Hybrid RL: Using both offline and online data can make RL efficient

Yuda Song *
Carnegie Mellon University
yudas@cs.cmu.edu

Yifei Zhou *
Cornell University
yz639@cornell.edu

Ayush Sekhari
MIT
sekhari@mit.edu

J. Andrew Bagnell
Carnegie Mellon University
dbagnell@aurora.tech

Akshay Krishnamurthy
Microsoft Research
akshaykr@microsoft.com

Wen Sun
Cornell University
ws455@cornell.edu

Abstract

We consider a hybrid reinforcement learning setting (Hybrid RL), in which an agent has access to an offline dataset and the ability to collect experience via real-world online interaction. The framework mitigates the challenges that arise in both pure offline and online RL settings, allowing for the design of simple and highly effective algorithms, in both theory and practice. We demonstrate these advantages by adapting the classical Q learning/iteration algorithm to the hybrid setting, which we call Hybrid Q-Learning or Hy-Q. In our theoretical results, we prove that the algorithm is both computationally and statistically efficient whenever the offline dataset supports a high-quality policy and the environment has bounded bilinear rank. Notably, we require no assumptions on the coverage provided by the initial distribution, in contrast with guarantees for policy gradient/iteration methods. In our experimental results, we show that Hy-Q with neural network function approximation outperforms state-of-the-art online, offline, and hybrid RL baselines on challenging benchmarks, including Montezuma's Revenge.

1 Introduction

Learning by interacting with an environment, in the standard online reinforcement learning (RL) protocol, has led to impressive results across a number of domains. State-of-the-art RL algorithms are quite general, employing function approximation to scale to complex environments with minimal domain expertise and inductive bias. However, online RL agents are also notoriously sample inefficient, often requiring billions of environment interactions to achieve suitable performance. This issue is particularly salient when the environment requires sophisticated exploration and a high quality reset distribution is unavailable to help overcome the exploration challenge. As a consequence, the practical success of online RL and related policy gradient/improvement methods has been largely restricted to settings where a high quality simulator is available.

To overcome the issue of sample inefficiency, attention has turned to the offline RL setting (Levine et al., 2020), where, rather than interacting with the environment, the agent trains on a large dataset of experience collected in some other manner (e.g., by a system running in production or an expert). While these methods still require a large dataset, they mitigate the sample complexity concerns of online RL, since the dataset can be collected without compromising system performance. However, offline RL methods can suffer from *distribution shift*, where the state distribution induced by the learned policy differs significantly from the offline distribution (Wang et al., 2021). Existing provable approaches for addressing distribution shift are computationally intractable, while empirical approaches rely on heuristics that can be sensitive to the domain and offline dataset (as we will see).

*equal contribution

Offline Reinforcement Learning Workshop at Neural Information Processing Systems, 2022.

In this paper, we focus on a hybrid reinforcement learning setting, which we call Hybrid RL, that draws on the favorable properties of both offline and online settings. In Hybrid RL, the agent has both an offline dataset and the ability to interact with the environment, as in the traditional online RL setting. The offline dataset helps address the exploration challenge, allowing us to greatly reduce the number of interactions required. Simultaneously, we can identify and correct distribution shift issues via online interaction. Variants of the setting have been studied in a number of empirical works (Rajeswaran et al., 2017; Hester et al., 2018; Nair et al., 2018, 2020; Vecerik et al., 2017) which mainly focus on using expert demonstrations as offline data. Our algorithmic development is closely related to these works, although our focus is on formalizing the hybrid setting and establishing theoretical guarantees against more general offline datasets.

Hybrid RL is closely related to the *reset setting*, where the agent can interact with the environment starting from a "nice" distribution. A number of simple and effective algorithms, including CPI (Kakade & Langford, 2002), PSDP (Bagnell et al., 2003), and policy gradient methods (Kakade, 2001; Agarwal et al., 2020b)—which have further inspired deep RL methods such as TRPO (Schulman et al., 2015) and PPO (Schulman et al., 2017)—are provably efficient in the reset setting. Yet, a nice reset distribution is a strong requirement (often tantamount to having access to a detailed simulation) and unlikely to be available in real world applications. Hybrid RL differs from the reset setting in that (a) we have an offline dataset, but (b) our online interactions start from the initial distribution of the environment, which is not assumed to have any nice properties. Both features (offline data and a nice reset distribution) facilitate algorithm design by de-emphasizing the exploration challenge. However, Hybrid RL is much more practical since an offline dataset is much easier to obtain in practice.

We showcase the Hybrid RL setting with a new algorithm, Hybrid Q learning or Hy-Q (pronounced: Haiku). The algorithm is a simple adaptation of the classical fitted Q-iteration algorithm (FQI) and accommodates value-based function approximation.[2] For our theoretical results, we prove that Hy-Q is both statistically and computationally efficient assuming that: (1) the offline distribution covers some high quality policy, (2) the MDP has low bilinear rank, (3) the function approximator is Bellman complete, and (4) we have a least squares regression oracle. The first three assumptions are standard statistical assumptions in the RL literature while the fourth is a widely used computational abstraction for supervised learning. No computationally efficient algorithms are known under these assumptions in pure offline or pure online settings, which highlights the advantages of the hybrid setting.

We also implement Hy-Q and evaluate it on two challenging RL benchmarks: a rich observation combination lock (Misra et al., 2020) and Montezuma's Revenge from the Arcade Learning Environment (Bellemare et al., 2013). Starting with an offline dataset that contains some transitions from a high quality policy, our approach outperforms: an online RL baseline with theoretical guarantees, an online deep RL baseline tuned for Montezuma's Revenge, pure offline RL baselines, imitation learning baselines, and existing hybrid methods. Compared to the online methods, Hy-Q requires only a small fraction of the online experience, demonstrating its sample efficiency. Compared to the offline and hybrid methods, Hy-Q performs most favorably when the offline dataset also contains many interactions from low quality policies, demonstrating its robustness. These results reveal the significant benefits that can be realized by combining offline and online data.

2 Related Works

We discuss related works from four categories: pure online RL, online RL with access to a reset distribution, offline RL, and prior work in hybrid settings. We note that pure online RL refers to the setting where one can only reset the system to initial state distribution of the environment, which is not assumed to provide any form of coverage.

Pure online RL Beyond tabular settings, many existing statistically efficient RL algorithms are not computationally tractable, due to the difficulty of implementing optimism. This is true in the linear MDP (Jin et al., 2020) with large action spaces, the linear Bellman complete model (Zanette et al., 2020; Agarwal et al., 2019), and in the general function approximation setting (Jiang et al., 2017; Sun et al., 2019; Du et al., 2021; Jin et al., 2021a). These computational challenges have inspired results on intractability of aspects of online RL (Dann et al., 2018; Kane et al., 2022).

[2]We use Q-learning and Q-iteration interchangeably, although they are not strictly speaking the same algorithm. Our theoretical results analyze Q-iteration, but we use an algorithm with an online/mini-batch flavor that is closer to Q-learning for our experiments.

There are several online RL algorithms that aim to tackle the computational issue via stronger structural assumptions and supervised learning-style computational oracles (Misra et al., 2020; Zhang et al., 2022c; Agarwal et al., 2020a; Uehara et al., 2021; Modi et al., 2021; Zhang et al., 2022a; Qiu et al., 2022). Compared to these oracle-based methods, our approach operates in the more general "bilinear rank" setting and relies on a standard supervised learning primitive: least squares regression. Notably, our oracle admits efficient implementation with linear function approximation, so we obtain an end-to-end computational guarantee; this is not true for prior oracle-based methods.

