Hybrid Reinforcement Learning from Offline Observation Alone

Yuda Song [1] **J. Andrew Bagnell** [1 2] **Aarti Singh** [1]

Abstract

We consider the hybrid reinforcement learning setting where the agent has access to both offline data and online interactive access. While RL research typically assumes offline data contains complete action, reward and transition information, datasets with only state information (also known as *observation-only* datasets) are more general, abundant and practical. This motivates our study of the *hybrid RL with observation-only offline dataset* framework. While the task of competing with the best policy "covered" by the offline data can be solved if a *reset* model of the environment is provided (i.e., one that can be reset to any state), we show evidence of hardness of competing when only given the weaker *trace* model (i.e., one can only reset to the initial states and must produce full traces through the environment), without further assumption of *admissibility* of the offline data. Under the admissibility assumptions– that the offline data could actually be produced by the policy class we consider– we propose the first algorithm in the trace model setting that provably matches the performance of algorithms that leverage a reset model. We also perform proof-of-concept experiments that suggest the effectiveness of our algorithm in practice.

1. Introduction

Recently, explosive growth in the availability of offline data for interactive decision making problems (Dasari et al., 2019; Qin et al., 2022; Mathieu et al., 2023; Padalkar et al., 2023), combined with an ability to interact with the environment for feedback, led to the advancement of hybrid Reinforcement Learning (hybrid RL) (Ross & Bagnell, 2012; Song et al., 2022). This setup blends the exploratory strengths of offline data with the ability to adapt the data distribution using online interaction with the environment. Previously, RL research has tended to focus on either purely offline or online regimes, each with its inherent challenges. Offline learning, while benefiting from the exploration and potential expert "advice" implicit in a large offline dataset, often suffers from instability due to distribution shifts (Wang et al., 2021). Online RL benefits from feedback from the environment but at the cost of increased complexity, both statistically and computationally (Du et al., 2020; Kane et al., 2022), due to the requirement of global exploration. Hybrid RL benefits from the synergy of combining both data sources. Earlier studies in this domain have predominantly utilized canonical offline datasets (Levine et al., 2020), with extensive information of state, action, reward, and subsequent state. This comprehensive data approach has proven beneficial, highlighting the statistical and computational superiority of hybrid RL (Song et al., 2022; Hu et al., 2023), and its robustness against distribution shift (Wagenmaker & Pacchiano, 2023; Ren et al., 2023).

However, the necessity for expansive datasets in such a rich format poses a significant barrier. In practice, most large-scale datasets exist in the format of videos (Grauman et al., 2022; 2023) (imagine using video demonstrations from Youtube). The requirement of annotated actions in the dataset is restrictive because actions do not generalize between different modalities: one should not expect to naively apply human actions to robot actuators, and different robots have different action spaces as well. The approach of collecting different actions for different modalities with human labors (Padalkar et al., 2023) is challenging to scale as the observation-only datasets.

This motivates a more general setting with a weaker offline data requirement without losing the statistical and computational benefit of hybrid RL. In this work, we initiate the study of Hybrid RL from (Offline) Observation Alone (HyRLO) framework where the offline data only contains state[1] information. Previous studies that fall into the HyRLO framework can be generally characterized in two ways: the first leverages the offline state data to perform representation learning (i.e., requires a separate pretraining stage) (Nair et al., 2018; Ma et al., 2022b; Ghosh et al., 2023), and then to use the learned feature map to speed up

[1]Carnegie Mellon University [2]Aurora Innovation. Correspondence to: Yuda Song <yudas@cs.cmu.edu>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

[1]We will use the terms observation and state interchangeably.

Table 1. Comparisons of hybrid RL frameworks and algorithms. We compare the sample complexity, problem setting and assumptions required by each work. Our setting assumes weaker requirement on the offline data and model access (defined in Section 3), while requiring the admissibility assumption (Assumption 5.1) on the offline data. We show that without the admissibility assumption, the problem might exhibit exponential sample complexity separation between the trace model and reset model in Proposition 5.1 and Proposition 5.2. Previous hybrid RL with trace model analysis (Song et al., 2022) requires *explicit structural assumptions* on the MDP and value function (Du et al., 2021) (characterized as the additional d parameter in their sample complexity result), while our analysis does not require such assumptions. Finally, we consider value-based learning, so Q^* realizability denotes the optimal Q-function is contained in the function class, while Q^π realizability requires the function class contains the Q-function for all deterministic policies (for example, all A^S many policies in tabular MDPs). Finally, we note that all methods only require single policy coverage. The original PSDPpaper (Bagnell et al., 2003) did not perform the analysis under single policy coverage, and the first PSDPwith single policy coverage analysis can be found in Bagnell (2004); Scherrer (2014).

	Sample Complexity	Offline Data	Admissibility	Model	Bellman Complete	Realizability
HYQ (Song et al., 2022)	$\frac{C_{\mathsf{be}}^2 H^5 A d \log(\lvert\mathcal{F}\rvert/\delta)}{\epsilon^2}$	Canonical	No	Trace	Yes	Q^*
PSDP (Bagnell et al., 2003)	$\frac{C_{\mathsf{pd}}^2 H^5 A \log(\lvert\mathcal{F}\rvert/\delta)}{\epsilon^2}$	State-only	No	Reset	No	Q^π
This work	$\frac{C_{\mathsf{pd}}^2 H^5 A \log(\lvert\mathcal{F}\rvert\lvert\Pi\rvert/\delta)}{\epsilon^2}$	State-only	Yes (Hardness examples if No)	Trace	Yes	Q^π

the downstream online RL training. However, we show that the state-only offline dataset, although less informative than the canonical offline data, still provides a rich signal for decision-making and not only representation learning. The second previous approach relies on a *reset model* (Kakade & Langford, 2002; Bagnell et al., 2003), which only holds true if a simulator is available and thus does not address many common real-world scenarios. In this work, we show that we can solve HYRLO *without* reset model access– i.e., with a trace model that only allows resets to the initial state. Our approach, in contrast with earlier methods, requires a notion of *admissibility* (Chen & Jiang, 2019) of the offline data which formalizes the idea that the offline data should have been generated by *some* policy or mixture of policies. Indeed, HYRLO fills in a missing piece theoretically where have neither complete data, as in between canonical hybrid RL, and access only to a trace model, a much weaker and more realistic access model for RL problems. We provide a comparison overview in Table 1.

Contributions. We initiate a theoretical study of HYRLO framework and provide the first provable algorithm for HYRLO. Specifically, we introduce:

- **Connections between reset model and trace model.** Given HYRLO can be solved efficiently when a reset model is available (Bagnell et al., 2003), we extend previous work with *a reduction from the trace model to the reset model setting* via an admissibility condition where the offline distribution is realizable by the policy class. Further, we demonstrate evidence that suggests statistical separation between trace model and reset model if the admissibility condition is violated.
- **Efficient algorithm.** We provide the first provably efficient algorithm for HYRLO with only trace model access, FOrward Observation-matching BAckward Reinforcement Learning (FOOBAR). With the admissibility assumption, FOOBAR requires the same order of samples as the previous algorithms (Kakade & Langford, 2002; Bagnell et al., 2003) that demand a reset model to compete with the best policy covered by the offline distribution.
- **General analysis.** Our approach does not require the strong *explicit structural assumptions* such as bilinear rank (Du et al., 2021) on the MDP and value function that previous hybrid RL analysis demanded (Song et al., 2022; Nakamoto et al., 2023). Relaxing this assumption allows our algorithm and analysis to be more general and applicable to a wider range of problems. In addition, we identify situations where FOOBAR succeeds under inadmissible offline data, and provide algorithms and analysis under stationary settings.
- **Empirical evaluation.** We perform experiments to show the effectiveness of our algorithm on two challenging benchmarks: the rich-observation combination lock (Misra et al., 2020) and high-dimensional robotics manipulation tasks (Rajeswaran et al., 2017). We compare with the state-of-the-art hybrid RL algorithms and investigate the gap due to the more limited information in the offline dataset.

2. Related Work

Hybrid RL. Hybrid RL defines the setting where the agent has access to both offline data (usually generated by policies with a mixture of qualities) (Levine et al., 2020) and online interaction access. This learning framework

has recently gained increasing interest due to its potential for efficient learning and practical values (Ross & Bagnell, 2012; Nair et al., 2020; Xie et al., 2021b; Song et al., 2022; Lee et al., 2022; Niu et al., 2022; Ball et al., 2023; Nakamoto et al., 2023; Wagenmaker & Pacchiano, 2023; Li et al., 2023b; Zhang et al., 2023a; Zhang & Zanette, 2023; Vemula et al., 2023; Swamy et al., 2023; Zhou et al., 2023). Previous works follow the standard offline RL setting where the offline dataset contains the state, action, reward and next state information, and they have shown the statistical and computational benefit of the hybrid setting over pure online or offline setting. In this work, we consider a more general and challenging setting where the offline dataset only contains the state information. Many previous works have also considered this setting (Machado et al., 2017; Nair et al., 2018; Schmeckpeper et al., 2020; Ma et al., 2022b; Baker et al., 2022; Seo et al., 2022; Ghosh et al., 2023), but the offline states are only used for representation learning in a separate pertaining stage, not for decision making via RL. In addition, most of the works assume that the offline data consists of the state and next state pair *collected from the same transition*. Instead of a heuristic application of the offline observations, our work conducts the first theoretical study in this setting that captures the minimal properties of the offline distribution, and our proposed algorithm utilizes the offline data for decision-making directly.

Learning from observation alone. Prior to the HyRLO setting, learning from state-only data has also been considered in other interactive decision-making problems, such as imitation learning and offline reinforcement learning. For example, Imitation from observation alone setting (ILfO) (Nair et al., 2017; Torabi et al., 2018; Sun et al., 2019; Smith et al., 2019; Song et al., 2020; Zhu et al., 2020; Radosavovic et al., 2021) considers learning from a dataset of expert states, and with online interaction. If one does not have online access, the offline counterpart is the offline imitation learning setting, where the agent has access to two datasets: one offline state-only dataset (which has a mixture of qualities) and another expert state-only dataset (Kim et al., 2021; Ma et al., 2022a; Yu et al., 2023; Pirotta et al., 2023). However, all these settings require explicitly labeled expert data, while our setting only requires unlabeled offline data that implicitly covers some good policy's trajectory. Recently Li et al. (2023a) removed the expert label requirement, but the offline data is still required to contain additional action or reward information.

RL with reset model. In the reset model setting, one assumes the ability to reset the dynamics to any state. With a reset model, the HyRLO problem can be solved using Policy Search by Dynamic Programming (PSDP) algorithm and others that share a similar core idea (Bagnell et al., 2003; Salimans & Chen, 2018; Uchendu et al., 2023). The reset model has been shown to have other favorable properties that contribute to overcoming the statistical hardness of the more commonly available trace model setting (Amortila et al., 2022; Weisz et al., 2021). On the empirical side, Sharma et al. (2022) shows that if expert data is available, one can learn to reset by training a policy that brings the current policy to the expert state distribution after the rollout. In this paper, we also demonstrate the benefit of such "reset policy". The previous work requires a non-stationary initial distribution for the reset policy due to the interleaving learning between the final policy and the rest policy. Our paper improves over the previous work by removing the non-stationarity with learning a reset policy before the "policy optimization" stage. In addition, the previous work does not apply to any non-reversible system, which restricts its application to real-world problems.

3. Preliminaries

We consider finite horizon MDPs $\mathcal{M} = \{\mathcal{S}, \mathcal{A}, H, R, P\}$, where H is the horizon, $\mathcal{S}$ is the state space with $|\mathcal{S}| = S$, $\mathcal{A}$ is the action space with $|\mathcal{A}| = A$, $R = \{R_h : \mathcal{S} \times \mathcal{A} \to [0,1]\}_{h=1}^H$ is the reward function, $P = \{P_h : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})\}_{h=1}^H$ is the state transition distribution, and $P_0(\emptyset)$ is the initial state distribution. We denote the model $\mathcal{M}$ as the trace model to distinguish it from the reset model that we will introduce later. Given a (potentially nonstationary) policy $\pi \in \Pi = \{\mathcal{S} \to \Delta(\mathcal{A})\}_{h=1}^H$, define the action-value Q-function of π at timestep h as $Q_h^\pi(s_h, a_h) = \mathbb{E}_{\pi,P}\left[\sum_{\tau=h}^H R_\tau(s_\tau, a_\tau)\right]$, and we define the optimal policy as π^*. We define the function class to estimate the Q function as $\mathcal{F} : \{\mathcal{F}_h : \mathcal{S} \times \mathcal{A} \to [0, H]\}_{h=1}^H$. We follow the conventional notation d_h^π to denote either the state (or state-action) occupancy measure induced by π at horizon h.

We study the hybrid RL setting (Song et al., 2022), where the agent has online interaction access to the environment, and in addition, has offline data set $\{\mathcal{D}_h\}_{h=1}^H$. In the canonical hybrid RL setting, each dataset $\mathcal{D}_h$ contains tuples $\{s_h^n, a_h^n, r_h^n, s_{h+1}^n\}_{n=1}^N$, where N is the size of the offline dataset. The data in $\mathcal{D}_h$ is drawn from some distribution μ_h, i.e., $s^n, a^n \sim \mu_h$: for example, μ_h can be the visitation distribution of some policy, and $r_h^n = R_h(s_h^n, a_h^n)$, $s_{h+1}^n \sim P_h(\cdot \mid s_h^n, a_h^n)$. Here we consider the HyRLO setting, where in the offline dataset we only have the single-timestep state data. That is, the offline dataset has the form $\mathcal{D}_h = \{s_h^n\}_{n=1}^N$, where $s_h \sim \mu_h$, and μ_h is some distribution over the states at timestep h.

Following the convention in hybrid RL, the learning goal is to compete with the best policy covered by the offline distribution. For the coverage notation, in the main text, we consider the density ratio coverage for simplicity: given any policy π, we define the density ratio coverage as $C_{\mathsf{cov}}(\pi) =$

$\min_{h\in[H]} \left\| \frac{d_h^\pi}{\mu_h} \right\|_\infty$, where the supremum norm is over states. [2]

To measure the difference between distributions, we define the Integral Probability Metric (IPM) (Müller, 1997) distance between two distributions $\mathbb{P}$ and $\mathbb{Q}$:

$$\mathsf{IPM}_{\mathcal{G}}(\mathbb{P},\mathbb{Q}) = \sup_{g\in\mathcal{G}} \left| \int g \, \mathrm{d}\mathbb{P} - \int g \, \mathrm{d}\mathbb{Q} \right|,$$

which is defined by the test function class $\mathcal{G}$. For example, when $\mathcal{G}$ is all bounded functions, the IPM recovers the Total Variation (TV) distance, and we denote the TV distance as $\|\cdot\|_{\mathsf{TV}}$. When $\mathcal{G}$ is the set of all 1-Lipschitz functions, this definition recovers the 1-Wasserstein distance.

We note the difference between the two different access models for an MDP: we will denote the canonical trace MDPs as we defined above as $\mathcal{M}$, where one can only reset at the initial state P_0 and simulate traces $\tau = \{s_1, a_1, r_1, s_2, \ldots, s_H, a_H, r_H\}$, where $s_1 \sim P_0$, $a_h \sim \pi_h(s_h)$, $r_h = R_h(s_h, a_h)$, $s_{h+1} \sim P_h(s_h, a_h)$. We also consider the *reset* access model with the ability to simulate a reward and transition from any state-action pair: at any horizon h, for any $s \in \mathcal{S}$, and any action $a \in \mathcal{A}$, we can query $r_h = R_h(s_h, a_h)$, and P_h to get a sample $s_{h+1} \sim P_h(\cdot \mid s_h, a_h)$. We denote this reset model $\mathcal{M}_{\mathsf{reset}}$.

For a more streamlined presentation, we will utilize the concept of a partial policy which operates over a sequential segment of time steps, specifically $[l \ldots r] \in [H]$. This is represented as $\Pi_{l:r} := \{\pi : \bigcup_{h=l}^{r} \mathcal{S} \to \Delta(\mathcal{A})\}$.

