text{(Jensen's inequality)}
\end{aligned}$$

and since $\mathbb{P}_{\mathcal{P}}\left[\widehat{\boldsymbol{\pi}}_\tau(x) = a \mid \mathcal{E}_x\right] = \mathbb{P}_{\mathcal{P}}\left[\tilde{\boldsymbol{\pi}}_\tau(x) = a \mid \mathcal{E}_x\right]$ $\forall a \in \mathcal{A}$,

$$\begin{aligned}
&= \frac{1}{2}\sum_{a\in\mathcal{A}} \mathbb{P}_{\mathcal{P}}[\mathcal{E}_x^c]\cdot\left|\mathbb{P}_{\mathcal{P}}\left[\widehat{\boldsymbol{\pi}}_\tau(x) = a \mid \mathcal{E}_x^c\right] - \mathbb{P}_{\mathcal{P}}\left[\tilde{\boldsymbol{\pi}}_\tau(x) = a \mid \mathcal{E}_x^c\right]\right|, \\
&\le \mathbb{P}_{\mathcal{P}}\left[\mathcal{E}_x^c\right], \\
&= \mathbb{P}_{\mathcal{P}}\left[\max_{a\in\mathcal{A}} \left|\widehat{\boldsymbol{\mathcal{P}}}_{\tau,\varepsilon',\delta}[\widehat{V}_{\tau+1}](x,a) - \mathcal{P}_\tau[V_{\tau+1}](x,a)\right| > 4\varepsilon'\right], \\
&\le \mathbb{P}_{\mathcal{P}}\left[\max_{a\in\mathcal{A}} \left|\widehat{\boldsymbol{\mathcal{P}}}_{\tau,\varepsilon',\delta}[\widehat{V}_{\tau+1}](x,a) - \mathcal{P}_\tau[\widehat{V}_{\tau+1}](x,a)\right| > \varepsilon'\right], \quad \text{(see below)} && (96) \\
&\le \delta, && (97)
\end{aligned}$$

where (96) follows from $x \in \mathcal{E}$ and the last inequality follows from Lemma G.2. Therefore, we have

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{x}_\tau\sim\mu}[D_{\mathrm{tv}}(\widehat{\pi}_\tau(\boldsymbol{x}_\tau), \tilde{\pi}_\tau(\boldsymbol{x}_\tau))] &\le \mathbb{P}_\mu[\mathcal{E}]\cdot\mathbb{E}_{\boldsymbol{x}_\tau\sim\mu}[D_{\mathrm{tv}}(\widehat{\pi}_\tau(\boldsymbol{x}_\tau), \tilde{\pi}_\tau(\boldsymbol{x}_\tau)) \mid \mathcal{E}] + \mathbb{P}_\mu[\mathcal{E}^c], \\
&\le \delta + \nu,
\end{aligned}$$

where the first inequality follows by the fact that the total variation is bounded by 1, and the last inequality follows by (95) and (97). □

Lemma K.3. *Let $x \in \mathbb{R}$ and $\nu \in (0, 1/2)$ be given. Further, let $\zeta \in (0, 1/2)$. Then,*

$$x + \zeta + \nu > \lceil x + \zeta\rceil \quad \textit{or} \quad x + \zeta - \nu \le \lceil x + \zeta \rceil - 1,$$

only if

$$\lceil x\rceil - \nu \le x + \zeta \le \lceil x\rceil + \nu \quad \textit{or} \quad \zeta \le \nu.$$

Proof of Lemma K.3. To prove the claim, it suffices to show the following items:

1. $x + \zeta + \nu > \lceil x + \zeta\rceil$ only if $\lceil x\rceil \ge x + \zeta > \lceil x\rceil - \nu$; and
2. $x + \zeta - \nu \le \lceil x + \zeta\rceil - 1$ only if $\lceil x\rceil < x + \zeta \le \lceil x\rceil + \nu$ or $\zeta \le \nu$.

We start by showing the first item. We proceed by showing the contrapositive; that is, we will show that if $x + \zeta \le \lceil x\rceil - \nu$ or $x + \zeta > \lceil x\rceil$, then $x + \zeta + \nu \le \lceil x + \zeta\rceil$. Suppose that $x + \zeta \le \lceil x\rceil - \nu$. This, together with the fact that $\zeta \ge 0$, implies that

$$\lceil x + \zeta\rceil = \lceil x\rceil \ge x + \zeta + \nu.$$

Now, suppose that $x + \zeta > \lceil x\rceil$. Then, we have

$$\lceil x + \zeta\rceil \ge \lceil x\rceil + 1 \ge \lceil x\rceil + \zeta + \nu \ge x + \zeta + \nu,$$

where the penultimate inequality follows by $\zeta, \nu \in (0, 1/2)$.