There are many deep RL methods for the online setting (e.g., Schulman et al. (2015, 2017); Lillicrap et al. (2016); Haarnoja et al. (2018); Schrittwieser et al. (2020)). Apart from a few exceptions (e.g., Burda et al. (2018); Badia et al. (2020); Guo et al. (2022)), most rely on random exploration and are not capable of strategic exploration. In our experiments, we test our approach on Montezuma's Revenge, and we pick RND (Burda et al., 2018) as a deep RL exploration baseline due to its effectiveness.

Online RL with reset distributions When an exploratory reset distribution is available, a number of statistically and computationally efficient algorithms are known. The classic algorithms are CPI (Kakade & Langford, 2002), PSDP (Bagnell et al., 2003), Natural Policy Gradient (Kakade, 2001; Agarwal et al., 2020b), and POLYTEX (Abbasi-Yadkori et al., 2019). Uchendu et al. (2022) recently demonstrated that algorithms like PSDP work well when equipped with modern neural network function approximators. However, these algorithms (and their analyses) heavily rely on the reset distribution to mitigate the exploration challenge, but such a reset distribution is typically unavailable in practice, unless one also has a simulator. In contrast, we assume the offline data covers some high quality policy, which helps with exploration, but we do not require an exploratory reset distribution. This makes the hybrid setting much more practically appealing.

Offline RL Offline RL methods learn policies solely from a given offline dataset, with no interaction whatsoever. When the dataset has global coverage, algorithms such as FQI (Munos & Szepesvári, 2008; Chen & Jiang, 2019) or certainty-equivalence model learning (Ross & Bagnell, 2012), can find near-optimal policies in an oracle-efficient manner, via least squares or model-fitting oracles. However, with only partial coverage, existing methods either (a) are not computationally efficient due to the difficulty of implementing pessimism both in linear settings with large action spaces (Jin et al., 2021b; Zhang et al., 2022b; Chang et al., 2021) and general function approximation settings (Uehara & Sun, 2021; Xie et al., 2021a; Jiang & Huang, 2020; Chen & Jiang, 2022; Zhan et al., 2022), or (b) require strong representation conditions such as policy-based Bellman completeness (Xie et al., 2021a; Zanette et al., 2021). In contrast, in the hybrid setting, we obtain an efficient algorithm under the more natural condition of completeness w.r.t., the Bellman optimality operator only.

Among the many empirical offline RL methods (e.g., Kumar et al. (2020); Yu et al. (2021); Kostrikov et al. (2021); Fujimoto & Gu (2021)), we use CQL (Kumar et al., 2020) as a baseline in our experiments, since it has been shown to work in image-based control settings such as Atari games.

Online RL with offline datasets Ross & Bagnell (2012) developed a model-based algorithm for a similar hybrid setting. In comparison, our approach is model-free and consequently more suitable for high-dimensional state spaces (e.g., raw-pixel images). Xie et al. (2021b) studied hybrid RL and show that offline data does not yield statistical improvements in tabular MDPs. Our work instead focuses on the function approximation setting and demonstrates computational benefits of hybrid RL.

On the empirical side, several works consider combining offline expert demonstrations with online interaction (Rajeswaran et al., 2017; Hester et al., 2018; Nair et al., 2018, 2020; Vecerik et al., 2017). A common challenge in offline RL is the robustness against low-quality offline dataset. Previous works mostly focus on expert demonstrations and have no rigorous guarantees for such robustness. In fact, Nair et al. (2020) showed that such degradation in performance indeed happens in practice with low-quality offline data. In our experiments, we observe that DQfD (Hester et al., 2018) also has a similar degradation. On the other hand, our algorithm is robust to the quality of the offline data. Note that the core idea of our algorithm is similar to that of Vecerik et al. (2017), who adapt DDPG to the setting of combining RL with expert demonstrations for continuous control. Although Vecerik et al. (2017) does not provide any theoretical results, it may be possible to combine our theoretical insights with existing analyses for policy gradient methods to establish some guarantees of the algorithm from Vecerik et al. (2017) for the hybrid RL setting. We also include a detailed comparison with previous empirical work in Appendix D.

3 Preliminaries

We consider finite horizon Markov Decision Process $M(\mathcal{S}, \mathcal{A}, H, R, P, d_0)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, H denotes the horizon, stochastic rewards $R(s,a) \in \Delta([0,1])$ and

Algorithm 1 Hybrid Q-learning using both offline and online data (Hy-Q)

Require: Value class: $\mathcal{F}$, #iterations: T, offline dataset $\mathcal{D}^{\nu}_h$ of size $m_{\text{off}} = HT/d$ for $h \in [H-1]$.

1: Initialize $f^1_h(s,a) = 0$.
2: **for** $t = 1, \ldots, T$ **do**
3: Let π^t be the greedy policy w.r.t. f^t i.e., $\pi^t_h(s) = \text{argmax}_a f^t_h(s,a)$.
4: For each h, collect $m_{\text{on}} = H^2$ online tuples $\mathcal{D}^t_h \sim d^{\pi^t}_h$. `// Online collection`
`// FQI using both online and offline data`
5: Set $f^{t+1}_H(s,a) = 0$.
6: **for** $h = H-1, \ldots, 0$ **do**
7: Estimate f^{t+1}_h using least squares regression on the aggregated data:

$$f^{t+1}_h \leftarrow \underset{f \in \mathcal{F}_h}{\text{argmin}} \left\{ \widehat{\mathbb{E}}_{\mathcal{D}^{\nu}_h}(f(s,a) - r - \max_{a'} f^{t+1}_{h+1}(s',a'))^2 + \sum_{\tau=1}^{t} \widehat{\mathbb{E}}_{\mathcal{D}^{\tau}_h}(f(s,a) - r - \max_{a'} f^{t+1}_{h+1}(s',a'))^2 \right\} \quad (1)$$

8: **end for**
9: **end for**

$P(s,a) \in \Delta(\mathcal{S})$ are the reward and transition distributions at (s,a), and $d_0 \in \Delta(\mathcal{S})$ is the initial distribution. We assume the agent can only reset from d_0 (at the beginning of each episode). Since the optimal policy is non-stationary in this setting, we define a policy $\pi := \{\pi_0, \ldots, \pi_{H-1}\}$ where $\pi_h : \mathcal{S} \mapsto \Delta(\mathcal{A})$. Given π, $d^{\pi}_h \in \Delta(\mathcal{S} \times \mathcal{A})$ denotes the state-action occupancy induced by π at step h.

Given π, we define the state and state-action value functions in the usual manner: $V^{\pi}_h(s) = \mathbb{E}[\sum_{\tau=h}^{H-1} r_\tau | \pi, s_h = s]$ and $Q^{\pi}_h(s,a) = \mathbb{E}[\sum_{\tau=h}^{H-1} r_\tau | \pi, s_h = s, a_h = a]$. $Q^\star$ and $V^\star$ denote the optimal value functions. We define the Bellman operator $\mathcal{T}$ such that for any $f : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}$,

$$\mathcal{T} f(s,a) = \mathbb{E}[R(s,a)] + \mathbb{E}_{s' \sim P(s,a)} \max_{a'} f(s',a') \qquad \forall s, a,$$

We assume that for each h we have an offline dataset of m_{off} samples (s,a,r,s') drawn iid via $(s,a) \sim \nu_h, r \in R(s,a), s' \sim P(s,a)$. Here $\nu = \{\nu_0, \ldots, \nu_{H-1}\}$ denote the corresponding offline data distributions. For a dataset $\mathcal{D}$, we use $\hat{\mathbb{E}}_{\mathcal{D}}[\cdot]$ to denote a sample average over this dataset. For our theoretical results, we will assume that ν covers some high-quality policy.