4. Algorithm

To provide an algorithm for HyRLO, in this section, we first see how this problem is solved in the reset model setting. Then we will derive a reduction from the trace model setting to the reset model setting. The resulting algorithm thus will be a two-phase algorithm: in the first phase, we run a careful reduction to the reset model setting, and in the second phase, we run the reset model algorithm to find the optimal policy.

4.1. Backward Algorithm: PSDP

Suppose we have a reset model $\mathcal{M}_{\mathsf{reset}}$, then as hinted above, we can simply apply an existing algorithm: Policy Search by Dynamic Programming (PSDP) (Bagnell et al., 2003). The PSDP algorithm proceeds in a backward fashion: from the last horizon H to the first horizon 1, at each horizon h, the algorithm first samples states from the offline dataset: $s_h \sim \mathcal{D}_h$, followed by sampling random action $a_h \sim \pi^{\mathsf{unif}}$, resets $\mathcal{M}_{\mathsf{reset}}$ to s_h, a_h, and samples $s_{h+1} \sim P_h(s_h, a_h)$. From s_{h+1} the algorithm will roll out $s_{h+1}, a_{h+1}, \ldots, s_H, a_H \sim \pi_{h+1:H}$ (which are already learned in the previous horizon). Now we have samples of the return information $\sum_{\tau=h}^{H} R(s_\tau, a_\tau)$ for each s_h, a_h we can use cost-sensitive classification to find the one-step optimal policy π_h which maximizes the returns following $\pi_{h+1:H}$, the previous one-step optimal policies. Here we provide a value-based version of the PSDP algorithm in Algorithm 2.

Now with PSDP, as long as the offline distribution μ that generates the offline dataset $\mathcal{D}$ covers some good policy's trajectory (for example, the density ratio coverage $C_{\mathsf{cov}}(\pi^*)$ is bounded), we will show in Section 5 that the returned policy is close to optimal with enough online data.

4.2. Trace to Reset

However, in the HyRLO framework, we do not have the reset model $\mathcal{M}_{\mathsf{reset}}$ but the more realistic trace model $\mathcal{M}$. What can we do in this case? It turns out that with the help of the offline dataset, we can learn a policy π^{f} that induces a state distribution similar to μ. Then suppose that for each timestep h, we have that $\|d_h^{\pi^{\mathsf{f}}} - \mu_h\|$ is small, where $\|\cdot\|$ is some distance metric that we care about. Then to reset to $s_h \sim \mu_h$, we can instead roll in the policy π^{f} to horizon h, and we will get samples $s_h \sim d_h^{\pi^{\mathsf{f}}}$ (as if we are sampling $s_h \sim \mu_h$), and then we can proceed to run PSDP. Thus we can build a reset model with π^{f}. The new algorithm is summarized in Algorithm 3. Note that the only change is in lines 3 and 4. We remark that a similar idea of using PSDP with a roll in policy has also been explored in previous work (Mhammedi et al., 2023), where the roll in policy is trained from reward-free exploration techniques. However, the goal of reward-free exploration is to ensure optimality instead of efficiency, since reward-free exploration has a similar lower bound as regular reward-based online RL (Jin et al., 2020a).

4.3. Forward Algorithm: FAIL

The technical problem remaining is to learn a policy π^{f} that induces a state distribution close to μ. Inspired by the idea of state-moment-matching in ILFO literature, we can adapt one such algorithm, Forward Adversarial Imitation Learning (FAIL) (Sun et al., 2019). FAIL learns a sequence of policies $\pi_{1:H}$ from $h = 1$ to H in sequence. At each timestep h, FAIL rolls in the previous policies $\pi_{1:h-1}$ and samples $s_h \sim d_h^\pi$ .It then takes a random action $a_h \sim \pi^{\mathsf{unif}}$ and samples $s_{h+1} \sim P_h(s_h, a_h)$. With the dataset $\{s_h, a_h, s_{h+1}\}$, FAIL solves the following minmax game by finding a one-step policy π_h that minimizes the IPM under discriminator class $\mathcal{G}$, between $\pi_{1:h}$ and μ_h, which we approximate the samples $\mathcal{D}_h$:

$$\min_{\pi_h\in\Pi_h} \max_{g\in\mathcal{G}_h} \left[\mathbb{E}_{s_h\sim d_h^\pi} g(s_h) - \mathbb{E}_{s_h\sim\mu_h} g(s_h)\right],$$

[2] We use the general density ratio coverage for simplicity of presentation for the main text, a tighter coverage similar to (Song et al., 2022) applies naturally but we defer it to Appendix E.

These can be estimated by the dataset collected online at time step h:

$$\min_{\pi_h \in \Pi} \max_{g \in \mathcal{G}_h} \left(\sum_{n=1}^{N} \frac{\pi_h(a_h^n \mid s_h^n)}{N/A} g(s_{h+1}^n) - \sum_{n=1}^{N'} \frac{g(s_{h+1}^n)}{N'} \right).$$

To solve the minmax game, we can use the common pattern of best response playing no-regret algorithm. We show one way to solve this in Algorithm 5. At the high level, the FAIL algorithm iteratively finds the solution of the minmax problem in a forward way, ensuring the policy induces similar state visitation distribution as the offline data on each horizon. We present the pseudocode of FAIL in Algorithm 4.

4.4. Forward-backward Algorithm: FOOBAR

We are now ready to present the proposed algorithm, FOward Observation-matching BAckward Reinforcement learning (FOOBAR). We present the pseudocode in Algorithm 1. In the forward phase, we run FAIL that outputs a sequence of policies $\pi^{\mathsf{f}}_{1:H}$ whose state visitation distribution is close to the offline distribution μ. Care is needed here because we have not defined the discriminator class we will use for the forward phase. It turns out that if we arbitrarily select the discriminator to be all bounded function, then a polynomial dependency on the number of states is required to imitate the offline distribution (see Theorem 3.2 of Sun et al. (2019)), which is unfavorable given the fact that we already have an offline state dataset to imitate so such dependency should be avoidable in the case with a relatively high-quality offline dataset (e.g., not a uniform distribution over states). Indeed the dependency on the state is avoidable by a careful construction of the discriminator class based on the value function class that we use in the backward pass (Eq. (1)). We provide the justification of such construction in Appendix C.2. Then in the backward phase, we run PSDP-trace Algorithm 3 with the roll-in policy $\pi^{\mathsf{f}}_{1:H}$. Algorithm 3 returns refined policies $\pi^{\mathsf{b}}_{1:H}$, which can compete with the best policy covered by μ. It's important to note that if the offline dataset consists of both sub-optimal and high-quality data, the refined policies $\pi^{\mathsf{b}}_{1:H}$ can be *dramatically* better than the initial policy $\pi^{\mathsf{f}}_{1:H}$ learned by moment matching that is used to simulate the reset model.

5. Analysis

In this section we provide the analysis of the proposed algorithm. The overall proof strategy follows the intuition of the algorithm itself: 1) we need a certain closeness guarantee (Theorem 5.1) between the forward policy and the offline distribution, and the major difficulty is to ensure that such requirement is not too strong (which will result in a suboptimal sample complexity, c.r. Section 4.4), but is sufficient to 2) show the guarantee of the downstream learning of the backward policy (Theorem 5.2). We will start with an essential assumption on a property of the offline distribution.

Algorithm 1 FOward Observation-matching BAckward Reinforcement learning (FOOBAR)

require Offline dataset $\mathcal{D}^{\mathsf{off}}$, value function class $\mathcal{F}$, policy class Π.

1: Define the discriminator class $\mathcal{G} := \{\mathcal{G}_h\}_{h=1}^H$: `// Discriminators take state as input while Q-functions take state-action as input.`

$$\mathcal{G}_h = \left\{ \max_a f(\cdot, a) - f(\cdot, a') \mid f \in \mathcal{F}_h, a' \in \mathcal{A} \right\}. \tag{1}$$

2: $\pi^{\mathsf{f}}_{1:H} \leftarrow$ Algorithm 4 with input $\left\{\mathcal{D}^{\mathsf{off}}, \mathcal{G}, \Pi\right\}$.

3: $\pi^{\mathsf{b}}_{1:H} \leftarrow$ Algorithm 3 with input $\left\{\pi^{\mathsf{f}}_{1:H}, \mathcal{F}\right\}$.

5.1. Admissibility

We follow the definition of admissibility from Chen & Jiang (2019):

Assumption 5.1 (Admissibility)**.** *We assume the offline distribution μ is admissible:*

$$\exists \pi \in \Pi,\ \forall h \in [H],\ \forall s, a \in \mathcal{S} \times \mathcal{A},\ \mu_h(s,a) = d_h^\pi(s,a).$$

Assumption 5.1 captures the situations where the offline data is generated by a single (possibly stochastic) policy, stitching policies (due to non-stationarity), or a mixture of such policies (Chapter 13 in (Sutton & Barto, 2018)), which is how most offline datasets are generated in practice. Practically, this assumption might be violated by artificial data filtering, data augmentation or other perturbation of the offline data.

Next, we provide examples in which if Assumption 5.1 fails, the problem is hard in trace model but remains easy with reset model access (Kakade & Langford, 2002; Bagnell et al., 2003):

Proposition 5.1. *For any algorithm* Alg*, denote the dataset collected by* Alg *as D^{Alg}, and let $\widehat{D}$ denote the empirical distribution of a dataset D. Then there exists an MDP $\mathcal{M}$ with deterministic transition and a set of offline datasets $\{D\}$, with arbitrary sample size $|D| = N \geq 2$, collected from the inadmissible offline distribution μ with constant coverage: $\max_h \left\| \frac{d_h^{\pi^*}}{\widehat{D}_h} \right\|_\infty = 2$ such that, unless $\left|D^{\mathsf{Alg}}\right| = \Omega(A^H)$, we have*

$$\max_D \left\| \widehat{D}_H^{\mathsf{Alg}} - \widehat{D}_H \right\|_{\mathsf{TV}} \geq \frac{1}{2}.$$

However, there exists an algorithm $\mathsf{Alg}^{\mathsf{reset}}$ that uses any offline dataset D and reset model $\mathcal{M}^{\mathsf{reset}}$ that returns optimal policy π^* with sample complexity $O(A)$.

The above statement is about the hardness of collecting a dataset that matches the offline dataset: when the admissibility assumption is violated, in the worst case we need to collect a dataset that is exponentially large in the horizon; otherwise, our dataset does not contain at least half of the states in the offline dataset (at least half of which is expert states), with probability 1 (recall that the construction is within a deterministic MDP). In the next proposition, we show another hardness result that has direct implications on the performance of the learned policy.

Proposition 5.2. *For any state distribution μ_h, $\forall h \in [H]$, let*

$$\pi_h^\mu := \operatorname*{argmin}_{\pi_h \in \Pi_h} \|d_h^\pi - \mu_h\|_{\mathsf{TV}},$$

i.e., the policy that induces the closest state distribution to the offline distribution in TV. Then there exists an MDP $\mathcal{M}$ and inadmissible offline distribution μ, such that $\max_h \left\| \frac{d_h^{\pi^*}}{\mu_h} \right\|_\infty = 18$, i.e., the offline distribution has a constant coverage over the optimal policy but

$$\max_h \left\| \frac{d_h^{\pi^*}}{d_h^{\tilde{\pi}}} \right\|_\infty = \infty, \quad \textit{and} \quad \max_{h, s_h} \left\| \frac{\pi_h^*(s_h)}{\tilde{\pi}_h(s_h)} \right\|_\infty = \infty.$$

i.e., the policy that minimizes the TV distance to the offline distribution does not cover some states from the optimal policy's trajectory, and the induced policy does not cover some actions that the optimal policy takes.

A direct implication of Proposition 5.2 is that, if reward 1 is assigned to the states that are not covered by the learned policy (or state only reachable from those states), and reward is 0 otherwise, then the policy that best mimics the offline distribution will have a constant gap to optimal policy, i.e., $J(\pi^*) - J(\pi^\mu) = 1$. Also, similar to Proposition 5.1, the setup in Proposition 5.2 is not hard in the reset model settings. These results suggest the potential for a separation between trace and result model, but they are not equivalent to an information-theoretical lower bound.

5.2. Performance Guarantee of the Forward Algorithm

Now we analyze Algorithm 4. We start with an assumption that is a relaxation of Assumption 5.1, which is sufficient for our analysis. Note that in the construction of the previous hardness results, this relaxed assumption is still violated.

Assumption 5.2 (Admissibility in IPM.)**.** *There exists a policy π such that, for all $h \in [H]$, $\mathsf{IPM}_{\mathcal{G}_h}(d_h^\pi, \mu_h) = 0$, where $\mathcal{G}_h$ is defined as in the Eq. (1).*

Note that this assumption is weaker because $\mathcal{G}$ is a subset of bounded functions, and Assumption 5.1 implies 0 TV distance, which implies Assumption 5.2. Next, we introduce the Bellman Completeness assumption, which is also commonly made in ILFO (Sun et al., 2019) and hybrid RL (Song et al., 2022; Nakamoto et al., 2023):

Assumption 5.3 (Completeness)**.** *For any $h \in [H]$, for any $g \in \mathcal{G}_{h+1}$, there exists $f \in \mathcal{G}_h$ such that $f = \mathcal{T}_h g$, where $\mathcal{T}_h$ is the Bellman operator with respect to the offline distribution at time h: $\mathcal{T}_h g(s_h) = \mathbb{E}_{a_h \sim \mu_h(s_h)} \mathbb{E}_{s_{h+1} \sim P_h(s_h, a_h)}[g(s_{h+1})]$. That is,*

$$\max_h \max_{g \in \mathcal{G}_{h+1}} \min_{f \in \mathcal{G}_h} \|f - \mathcal{T}_h g\|_\infty = 0.$$

Note that the previous two assumptions can both hold approximately, and here we assume that they hold exactly for simplicity. Now we can state the performance guarantee of the forward algorithm. The result is characterized in the IPM between the learned policy and the offline distribution.

Theorem 5.1 (Guarantee of Algorithm 4)**.** *Assume Assumptions 5.2 and 5.3 hold. Suppose $|\mathcal{D}^{\mathsf{off}}| = |\mathcal{D}^{\mathsf{on}}| = N$, then with probability $1 - \delta$, the returned policy π^{f} satisfies that, for any $h \in [H]$,*

$$\mathsf{IPM}_{\mathcal{G}_h}(d_h^{\pi^{\mathsf{f}}}, \mu_h) \le h \varepsilon^{\mathsf{for}}(\delta, N),$$

where $\varepsilon^{\mathsf{for}}(\delta, N) :=$

$$8\sqrt{\frac{2A \log(2|\mathcal{G}_h||\Pi_h|/\delta)}{N}} + \frac{16A \log(2|\mathcal{G}_h||\Pi_h|/\delta)}{N} + \sqrt{\frac{A^2}{T}},$$

where T is the number of iterations in Algorithm 5.

This result indicates that if we have equally enough samples from both online and offline (which is one of the key features of hybrid RL), and we perform enough iterations of the minmax game, then we will have the guarantee that the learned forward policy will be close to the offline distribution under any discriminator in $\mathcal{G}$. Note that this result does not imply that the learned policy is close to the offline distribution in a stronger sense such as TV distance, and we emphasize that such a stronger notion of closeness is not necessary for learning a policy that can compare with the best policy covered by the offline distribution.

5.3. Performance Guarantee of FOOBAR

With the guarantee of the forward algorithm, we can show the performance guarantee of FOOBAR. Different from the analysis of the forward algorithm, whose result is to compare with the offline distribution, the final result of FOOBAR is to compare with the performance of other policies. Therefore, following the convention common in hybrid RL literature (Bagnell et al., 2003; Ross & Bagnell, 2012; Xie et al., 2021b; Song et al., 2022), we state the performance guarantee of FOOBAR with respect to any policy that is covered by the offline distribution, i.e., we can compare with any policy π^{comp} with $C_{\mathsf{cov}}(\pi^{\mathsf{comp}}) < \infty$.