We now prove the second claim. Again, we proceed by showing the contrapositive; that is, we will show that if $\{\lceil x\rceil + \nu < x + \zeta$ or $\lceil x\rceil \geq x + \zeta\}$ and $\zeta > \nu$, then $x + \zeta - \nu > \lceil x + \zeta\rceil - 1$.

Suppose that $\lceil x\rceil + \nu < x + \zeta$ and $\zeta > \nu$. The first inequality together with $\nu \geq 0$ implies that $\lceil x + \zeta\rceil > \lceil x\rceil$. On the other hand, since $\zeta \leq 1/2$, we have $\lceil x + \zeta\rceil \leq \lceil x\rceil + 1$, and so

$$\lceil x + \zeta\rceil - 1 = \lceil x\rceil < x + \zeta - \nu,$$

where the last inequality follows by the current assumption that $\lceil x\rceil + \nu < x + \zeta$.

Now, suppose that $\lceil x\rceil \geq x + \zeta$ and that $\zeta > \nu$. Then, we have

$$\lceil x + \zeta\rceil \leq \lceil x\rceil \leq x + 1 < x + \zeta - \nu + 1, \tag{98}$$

where the last inequality follows by $\zeta > \nu$. Rearranging (98) completes the proof. □

L BehaviorCloning Algorithm and Analysis

In this section, we give a self-contained presentation and analysis for the standard *behavior cloning* algorithm for imitation learning (e.g., Ross and Bagnell (2010)), displayed in Algorithm 8. Given access to trajectories from an expert policy $\widehat{\pi}_{1:H}$ (which may be non-executable in the sense of Definition 2.1) the algorithm learns an executable policy π^{bc} with similar performance. We use this scheme within RVFS.bc and RVFS$^{\mathsf{exo}}$.bc.

Algorithm 8 BehaviorCloning: Imitation Learning Algorithm.

```
input: Policy class $\Pi \subseteq \Pi_{\mathsf{S}}$, expert policy $\widehat{\pi}_{1:H}$, suboptimality $\varepsilon \in (0,1)$, and confidence $\delta \in (0,1)$.
Set $N_{\mathsf{bc}} = 16H^2 \log(|\Pi|/\delta)/\varepsilon$.
Set $\mathcal{D} \leftarrow \emptyset$.
for $i = 1, \dots, N_{\mathsf{bc}}$ do
    Generate trajectory $\boldsymbol{\tau} = ((\boldsymbol{x}_1, \boldsymbol{a}_1), \dots, (\boldsymbol{x}_H, \boldsymbol{a}_H)) \sim \mathbb{P}^{\widehat{\pi}}$.
    Update $\mathcal{D} \leftarrow \mathcal{D} \cup \{\boldsymbol{\tau}\}$.
Compute $\pi^{\mathsf{bc}} \in \arg\min_{\pi \in \Pi} \sum_{((x_1,a_1),\dots,(x_H,a_H)) \in \mathcal{D}} \sum_{h \in [H]} \mathbb{I}\{\boldsymbol{\pi}_h(x_h) \neq a_h\}$.
Return $\pi^{\mathsf{bc}}$.
```

Proposition L.1. *Let $\varepsilon, \delta \in (0,1)$ be given and let $\Pi \subseteq \Pi_{\mathsf{S}}$ and $\widehat{\pi}_{1:H}$ be an expert policy such that*

$$\inf_{\pi \in \Pi} \sum_{h=1}^{H} \mathbb{P}^{\widehat{\pi}}[\widehat{\boldsymbol{\pi}}_h(\boldsymbol{x}_h) \neq \boldsymbol{\pi}_h(\boldsymbol{x}_h)] \leq \varepsilon_{\mathtt{mis}}. \tag{99}$$

Then, the policy $\pi^{\mathsf{bc}}_{1:H} =$ BehaviorCloning$(\Pi, \varepsilon, \widehat{\pi}_{1:H}, \delta)$ returned by Algorithm 8 satisfies, with probability at least $1 - \delta$,

$$J(\widehat{\pi}) - J(\pi^{\mathsf{bc}}) \leq 4H\varepsilon_{\mathtt{mis}} + \varepsilon/2.$$

Proof of Proposition L.1. First, by the performance difference lemma, we have

$$\begin{aligned}
\mathbb{E}[V_1^{\widehat{\pi}}(\boldsymbol{x}_1)] - \mathbb{E}[V_1^{\pi^{\mathsf{bc}}}(\boldsymbol{x}_1)] &= \sum_{h=1}^{H} \mathbb{E}^{\widehat{\pi}}[Q_h^{\pi^{\mathsf{bc}}}(\boldsymbol{x}_h, \widehat{\boldsymbol{\pi}}_h(\boldsymbol{x}_h)) - Q_h^{\pi^{\mathsf{bc}}}(\boldsymbol{x}_h, \boldsymbol{\pi}_h^{\mathsf{bc}}(\boldsymbol{x}_h))], \\
&\leq H \sum_{h=1}^{H} \mathbb{P}^{\widehat{\pi}}[\widehat{\boldsymbol{\pi}}_h(\boldsymbol{x}_h) \neq \boldsymbol{\pi}_h^{\mathsf{bc}}(\boldsymbol{x}_h)]. \qquad (100)
\end{aligned}$$