We consider the value-based function approximation setting, where we are given a function class $\mathcal{F} = \mathcal{F}_0 \times \cdots \times \mathcal{F}_{H-1}$ with $\mathcal{F}_h \subset \mathcal{S} \times \mathcal{A} \mapsto [0, V_{\max}]$ that we use to approximate the value functions for the underlying MDP. For ease of notation, we define $f = \{f_0, \ldots, f_{H-1}\}$ and define π^f to be the greedy policy w.r.t., f, which chooses actions as $\pi^f_h(s) = \text{argmax}_a f_h(s,a)$.

4 Hybrid Q-Learning

In this section, we present our algorithm *Hybrid Q Learning* – Hy-Q in Algorithm 1. Hy-Q takes an offline dataset $\mathcal{D}^{\nu}$ that contains (s,a,r,s') tuples and a Q function class $\mathcal{F} \subset \mathcal{S} \times \mathcal{A} \mapsto [0,H]$ as inputs, and outputs a policy that optimizes the given reward function. The algorithm is conceptually simple: it iteratively executes the Fitted Q Iteration procedure (line 6) using the offline dataset *and* on-policy samples generated by the learned policies.

Specifically, at iteration t, we have an estimate f^t of the $Q^\star$ function and we set π^t to be the greedy policy for f^t. We execute π^t to collect a dataset $\mathcal{D}^t_h$ of online samples in line 4. Then we run FQI, a dynamic programming style algorithm on both the offline dataset $\mathcal{D}^{\nu}$ and all previously collected online samples $\{\mathcal{D}^{\tau}_h\}_{\tau=1}^t$. The FQI update works backward from time step H to 0 and computes f^{t+1}_h via least squares regression with input (s,a) and regression target $r + \max_{a'} f^{t+1}_{h+1}(s',a')$.[3]

Let us make several remarks. Intuitively, the FQI updates in Hy-Q try to ensure that the estimate f^t has small Bellman error under both the offline distribution ν and the online distributions d^{π^t}. The standard offline version of FQI ensures the former, but this alone is insufficient when the offline dataset has poor coverage. Indeed FQI may have poor performance in such cases (see examples in Zhan et al., 2022; Chen & Jiang, 2022). The key insight in Hy-Q is to use online interaction to ensure that we also have small Bellman error on d^{π^t}. As we will see, the moment we find an f^t that has

[3]Note that FQI and Hy-Q extend to the infinite horizon discounted setting (Munos & Szepesvári, 2008).

small Bellman error on the offline distribution ν and *its own greedy policy's distribution* d^{π^t}, FQI guarantees that π^t will be at least as good as *any* policy covered by ν. This observation results in an explore-or-terminate phenomenon: either f^t has small Bellman error on its distribution and we are done, or d^{π^t} must be significantly different from distributions we have seen previously and we make progress. Crucially, no explicit exploration is required for this argument, which is precisely how we avoid the computational difficulties with implementing optimism.

Another important point pertains to *catastrophic forgetting*. We will see that the size of the offline dataset m_{off} should be comparable to the total amount of online data $\{\mathcal{D}_h^\tau\}_{\tau=1}^T$, so that the two terms in Eq. 1 have similar weight and we ensure low Bellman error on ν throughout the learning process. In practice, we implement this by having all model updates use a fixed proportion of offline samples even as we collect more online data, so that we do not "forget" the distribution ν. This is quite different from warm-starting with $\mathcal{D}^\nu$ and then switching to online RL, which may result in catastrophic forgetting due to a vanishing proportion of offline samples being used for model training as we collect more online samples. We note that this balancing scheme is analogous to and inspired by the one used by Ross & Bagnell (2012) in the context of model-based RL with a reset distribution. As in their work, a key practical insight from our analysis is that the offline data should be used throughout training to avoid catastrophic forgetting.

5 Theoretical Analysis: Low Bilinear Rank Models

In this section we present the main theoretical guarantees for Hy-Q. We start by stating the main assumptions and definitions for the function approximator, the offline data distribution, and the MDP. We state the key definitions and then provide some discussion.

Assumption 1 (Realizability and Bellman completeness)**.** *For any h, we have $Q_h^\star \in \mathcal{F}_h$. Additionally, for any $f_{h+1} \in \mathcal{F}_{h+1}$, we have $\mathcal{T} f_{h+1} \in \mathcal{F}_h$.*

Definition 1 (Bellman error transfer coefficient)**.** *For any policy π, define the transfer coefficient as*

$$C_\pi := \max\left\{0,\ \max_{f\in\mathcal{F}} \frac{\sum_{h=0}^{H-1} \mathbb{E}_{s,a\sim d_h^\pi}\left[\mathcal{T} f_{h+1}(s,a) - f_h(s,a)\right]}{\sqrt{\sum_{h=0}^{H-1} \mathbb{E}_{s,a\sim \nu_h}\left(\mathcal{T} f_{h+1}(s,a) - f_h(s,a)\right)^2}}\right\}. \tag{2}$$

Definition 2 (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 $\max_f \|X_h(f)\|_2 \le B_X$ and $\max_f \|W_h(f)\|_2 \le B_W$ such that:*

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

All concepts defined above are frequently used in the statistical analysis of RL methods with function approximation. Realizability is the most basic function approximation assumption, but is known to be insufficient for offline RL (Foster et al., 2021) unless other strong assumptions hold (Xie & Jiang, 2021; Zhan et al., 2022; Chen & Jiang, 2022). Completeness is the most standard strengthening of realizability that is used routinely in both online (Jin et al., 2021a) and offline RL (Munos & Szepesvári, 2008; Chen & Jiang, 2019) and is known to hold in several settings including the linear MDP and the linear quadratic regulator. These assumptions ensure that the dynamic programming updates of FQI are stable in the presence of function approximation.

The transfer coefficient definition above is somewhat non-standard, but is actually weaker than related notions used in prior offline RL results. First, the average Bellman error appearing in the numerator is weaker than the squared Bellman error notion of (Xie et al., 2021a); a simple calculation shows that C_π^2 is upper bounded by their coefficient. Second, by using Bellman errors, both of these are bounded by notions involving density ratios (Kakade & Langford, 2002; Munos & Szepesvári, 2008; Chen & Jiang, 2019). Finally, many works, particularly those that do not employ pessimism (Munos & Szepesvári, 2008; Chen & Jiang, 2019), require "all-policy" analogs, which places a much stronger requirement on the offline data distribution ν. In contrast, we will only ask that C_π is small for *some* high-quality policy that we hope to compete with (see Appendix A.5 for more details).