In addition to the above offline coverage condition, since our algorithm involves function approximation (i.e., we use $\mathcal{F}$ to estimate the value functions), we also require the following standard realizability assumption:

Assumption 5.4 (Realizability)**.** *For any deterministic policy π, $h \in [H]$, we have $Q_h^\pi \in \mathcal{F}_h$.*

Note that here we state the most general form of realizability assumption for simplicity. In the proof (Appendix C.2), we use a relaxed version where the realizability holds for a subset of policies and states. We also assume the function class $|\mathcal{F}|$ is finite [3]. Now we are ready to present our main result:

Theorem 5.2. *Suppose Assumption 5.2, Assumption 5.3 and Assumption 5.4 hold. Then with probability $1-\delta$, the returned policy $\pi^{\mathsf{b}}_{1:H}$ from Algorithm 1 with discriminator constructed from Eq. (1), N^{for} offline and forward samples, and N^{back} backward samples, satisfies that for any comparator policy π^{comp} such that $C_{\mathsf{cov}}(\pi^{\mathsf{comp}}) < \infty$,*

$$V^{\pi^{\mathsf{comp}}} - V^{\pi^{\mathsf{b}}} \leq \varepsilon,$$

when

$$H \cdot N^{\mathsf{for}} = O\left(\frac{C^2_{\mathsf{cov}}(\pi^{\mathsf{comp}}) H^5 A \log(|\mathcal{F}_h||\Pi_h|/\delta)}{\varepsilon^2}\right) \quad \textit{and}$$

$$H \cdot N^{\mathsf{back}} = O\left(\frac{C^2_{\mathsf{cov}}(\pi^{\mathsf{comp}}) H^5 A \log(|\mathcal{F}_h|/\delta)}{\varepsilon^2}\right).$$

A few remarks are in order:

Remark 5.1 (Reduction from trace to reset)**.** We can see that the samples required for the forward algorithm and backward algorithm are only different by a factor of $\log(|\Pi|)$. If we consider a policy class with the same expressiveness as the value function class (which is generally true in practice), i.e., $\log(|\mathcal{F}||\Pi|) \approx 2\log(|\mathcal{F}|)$, then our algorithm performs *a reduction from the trace model setting to the reset model setting* with constant overhead.

Remark 5.2 (Removing explicit structural assumptions)**.** Note that our result is not specific to tabular MDPs. In fact, compared with previous hybrid RL (or online RL) analysis (Song et al., 2022; Wagenmaker & Pacchiano, 2023; Nakamoto et al., 2023), our analysis is agnostic to the structural complexity measure d (Jin et al., 2020b; Du et al., 2021) of the MDPs and thus applies to any MDP with finite action space. For example, in the tabular setting where $d = SA$, our result has no explicit dependency on the number of states S, and recall in Proposition 5.1 we showed that a polynomial dependency on the state space size ($S = A^H$) is difficult to avoid without the admissibility assumption. Our result has a worse dependency on A but we conjecture this is fundamental in the observation-only setting. We provide a thorough discussion on this topic in Appendix E.

[3] This is without loss of generality and we can also use $|\mathcal{F}|$ to denote similar measures such as covering number or VC-dimension of the function class.

Remark 5.3 (Significance of the discriminator class)**.** One might think that the positive result from Theorem 5.2 is a natural byproduct of the positive results from FAIL and PSDP. However, we note that FAIL only guarantees to return a policy that is comparable to the behavior policy (offline distribution), but the learned policy can induce different visitation distribution from the behavior policy. Thus the guarantee to compare with any covered policy is not trivial, and this is addressed by the careful construction of the discriminator class $\mathcal{G}$.

Further practical considerations. Finally we state two additional results that will have direct implications on the practicality of the algorithm. First, if the guarantee of Theorem 5.1 breaks (i.e., $\max_h \mathsf{IPM}_{\mathcal{G}_h}(d_h^{\pi^{\mathsf{f}}}, \mu_h) = c$, where c is not small), which may be caused by inadmissible offline data, violation of completeness assumption, or optimization error, we show that in Appendix D.1 that FOOBAR can still compare with *the best policy covered by the forward policy*. In Section 6.2, we verify empirically robustness of FOOBAR against different levels of inadmissibility.

Second, it might be computationally and memory intensive to perform non-stationary algorithms such as FOOBAR, or limiting if the horizon of the problem is not known or fixed. As such, in Appendix D.2, we provide algorithms and analysis in stationary setting but with interactive offline distribution. In the robotics simulation in Section 6, we witness the practical value brought by both results: we obtain the optimal policy with a stationary backward policy, while the forward policy does not perfectly mimic the offline states.

6. Experiments

In the experiments, we analyze the following questions: (1) Does FOOBAR still demonstrate the benefit of hybrid RL framework? For example, does it still efficiently solve exploration-heavy problems without explicit exploration? (2) How does FOOBAR compare to the canonical hybrid RL algorithms, i.e., what is the price for the missing information in the offline dataset? (3) How does the performance compare with PSDP if a reset model is available, and how robust is FOOBAR against inadmissibility in practice?

We use the following two benchmarks: the combination lock (Misra et al., 2020) and the hammer task of the Adroit robotics from the D4RL benchmark (Fu et al., 2020). The visualization can be found in Figure 4. Both environments are challenging: the combination lock requires careful exploration and previous online RL algorithms require additional representation learning in addition to RL (Misra et al., 2020; Zhang et al., 2022; Mhammedi et al., 2023) due to its high-dimensional observation space, which also poses challenges for our forward state-moment-matching algorithm. The hammer task has high-dimension state and action space and

difficult success conditions. Similar to Ball et al. (2023), we use the binary reward version of the environment.

6.1. Comparing to Hybrid RL

Combination locks. In this section, we investigate the first two questions. We first provide a brief description of the combination lock environment, and more details can be found in Appendix F.1. For our experiment, we set the horizon $H = 100$. In each horizon h there are three latent states: two good states and one bad state. Taking only one correct action (out of 10 actions in total) makes the agent proceed to the good states in the next horizon, otherwise, it proceeds to the bad state, and bad states only transit to bad states. The agent receives a reward of 1 if it stays at the good states at $h = H$ so the reward signal is sparse, and random exploration requires 10^{100} episodes to receive a reward signal for the first time.

We collect the offline dataset with a ε-greedy version of π^*, where $\varepsilon = \frac{1}{H}$. This guaranteed us $C_{\mathsf{cov}}(\pi^*) \approx 2.5$. We collect 2000 samples per horizon for both FOOBAR and Hybrid Q-Learning (HYQ) (Song et al., 2022), the hybrid RL algorithm that solved this task using the canonical offline dataset. We also compare with pure online RL, and we compare with the state-of-the-art algorithm in combination lock, BRIEE (Zhang et al., 2022). We show the result on the left of Figure 1. We see that compared to the online RL method, FOOBAR is still much more efficient, and compared with HYQ, FOOBAR indeed takes more samples but the overall sample efficiency is very comparable to the canonical hybrid RL algorithms that enjoy more information in the offline dataset.

Our practical implementation follows our description in Algorithm 1, and we use Maximum Mean Discrepancy (MMD) (Gretton et al., 2012) with RBF Kernel for the discriminator class, and we parameterize the policy and value functions with neural networks. We defer most implementation details to Appendix F.

Hammer. We also test FOOBAR on a more popular D4RL (Fu et al., 2020) robotics benchmark hammer (Rajeswaran et al., 2017). Following the evaluation protocol from Ball et al. (2023), we use the binary reward version of the environment (reward of 1 if fully complete the task, -1 otherwise): thus the environment delivers sparse reward signal and the evaluation is based on how fast the agent finishes the task. To our best knowledge, we are not aware of any pure online RL method that reported solving the binary version of hammer thus we will focus our comparison to hybrid RL. We also observe that an optimal policy for this task can finish it within 50 timesteps, so we truncate the environment to $H = 50$ for computational consideration.

For the implementation of FOOBAR, we use the same implementation as the combination lock, but instead of taking random action (which is hard over the 26-dimensional continuous action space), we interleave the policy update and data collection with the latest policy. For the backward pass, we follow our stationary algorithm described in Appendix D.2 and use SAC (Haarnoja et al., 2018) as the policy optimization subprotocol. We present the result in Figure 1 (right). In the plot we only plotted the evaluation curve along the backward run for a cleaner comparison, and we use 100k samples for each horizon in the forward run (in total the forward run requires 10 times more samples than the backward run). Our backward run is comparable to RLPD (Ball et al., 2023), but we hypothesize two reasons why our method is slightly slower: first, our forward policy does not recover the offline distribution perfectly[4], and second, we choose to avoid some practical design choices that deviate from our theoretical algorithm but are potentially beneficial to the practical sample efficiency. Although this is a prototypical comparison, we believe the result suggests that HYRLO still demonstrates the superiority of the hybrid RL setting, but the result also suggests the gap from lacking action, reward and dynamics information in the offline dataset. Regarding the less efficient forward phase, our result in combination lock suggests that in a more controlled setting, the sample efficiency of the forward and backward run are similar (Figure 5), and we believe this encourages the community to design a better algorithm for state-moment-matching to close the gap further.

	FOOBAR (Forward)	FOOBAR	PSDP
Benign	0.12 $(0.1, 0.135)$	1 $(1, 1)$	1 $(1, 1)$
Adversarial	0 $(0, 0)$	0 $(0, 0)$	1.1 $0.95, 1.15)$

Table 2. Comparison between FOOBAR and PSDP under inadmissible setting. We show the median of the relative success rate (over optimal policy), and 25% and 75% percentile in the parentheses, over 10 random seeds. Note that the relative success rate in the adversarial case can exceed 1 because the environment is stochastic and the theoretical optimal policy has a success rate of 10%.

6.2. Inadmissible Offline Distribution

To answer the last question, we construct inadmissible offline datasets in the combination lock environment with $H = 10$. Specifically, we test on two inadmissible datasets: a benign dataset where the proportion of good states increases along the horizon (which is impossible for any policy to collect in a trace model setting), and an adversarial inad-

[4] As we suggest in Theorem D.1 in Appendix D.1, FOOBAR is robust to an imperfect forward run, and we provide more discussion on the empirical results in Appendix F.2.

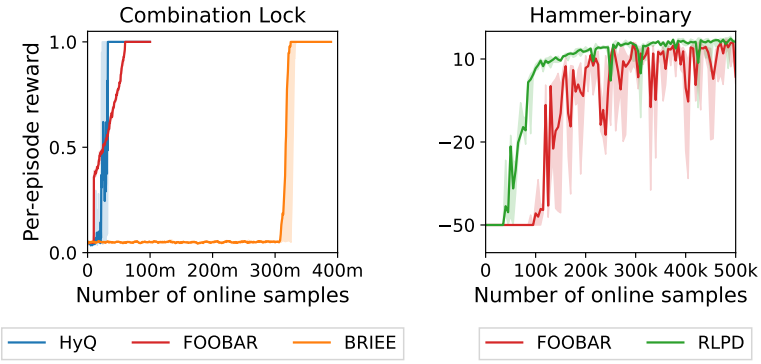

Figure 1. Comparison with hybrid RL and online RL. **Left**: evaluation curve along the training process in the combination lock task. The plot for FOOBAR combines the forward and backward passes: during the forward pass, the evaluation result is from all the forward policies (trained and untrained). During the backward pass, after training at horizon h, the evaluation is from the policy $\pi^{\mathsf{f}} \circ_h \pi^{\mathsf{b}}$. **Right**: evaluation curve along the training process in the hammer-binary task. The plot for FOOBAR shows the performance of the stationary backward policy in the backward phase. We repeat the experiment for 10 random seeds and plot the median and 25% to 75% percentiles.

missible dataset where we inject the hardness construction of Proposition 5.2 into the first horizon of the environment. We compare FOOBAR and PSDP and we present the results in Table 2: we can see in practice, FOOBAR is still robust under a certain level of inadmissibility (it still solves the combination lock with the benign inadmissible dataset), but can not solve the provably hard example compared to the reset model algorithms. In this case of benign inadmissibility, the forward policy still covers the distribution of the optimal policy (the minimum coverage of good states of the offline data over the horizon is 15%, and the forward policy has a median success rate of 12%), leading to the final success of the whole algorithm. This again corresponds to the result of Theorem D.1 that characterizes the success condition of FOOBAR under inadmissibility.

7. Discussion

Our work initiates the theoretical study of HYRLO, a new theoretical paradigm with promising practical potential. Here we discuss some theoretical and practical open problems for future research:

- Although we provide two hardness examples in the trace model setting when the admissibility assumption fails, it will be interesting to understand if fundamental separations exist between the trace and reset model.
- Previous hybrid RL method (Song et al., 2022) works under inadmissible offline distribution but requires structural assumption. Is there any tradeoff or connection between these two assumptions?
- Our analysis gives partial answers towards a stationary solution to the HYRLOproblem, and it will be interesting to design a fully stationary algorithm for HYRLO.
- Our theory suggests that a better practical implementation state-moment-matching algorithm is possible and we believe this is an important practical problem to solve.

Acknowledgments

YS thanks Wen Sun for detailed feedback on the draft. The authors thank and acknowledge the support of ONR grant N000142212363 and NSF AI Institute for Societal Decision Making AI-SDM grant IIS2229881.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

References

Agarwal, A., Jiang, N., Kakade, S. M., and Sun, W. Reinforcement learning: Theory and algorithms. *CS Dept., UW Seattle, Seattle, WA, USA, Tech. Rep*, 32, 2019.

Agarwal, A., Kakade, S., Krishnamurthy, A., and Sun, W. FLAMBE: Structural complexity and representation learning of low rank MDPs. *Neural Information Processing Systems (NeurIPS)*, 2020a.

Agarwal, A., Kakade, S. M., Lee, J. D., and Mahajan, G. Optimality and approximation with policy gradient methods

in markov decision processes. In *Conference on Learning Theory*, pp. 64–66. PMLR, 2020b.

Amortila, P., Jiang, N., Madeka, D., and Foster, D. P. 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.

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

Bagnell, J., Kakade, S. M., Schneider, J., and Ng, A. Policy search by dynamic programming. *Advances in neural information processing systems*, 16, 2003.

Bagnell, J. A. *Learning decisions: Robustness, uncertainty, and approximation*. Carnegie Mellon University, 2004.

Baker, B., Akkaya, I., Zhokov, P., Huizinga, J., Tang, J., Ecoffet, A., Houghton, B., Sampedro, R., and Clune, J. Video pretraining (vpt): Learning to act by watching unlabeled online videos. *Advances in Neural Information Processing Systems*, 35:24639–24654, 2022.

Ball, P. J., Smith, L., Kostrikov, I., and Levine, S. Efficient online reinforcement learning with offline data. *arXiv preprint arXiv:2302.02948*, 2023.

Block, A., Jadbabaie, A., Pfrommer, D., Simchowitz, M., and Tedrake, R. Provable guarantees for generative behavior cloning: Bridging low-level stability and high-level behavior. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

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

Dasari, S., Ebert, F., Tian, S., Nair, S., Bucher, B., Schmeckpeper, K., Singh, S., Levine, S., and Finn, C. Robonet: Large-scale multi-robot learning. *arXiv preprint arXiv:1910.11215*, 2019.

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

Du, S. S., Kakade, S. M., Lee, J. D., Lovett, S., Mahajan, G., Sun, W., and Wang, R. Bilinear classes: A structural framework for provable generalization in RL. *International Conference on Machine Learning*, 2021.

Fu, J., Kumar, A., Nachum, O., Tucker, G., and Levine, S. D4rl: Datasets for deep data-driven reinforcement learning. *arXiv preprint arXiv:2004.07219*, 2020.

Fukumizu, K., Gretton, A., Lanckriet, G., Schölkopf, B., and Sriperumbudur, B. K. Kernel choice and classifiability for rkhs embeddings of probability distributions. *Advances in neural information processing systems*, 22, 2009.

Ghosh, D., Bhateja, C. A., and Levine, S. Reinforcement learning from passive data via latent intentions. In *International Conference on Machine Learning*, pp. 11321–11339. PMLR, 2023.

Grauman, K., Westbury, A., Byrne, E., Chavis, Z., Furnari, A., Girdhar, R., Hamburger, J., Jiang, H., Liu, M., Liu, X., et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18995–19012, 2022.