We now bound the probability terms on the right-hand side. Fix $h \in [H]$ and let $\mathcal{D}$ be the dataset in Algorithm 8, which consists of N_{bc} i.i.d. trajectories $((\boldsymbol{x}_1, \boldsymbol{a}_1), \dots, (\boldsymbol{x}_H, \boldsymbol{a}_H))$ generated by rolling with $\widehat{\pi}_{1:H}$.

By Lemma B.4 (with i.i.d. data, $B = H$, and $\mathcal{Q} = \Pi$), we have that, with probability at least $1 - \delta$,

$$\begin{aligned}\forall \pi \in \Pi, \quad \sum_{((x_1,a_H),\dots,(x_H,a_H))\in\mathcal{D}} \sum_{h\in[H]} \mathbb{I}\{\boldsymbol{\pi}_h(x_h) \neq \widehat{\boldsymbol{\pi}}(x_h)\} &\leq 2\sum_{h\in[H]} \mathbb{P}^{\widehat{\pi}}[\boldsymbol{\pi}_h(\boldsymbol{x}_h) \neq \widehat{\boldsymbol{\pi}}_h(\boldsymbol{x}_h)] \\ &\quad + \frac{2H\log(2|\Pi|/\delta)}{N_{\mathsf{bc}}},\end{aligned} \tag{101}$$

and

$$\begin{aligned}\forall \pi \in \Pi, \quad \sum_{h\in[H]} \mathbb{P}^{\widehat{\pi}}[\boldsymbol{\pi}_h(\boldsymbol{x}_h) \neq \widehat{\boldsymbol{\pi}}_h(\boldsymbol{x}_h)] &\leq 2 \sum_{((x_1,a_H),\dots,(x_H,a_H))\in\mathcal{D}} \sum_{h\in[H]} \mathbb{I}\{\boldsymbol{\pi}_h(x_h) \neq \widehat{\boldsymbol{\pi}}(x_h)\} \\ &\quad + \frac{4H\log(2|\Pi|/\delta)}{N_{\mathsf{bc}}}.\end{aligned} \tag{102}$$

Taking the infimum over π on both sides of (101) and using the definition of π^{bc}_h in Algorithm 8 gives:

$$\begin{aligned}\sum_{((x_1,a_H),\dots,(x_H,a_H))\in\mathcal{D}} \sum_{h\in[H]} \mathbb{I}\{\boldsymbol{\pi}^{\mathsf{bc}}_h(x_h) \neq \widehat{\boldsymbol{\pi}}(x_h)\} &\leq 2\inf_{\pi\in\Pi}\sum_{h\in[H]} \mathbb{P}^{\widehat{\pi}}[\boldsymbol{\pi}_h(\boldsymbol{x}_h) \neq \widehat{\boldsymbol{\pi}}_h(\boldsymbol{x}_h)] \\ &\quad + \frac{2H\log(2|\Pi|/\delta)}{N_{\mathsf{bc}}}, \\ &\leq 2\varepsilon_{\mathtt{mis}} + \frac{2H\log(2|\Pi|/\delta)}{N_{\mathsf{bc}}},\end{aligned}$$

where the last inequality follows from (99). Using this together with (102), instantiated with $\pi \equiv \pi^{\mathsf{bc}}$, we get that with probability at least $1 - \delta$:

$$\sum_{h\in[H]} \mathbb{P}^{\widehat{\pi}}[\boldsymbol{\pi}^{\mathsf{bc}}_h(\boldsymbol{x}_h) \neq \widehat{\boldsymbol{\pi}}_h(\boldsymbol{x}_h)] \leq 4\varepsilon_{\mathtt{mis}} + \frac{8H\log(2|\Pi|/\delta)}{N_{\mathsf{bc}}}.$$

Plugging this into (100), we get that with probability at least $1 - \delta$:

$$\mathbb{E}[V_1^{\widehat{\pi}}(\boldsymbol{x}_1)] - \mathbb{E}[V_1^{\pi^{\mathsf{bc}}}(\boldsymbol{x}_1)] \leq 4H\varepsilon_{\mathtt{mis}} + \frac{8H^2\log(2|\Pi|/\delta)}{N_{\mathsf{bc}}} \leq 4H\varepsilon_{\mathtt{mis}} + \varepsilon/2,$$

where the last inequality follows by the fact that $N_{\mathsf{bc}} = 16H^2\log(2|\Pi|/\delta)/\varepsilon$. This completes the proof. □