Lastly, the bilinear model was developed in a series of works (Jiang et al., 2017; Jin et al., 2021a; Du et al., 2021) on sample efficient online RL.[4] The setting is known to capture a wide class of models

[4] Jin et al. (2021a) consider the Bellman Eluder dimension, which is related but distinct from the Bilinear model. However, our proofs can be easily translated to this setting; see Appendix B for more details.

including linear MDPs, linear Bellman complete models, low-rank MDPs, reactive POMDPs, and more. As a technical note, the main paper focuses on the "Q-type" version of the bilinear model, but the algorithm and proofs easily extend to the "V-type" version. See Appendix B for details.

Theorem 1 (Cumulative subpotimality). *Fix $\delta \in (0,1)$, number of total offline samples $m_{\text{off}} = HT/d$ and number of online samples at each iteration $m_{\text{on}} = H^2$, and suppose that the underlying MDP admits Bilinear rank d, and the function class $\mathcal{F}$ satisfies Assumption 1. Then with probability at least $1-\delta$, Algorithm 1 obtains the following bound on cumulative subpotimality w.r.t. any comparator policy π^e,*

$$\sum_{t=1}^{T} V^{\pi^e} - V^{\pi^t} = \widetilde{O}\Big(V_{\max} \max\{C_{\pi^e}, 1\} \sqrt{dT \cdot \log(|\mathcal{F}|/\delta)}\Big),$$

where $\pi^t = \pi^{f^t}$ is the greedy policy w.r.t. f^t at round t.

A standard online-to-batch conversion (Shalev-Shwartz & Ben-David, 2014) immediately gives the following sample complexity guarantee for Algorithm 1 for finding an ϵ-suboptimal policy w.r.t. the optimal policy π^* for the underlying MDP.

Corollary 1 (Sample complexity). *Under the assumptions of Theorem 1 if $C_{\pi^*} < \infty$ then Algorithm 1 can find an ϵ-suboptimal policy $\widehat{\pi}$ for which $V^{\pi^*} - V^{\widehat{\pi}} \le \epsilon$ with total sample complexity:*

$$n = \widetilde{O}\big(V_{\max}^2 C_{\pi^*}^2 H^3 d \log(|\mathcal{F}|/\delta)/\epsilon^2\big)$$

The results formalize the statistical properties of Hy-Q. In terms of sample complexity, a somewhat unique feature of the hybrid setting is that both transfer coefficient and bilinear rank parameters are relevant, whereas these (or related) parameters typically appear in isolation in offline and online RL respectively. In terms of coverage, Theorem 1 highlights an "oracle property" of Hy-Q: it competes with *any* policy that is sufficiently covered by the offline dataset.

We also highlight the computational efficiency of Hy-Q: it only requires solving least squares problems over the function class $\mathcal{F}$. To our knowledge, no purely online or purely offline methods are known to be efficient in this sense, except under much stronger "uniform" coverage conditions.

5.1 The Linear Bellman Completeness Model

We next showcase one example of low bilinear rank models: the popular linear Bellman complete model which captures the linear MDP model (Yang & Wang, 2019; Jin et al., 2020), and instantiate the sample complexity bound in Corollary 1.

Definition 3. *Given a feature function $\phi : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{B}_d(1)$, a model admits linear Bellman completeness if for any $w \in \mathbb{B}_d(B_W)$, there exists a $w' \in \mathbb{B}_d(B_W)$ such that*

$$\forall s, a: \qquad \langle w', \phi(s,a)\rangle = \mathbb{E}[R(s,a)] + \mathbb{E}_{s' \sim P(s,a)} \max_{a'} \langle w, \phi(s', a')\rangle.$$

Note that the above condition implies that $Q_h^\star(s,a) = \langle w_h^\star, \phi(s,a)\rangle$ with $\|w_h^\star\|_2 \le B_W$. Thus, we can define a function class $\mathcal{F}_h = \{\langle w_h, \phi(s,a)\rangle : w_h \in \mathbb{R}^d, \|w_h\|_2 \le B_W\}$ which by inspection satisfies Assumption 1. Additionally, this model is also known to have bilinear rank at most d (Du et al., 2021). Thus, using Corollary 1 we immediately get the following guarantee:

Lemma 1. *Let $\delta \in (0,1)$, suppose the MDP is linear Bellman complete, $C_{\pi^*} < \infty$, and consider $\mathcal{F}_h$ defined above. Then, with probability $1-\delta$, Algorithm 1 finds an ϵ-suboptimal policy with total sample complexity:*

$$n = \widetilde{O}\big(B_W^2 C_{\pi^*}^2 H^4 d^2 \log(1/\delta)/\epsilon^2\big).$$

On the computational side, with $\mathcal{F}$ as in Lemma 1, the regression problem in Algorithm 1 reduces to a *least squares linear* regression with a norm constraint on the weight vector. This can be solved by convex programming with complexity scaling polynomially in the parameters (Bubeck et al., 2015).

Remark 1 (Efficient implementation). *Algorithm 1 is both statistically and computationally efficient for linear Bellman complete models. This shows a computational benefit of hybrid RL over both pure online and pure offline RL. To the best of our knowledge, no prior methods are known to be computationally efficient in this setting.*

Remark 2. *(Linear MDPs) Since linear Bellman complete models generalize linear MDPs (Yang & Wang, 2019; Jin et al., 2020), Algorithm 1 is computationally efficient for linear MDPs as well even when the action space is large or continuous. In contrast, prior algorithms rely on optimism or pessimism, which requires solving NP-hard optimization problems (particularly with $|\mathcal{A}|$ is large).*

Remark 3. *(Relative condition number) A common coverage metric in these linear MDP models is the relative condition number. In Appendix A.5, we show that our coefficient C_π is upper bounded by the relative condition number of π with respect to ν: $\mathbb{E}_{d^\pi}\|\phi\|_{\Sigma_\nu^{-1}}$, where $\Sigma_\nu = \mathbb{E}_{s,a\sim\nu}\phi(s,a)\phi^\top(s,a)$. Concretely, we have $C_\pi \leq \sqrt{\max_h \mathbb{E}_{d_h^\pi}\|\phi\|^2_{\Sigma_{\nu_h}^{-1}}}$.*

5.2 Why don't offline RL methods work?

One may wonder why do pure offline RL methods fail to learn when the transfer coefficient is bounded, and why does online access help? We illustrate with the MDP construction developed by Zhan et al. (2022); Chen & Jiang (2022), visualized in Figure 1.

Consider two MDPs $\{M_1, M_2\}$ with $H = 2$, three states $\{A, B, C\}$, two actions $\{L, R\}$ and the fixed start state A. The two MDPs have the same dynamics but different rewards. In both, actions from state B yield reward 1. In M_1, (C, R) yields reward 1 while (C, L) yields reward 1 in M_2. All other rewards are 0. In both M_1 and M_2, an optimal policy is $\pi^*(A) = L$ and $\pi^*(B) = \pi^*(C) = \text{Uniform}(\{L, R\})$. With $\mathcal{F} = \{Q_1^\star, Q_2^\star\}$ where $Q_j^\star$ is the optimal Q function for M_j, then one can easily verify that $\mathcal{F}$ satisfies Bellman completeness, for both MDPs. Finally with offline distribution ν supported on states A and B only (with no coverage on state C), we have sufficient coverage over $d^{\pi^\star}$. However, samples from ν are unable to distinguish between f_1 and f_2 or (M_1 and M_2), since state C is not supported by ν. Unfortunately, adversarial tie-breaking may result the greedy policies of f_1 and f_2 visiting state C, where we have no information about the correct action.