Grauman, K., Westbury, A., Torresani, L., Kitani, K., Malik, J., Afouras, T., Ashutosh, K., Baiyya, V., Bansal, S., Boote, B., et al. Ego-exo4d: Understanding skilled human activity from first-and third-person perspectives. *arXiv preprint arXiv:2311.18259*, 2023.

Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B., and Smola, A. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012.

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pp. 1861–1870. PMLR, 2018.

Hu, Z., Rovinsky, A., Luo, J., Kumar, V., Gupta, A., and Levine, S. Reboot: Reuse data for bootstrapping efficient real-world dexterous manipulation. *arXiv preprint arXiv:2309.03322*, 2023.

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

Jin, C., Yang, Z., Wang, Z., and Jordan, M. I. Provably efficient reinforcement learning with linear function approximation. In *Conference on Learning Theory*, pp. 2137–2143, 2020b.

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

Kakade, S. and Langford, J. Approximately optimal approximate reinforcement learning. In *Proceedings of the Nineteenth International Conference on Machine Learning*, pp. 267–274, 2002.

Kane, D., Liu, S., Lovett, S., and Mahajan, G. Computational-statistical gap in reinforcement learning. In *Conference on Learning Theory*, pp. 1282–1302. PMLR, 2022.

Kim, G.-H., Seo, S., Lee, J., Jeon, W., Hwang, H., Yang, H., and Kim, K.-E. Demodice: Offline imitation learning with supplementary imperfect demonstrations. In *International Conference on Learning Representations*, 2021.

Lee, S., Seo, Y., Lee, K., Abbeel, P., and Shin, J. Offline-to-online reinforcement learning via balanced replay and pessimistic q-ensemble. In *Conference on Robot Learning*, pp. 1702–1712. PMLR, 2022.

Levine, S., Kumar, A., Tucker, G., and Fu, J. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

Li, A., Boots, B., and Cheng, C.-A. Mahalo: Unifying offline reinforcement learning and imitation learning from observations. *arXiv preprint arXiv:2303.17156*, 2023a.

Li, G., Zhan, W., Lee, J. D., Chi, Y., and Chen, Y. Reward-agnostic fine-tuning: Provable statistical benefits of hybrid reinforcement learning. *arXiv preprint arXiv:2305.10282*, 2023b.

Ma, Y., Shen, A., Jayaraman, D., and Bastani, O. Versatile offline imitation from observations and examples via regularized state-occupancy matching. In *International Conference on Machine Learning*, pp. 14639–14663. PMLR, 2022a.

Ma, Y. J., Sodhani, S., Jayaraman, D., Bastani, O., Kumar, V., and Zhang, A. Vip: Towards universal visual reward and representation via value-implicit pre-training. *arXiv preprint arXiv:2210.00030*, 2022b.

Machado, M. C., Rosenbaum, C., Guo, X., Liu, M., Tesauro, G., and Campbell, M. Eigenoption discovery through the deep successor representation. *arXiv preprint arXiv:1710.11089*, 2017.

Mathieu, M., Ozair, S., Srinivasan, S., Gulcehre, C., Zhang, S., Jiang, R., Paine, T. L., Powell, R., Żołna, K., Schrittwieser, J., et al. Alphastar unplugged: Large-scale offline reinforcement learning. *arXiv preprint arXiv:2308.03526*, 2023.

Mhammedi, Z., Foster, D. J., and Rakhlin, A. Representation learning with multi-step inverse kinematics: An efficient and optimal approach to rich-observation rl. *arXiv preprint arXiv:2304.05889*, 2023.

Misra, D., Henaff, M., Krishnamurthy, A., and Langford, J. Kinematic state abstraction and provably efficient rich-observation reinforcement learning. In *International conference on machine learning*, pp. 6961–6971. PMLR, 2020.

Müller, A. Integral probability metrics and their generating classes of functions. *Advances in applied probability*, 29 (2):429–443, 1997.

Nair, A., Chen, D., Agrawal, P., Isola, P., Abbeel, P., Malik, J., and Levine, S. Combining self-supervised learning and imitation for vision-based rope manipulation. In *2017 IEEE international conference on robotics and automation (ICRA)*, pp. 2146–2153. IEEE, 2017.

Nair, A., Gupta, A., Dalal, M., and Levine, S. Awac: Accelerating online reinforcement learning with offline datasets. *arXiv preprint arXiv:2006.09359*, 2020.

Nair, A. V., Pong, V., Dalal, M., Bahl, S., Lin, S., and Levine, S. Visual reinforcement learning with imagined goals. *Advances in neural information processing systems*, 31, 2018.

Nakamoto, M., Zhai, Y., Singh, A., Mark, M. S., Ma, Y., Finn, C., Kumar, A., and Levine, S. Cal-ql: Calibrated offline rl pre-training for efficient online fine-tuning. *arXiv preprint arXiv:2303.05479*, 2023.

Niu, H., Qiu, Y., Li, M., Zhou, G., HU, J., Zhan, X., et al. When to trust your simulator: Dynamics-aware hybrid offline-and-online reinforcement learning. *Advances in Neural Information Processing Systems*, 35: 36599–36612, 2022.

Padalkar, A., Pooley, A., Jain, A., Bewley, A., Herzog, A., Irpan, A., Khazatsky, A., Rai, A., Singh, A., Brohan, A., et al. Open x-embodiment: Robotic learning datasets and rt-x models. *arXiv preprint arXiv:2310.08864*, 2023.

Pirotta, M., Tirinzoni, A., Touati, A., Lazaric, A., and Ollivier, Y. Fast imitation via behavior foundation models. In *NeurIPS 2023 Foundation Models for Decision Making Workshop*, 2023.

Qin, R.-J., Zhang, X., Gao, S., Chen, X.-H., Li, Z., Zhang, W., and Yu, Y. Neorl: A near real-world benchmark for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 35:24753–24765, 2022.

Radosavovic, I., Wang, X., Pinto, L., and Malik, J. State-only imitation learning for dexterous manipulation. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 7865–7871. IEEE, 2021.

Rajeswaran, A., Kumar, V., Gupta, A., Vezzani, G., Schulman, J., Todorov, E., and Levine, S. Learning complex

dexterous manipulation with deep reinforcement learning and demonstrations. *arXiv preprint arXiv:1709.10087*, 2017.

Ren, J., Swamy, G., Wu, S. Z., Bagnell, J. A., and Sanjiban, C. Hybrid inverse reinforcement learning. *NeurIPS 2023 Workshop on Robot Learning: Pretraining, Fine-Tuning, and Generalization with Large Scale Models*, 2023.

Ross, S. and Bagnell, J. A. Agnostic system identification for model-based reinforcement learning. *arXiv preprint arXiv:1203.1007*, 2012.

Ross, S., Gordon, G., and Bagnell, D. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 627–635. JMLR Workshop and Conference Proceedings, 2011.

Salimans, T. and Chen, R. Learning montezuma's revenge from a single demonstration. *arXiv preprint arXiv:1812.03381*, 2018.

Scherrer, B. Approximate policy iteration schemes: A comparison. In *International Conference on Machine Learning*, pp. 1314–1322. PMLR, 2014.

Schmeckpeper, K., Rybkin, O., Daniilidis, K., Levine, S., and Finn, C. Reinforcement learning with videos: Combining offline observations with interaction. *arXiv preprint arXiv:2011.06507*, 2020.

Seo, Y., Lee, K., James, S. L., and Abbeel, P. Reinforcement learning with action-free pre-training from videos. In *International Conference on Machine Learning*, pp. 19561–19579. PMLR, 2022.

Sharma, A., Ahmad, R., and Finn, C. A state-distribution matching approach to non-episodic reinforcement learning. *arXiv preprint arXiv:2205.05212*, 2022.

Smith, L., Dhawan, N., Zhang, M., Abbeel, P., and Levine, S. Avid: Learning multi-stage tasks via pixel-level translation of human videos. *arXiv preprint arXiv:1912.04443*, 2019.

Song, Y., Mavalankar, A., Sun, W., and Gao, S. Provably efficient model-based policy adaptation. In *International Conference on Machine Learning*, pp. 9088–9098. PMLR, 2020.

Song, Y., Zhou, Y., Sekhari, A., Bagnell, J. A., Krishnamurthy, A., and Sun, W. Hybrid rl: Using both offline and online data can make rl efficient. *arXiv preprint arXiv:2210.06718*, 2022.

Sun, W., Vemula, A., Boots, B., and Bagnell, D. Provably efficient imitation learning from observation alone. In *International conference on machine learning*, pp. 6036–6045. PMLR, 2019.

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

Swamy, G., Wu, D., Choudhury, S., Bagnell, D., and Wu, S. Inverse reinforcement learning without reinforcement learning. In *International Conference on Machine Learning*, pp. 33299–33318. PMLR, 2023.

Torabi, F., Warnell, G., and Stone, P. Generative adversarial imitation from observation. *arXiv preprint arXiv:1807.06158*, 2018.

Uchendu, I., Xiao, T., Lu, Y., Zhu, B., Yan, M., Simon, J., Bennice, M., Fu, C., Ma, C., Jiao, J., et al. Jump-start reinforcement learning. In *International Conference on Machine Learning*, pp. 34556–34583. PMLR, 2023.

Vemula, A., Song, Y., Singh, A., Bagnell, D., and Choudhury, S. The virtues of laziness in model-based rl: A unified objective and algorithms. In *International Conference on Machine Learning*, pp. 34978–35005. PMLR, 2023.

Wagenmaker, A. and Pacchiano, A. Leveraging offline data in online reinforcement learning. In *International Conference on Machine Learning*, pp. 35300–35338. PMLR, 2023.

Wang, R., Wu, Y., Salakhutdinov, R., and Kakade, S. Instabilities of offline rl with pre-trained neural representation. In *International Conference on Machine Learning*, pp. 10948–10960. PMLR, 2021.

Weisz, G., Amortila, P., Janzer, B., Abbasi-Yadkori, Y., Jiang, N., and Szepesvári, C. On query-efficient planning in mdps under linear realizability of the optimal state-value function. In *Conference on Learning Theory*, pp. 4355–4385. PMLR, 2021.

Xie, T., Cheng, C.-A., Jiang, N., Mineiro, P., and Agarwal, A. Bellman-consistent pessimism for offline reinforcement learning. *Advances in neural information processing systems*, 34:6683–6694, 2021a.

Xie, T., Jiang, N., Wang, H., Xiong, C., and Bai, Y. Policy finetuning: Bridging sample-efficient offline and online reinforcement learning. *Advances in neural information processing systems*, 34:27395–27407, 2021b.

Xie, T., Foster, D. J., Bai, Y., Jiang, N., and Kakade, S. M. The role of coverage in online reinforcement learning. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=LQIjzPdDt3q.

Yu, L., Yu, T., Song, J., Neiswanger, W., and Ermon, S. Offline imitation learning with suboptimal demonstrations via relaxed distribution matching. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 11016–11024, 2023.

Zhang, H., Xu, W., and Yu, H. Policy expansion for bridging offline-to-online reinforcement learning. *arXiv preprint arXiv:2302.00935*, 2023a.

Zhang, R. and Zanette, A. Policy finetuning in reinforcement learning via design of experiments using offline data. *arXiv preprint arXiv:2307.04354*, 2023.

Zhang, X., Song, Y., Uehara, M., Wang, M., Agarwal, A., and Sun, W. Efficient reinforcement learning in block mdps: A model-free representation learning approach. In *International Conference on Machine Learning*, pp. 26517–26547. PMLR, 2022.

Zhang, Z., Chen, Y., Lee, J. D., and Du, S. S. Settling the sample complexity of online reinforcement learning. *arXiv preprint arXiv:2307.13586*, 2023b.

Zhou, Y., Sekhari, A., Song, Y., and Sun, W. Offline data enhanced on-policy policy gradient with provable guarantees. *arXiv preprint arXiv:2311.08384*, 2023.

Zhu, Z., Lin, K., Dai, B., and Zhou, J. Off-policy imitation learning from observations. *Advances in Neural Information Processing Systems*, 33:12402–12413, 2020.

A. Omitted Pseudocodes

In the following, we will utilize the concept of a partial policy (defined in the main text already but we will repeat here for completeness) which operates over a sequential segment of time steps, specifically $[l \dots r] \in [H]$. This is represented as $\Pi_{l:r} := \{\pi : \bigcup_{h=l}^{r} \mathcal{S} \to \Delta(\mathcal{A})\}$. Given any two intervals $1 \le t \le h \le H$, we consider two partial policies: $\pi \in \Pi_{1:t-1}$ and $\pi' \in \Pi_{t:h}$. The composition $\pi \circ_t \pi'$ represents a policy that follows π for the initial $t-1$ steps and switches to π' for the subsequent $h-t+1$ steps. Formally, this is defined as $(\pi \circ_t \pi')(s_l) = \pi(s_l)$ when $l < t$ and $(\pi \circ_t \pi')(s_l) = \pi'(s_l)$ for $t \le l \le h$. The notation $s_h \sim \pi$ implies that the state s_h is selected according to the distribution defined by the law of π and P, and we extend this notation to include the action a_h as well, denoted as $s_h, a_h \sim \pi$.

Algorithm 2 Policy Search by Dynamic Programming (PSDP)

require Offline dataset $\mathcal{D}_h$, online sample size N.

1: **for** $h = H, \dots, 1$ **do**
2: **for** $n = 1, \dots, N$ **do**
3: Sample $s_h^n \sim D_h, a_h^n \sim \pi^{\mathsf{unif}}$.
4: Reset $\mathcal{M}_{\mathsf{reset}}$ to s_h^n, a_h^n and sample $s_{h+1}^n \sim P_h(s_h^n, a_h^n)$, $r_h^n = R_h(s_h^n, a_h^n)$.
5: Follow $\pi_{h+1:H}$ and get sample $r_{h+1:H}^n \sim \pi_{h+1:H}$.
6: Train regressor $f_h(s_h, a_h)$ on $r_{h:H}$: `// Estimate Q function.`

$$f_h = \operatorname*{argmin}_{f \in \mathcal{F}} \sum_{n=1}^{N} \left(f(s_h^n, a_h^n) - \sum_{\tau=h}^{H} r_\tau^n \right)^2 .$$

7: Get one-step greedy policy $\pi_h(s_h) = \operatorname{argmax}_{a_h} f_h(s_h, a_h)$.
return Non-stationary backward policy $\pi_{1:H}$.

Algorithm 3 Policy Search by Dynamic Programming (PSDP) with trace model

require Roll in policy π^{f}, online sample size N.

1: **for** $h = H, \dots, 1$ **do**
2: **for** n = $1, \dots, N$ **do**
3: Sample $s_h^n \sim \pi_{1:h}^{\mathsf{f}}, a_h^n \sim \pi^{\mathsf{unif}}$, $r_h^n = R_h(s_h^n, a_h^n)$.
4: Sample $s_{h+1}^n \sim P_h(s_h^n, a_h^n)$.
5: Follow $\pi_{h+1:H}$ and get sample $r_{h+1:H}^n \sim \pi_{h+1:H}$.
6: Train regressor $f_h(s_h, a_h)$ on $r_{h:H}$: `// Estimate Q function`

$$f_h = \operatorname*{argmin}_{f \in \mathcal{F}} \sum_{n=1}^{N} \left(f(s_h^n, a_h^n) - \sum_{\tau=h}^{H} r_\tau^n \right)^2 .$$

7: Get one-step greedy policy $\pi_h(s_h) = \operatorname{argmax}_{a_h} f_h(s_h, a_h)$.
return Non-stationary backward policy $\pi_{1:H}$.

Algorithm 4 Forward Adversarial Imitation Learning (FAIL)

require Offline dataset $\{\mathcal{D}_h^{\mathsf{off}}\}$, discriminator class $\mathcal{G} = \{\mathcal{G}_h\}_{h=1}^H$, policy class $\Pi = \{\Pi_h\}_{h=1}^H$, number of online samples N, number of iterations of minmax game T.

for $h = 1, \dots, H$ **do**
  $\mathcal{D}_h^{\mathsf{on}} \leftarrow \emptyset$.
  for $n = 1, \dots, N$ **do**
    Sample $s_{h-1}^n, a_{h-1}^n, s_h^n \sim \pi \circ_h \pi^{\mathsf{unif}}$.
    Add $(s_{h-1}^n, a_{h-1}^n, s_h^n)$ to $\mathcal{D}_h^{\mathsf{on}}$.
  Get π_h from the return of Algorithm 5 with inputs $\left\{\Pi_h, \mathcal{G}_h, T, \mathcal{D}_h^{\mathsf{on}}, \mathcal{D}_h^{\mathsf{off}}\right\}$.
return $\pi_{1:H}$.