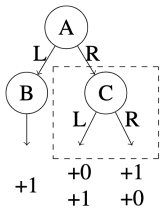

Figure 1: A hard instance for offline RL (Zhan et al., 2022, reproduced with permission)

This issue has been documented before, and in order to address it with pure offline RL, existing approaches require additional structural assumptions. For instance, Chen & Jiang (2022) assume that $Q^\star$ has a gap, which usually does not hold when action space is large or continuous. Xie et al. (2021a) assumes policy-dependent Bellman completeness for every possible policy $\pi \in \Pi$ (which is much stronger than our assumption), and Zhan et al. (2022) assumes a somewhat non-interpretable realizability assumption on some "value" function that does not obey the standard Bellman equation. In contrast, by combining offline data and online data, our approach focuses on functions that have small Bellman residual under both the offline distribution and the on-policy distributions, which together with the offline data coverage assumption, ensures near optimality. It is easy to see that the hybrid approach will succeed Figure 1.

6 Experiments

In this section we discuss empirical results comparing Hy-Q to several representative RL methods on two challenging benchmarks. Our experiments focus on answering the following questions:

1. Can Hy-Q efficiently solve problems that SOTA offline RL methods simply cannot?
2. Can Hy-Q, via the use of offline data, significantly improve the sample efficiency of online RL?
3. Does Hy-Q scale to challenging deep-RL benchmarks?

Our empirical results provide positive answers to all of these questions. To study the first two, we consider the diabolical combination lock environment (Misra et al., 2020; Zhang et al., 2022c), a synthetic environment designed to be particularly challenging for online exploration. The synthetic nature allows us to carefully control the offline data distribution to modulate the difficulty of the setup and also to compare with a provably efficient baseline (Zhang et al., 2022c). To study the third question, we consider the Montezuma's Revenge benchmark from the Arcade Learning environment, which is one of the most challenging empirical benchmarks with high-dimensional image inputs, largely due to the difficulties of exploration. Additional details are deferred to Appendix E.

Hy-Q implementation. We largely follow Algorithm 1 in our implementation for the combination lock experiment. Particularly, we use a similar function approximation to Zhang et al. (2022c), and a minibatch Adam update on Eq. (1) with the same sampling proportions as in the pseudocode. For

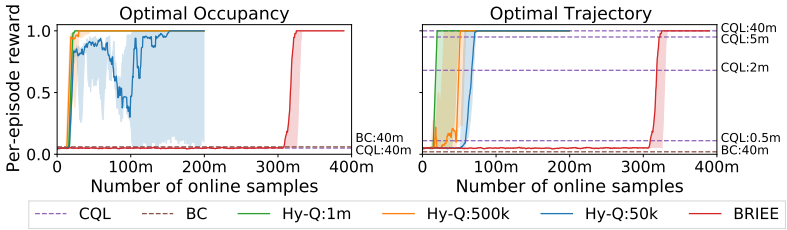

Figure 2: The learning curve for combination lock with $H = 100$. The plots show the median and 80th/20th quantile for 5 replicates. Pure offline and IL methods are visualized as dashed horizontal lines (in the left plot, CQL overlaps with BC). Note that we report the number of samples while Zhang et al. (2022c) report the number of episodes.

Montezuma's Revenge, in addition to minibatch optimization, since the horizon of the environment is not fixed, we deploy a discounted version of Hy-Q. Concretely, the target value in the Bellman error is calculated from the output of a target network, which is periodically updated, times a discount factor. We refer the readers to Appendix E for more details.

Baselines. We include representative algorithms from four categories: (1) for imitation learning we use Behavior Cloning (BC) (Bain & Sammut, 1995), (2) for offline RL we use Conservative Q-Learning (CQL) (Kumar et al., 2020), (3) for online RL we use BRIEE (Zhang et al., 2022c) for combination lock[5] and Random Network Distillation (RND) (Burda et al., 2018) for Montezuma's Revenge, and (4) as a Hybrid-RL baseline we use Deep Q-learning from Demonstrations (DQFD) (Hester et al., 2018). We note that DQFD and prior hybrid RL methods combine expert demonstrations with online interactions, but are not necessarily designed to work with general offline datasets.

Results summary. Overall, we find that Hy-Q performs favorably against all of these baselines. Compared with offline RL, imitation learning, and prior hybrid methods, Hy-Q is significantly more robust in the presence of a low quality offline data distribution. Compared with online methods, Hy-Q offers order-of-magnitude savings in the total experience.

6.1 Combination Lock

The combination lock benchmark is depicted in Figure 3 and consists of horizon $H = 100$, three latent states for each time step and 10 actions in each state. Each state has a single "good" action that advances down a chain of favorable states from which optimal reward can be obtained. A single incorrect action transitions to an absorbing chain with suboptimal value. The agent operates on high dimensional observations and must use function approximation to succeed. This is an extremely challenging problem for which many Deep RL methods are known to fail (Misra et al., 2020), in part because (uniform) random exploration only has 10^{-H} probability of obtaining the optimal reward.

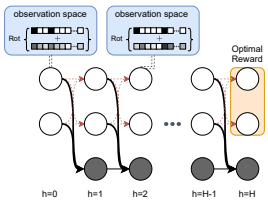

Figure 3: The combination lock (Zhang et al., 2022c), reproduced with permission.

On the other hand, the model has low bilinear rank, so we do have online RL algorithms that are provably sample-efficient: BRIEE currently obtains state of the art sample complexity. However, its sample complexity is still quite large, and we hope that Hybrid RL can address this shortcoming. We are not aware of any experiments with offline RL methods on this benchmark.

We construct two offline datasets for the experiments, both of which are derived from the optimal policy $\pi^\star$. In the **optimal trajectory** dataset we collect full trajectories by following $\pi^\star$ with ϵ-greedy exploration with $\epsilon = 1/H$. In the **optimal occupancy** dataset we collect transition tuples from the state-occupancy measure of $\pi^\star$ with random actions.[6] Both datasets have bounded concentrability coefficients (and hence transfer coefficients) with respect to $\pi^\star$, but the second dataset is much more challenging since the actions do not directly provide information about $\pi^\star$, as they do in the former.

The results are presented in Figure 2. First, we observe that Hy-Q can reliably solve the task under both offline distributions with relatively low sample complexity (500k offline samples and $\leq$ 25m

[5]We note that BRIEE is currently the state-of-the-art method for the combination lock environment. In particular, Misra et al. (2020) show that many Deep RL baselines fail in this enviroment.

[6]Formally, we sample $h \sim \text{Unif}([H])$, $s \sim d_h^{\pi^\star}$, $a \sim \text{Unif}(\mathcal{A})$, $r \sim R(s,a)$, $s' \sim P(s,a)$.

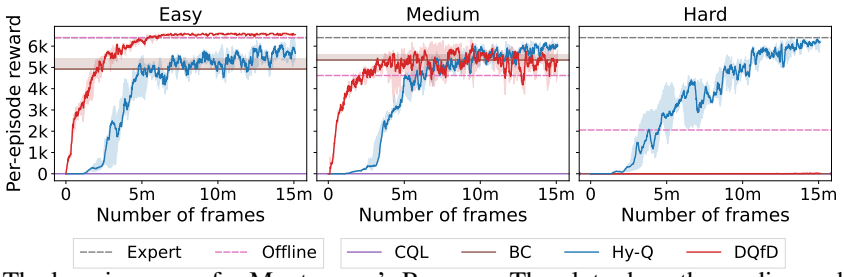

Figure 4: The learning curve for Montezuma's Revenge. The plots show the median and 80th/20th quantile for 5 replicates. Pure offline, IL methods and dataset qualities are visualized as dashed horizontal lines. "Expert" denotes V^{π^e} and "Offline" denotes the average trajectory reward in the offline dataset. The y-axis denotes the (moving) average of 100 episodes for the methods involving online interactions. Note that CQL and BC overlap on the last plot.

online samples). In comparison, BC fails completely since both datasets contain random actions. CQL can solve the task using the trajectory-based dataset with a sample complexity that is comparable to the combined sample size of Hy-Q. However, CQL fails on the occupancy-based dataset since the actions themselves are not informative. Indeed the pessimism-inducing regularizer of CQL is constant on this dataset and so the algorithm reduces to FQI. Finally, Hy-Q can solve the task with a factor of 5-10 reduction in (online and offline) samples when compared with BRIEE. This demonstrates the robustness and sample efficiency provided by hybrid RL.

6.2 Montezuma's Revenge

To answer the third question, we turn to Montezuma's Revenge, an extremely challenging image-based benchmark environment with sparse rewards. We follow the setup from Burda et al. (2018) and introduce stochasticity to the original dynamics: with probability 0.25 the environment executes the previous action instead of the current one. For offline datasets, we first train an "expert policy" π^e via RND to achieve $V^{\pi^e} \approx 6400$. We create three datasets by mixing samples from π^e with those from a random policy: the **easy dataset** contains only samples from π^e, the **medium dataset** mixes in a 80/20 proportion (80 from π^e), and the **hard dataset** mixes in a 50/50 proportion. Here we record full trajectories from both policies in the offline dataset, but measure the proportion using the number of transition tuples instead of trajectories. We provide 0.1 million offline samples for the hybrid methods, and 1 million samples for the offline and IL methods.

Results are displayed in Figure 4. CQL fails completely on all datasets. DQFD performs well on the easy dataset due to the large margin loss (Piot et al., 2014) that imitates the policies in the offline dataset. However, DQFD's performance drops as the quality of the offline dataset degrades (medium), and fails when the offline dataset is low quality (hard). We also observe that BC is a competitive baseline in the first two settings, and thus we view these problems as relatively easy to solve. Hy-Q is the only method that performs well on the hard dataset. Note that here, BC's performance is quite poor. We also include the comparison with RND in Figure 5: with only 100k offline samples from any of the three datasets, Hy-Q is over 10x more efficient in terms of online sample complexity.

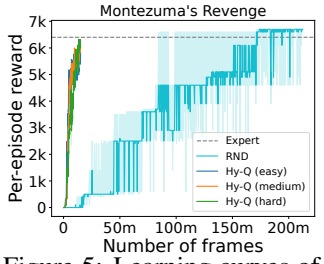

Figure 5: Learning curves of Hy-Q and RND. Metric follows Figure 4.

7 Conclusion

We demonstrate the potential of hybrid RL with Hy-Q, a simple, theoretically principled, and empirically effective algorithm. Our theoretical results showcase how Hy-Q circumvents the computational issues of pure offline or online RL, while our empirical results highlight its robustness and sample efficiency. Yet, Hy-Q is perhaps the most natural hybrid algorithm, and we are optimistic that there is much more potential to unlock from the hybrid setting. We look forward to studying this in the future.

Reproducibility Statement. For our theory results, we provide detailed proof in the Appendices. For experiments, we submit anonymous code in the supplemental materials. Our (offline) dataset can be reproduced with the attached instructions, and our results could be reproduced with the given random seeds. For more details, we include implementation, environment and computation hardware details in the Appendices, along with hyperparameters for both our method and the baselines.

References

Yasin Abbasi-Yadkori, Peter Bartlett, Kush Bhatia, Nevena Lazic, Csaba Szepesvari, and Gellért Weisz. Politex: Regret bounds for policy iteration using expert prediction. In *International Conference on Machine Learning*, 2019.

Alekh Agarwal, Nan Jiang, Sham M Kakade, and Wen Sun. *Reinforcement learning: Theory and algorithms*. 2019. URL https://rltheorybook.github.io/.

Alekh Agarwal, Sham Kakade, Akshay Krishnamurthy, and Wen Sun. Flambe: Structural complexity and representation learning of low rank MDPs. In *Advances in Neural Information Processing Systems*, 2020a.

Alekh Agarwal, Sham M Kakade, Jason D Lee, and Gaurav Mahajan. Optimality and approximation with policy gradient methods in markov decision processes. In *Conference on Learning Theory*, 2020b.

Adrià Puigdomènech Badia, Bilal Piot, Steven Kapturowski, Pablo Sprechmann, Alex Vitvitskyi, Zhaohan Daniel Guo, and Charles Blundell. Agent57: Outperforming the Atari human benchmark. In *International Conference on Machine Learning*, 2020.

James Bagnell, Sham M Kakade, Jeff Schneider, and Andrew Ng. Policy search by dynamic programming. *Advances in Neural Information Processing Systems*, 2003.

Michael Bain and Claude Sammut. A framework for behavioural cloning. In *Machine Intelligence 15*, 1995.

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

Alina Beygelzimer, John Langford, Lihong Li, Lev Reyzin, and Robert Schapire. Contextual bandit algorithms with supervised learning guarantees. In *International Conference on Artificial Intelligence and Statistics*, 2011.

Sébastien Bubeck et al. Convex optimization: Algorithms and complexity. *Foundations and Trends® in Machine Learning*, 2015.

Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. Exploration by random network distillation. In *International Conference on Learning Representations*, 2018.

Jonathan Chang, Masatoshi Uehara, Dhruv Sreenivas, Rahul Kidambi, and Wen Sun. Mitigating covariate shift in imitation learning via offline data with partial coverage. *Advances in Neural Information Processing Systems*, 2021.

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

Jinglin Chen and Nan Jiang. Offline reinforcement learning under value and density-ratio realizability: the power of gaps. *arXiv:2203.13935*, 2022.

Christoph Dann, Nan Jiang, Akshay Krishnamurthy, Alekh Agarwal, John Langford, and Robert E Schapire. On oracle-efficient PAC RL with rich observations. In *Advances in Neural Information Processing Systems*, 2018.

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

Dylan J Foster, Akshay Krishnamurthy, David Simchi-Levi, and Yunzong Xu. Offline reinforcement learning: Fundamental barriers for value function approximation. In *Conference on Learning Theory*, 2021.

Scott Fujimoto and Shixiang Shane Gu. A minimalist approach to offline reinforcement learning. In *Advances in Neural Information Processing Systems*, 2021.

Zhaohan Daniel Guo, Shantanu Thakoor, Miruna Pîslar, Bernardo Avila Pires, Florent Altché, Corentin Tallec, Alaa Saade, Daniele Calandriello, Jean-Bastien Grill, Yunhao Tang, Michal Valko, R'emi Munos, Mohammad Gheshlaghi Azar, and Bilal Piot. BYOL-explore: Exploration by bootstrapped prediction. *arXiv:2206.08332*, 2022.

Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, and Sergey Levine. Soft actor-critic algorithms and applications. *arXiv:1812.05905*, 2018.