Algorithm 5 Min-Max Game

require Policy class Π, discriminator class $\mathcal{G}$, number of iterations T, online dataset $\mathcal{D}^{\mathsf{on}}$, offline dataset $\mathcal{D}^{\mathsf{off}}$.

Randomly initialize $\pi_0 \in \Pi$.
Define loss function

$$u(\pi, g) := \left(\widehat{\mathbb{E}}_{\mathcal{D}^{\mathsf{on}}} \left[\frac{\pi(a_{h-1} \mid s_{h-1})}{1/A} g(s_h) \right] - \widehat{\mathbb{E}}_{\mathcal{D}^{\mathsf{off}}} \left[g(s_h) \right] \right).$$

for $t = 1, \dots, T$ **do**
  $g^t = \operatorname{argmax}_{g \in \mathcal{G}} u(\pi^t, g)$.// Linear programming oracle.
  $u^t := u(\pi^t, g^t)$.
  $\pi^{t+1} = \operatorname{argmin}_{\pi \in \Pi} \sum_{\tau=1}^t u(\pi, g^t) + \phi(\pi)$. // Regularized cost-sensitive oracle.
return π^{t^*} with $t^* = \operatorname{argmin}_{t \in [T]} u^t$.

B. Proof of Inadmissibility Hardness

In this section we prove the proofs for the two hardness examples we constructed in Proposition 5.1 and Proposition 5.2.

Proposition B.1. *For any algorithm* Alg*, denote the dataset collected by* Alg *as* D^{Alg}*, and let* $\widehat{D}$ *denote the empirical distribution of a dataset* D*. Then there exists an MDP* $\mathcal{M}$ *with deterministic transition and a set of offline datasets* $\{D\}$*, with arbitrary sample size* $|D| = N \geq 2$*, collected from the inadmissible offline distribution* μ *with constant coverage:* $\max_h \left\| \frac{d_h^{\pi^*}}{\widehat{D}_h} \right\|_\infty = 2$ *such that, unless* $\left|D^{\mathsf{Alg}}\right| = \Omega(A^H)$*, we have*

$$\max_D \left\| \widehat{D}_H^{\mathsf{Alg}} - \widehat{D}_H \right\|_{TV} \geq \frac{1}{2}.$$

However, there exists an algorithm $\mathsf{Alg}^{\mathsf{reset}}$ *that uses any offline dataset* D *and reset model* $\mathcal{M}^{\mathsf{reset}}$ *that returns optimal policy* π^* *with sample complexity* $O(A)$*.*

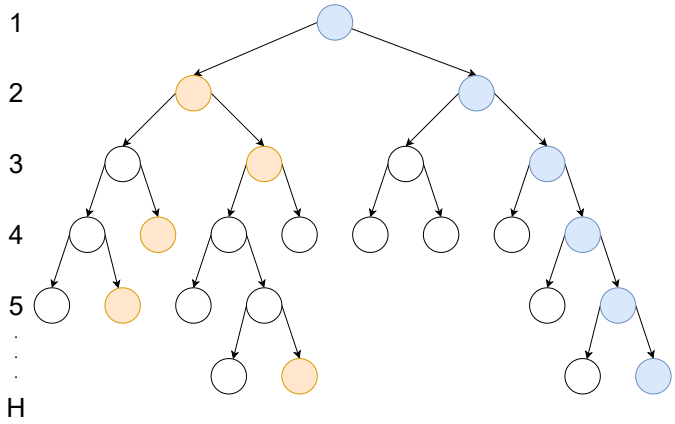

Figure 2. Construction for Proposition 5.1. The blue notes correspond to the offline data's coverage of the optimal policy. The orange note corresponds to the inadmissible part of the offline data.

Proof. Consider a binary tree MDP $\mathcal{M}$, with two actions and deterministic transitions. Now we construct the offline dataset $\mathcal{D}$ as follows: for each horizon h, the dataset $\mathcal{D}_h$ contains one sample from the optimal path and one sample from the other half of the tree. Thus by construction the dataset satisfies the coverage assumption. On the non-optimal data, we select them in the following way: on each level $h \geq 4$, the non-optimal data s_h^{no} is an arbitrary non-child node of the last horizon s_{h-1}^{no}. Thus we see the previous non-optimal states provide no information for the current horizon, and thus the problem of finding s_{h-1}^{no} is equivalent to a random search over an arbitrary leaf node. However, unless s_h^{no} is added to the dataset, we will have $\|D_h - \mathcal{D}_h\|_{\mathsf{TV}} \geq \frac{1}{2}$, and thus we complete the proof. We show an example of such construction in Figure 2, where the orange states denote the non-optimal states covered by the offline dataset and the blue states denote the optimal states covered by the offline dataset. □

Proposition B.2. *For any state distribution* μ_h*, let*

$$\pi_h^\mu = \operatorname*{argmin}_{\pi_h \in \Pi} \|d_h^\pi - \mu_h\|_{\mathsf{TV}},$$

i.e., the policy that induces the closest state distribution to the offline distribution in TV. Then there exists an MDP $\mathcal{M}$ *and inadmissible offline distribution* μ*, such that* $\exists h$ *such that* $\max_h \left\| \frac{d_h^{\pi^*}}{\mu_h} \right\|_\infty = 18$*, i.e., the offline distribution has a constant coverage and we have*

$$\max_h \left\| \frac{d_h^{\pi^*}}{d_h^{\tilde{\pi}}} \right\|_\infty = \infty, \quad \textit{and} \quad \max_{h, s_h} \left\| \frac{\pi_h^*(s_h)}{\tilde{\pi}_h(s_h)} \right\|_\infty = \infty.$$

i.e., the policy that minimizes the TV distance to the offline distribution does not cover some states from the optimal policy's trajectory, and the induced policy does not cover some actions that the optimal policy takes.

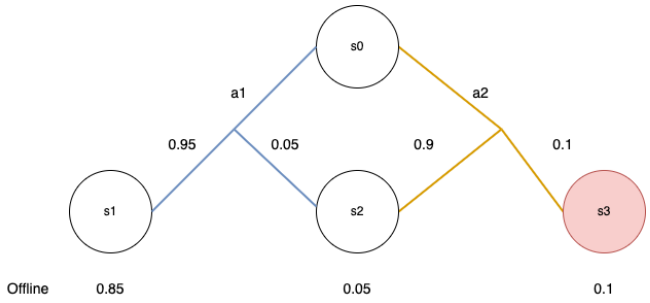

Figure 3. Construction for Proposition 5.2. The blue transition corresponds to the dynamics after taking the action a_1, and the orange transition corresponds to the dynamics after taking the action a_2. The red node denotes the node with rewards.

Proof. Consider the following one-step transition: where we start from s_0, and action a_1, a_2, and next states s_1, s_2, s_3, We have the following transition: $P(\cdot \mid s_0, a_1) = [0.95, 0.05, 0]$ and $P(\cdot \mid s_0, a_2) = [0, 0.9, 0.1]$. Consider the inadmissible offline state distribution over s_1, s_2, s_3 : $\mu = [0.85, 0.05, 0.1]$. We can verify that the coverage assumption holds. Also suppose $\pi^*(s_0) = a_2$. Then by some calculation we have $\pi^\mu(s_0) = a_1$, and thus we have $\left\| \frac{d^{\pi^*}}{d^{\tilde{\pi}}} \right\|_\infty = \infty$, i.e., $\tilde{\pi}$ does not visit s_3 but π^* does. An illustration of the construction can be found in Figure 3. □

C. Proof of FOOBAR

C.1. Proof of Theorem 5.1

For the proof, we define a shorthand notation for the IPM distance as follows:

$$d_{\mathcal{G}_h}(\pi \mid \rho_{h-1}, \mu_h) := \mathsf{IPM}_{\mathcal{G}_h}(\rho \circ_h \pi, \mu_h).$$

Then we have the following guarantee of the Algorithm 5:

Lemma C.1 (Guarantee of Algorithm 5 (Theorem 3.1 of Sun et al. (2019))). *Assume Assumption 5.2 holds. Suppose that $\mathcal{G}_h$ is the discriminator class, ρ is the roll in distribution, and μ is the offline distribution. Let $|\mathcal{D}^{on}| = |\mathcal{D}^{on}| = N$, then with probability $1-\delta$, the returned policy π satisfies:*

$$d_{\mathcal{G}_h}(\pi \mid \rho_{h-1}, \mu_h) \le \min_{\pi' \in \Pi} d_{\mathcal{G}_h}(\pi' \mid \rho_{h-1}, \mu_h) + \varepsilon^{\mathsf{for}}(\delta, N),$$

where

$$\varepsilon^{\mathsf{for}}(\delta, N) = 8\sqrt{\frac{2A\log(2|\mathcal{G}_h||\Pi|/\delta)}{N}} + \frac{16A\log(2|\mathcal{G}_h||\Pi|/\delta)}{N} + \sqrt{\frac{A^2}{T}}.$$

The proof of Lemma C.1 can be found in Sun et al. (2019). The result is the standard concentration argument and taking union bound over the discriminator class and policy class.

Now we can prove the guarantee of Algorithm 4 in IPM, which is the property required in proving the final result.

Theorem C.1 (Guarantee of Algorithm 4). *Assume Assumption 5.3 and Lemma C.1 hold. Suppose $|\mathcal{D}^{on}| = |\mathcal{D}^{on}| = N$, then with probability $1-\delta$, the returned policy π^{f} satisfies that, for any $h \in [H]$,*

$$\mathsf{IPM}_{\mathcal{G}_h}(\pi^{\mathsf{f}}_{1:h}, \mu_h) \le h\varepsilon^{\mathsf{for}}(\delta, N).$$

Proof. We can prove by induction. Consider the h timesteps, where we have

$$\mathsf{IPM}_{\mathcal{G}_{h-1}}(\pi^{\mathsf{f}}_{1:h-1}, \mu_{h-1}) \le (h-1)\varepsilon^{\mathsf{for}}(\delta, N).$$

Let $\pi^* := \operatorname{argmin}_{\pi\in\Pi} d_{\mathcal{G}_h}(\pi \mid \pi^{\mathsf{f}}_{1:h-1}, \mu_h)$, we have:

$$\begin{aligned}
\mathsf{IPM}_{\mathcal{G}_h}(\pi^{\mathsf{f}}_{1:h}, \mu_h) &\le d_{\mathcal{G}_h}(\pi^* \mid \pi^{\mathsf{f}}_{1:h-1}, \mu_h) + \varepsilon^{\mathsf{for}}(\delta, N) && \text{(Lemma C.1)}\\
&= \max_{g\in\mathcal{G}_h} \left|\mathbb{E}_{s_h\sim\pi_{h-1}, a_h\sim\pi^*(s_h), s_{h+1}\sim P_h(s_h,a_h)} g(s_{h+1}) - \mathbb{E}_{s_h\sim\mu_h, a_h\sim\mu(s_h), s_{h+1}\sim P_h(s_h,a_h)} g(s_{h+1})\right| \\
&\qquad + \varepsilon^{\mathsf{for}}(\delta, N)\\
&\le \max_{g\in\mathcal{G}_h} \left|\mathbb{E}_{s_h\sim\pi_{h-1}, a_h\sim\mu(s_h), s_{h+1}\sim P_h(s_h,a_h)} g(s_{h+1}) - \mathbb{E}_{s_h\sim\mu_h, a_h\sim\mu(s_h), s_{h+1}\sim P_h(s_h,a_h)} g(s_{h+1})\right| \\
&\qquad + \varepsilon^{\mathsf{for}}(\delta, N).
\end{aligned}$$

Now denote $g^*_h = \operatorname{argmax}_{g\in\mathcal{G}_h} \left|\mathbb{E}_{s_h\sim\pi_{h-1}, a_h\sim\mu(s_h), s_{h+1}\sim P_h(s_h,a_h)} g(s_{h+1}) - \mathbb{E}_{s_h\sim\mu_h, a_h\sim\mu(s_h), s_{h+1}\sim P_h(s_h,a_h)} g(s_{h+1})\right|$, we denote

$$g^*_{h-1} = \operatorname*{argmin}_{g\in\mathcal{G}_{h-1}} \|g - \mathcal{T}_{h-1} g^*_h\|_\infty,$$

then we have

$$\begin{aligned}
&\left|\mathbb{E}_{s_h\sim\pi_{h-1}, a_h\sim\mu(s_h), s_{h+1}\sim P_h(s_h,a_h)} g^*_h(s_{h+1}) - \mathbb{E}_{s_h\sim\mu_h, a_h\sim\mu(s_h), s_{h+1}\sim P_h(s_h,a_h)} g^*_h(s_{h+1})\right| \\
\le& \left|\mathbb{E}_{s_h\sim\pi_{h-1}} g^*_{h-1}(s_h) - \mathbb{E}_{s_h\sim\mu_h} g^*_{h-1}(s_h)\right| + \varepsilon^{\mathsf{be}} && \text{(Assumption 5.3)}\\
\le& \mathsf{IPM}_{\mathcal{G}_{h-1}}(\pi^{\mathsf{f}}_{1:h-1}, \mu_{h-1}) + \varepsilon^{\mathsf{be}}\\
\le& (h-1)\varepsilon^{\mathsf{for}}(\delta, N) + \varepsilon^{\mathsf{be}}, && \text{(Inductive hypothesis)}
\end{aligned}$$

and thus we complete the proof. □

C.2. Proof of Theorem 5.2.

Now we prove the guarantee of Algorithm 1. We start with the guarantee of the value function estimation in Algorithm 3.

Lemma C.2. *Suppose Assumption 5.4 holds. For any $h \in [H]$, let f_h be the returned value function from running Algorithm 3 With $|\mathcal{D}_h^{\mathsf{on}}| = N$, then with probability at least $1-\delta$, we have:*

$$\mathbb{E}_{s_h \sim \pi^{\mathsf{f}}_{1:h-1}} \max_{a_h} \left[\left(f_h(s_h, a_h) - Q_h^{\pi^{\mathsf{b}}}(s_h, a_h) \right)^2 \right] \lesssim \frac{H^2 A \log(|\mathcal{F}|/\delta)}{n} := \left(\varepsilon^{\mathsf{back}}(\delta, N) \right)^2.$$

Proof. First we have

$$\begin{aligned}\mathbb{E}_{s_h \sim \pi^{\mathsf{f}}_{1:h-1}} \max_{a_h} \left[\left(f_h(s_h, a_h) - Q_h^{\pi^{\mathsf{b}}}(s_h, a_h) \right)^2 \right] &\leq \mathbb{E}_{s_h \sim \pi^{\mathsf{f}}_{1:h-1}} \sum_{a \in \mathcal{A}} \left[\left(f_h(s_h, a) - Q_h^{\pi^{\mathsf{b}}}(s_h, a) \right)^2 \right] \\ &= A \mathbb{E}_{s_h, a_h \sim \pi^{\mathsf{f}} \circ_h \pi^{\mathsf{unif}}} \left[\left(f_h(s_h, a_h) - Q_h^{\pi^{\mathsf{b}}}(s_h, a_h) \right)^2 \right].\end{aligned}$$

Then follows standard least-square analysis (Lemma A.11, Agarwal et al. (2019)), since $\pi^{\mathsf{b}} \circ_h \pi^{\mathsf{unif}}$ is our roll-in distribution, we have

$$\mathbb{E}_{s_h, a_h \sim \pi^{\mathsf{f}} \circ_h \pi^{\mathsf{unif}}} \left[\left(f_h(s_h, a_h) - Q_h^{\pi^{\mathsf{b}}}(s_h, a_h) \right)^2 \right] \lesssim \frac{H^2 \log(|\mathcal{F}|/\delta)}{N},$$

where H is the range of the regression target. □

Recall our construction of the discriminator class $\mathcal{G}_h$:

$$\mathcal{G}_h = \left\{ \max_a f(\cdot, a) - f(\cdot, a') \mid f \in \mathcal{F}_h, a' \in \mathcal{A} \right\}.$$

The reason for such construction will be clear in the proof of Theorem 5.2. But we first show that the size of the discriminator class is bounded by the size of the value function class so it is not big. We assume that the value function class is finite for simplicity, but the results can be easily extended to the infinite case.