Todd Hester, Matej Vecerik, Olivier Pietquin, Marc Lanctot, Tom Schaul, Bilal Piot, Dan Horgan, John Quan, Andrew Sendonaris, Ian Osband, John Agapiou, Joel Z. Leibo, and Audrunas Gruslys. Deep Q-learning from demonstrations. In *AAAI Conference on Artificial Intelligence*, 2018.

Zhiwei Jia, Xuanlin Li, Zhan Ling, Shuang Liu, Yiran Wu, and Hao Su. Improving policy optimization with generalist-specialist learning. In *International Conference on Machine Learning*, pp. 10104–10119. PMLR, 2022.

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

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

Chi Jin, Zhuoran Yang, Zhaoran Wang, and Michael I Jordan. Provably efficient reinforcement learning with linear function approximation. In *Conference on Learning Theory*, 2020.

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

Ying Jin, Zhuoran Yang, and Zhaoran Wang. Is pessimism provably efficient for offline RL? In *International Conference on Machine Learning*, 2021b.

Sham M Kakade. A natural policy gradient. *Advances in Neural Information Processing Systems*, 2001.

Sham M Kakade and John Langford. Approximately optimal approximate reinforcement learning. In *International Conference on Machine Learning*, 2002.

Daniel Kane, Sihan Liu, Shachar Lovett, and Gaurav Mahajan. Computational-statistical gaps in reinforcement learning. In *Conference on Learning Theory*, 2022.

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit Q-learning. *arXiv:2110.06169*, 2021.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative Q-learning for offline reinforcement learning. In *Advances in Neural Information Processing Systems*, 2020.

Tor Lattimore and Csaba Szepesvári. *Bandit algorithms*. Cambridge University Press, 2020.

Seunghyun Lee, Younggyo Seo, Kimin Lee, Pieter Abbeel, and Jinwoo Shin. Offline-to-online reinforcement learning via balanced replay and pessimistic q-ensemble. In *Conference on Robot Learning*, pp. 1702–1712. PMLR, 2022.

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

Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. In *International Conference on Learning Representations*, 2016.

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

Aditya Modi, Jinglin Chen, Akshay Krishnamurthy, Nan Jiang, and Alekh Agarwal. Model-free representation learning and exploration in low-rank MDPs. *arXiv:2102.07035*, 2021.

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

Ashvin Nair, Bob McGrew, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. Overcoming exploration in reinforcement learning with demonstrations. In *IEEE International Conference on Robotics and Automation*, 2018.

Ashvin Nair, Murtaza Dalal, Abhishek Gupta, and Sergey Levine. Accelerating online reinforcement learning with offline datasets. *arXiv:2006.09359*, 2020.

Haoyi Niu, Shubham Sharma, Yiwen Qiu, Ming Li, Guyue Zhou, Jianming Hu, and Xianyuan Zhan. When to trust your simulator: Dynamics-aware hybrid offline-and-online reinforcement learning. *arXiv preprint arXiv:2206.13464*, 2022.

Bilal Piot, Matthieu Geist, and Olivier Pietquin. Boosted bellman residual minimization handling expert demonstrations. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 2014.

Shuang Qiu, Lingxiao Wang, Chenjia Bai, Zhuoran Yang, and Zhaoran Wang. Contrastive UCB: Provably efficient contrastive self-supervised learning in online reinforcement learning. In *International Conference on Machine Learning*, 2022.

Aravind Rajeswaran, Vikash Kumar, Abhishek Gupta, Giulia Vezzani, John Schulman, Emanuel Todorov, and Sergey Levine. Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. *arXiv:1709.10087*, 2017.

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

Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay. *arXiv:1511.05952*, 2015.

Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, Timothy Lillicrap, and David Silver. Mastering Atari, Go, chess and Shogi by planning with a learned model. *Nature*, 2020.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International Conference on Machine Learning*, 2015.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv:1707.06347*, 2017.

Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge University Press, 2014.

Wen Sun, Nan Jiang, Akshay Krishnamurthy, Alekh Agarwal, and John Langford. Model-based RL in contextual decision processes: PAC bounds and exponential improvements over model-free approaches. In *Conference on learning theory*, 2019.

Ikechukwu Uchendu, Ted Xiao, Yao Lu, Banghua Zhu, Mengyuan Yan, Joséphine Simon, Matthew Bennice, Chuyuan Fu, Cong Ma, Jiantao Jiao, Sergey Levine, and Karol Hausman. Jump-start reinforcement learning. *arXiv:2204.02372*, 2022.

Masatoshi Uehara and Wen Sun. Pessimistic model-based offline reinforcement learning under partial coverage. In *International Conference on Learning Representations*, 2021.

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

Hado Van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double Q-learning. In *AAAI Conference on Artificial Intelligence*, 2016.

Mel Vecerik, Todd Hester, Jonathan Scholz, Fumin Wang, Olivier Pietquin, Bilal Piot, Nicolas Heess, Thomas Rothörl, Thomas Lampe, and Martin Riedmiller. Leveraging demonstrations for deep reinforcement learning on robotics problems with sparse rewards. *arXiv:1707.08817*, 2017.

Ruosong Wang, Yifan Wu, Ruslan Salakhutdinov, and Sham Kakade. Instabilities of offline rl with pre-trained neural representation. In *International Conference on Machine Learning*, 2021.

Ziyu Wang, Tom Schaul, Matteo Hessel, Hado Hasselt, Marc Lanctot, and Nando Freitas. Dueling network architectures for deep reinforcement learning. In *International Conference on Machine Learning*, 2016.

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

Tengyang Xie, Ching-An Cheng, Nan Jiang, Paul Mineiro, and Alekh Agarwal. Bellman-consistent pessimism for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 2021a.

Tengyang Xie, Nan Jiang, Huan Wang, Caiming Xiong, and Yu Bai. Policy finetuning: Bridging sample-efficient offline and online reinforcement learning. *Advances in Neural Information Processing Systems*, 2021b.

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

Tianhe Yu, Aviral Kumar, Rafael Rafailov, Aravind Rajeswaran, Sergey Levine, and Chelsea Finn. Combo: Conservative offline model-based policy optimization. In *Advances in Neural Information Processing Systems*, 2021.

Andrea Zanette, Alessandro Lazaric, Mykel Kochenderfer, and Emma Brunskill. Learning near optimal policies with low inherent Bellman error. In *International Conference on Machine Learning*, 2020.

Andrea Zanette, Martin J Wainwright, and Emma Brunskill. Provable benefits of actor-critic methods for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 2021.

Wenhao Zhan, Baihe Huang, Audrey Huang, Nan Jiang, and Jason Lee. Offline reinforcement learning with realizability and single-policy concentrability. In *Conference on Learning Theory*, pp. 2730–2775. PMLR, 2022.

Tianjun Zhang, Tongzheng Ren, Mengjiao Yang, Joseph Gonzalez, Dale Schuurmans, and Bo Dai. Making linear MDPs practical via contrastive representation learning. In *International Conference on Machine Learning*, 2022a.

Xuezhou Zhang, Yiding Chen, Xiaojin Zhu, and Wen Sun. Corruption-robust offline reinforcement learning. In *International Conference on Artificial Intelligence and Statistics*, 2022b.