Lemma C.3. $|\mathcal{G}_h| \leq |\mathcal{F}_h||\mathcal{A}|$.

The proof follows immediately from the construction of the discriminator class.

Then we can show the performance guarantee of Algorithm 1:

Theorem C.2 (Restatement of Theorem 5.2)**.** *Suppose Assumption 5.2, Assumption 5.3 and Assumption 5.4 hold. Then with probability at least $1-\delta$, the returned policy $\pi^{\mathsf{b}}_{1:H}$ from Algorithm 1 with discriminator constructed from Eq. (1), N^{for} offline and forward samples, and N^{back} backward samples, satisfies that for any comparator policy π^{comp} such that $C_{\mathsf{cov}}(\pi^{\mathsf{comp}}) < \infty$,*

$$V^{\pi^{\mathsf{comp}}} - V^{\pi^{\mathsf{b}}} \leq \varepsilon,$$

when

$$N^{\mathsf{for}} = O\left(\frac{C^2_{\mathsf{cov}}(\pi^{\mathsf{comp}}) H^4 A \log(|\mathcal{F}||\Pi|/\delta)}{\varepsilon^2} \right), \quad N^{\mathsf{back}} = O\left(\frac{C^2_{\mathsf{cov}}(\pi^{\mathsf{comp}}) H^4 A \log(|\mathcal{F}|/\delta)}{\varepsilon^2} \right).$$

Proof. By performance difference lemma (Kakade & Langford, 2002), we have that

$$
\begin{aligned}
& V^{\pi^{\mathsf{comp}}} - V^{\pi^{\mathsf{b}}_{1:H}} \\
&= \sum_{h=1}^{H} \mathbb{E}_{s_h,a_h \sim d_h^{\pi^{\mathsf{comp}}}} \left[Q_h^{\pi^{\mathsf{b}}}(s_h,a_h) - Q_h^{\pi^{\mathsf{b}}}(s_h,\pi_h^{\mathsf{b}}(s_h)) \right] \\
&\leq \sum_{h=1}^{H} \mathbb{E}_{s_h \sim d_h^{\pi^{\mathsf{comp}}}} \left[\max_a Q_h^{\pi^{\mathsf{b}}}(s_h,a) - Q_h^{\pi^{\mathsf{b}}}(s_h,\pi_h^{\mathsf{b}}(s_h)) \right] \\
&\leq C_{\mathsf{cov}}(\pi^{\mathsf{comp}}) \sum_{h=1}^{H} \mathbb{E}_{s_h \sim \mu_h} \left[\max_a Q_h^{\pi^{\mathsf{b}}}(s_h,a) - Q_h^{\pi^{\mathsf{b}}}(s_h,\pi_h^{\mathsf{b}}(s_h)) \right] && \text{(Non-negativity)} \\
&\leq C_{\mathsf{cov}}(\pi^{\mathsf{comp}}) \left(\sum_{h=1}^{H} \mathbb{E}_{s_h \sim d_h^{\pi^{\mathsf{f}}}} \left[\max_a Q_h^{\pi^{\mathsf{b}}}(s_h,a) - Q_h^{\pi^{\mathsf{b}}}(s_h,\pi_h^{\mathsf{b}}(s_h)) \right] + \mathsf{IPM}_{\mathcal{G}_h}\left(d_h^{\pi^{\mathsf{f}}} || \mu_h \right) \right) && \text{(Construction of } \mathcal{G}) \\
&\leq C_{\mathsf{cov}}(\pi^{\mathsf{comp}}) \cdot \\
&\left(\sum_{h=1}^{H} \mathbb{E}_{s_h \sim d_h^{\pi^{\mathsf{f}}}} \left[\left| \max_a Q_h^{\pi^{\mathsf{b}}}(s_h,a) - f_h(s_h,\pi_h^{\mathsf{b}}(s_h)) \right| + \left| f_h(s_h,\pi_h^{\mathsf{b}}(s_h)) - Q_h^{\pi^{\mathsf{b}}}(s_h,\pi_h^{\mathsf{b}}(s_h)) \right| \right] + \mathsf{IPM}_{\mathcal{G}_h}\left(d_h^{\pi^{\mathsf{f}}} || \mu_h \right) \right) \\
&\leq C_{\mathsf{cov}}(\pi^{\mathsf{comp}}) \cdot \\
&\left(\sum_{h=1}^{H} \mathbb{E}_{s_h \sim d_h^{\pi^{\mathsf{f}}}} \left[\max_a \left| Q_h^{\pi^{\mathsf{b}}}(s_h,a) - f_h(s_h,a) \right| + \left| f_h(s_h,\pi_h^{\mathsf{b}}(s_h)) - Q_h^{\pi^{\mathsf{b}}}(s_h,\pi_h^{\mathsf{b}}(s_h)) \right| \right] + \mathsf{IPM}_{\mathcal{G}_h}\left(d_h^{\pi^{\mathsf{f}}} || \mu_h \right) \right) \\
& && (\max_x |f(x) - g(x)| \geq |\max_x f(x) - \max_x g(x)|) \\
&\lesssim C_{\mathsf{cov}}(\pi^{\mathsf{comp}}) H \varepsilon^{\mathsf{back}}(\delta, N^{\mathsf{back}}) + C_{\mathsf{cov}}(\pi^{\mathsf{comp}}) H^2 \varepsilon^{\mathsf{for}}(\delta, N^{\mathsf{for}}).
\end{aligned}
$$

The last step is by Jensen's inequality, Lemma C.2 and Theorem 5.1. By plugging in the defintion of $\varepsilon^{\mathsf{back}}$ and $\varepsilon^{\mathsf{for}}$, we have

$$
V^{\pi^{\mathsf{comp}}} - V^{\pi^{\mathsf{b}}_{1:H}} \leq O\left(C_{\mathsf{cov}}(\pi^{\mathsf{comp}}) H \sqrt{\frac{H^2 A \log(|\mathcal{F}|/\delta)}{N^{\mathsf{back}}}} + C_{\mathsf{cov}}(\pi^{\mathsf{comp}}) H \sqrt{\frac{A \log(|\mathcal{F}||\Pi|/\delta)}{N^{\mathsf{for}}}} \right),
$$

by setting $T = AN^{\mathsf{for}}$ and by Lemma C.3, finally by setting N properly, we have the desired result. □

Here we remark that in order for our proof to hold, we only require the following weaker notion of realizability of value function class $\mathcal{F}$: we only require that $Q^{\pi^{\mathsf{b}}} \in \mathcal{F}$, and in fact we only require it to hold under state visited by π^{f} and μ.

D. Pracical Considerations

D.1. Imperfect Forward Run

One advantage of this forward-backward algorithm is that, either due to optimization error or insufficient data size, if $\varepsilon^{\mathsf{for}}$ is not small, we can still guarantee the final performance as long as the following holds: define the coverage with respect to the forward policy as

$$C^{\mathsf{for}}_{\mathsf{cov}}(\pi) := \max_h \left\| \frac{d^{\pi}_h}{d^{\pi^{\mathsf{f}}}_h} \right\|_{\infty}.$$

and we can have the following guarantee:

Theorem D.1 (FOOBAR guarantee for imperfect forward run)**.** *With probability at least $1-\delta$, for any comparator policy π^{comp}, we have*

$$V^{\pi^{\mathsf{comp}}} - V^{\pi^{\mathsf{b}}_{1:H}} \le O\left(C^{\mathsf{for}}_{\mathsf{cov}}(\pi^{\mathsf{comp}}) H \sqrt{\frac{H^2 A \log(|\mathcal{F}|/\delta)}{N^{\mathsf{back}}}} \right).$$

Note that the number of offline and online forward samples will contribute to the term $C^{\mathsf{for}}_{\mathsf{cov}}(\pi^{\mathsf{comp}})$, but here we make their relationship implicit. The theorem states that by paying $C^{\mathsf{for}}_{\mathsf{cov}}(\pi^{\mathsf{comp}})$ (but potentially $C^{\mathsf{for}}_{\mathsf{cov}}(\pi^{\mathsf{comp}}) > C_{\mathsf{cov}}(\pi^{\mathsf{comp}})$), we can avoid paying the additive term $C_{\mathsf{cov}}(\pi^{\mathsf{comp}}) H \sqrt{\frac{A \log(|\mathcal{F}||\Pi|/\delta)}{N^{\mathsf{for}}}}$. The practical application of this theorem can be demonstrated by our result for the robotics task in Section 6, where perfectly mimicking the offline distribution is hard duo to the high-dimensional continuous action space. Nevertheless, the forward policy still covers the optimal policy, and thus FOOBAR returns the optimal policy after the backward phase. We show the proof below:

Proof. We have again by performance difference lemma,

$$\begin{aligned}
& V^{\pi^{\mathsf{comp}}} - V^{\pi^{\mathsf{b}}_{1:H}} \\
&= \sum_{h=1}^{H} \mathbb{E}_{s_h, a_h \sim d^{\pi^{\mathsf{comp}}}_h} \left[Q^{\pi^{\mathsf{b}}}_h(s_h, a_h) - Q^{\pi^{\mathsf{b}}}_h(s_h, \pi^{\mathsf{b}}_h(s_h)) \right] \\
&\le \sum_{h=1}^{H} \mathbb{E}_{s_h \sim d^{\pi^{\mathsf{comp}}}_h} \left[\max_a Q^{\pi^{\mathsf{b}}}_h(s_h, a) - Q^{\pi^{\mathsf{b}}}_h(s_h, \pi^{\mathsf{b}}_h(s_h)) \right] \\
&\le \sum_{h=1}^{H} \mathbb{E}_{s_h \sim d^{\pi^{\mathsf{comp}}}_h} \left[\max_a \left| Q^{\pi^{\mathsf{b}}}_h(s_h, a) - f_h(s_h, a) \right| + \left| f_h(s_h, \pi^{\mathsf{b}}_h(s_h)) - Q^{\pi^{\mathsf{b}}}_h(s_h, \pi^{\mathsf{b}}_h(s_h)) \right| \right] \\
&\le C^{\mathsf{for}}_{\mathsf{cov}}(\pi^{\mathsf{comp}}) \left(\sum_{h=1}^{H} \mathbb{E}_{s_h \sim d^{\pi^{\mathsf{f}}}_h} \left[\max_a \left| Q^{\pi^{\mathsf{b}}}_h(s_h, a) - f_h(s_h, a) \right| + \left| f_h(s_h, \pi^{\mathsf{b}}_h(s_h)) - Q^{\pi^{\mathsf{b}}}_h(s_h, \pi^{\mathsf{b}}_h(s_h)) \right| \right] \right) \\
&\le C^{\mathsf{for}}_{\mathsf{cov}}(\pi^{\mathsf{comp}}) H \varepsilon^{\mathsf{back}}(\delta, N^{\mathsf{back}}),
\end{aligned}$$

note that in this case, we can directly shift the distribution from π^{comp} to π^{f} in line 3. The rest of the proof is the same as the proof of Theorem 5.2. □

D.2. Stationary Results

In this section we introduce a variant of our algorithm and analysis in the stationary setting. We will start with introducing new notations for the stationary setting, and like the procedure in our main text, we will first introduce the backward phase of the algorithm, which *requires no additional assumptions and directly extends to the stationary setting*. We will end up with our forward phase algorithm, which requires an additional assumption that the offline dataset is interactive.

Notations. We start with introducing the notations in the stationary setting. In the stationary setting, we are interested in the finite horizon discounted MDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, R, \gamma)$, where the transition kernel $P : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ and reward

Algorithm 6 Conservative Policy Iteration (CPI) with trace model

require Roll in policy π^{f}, accuracy parameter ε, step size α.

1: Initialize π^1 randomly.
2: **for** $t = 1, 2, \ldots$ **do**
3: $\quad \pi' \leftarrow \mathsf{Greedy}_{\varepsilon}(\pi^t, \Pi, d^{\pi^{\mathsf{f}}})$.
4: $\quad$ **if** $\mathbb{E}_{s\sim\pi^t} A^{\pi^t}(s, \pi'(s)) \le \varepsilon$ **then**
5: $\quad\quad$ **return** π^t.
6: $\quad$ Update policy conservatively:

$$\pi^{t+1} \leftarrow (1-\alpha)\pi^t + \alpha\pi'. \tag{2}$$

function $R : \mathcal{S} \times \mathcal{A} \to [0, 1]$ are stationary. We denote γ as the discounted factor. For any policy π, we denote the value function as $V^\pi(s) = \mathbb{E}_{\pi,P}\left[\sum_{h=1}^\infty \gamma^h R(s_h, a_h) \mid s_1 = s\right]$ and $Q^\pi(s) = \mathbb{E}_{\pi,P}\left[\sum_{h=1}^\infty \gamma^h R(s_h, a_h) \mid s_1 = s, a_1 = a\right]$. We still denote d_h^π as the visitation distribution of policy π at horizon h, and we will often use $d^\pi = (1-\gamma)\sum_{h=1}^\infty \gamma^h d_h^\pi$ as the stationary visitation distribution (or occupancy measure) of policy π. We denote μ as the offline distribution that is constructed in a similar manner, i.e., $\mu = (1-\gamma)\sum_{h=1}^\infty \gamma^h \mu_h$, and we denote the coverage of μ as $C_{\mathsf{cov}}(\pi) = \left\|\frac{d^\pi}{\mu}\right\|_\infty$. Finally in this section, to simply the notation, we will make extensive use of the notion of advantage, which is defined as $A^\pi(s, \pi'(s)) = \mathbb{E}_{a\sim\pi'(s)} Q^\pi(s, a) - Q^\pi(s, \pi(s))$.

Backward phase. For the backward phase, we will use the classic Conservative Policy Iteration (CPI) (Kakade & Langford, 2002) algorithm, which is a stationary algorithm that guarantees the optimality of the returned policy under an exploratory reset distribution, which in our case will be our forward policy's state visitation distribution. The intuition of CPI is similar to the backward pass of FOOBAR, where we first roll in the forward policy π^{f}, and then we will update our backward policy by rolling out and perform policy optimization. Specifically, given a policy π, a policy class Π and an initial distribution μ, the output of the greedy policy selector $\pi' \leftarrow \mathsf{Greedy}_{\varepsilon}(\pi, \Pi, \mu)$ has the following guarantee:

$$\mathbb{E}_{s\sim d_\mu^\pi} A^\pi(s, \pi'(s)) \ge \max_{\widetilde{\pi}\in\Pi} \mathbb{E}_{s\sim d_\mu^\pi} A^\pi(s, \widetilde{\pi}(s)) - \varepsilon,$$

where d_μ^π is the state visitation distribution of policy π under the initial distribution μ. In practice, to ensure that the initial distribution of the policy optimization problem is $d^{\pi^{\mathsf{f}}}$, we can start to roll in π^{f}, and at each timestep, we will start to switch to roll out policy π with probability $1-\gamma$ (Agarwal et al., 2020b). In the stationary setting, however, we do not use the greedy policy as the next policy, because we can not guarantee the optimality in an inductive way, but we can still ensure a local improvement by performing a conservative policy update (Eq. (2)). We provide the pseudocode of CPI in Algorithm 6.

For simplicity, we will not perform the finite sample analysis on the $\mathsf{Greedy}_{\varepsilon}$ subprocedure, but we will assume that the greedy policy selector guarantee always holds, and we will prove the final optimality result based on it. We first state a critical lemma that is useful for the analysis of CPI:

Lemma D.1 (Local optimality of CPI, Theorem 14.3 of Agarwal et al. (2019))**.** *Algorithm 6 terminates in at most* $8\gamma/\varepsilon^2$ *steps and the output policy* π *satisfies that*

$$\max_{\pi'\in\Pi} \mathbb{E}_{s\sim d_\mu^\pi} A^\pi(s, \pi'(s)) \le 2\varepsilon,$$

where $\mu = d^{\pi^{\mathsf{f}}}$.

With the local optimality guarantee, we can show that the result for Algorithm 6:

Theorem D.2 (Guarantee of CPI-trace)**.** *Let the returned policy of Algorithm 6 be* π^{b}*, suppose policy class* Π *is realizable in the sense that*

$$\mathbb{E}_{s\sim d^{\pi^{\mathsf{b}}}}\left[\max_{a\in\mathcal{A}} A^{\pi^{\mathsf{b}}}(s, a)\right] - \mathbb{E}_{s\sim d^{\pi^{\mathsf{b}}}}\left[A^{\pi^{\mathsf{b}}}(s, \pi(s))\right] = 0.$$

Then we have that

$$V^{\pi^{\mathsf{comp}}} - V^{\pi^{\mathsf{b}}} \le \frac{C_{\mathsf{cov}}(\pi^{\mathsf{comp}})}{(1-\gamma)^2}(2\varepsilon) + \frac{C_{\mathsf{cov}}(\pi^{\mathsf{comp}})}{(1-\gamma)}\mathsf{IPM}_{\mathcal{G}}\left(d_h^{\pi^{\mathsf{f}}} || \mu_h\right).$$

Proof. By performance difference lemma (Kakade & Langford, 2002), we have that

$$
\begin{aligned}
V^{\pi^{\mathsf{comp}}} - V^{\pi^{\mathsf{b}}} &= \frac{1}{1-\gamma}\mathbb{E}_{s\sim d^{\pi^{\mathsf{comp}}}}\left[A^{\pi^{\mathsf{b}}}(s,\pi^{\mathsf{comp}}(s))\right] \\
&\le \frac{1}{1-\gamma}\mathbb{E}_{s\sim d^{\pi^{\mathsf{comp}}}}\left[\max_{a\in\mathcal{A}} A^{\pi^{\mathsf{b}}}(s,a)\right] \\
&\le \frac{1}{1-\gamma}C_{\mathsf{cov}}(\pi^{\mathsf{comp}})\mathbb{E}_{s\sim\mu}\left[\max_{a\in\mathcal{A}} A^{\pi^{\mathsf{b}}}(s,a)\right] \\
&\le \frac{1}{1-\gamma}C_{\mathsf{cov}}(\pi^{\mathsf{comp}})\left(\mathbb{E}_{s\sim\pi^{\mathsf{f}}}\left[\max_{a\in\mathcal{A}} A^{\pi^{\mathsf{b}}}(s,a)\right] + \mathsf{IPM}_{\mathcal{G}}\left(d^{\pi^{\mathsf{f}}}||\mu\right)\right) \\
&\le \frac{C_{\mathsf{cov}}(\pi^{\mathsf{comp}})}{(1-\gamma)^2}\mathbb{E}_{s\sim d^{\pi^{\mathsf{b}}}}\left[\max_{a\in\mathcal{A}} A^{\pi^{\mathsf{b}}}(s,a)\right] + \frac{C_{\mathsf{cov}}(\pi^{\mathsf{b}})}{(1-\gamma)}\mathsf{IPM}_{\mathcal{G}}\left(d^{\pi^{\mathsf{f}}}||\mu\right) \qquad (d^{\pi^{\mathsf{b}}}(s)\ge(1-\gamma)\pi^{\mathsf{f}}(s)) \\
&\le \frac{C_{\mathsf{cov}}(\pi^{\mathsf{b}})}{(1-\gamma)^2}\mathbb{E}_{s\sim d^{\pi^{\mathsf{b}}}}\left[\max_{pi'\in\Pi}\mathbb{E}_{s\sim d^{\pi^{\mathsf{b}}}}\left[A^{\pi^{\mathsf{b}}}(s,\pi'(s))\right] - \max_{pi'\in\Pi}\mathbb{E}_{s\sim d^{\pi^{\mathsf{b}}}}\left[A^{\pi^{\mathsf{b}}}(s,\pi'(s))\right] + \max_{a\in\mathcal{A}} A^{\pi^{\mathsf{b}}}(s,a)\right] \\
&\quad + \frac{C_{\mathsf{cov}}(\pi^{\mathsf{b}})}{(1-\gamma)}\mathsf{IPM}_{\mathcal{G}}\left(d_h^{\pi^{\mathsf{f}}}||\mu_h\right) \\
&\le \frac{C_{\mathsf{cov}}(\pi^{\mathsf{b}})}{(1-\gamma)^2}(2\varepsilon) + \frac{C_{\mathsf{cov}}(\pi^{\mathsf{b}})}{(1-\gamma)}\mathsf{IPM}_{\mathcal{G}}\left(d^{\pi^{\mathsf{f}}}||\mu\right). \qquad \text{(Lemma D.1 and realizability)}
\end{aligned}
$$

Note that for the simplicity of the presentation, we denote $d^{\pi^{\mathsf{b}}} := d_{\mu}^{\pi^{\mathsf{b}}}$ where $\mu := d^{\pi^{\mathsf{f}}}$. □

Theorem D.2 states that, as long as the forward policy π^{f} covers the comparator policy, we can guarantee the performance of the returned policy π^{b} is close to the best comparator policy. Next we see how we can achieve the guarantee of the forward policy in the stationary setting.

Forward phase. In the forward phase, we assume that we have an interactive offline distribution μ^{it}, which for any state s, if we query the offline distribution μ^{it} with s, we will return a sample s' by $a\sim\mu^{\mathsf{it}}(s), s'\sim P(s,a)$. However, since we are in the observation-only setting, we only observe s' but not a, and thus this is a relaxation from the previous works that assume interactive experts which also provide the action information (Ross et al., 2011; Ross & Bagnell, 2012).

Our roll-in procedure is similar to the backward phase, where for each horizon, we will have probability $(1-\gamma)$ to terminate the roll in on that horizon. Denote the state at termination as s, we will take a random action $a\sim\pi^{\mathsf{unif}}$ and observe s', and we add the tuple (s,a,s') to the online dataset $\mathcal{D}^{\mathsf{on}}$, and similarly, we query μ^{it} with s, and get $s'\sim\mu^{\mathsf{it}}(s)$, and add (s') to offline dataset $\mathcal{D}^{\mathsf{off}}$. Then we perform the best-response playing no-regret algorithm to iteratively update our policy, similar to Algorithm 5. The full pseudocode is in Algorithm 7.

Algorithm 7 Interactive Forward Adversarial Imitation Learning (Inter-FAIL)

require Discriminator class $\mathcal{G}$, policy class Π, number of iterations T.

$\mathcal{D}^{\mathsf{on}}\leftarrow\emptyset, \mathcal{D}^{\mathsf{off}}\leftarrow\emptyset$.
Randomly initialize π^1.
for $t=1$ to T **do**
  Sampling stopping time $h\sim\mathsf{Geom}(1-\gamma)$.
  Sample $s,a,s'\sim\pi^t\circ_h\pi^{\mathsf{unif}}$, and add (s,a,s') to $\mathcal{D}^{\mathsf{on}}$.
  Sample $s'\sim\mu(s)$, and add (s') to $\mathcal{D}^{\mathsf{off}}$.

$$u(\pi,g) := \left(\widehat{\mathbb{E}}_{\mathcal{D}^{\mathsf{on}}}\left[\frac{\pi(a\mid s)}{1/A}g(s')\right] - \widehat{\mathbb{E}}_{\mathcal{D}^{\mathsf{off}}}\left[g(s')\right]\right).$$

  $g^t = \mathrm{argmax}_{g\in\mathcal{G}}\, u(\pi^t,g)$.// Linear programming oracle.
  $u^t := u(\pi^t,g^t)$.
  $\pi^{t+1} = \mathrm{argmin}_{\pi\in\Pi}\sum_{\tau=1}^{t} u(\pi,g^t)+\phi(\pi)$. // Regularized cost-sensitive oracle.
return π^{t^*} with $t^* = \mathrm{argmin}_{t\in[T]}\, u^t$.

Now we first show the guarantee on the no-regret procedure. Note that this result does not immediately implies that the

return policy to the offline distribution, because our data collection distribution is: we first roll in our policy, and then switch to the offline distribution for one-step. And the no-regret guarantee is that the policy distribution will be close to this one-step-shift distribution. In the following, whenever we refer to the admissibility assumption Assumption 5.2 or bellman completeness assumption Assumption 5.3, we refer to their stationary version analog.

Lemma D.2 (Guarantee of the no-regret procedure)**.** *Assume Assumption 5.2 holds. Suppose that $\mathcal{G}$ is the discriminator class, ρ is the roll in distribution, then let π^{f} be the return policy from running Algorithm 7 for T iterations, we have with probability at least $1-\delta$,*

$$d_{\mathcal{G}_h}(\pi \mid \pi, \pi \circ \mu) \leq \min_{\pi' \in \Pi} d_{\mathcal{G}_h}(\pi' \mid \pi', \pi' \circ) + \varepsilon^{\mathsf{for}}(\delta, T),$$

where

$$\varepsilon^{\mathsf{for}}(\delta, T) = 8\sqrt{\frac{2A^2 \log(2|\mathcal{G}_h||\Pi|/\delta)}{T}}.$$

The proof is using the same concentration argument used in the proof of Lemma C.2 and the same no-regret techniques that handle non-stationary distributions in (Vemula et al., 2023) so we omit proof here.

Then we can use the no-regret guarantee for the final result for the forward algorithm:

Theorem D.3 (Guarantee of Algorithm 7)**.** *Assume Assumption 5.3 and Lemma D.2 hold with probability at least $1-\delta$. Then the returned policy π^{f} of Algorithm 7 after T iterations satisfies that,*

$$\mathsf{IPM}_{\mathcal{G}}(\pi^{\mathsf{f}}, \mu) \leq \frac{\varepsilon^{\mathsf{for}}(\delta, T) + \varepsilon^{\mathsf{be}}}{1-\gamma}.$$

Proof. We start with an important identity: for any stationary policy π, we have

$$\mathbb{E}_{s\sim d^{\pi}}[f(s)] = (1-\gamma)\mathbb{E}_{s\sim P_0}[f(s)] + \gamma \mathbb{E}_{s\sim d^{\pi}, a\sim\mu, s'\sim P(s,a)}[f(s')].$$

Then we have

$$\begin{aligned}
\mathsf{IPM}_{\mathcal{G}}(\pi^{\mathsf{f}}, \mu) =& \max_{g\in\mathcal{G}} \left|\mathbb{E}_{s\sim d^{\pi^{\mathsf{f}}}}[g(s)] - \mathbb{E}_{s\sim\mu}[g(s)]\right| \\
=& \max_{g\in\mathcal{G}} \left|\gamma \mathbb{E}_{s\sim d^{\pi^{\mathsf{f}}}, a\sim\pi^{\mathsf{f}}(s), s'\sim P(s,a)}[g(s')] - \gamma \mathbb{E}_{s\sim\mu, a\sim\mu(s), s'\sim P(s,a)}[g(s')]\right| \\
\leq& \max_{g\in\mathcal{G}} \left|\gamma \mathbb{E}_{s\sim d^{\pi^{\mathsf{f}}}, a\sim\pi^{\mathsf{f}}(s), s'\sim P(s,a)}[g(s')] - \gamma \mathbb{E}_{s\sim d^{\pi^{\mathsf{f}}}, a\sim\mu(s), s'\sim P(s,a)}[g(s')]\right| + \\
& \max_{g\in\mathcal{G}} \left|\gamma \mathbb{E}_{s\sim d^{\pi^{\mathsf{f}}}, a\sim\mu(s), s'\sim P(s,a)}[g(s')] - \gamma \mathbb{E}_{s\sim\mu, a\sim\mu(s), s'\sim P(s,a)}[g(s')]\right|.
\end{aligned}$$

Note that the first term, by the no-regret guarantee in Lemma D.2, is bounded by $\varepsilon^{\mathsf{for}}$, and the second term we can bound by the following, which is similar to the technique we use in the proof for the forward run in the non-stationary setting:

Now denote

$$g^* := \operatorname*{argmax}_{g\in\mathcal{G}} \left|\gamma \mathbb{E}_{s\sim d^{\pi^{\mathsf{f}}}, a\sim\mu(s), s'\sim P(s,a)}[g(s')] - \gamma \mathbb{E}_{s\sim\mu, a\sim\mu(s), s'\sim P(s,a)}[g(s')]\right|$$

we let

$$g^{\mathsf{b}} = \operatorname*{argmin}_{g\in\mathcal{G}} \|g - \mathcal{T} g^*\|_{\infty},$$

the bellman backup of g^* under the offline distribution backup, then we have

$$\begin{aligned}
& \max_{g\in\mathcal{G}} \left|\gamma \mathbb{E}_{s\sim d^{\pi^{\mathsf{f}}}, a\sim\mu(s), s'\sim P(s,a)}[g(s')] - \gamma \mathbb{E}_{s\sim\mu, a\sim\mu(s), s'\sim P(s,a)}[g(s')]\right| \\
=& \gamma \left|\mathbb{E}_{s\sim d^{\pi^{\mathsf{f}}}, a\sim\mu(s), s'\sim P(s,a)}[g^*(s')] - \mathbb{E}_{s\sim\mu, a\sim\mu(s), s'\sim P(s,a)}[g^*(s')]\right| \\
\leq& \gamma \left|\mathbb{E}_{s\sim d^{\pi^{\mathsf{f}}}}[g^*(s)] - \mathbb{E}_{s\sim\mu}[g^*(s)]\right| + \varepsilon^{\mathsf{be}} \\
=& \gamma \mathsf{IPM}_{\mathcal{G}}(\pi^{\mathsf{f}}, \mu) + \varepsilon^{\mathsf{be}}.
\end{aligned}$$

Finally, putting everything together we will get:

$$\mathsf{IPM}_{\mathcal{G}}(\pi^{\mathsf{f}}, \mu) \leq \varepsilon^{\mathsf{for}} + \gamma \mathsf{IPM}_{\mathcal{G}}(\pi^{\mathsf{f}}, \mu) + \varepsilon^{\mathsf{be}},$$

which by rearranging we get:

$$\mathsf{IPM}_{\mathcal{G}}(\pi^{\mathsf{f}}, \mu) \leq \frac{1}{1-\gamma}\varepsilon^{\mathsf{for}} + \varepsilon^{\mathsf{be}}.$$

□

Finally, to obtain the result for the stationary version, we can simply combine the result of Theorem D.2 and Theorem D.3. And we obtain the stationary analog of Theorem 5.2 by replacing the horizon dependency to the effective horizon $\frac{1}{1-\gamma}$. We remark that, in the stationary setting, the name "forward phase" and "backward phase" may not be as clear as in the non-stationary setting, but one can interpret the "forward phase" as the offline distribution matching phase, and the "backward phase" as the policy refinement (optimization) phase. Here to contextualize the discussion, we introduce the definition of the bilinear class model:

E. Discussion on the Structural Assumption

Here we give the formal introduction of the structural assumption. We adopt the one from Du et al. (2021) as it is the structural assumption made in the most hybrid RL analysis (Song et al., 2022; Nakamoto et al., 2023). However, the results will transfer trivially to similar structural assumptions like Bellman Eluder dimension(Jin et al., 2021) or coverability (Xie et al., 2023). In other hybrid RL works, Wagenmaker & Pacchiano (2023) assumes linear MDPs structure (Jin et al., 2020b) and Li et al. (2023b) assumes tabular MDPs.

Definition E.1 (Bilinear model (Du et al., 2021))**.** *We say that the MDP together with the function class $\mathcal{F}$ is a bilinear model of rank d if for any $h \in [H-1]$, there exist two (unknown) mappings $X_h, W_h : \mathcal{F} \mapsto \mathbb{R}^d$ with $\max_f \|X_h(f)\|_2 \leq B_X$ and $\max_f \|W_h(f)\|_2 \leq B_W$ such that:*

$$\forall f, g \in \mathcal{F} : \left| \mathbb{E}_{s,a \sim d_h^{\pi^f}} \left[g_h(s,a) - \mathcal{T} g_{h+1}(s,a) \right] \right| = \left| \langle X_h(f), W_h(g) \rangle \right|.$$

Note that the dimension of the mapping X and W are called the bilinear rank, which is bounded by d. For example, in tabular MDPs, $d = SA$, and in linear MDPs (Jin et al., 2020b) and low-rank MDPs (Agarwal et al., 2020a), d is the dimension of the feature vector.

Continuing from Remark 5.2, suppose we are in the tabular setting, since we involve function approximation, the worst-case log size of the function class will still be bounded by SA, and then the final bound will be worse than the tightest bound in the tabular case (Zhang et al., 2023b). Note that in the worst case, the S dependency is unavoidable in the hybrid RL setting even with canonical offline data (see Theorem 3 of Xie et al. (2021b)). However, we argue that the dependency of SA has a different source compared to the tightest analysis in tabular MDPs such as Azar et al. (2017); Zhang et al. (2023b): the SA dependency in these analyses is from the fundamental complexity measure $d = SA$ in the MDP itself. For example, in the worst case, one has to hit each state-action pairs enough times such that the confidence intervals shrink. On the other hand, the size of the function class is not necessarily tied to the complexity of dynamics or rewards, and the SA dependency of the log size of the function class is always avoidable with the right choice of function class (inductive bias). However, unlike our analysis, such SA dependency still shows up in the current hybrid RL analysis, where their suboptimality scales in (ignoring irrelevant terms):

$$V^{\pi^{\mathsf{comp}}} - V^{\pi} \leq O\left(C_{\mathsf{cov}}(\pi^{\mathsf{comp}}) \sqrt{\frac{d \log(|\mathcal{F}|/\delta)}{N}} \right) = O\left(C_{\mathsf{cov}}(\pi^{\mathsf{comp}}) \sqrt{\frac{SA \log(|\mathcal{F}|/\delta)}{N}} \right),$$

i.e., previous results pay for both SA and $\log(|\mathcal{F}|)$.

Our result is even more favorable in the more general cases. In the main text, we use the ℓ_∞ coverage for the simplicity of presentation, which may be unbounded when the state space is not finite. Here we introduce a tighter coverage coefficient that is similar to the previous *expected* Bellman error coverage used in offline RL (Xie et al., 2021a) and hybrid RL (Song et al., 2022; Nakamoto et al., 2023), which we called *performance difference coverage*:

Definition E.2 (Performance difference coverage)**.** *For the given offline distributino ρ, and for any policy π, the performance difference coverage coefficient is define as*

$$C_{\mathsf{cov}}^{\mathsf{pd}}(\pi) = \max_{\pi' \in \Pi^{\mathsf{det}}} \frac{\sum_{h=1}^H \mathbb{E}_{s_h \sim d_h^\pi} \left[\max_a A_h^{\pi'}(s_h, a) \right]}{\sum_{h=1}^H \mathbb{E}_{s_h \sim \mu_h} \left[\max_a A_h^{\pi'}(s_h, a) \right]}.$$

With this we can state the following more refined result:

Theorem E.1. *Suppose Assumption 5.2, Assumption 5.3 and Assumption 5.4 hold. Then with probability $1-\delta$, the returned policy $\pi^{\mathsf{b}}_{1:H}$ from Algorithm 1 with discriminator constructed from Eq. (1), N^{for} offline and forward samples, and N^{back} backward samples, satisfies that for any comparator policy π^{comp} such that $C_{\mathsf{cov}}^{\mathsf{pd}}(\pi^{\mathsf{comp}}) < \infty$,*

$$V^{\pi^{\mathsf{comp}}} - V^{\pi^{\mathsf{b}}} \leq \varepsilon,$$

when

$$N^{\mathsf{for}} = O\left(\frac{{C_{\mathsf{cov}}^{\mathsf{pd}}}^2(\pi^{\mathsf{comp}}) H^4 A \log(|\mathcal{F}||\Pi|/\delta)}{\varepsilon^2} \right), \quad N^{\mathsf{back}} = O\left(\frac{{C_{\mathsf{cov}}^{\mathsf{pd}}}^2(\pi^{\mathsf{comp}}) H^4 A \log(|\mathcal{F}|/\delta)}{\varepsilon^2} \right).$$

The proof is the same as the proof of Theorem 5.2, and one can check we can safely replace C_{cov} with $C^{\mathsf{pd}}_{\mathsf{cov}}$ during the distribution shift step. Note that this result does not depend on specific structural complexity measures of the MDPs (e.g., the bilinear rank (Du et al., 2021; Song et al., 2022)). On the other hand, one advantage of previous hybrid RL algorithms is that they work under situations where the offline data is inadmissible (c.r. Table 1).

Intuitively, the bilinear rank assumption captures the following idea: the rank d denotes the number of "distribution shift" that the algorithm will encounter during the online policy or value function update, i.e., how many times the algorithms have to roll out so that the previous data distribution will cover the current policy's visitation distribution. However, in FOOBAR, there is no distribution issue (because for every horizon, we will collect some data, train a one-step policy, commit to the policy, and not update it anymore). We believe the absence of the distribution shift problem is partially due to the admissibility assumption we make for the offline dataset, but an understanding of the fundamental connections between the admissibility and structural assumptions remains an interesting open problem.

Finally, we remark that there is one previous hybrid RL work that is also free from the structural assumption, which is Zhou et al. (2023). However, like the previous line of works that study RL in the reset model (Kakade & Langford, 2002; Bagnell et al., 2003), their analysis requires an exploratory reset distribution, which is as strong as having a reset model.

F. Experiment Details

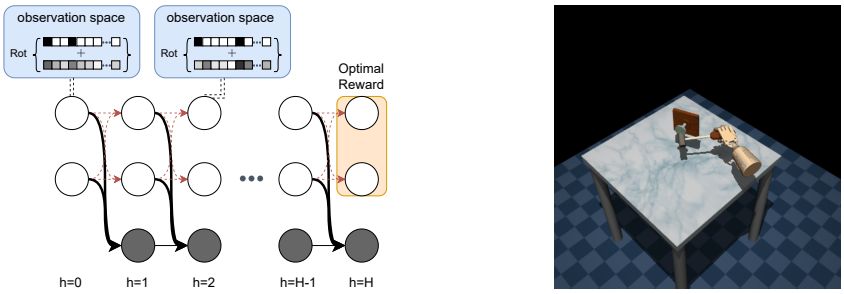

Figure 4. Visualization of the environment. Left: combination lock. Right: hammer. The left figure is reproduced from Zhang et al. (2022) with permission from the authors.

F.1. Combination Lock Environment

Detailed environment description. In our experiment, the *diabolical combination lock* problem (referenced as comb-lock) served as the testing ground for our algorithm. This scenario is defined by a horizon of length H and involves a selection from 10 distinct actions. At each point in the sequence, denoted as step h, the system can be in one of three potential hidden states, symbolized as $z_{i;h}$ for i values in the set $0, 1, 2$. States $z_{i;h}$, where i falls within $0, 1$, are considered advantageous, while the state $z_{2;h}$ is categorized as disadvantageous.

For each advantageous state $z_{i;h}$ (where i is either 0 or 1), an action, denoted as $a_{i;h}$, is chosen randomly from the pool of 10 actions. In such states, executing the action $a_{i;h}$ leads to a transition to either state $z_{0;h+1}$ or $z_{1;h+1}$, with each possibility having an equal chance of occurrence. Choosing any action other than $a^*_{i;h}$ in these states ensures a move to the state $z_{2;h+1}$. In the state $z_{2;h}$, the agent's action choice does not affect its transition, which is always to $z_{2;h+1}$.

The reward structure is such that a reward of 1 is assigned at state $z_{i;H}$ for $i \in 0, 1$. There is also a 50% probability of receiving a minor, inverted reward of 0.1 when transitioning from a favorable to an unfavorable state. All other state transitions or states do not yield any reward.

Observations in this problem, denoted as s, have a dimensionality of $2^{\lceil \log(H+4) \rceil}$. This is formulated by concatenating the one-hot vectors representing the hidden state z and the horizon h, to which noise from the distribution $\mathcal{N}(0, 0.1)$ is added for each dimension. This is then adjusted with zeroes where necessary and processed through a Hadamard matrix. The starting state distribution is uniformly divided among $z_{i;0}$ for $i \in 0, 1$. An important aspect to note is that the ideal strategy involves consistently selecting the action $a^*_{i;h}$ at each step h. Once the agent enters a disadvantageous state, it remains in such state till the episode concludes, thus forfeiting the opportunity for a significant end reward. This presents a significant challenge in terms of exploration, as a strategy based on random uniform selection yields only a 10^{-H} chance of reaching the intended goals.

Implementation details. We parametrize the forward policy with a 2-layer neural network with Tanh activation and we model the action distribution with diagonal Gaussian. For the backward policy, we use least square regression to estimate the Q-functions where we follow the same parametrization as in Song et al. (2022). We use the median trick (Fukumizu et al., 2009) to set up the bandwidth for the RBF kernel. Hyperparameters for the combination lock experiment are presented in Table 3.

For completeness, here we also provide a zoomed-in training curve for FOOBAR with both forward phase and backward phase labeled in Figure 5.

F.2. Hammer

For the offline dataset construction of the hammer environment, we use the expert offline dataset provided in the d4rl benchmark. Take the first 2000 trajectories and extract the first 50 horizons for each trajectory for the offline dataset. Note that for the hammer environment, the expert dataset does not contain the optimal policy, and in fact only 80% trajectories of offline datasets contain a successful state at horizon 50. We use the expert offline dataset mainly due to the fact that this

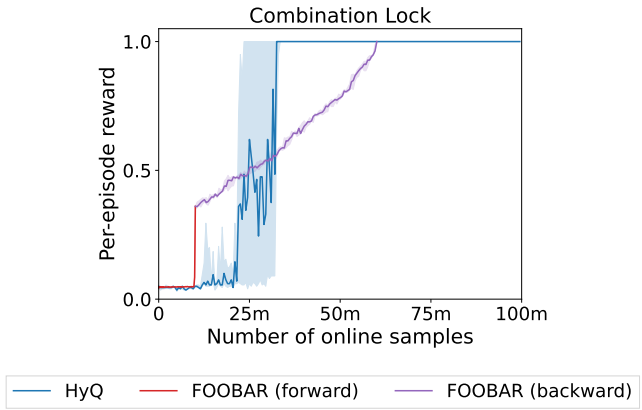

Figure 5. Zoomed-in training curve of FOOBAR.

Table 3. Hyperparameters for combination lock

	Value
Offline sample size (per horizon)	2000
Online forward sample size (per horizon)	2000
Forward policy hidden layer size	128
Min-max game iteration	1000
Online backward sample size (per horizon)	5000
Backward number of gradient descent updates	1500
Backward minibatch size	128
Learning rate	0.001

dataset is collected by a diagonal Gaussian policy, which is the same as our parametrization of the policy so admissibility holds gracefully. However, we believe using the recently proposed diffusion policy (Block et al., 2023) will address this issue since diffusion policies can parameterize multimodal distributions.

As mentioned in the main text, we make one modification on the forward phase that for each horizon, we iterate between optimizing the forward policy and using the latest policy to collect more data. During training, we still use the same min-max objective and instead of performing importance weighting on the uniform policy, we adjust the importance weight with respect to the data collection policy. For the backward run, we follow Algorithm 6: we roll-in the forward policy to a random horizon, and we switch to the current stationary SAC policy to roll out and only update the SAC policy using the data collected during the roll out. We provide the hyperparameter table in Table 4.

To show the performance of the forward run, we notice that on average the forward policies will have 10% success rate at the end of the forward phase (compared to the 80% success rate in the offline dataset). However, it is due to the strict success evaluation of the hammer-binary environment, and we note that even if the policy fails to solve the task, it still covers the optimal policy reasonably, and thus although in theory, the IPM between forward policy and offline distribution may not be small, the forward policy still covers the optimal policy, and the learning will success due to Theorem E.1. Here we give a qualitative and quantitative evaluation of the forward policy. For the qualitative evaluation, we visualize a typical failure trajectory of the forward policy in Figure 6 and note that the hammer hits the nail but does not fully push the nail into the board. For the qualitative evaluation, we test the empirical Jensen-Shannon divergence between the dataset induced by the forward policy and the offline dataset, and we plot the average across the 10 runs in natural log scale in Figure 7.

F.3. Inadmissble Offline

In this section, we describe the construction of the experiments in Section 6.2. For the benign inadmissibility setting, we collect the offline data in the following way: we reset the initial state distribution the same way as regular combination lock, and for horizon $h = 1$, we generate the observation of state 0 (good state), state 1 (good state) and state 2 (bad state) with

Table 4. Hyperparameters for hammer

	Value
Offline sample size (per horizon)	2000
Online forward sample size (per horizon)	2000
Forward policy hidden layer size	128
Min-max game iteration	1000
Online backward sample size (per horizon)	5000
Backward number of gradient descent updates	1500
Backward minibatch size	128
Learning rate	0.001

Figure 6. Visualization of a typical failure trajectory of the forward policy. Note that the hammer hits the nail but does not fully push the nail into the board.

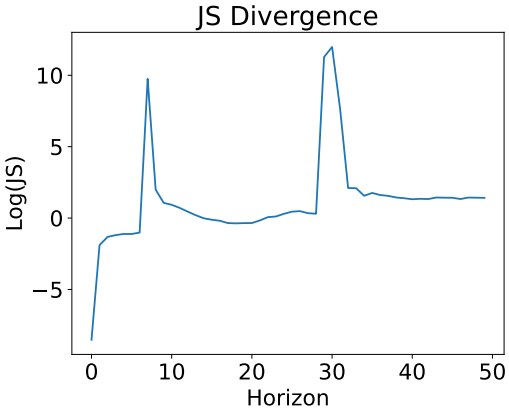

Figure 7. Plot of empirical JS divergence between forward policy and offline data for each horizon. The y-axis is in the natural log scale.

probability $(0.1, 0.05, 0.85)$ respectively. For $h \geq 2$, we generate the observation of state 0 (good state), state 1 (good state) and state 2 (bad state) with probability $(0.5 \cdot h,\ 0.05 \cdot h,\ 1 - 0.1 \cdot h)$ respectively. Note that this is an inadmissible offline dataset because the probability of visiting good states is non-increasing over the horizon for any admissible distribution.

For the adversarial inadmissibility setting, the offline distribution follows the same construction as the benign setting: we reset the initial state distribution the same way as regular combination lock, and for horizon $h = 1$, we generate the observation of state 0 (good state), state 1 (good state) and state 2 (bad state) with probability $(0.1, 0.05, 0.85)$ respectively. For $h \geq 2$, we generate the observation of state 0 (good state), state 1 (good state) and state 2 (bad state) with probability $(0.5 \cdot h,\ 0.05 \cdot h,\ 1 - 0.1 \cdot h)$ respectively. However, we modify P_1 and P_2 of the combination lock in the following way: at horizon 1, taking good actions in either state 0 or state 1 will have a 0.1 probability transiting to state 0 in timestep 2, and 0.9 probability to state 1; taking any bad action will have a probability of 0.05 transiting to state 1, and 0.85 probability to transit to state 2. However, in timestep 2, only state 0 will be treated as a good state, and state 1 will be treated as a bad state and thus taking any action in state 1 in timestep 2 will transit to state 2 deterministically. All the remaining dynamics are the same as the regular combination lock. We note that this is exactly the same construction as in Proposition 5.2, and the optimal policy will have a success rate of 10% due to the stochasticity of the environment.

Finally, we include the hyperparameters for each baseline in Table 5 and Table 6.

Table 5. Hyperparameters for FOOBAR

	Value
Offline sample size (per horizon)	2000
Online forward sample size (per horizon)	2000
Forward policy hidden layer size	128
Min-max game iteration	1000
Online backward sample size (per horizon)	4000
Backward number of gradient descent updates	2000
Backward minibatch size	128
Learning rate	0.001

Table 6. Hyperparameters for PSDP

	Value
Offline sample size (per horizon)	2000
Online sample size (per horizon)	4000
Number of gradient descent updates	2000
Minibatch size	128
Learning rate	0.001