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

A Proofs for Section 5

Throughout the appendix, we give the proofs under generalizations of the assumptions made in the main body. Firstly, we assume that Realizability and Bellman completeness only hold approximately:

Assumption 2 (Approximate Realizability and Bellman completeness)**.** *For any h, there exists a $\widetilde{Q}_h^\star \in \mathcal{F}_h$ such that $\|\widetilde{Q}_h^\star - Q_h^\star\|_\infty \leq \gamma$. Additionally, for any $f_{h+1} \in \mathcal{F}_{h+1}$, there exists a $\widehat{f}_h \in \mathcal{F}_h$ such that $\|\widehat{f}_h - \mathcal{T} f_{h+1}\|_\infty \leq \gamma$.*

Additional notation. Throughout the appendix, we define the feature covariance matrix $\Sigma_{t;h}$ as

$$\Sigma_{t;h} = \sum_{\tau=1}^{t} X_h(f^\tau)(X_h(f^\tau))^\top + \lambda \mathbb{I}. \tag{3}$$

Furthermore, given a distribution $\beta \in \Delta(\mathcal{S} \times \mathcal{A})$ and a function $f : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}$, we denote its weighted ℓ_2 norm as $\|f\|_\beta^2 := \sqrt{\mathbb{E}_{s,a\sim\beta} f^2(s,a)}$.

A.1 Supporting technical results

We first develop some useful technical results for our main proofs. The following form of Freedman's inequality is a modification of a similar inequality in (Beygelzimer et al., 2011).

Lemma 2 (Freedman's Inequality)**.** *Let $\{X_1, \ldots, X_T\}$ be a sequence of non-negative random variables adapted to the increasing filtration $\{\mathcal{G}_1, \ldots, \mathcal{G}_T\}$ of the σ-algebra $\mathcal{G}$. Further, suppose that $|X_t| \leq R$ almost surely for all $t \leq T$. Then, for any $\delta > 0$ and $\lambda \in [0, 1/2R]$, with probability at least $1-\delta$,*

$$\left|\sum_{t=1}^{T} X_t - \mathbb{E}_{t-1}[X_t]\right| \leq \lambda \sum_{t=1}^{T} \left(2R|\mathbb{E}_{t-1}[X_t]| + \mathbb{E}_{t-1}\left[X_t^2\right]\right) + \frac{\log(2/\delta)}{\lambda}$$

where $\mathbb{E}_t = \mathbb{E}[\cdot \mid \mathcal{G}_t]$ denotes the conditional expectation w.r.t. the filtration $\mathcal{G}_t$.

Proof. Define the random variable $Z_t = X_t - \mathbb{E}_{t-1}[X_t]$. Clearly, $\{Z_t\}_{t=1}^T$ is a martingale difference sequence adapted to the filtration $\{\mathcal{F}_t\}_{t=1}^T$. Furthermore, we have that for any t, $|Z_t| \leq 2R$ and that

$$\mathbb{E}_{t-1}\left[Z_t^2\right] = \mathbb{E}_{t-1}\left[\left(X_t - \mathbb{E}_{t-1}[X_t]\right)^2\right] \leq 2R|\mathbb{E}_{t-1}[X_t]| + \mathbb{E}_{t-1}\left[X_t^2\right]. \tag{4}$$

where the last inequality holds because $|X_t| \leq R$.

Using the form of Freedman's inequality in Beygelzimer et al. (2011, Lemma 9), we get that for any $\lambda \in [0, 1/2R]$,

$$\left|\sum_{t=1}^{T} Z_t\right| \leq \lambda \sum_{t=1}^{T} \mathbb{E}_{t-1}\left[Z_t^2\right] + \frac{\log(2/\delta)}{\lambda}.$$

Plugging in the form of Z_t and using (4), we get the desired statement. □

The following gives a generalization bound for least squares regression when the samples are adapted to an increasing filtration (and are not necessarily i.i.d.). The proof follows similarly to Agarwal et al. (2019, Lemma A.11).

Lemma 3. *(Least squares generalization bound) Let $R > 0$, $\delta \in (0,1)$ and let $\mathcal{H} : \mathcal{X} \mapsto [-R, R]$ be a class of real valued functions. Let $\mathcal{D} = \{(x_1, y_1), \ldots, (x_T, y_T)\}$ be a dataset of n points where $(x_T, \ldots, x_T)$ is adapted to the increasing filtration $\{\mathcal{G}_1, \ldots, \mathcal{G}_T\}$ of the σ-algebra $\mathcal{G}$, and y_t is sampled via*

$$y_t \sim h^*(x_t) + \varepsilon_t,$$

where the function h^* satisfies approximate realizability i.e.

$$\inf_{h \in \mathcal{H}} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}_{t-1}\left[\left(h^*(x) - h(x)\right)^2\right] \leq \gamma,$$

and $\{\epsilon_i\}_{i=1}^n$ *are independent random variables adapted to the filtration* $\mathcal{G}_t$*. Additionally, suppose that* $\max_t |y_t| \le R$ *and* $\max_x |h^*(x)| \le R$*. Then the least square solution* $\widehat{h} \leftarrow \operatorname{argmin}_{h\in\mathcal{H}} \sum_{t=1}^T (h(x_t) - y_t)^2$ *satisfies with probability at least* $1-\delta$*,*

$$\sum_{t=1}^T \mathbb{E}_{t-1}\Big[\big(\widehat{h}(x_t) - h^*(x_t)\big)^2\Big] \le 3\gamma T + 256R^2 \log(2|\mathcal{H}|/\delta).$$

Proof. Consider any fixed function $h \in \mathcal{H}$ and define the random variable

$$Z_t^h := (h(x_t) - y_t)^2 - (h^*(x_t) - y_t)^2$$

and note that

$$\mathbb{E}_{t-1}\big[Z_t^h\big] = \mathbb{E}_{t-1}[(h(x_t) - h^*(x_t))(h(x_t) + h^*(x_t) - 2y_i)] = \mathbb{E}_{t-1}\Big[\big(h(x_t) - h^*(x_t)\big)^2\Big], \quad (5)$$

where the last line holds because $\mathbb{E}[y_t \mid x_t] = h^*(x_t)$ and the notation $\mathbb{E}_{t-1}[\cdot] := \mathbb{E}[\cdot \mid \mathcal{F}_t]$. Furthermore, we also have that

$$\begin{aligned}\mathbb{E}_{t-1}\big[(Z_t^h)^2\big] &= \mathbb{E}_{t-1}\Big[\big(h(x_t) - h^*(x_t)\big)^2 \big(h(x_t) + h^*(x_t) - 2y_t\big)^2\Big] \\ &\le 16R^2\, \mathbb{E}_{t-1}\Big[\big(h(x_t) - h^*(x_t)\big)^2\Big]. \end{aligned} \quad (6)$$

Note that the sequence of random variables $\big\{Z_1^h, \ldots, Z_T^h\big\}$ are adapted to the filtration $\{\mathcal{G}_1, \ldots, \mathcal{G}_T\}$ and that $\big|Z_t^h\big| \le 4R^2$. Thus, using Lemma 2 we get that for any $\lambda \in [0, 1/8R^2]$ and $\delta > 0$, with probability at least $1-\delta$,

$$\begin{aligned}\left|\sum_{t=1}^T Z_t^h - \mathbb{E}_{t-1}\big[Z_t^h\big]\right| &\le \lambda \sum_{t=1}^T \Big(8R^2 \big|\mathbb{E}_{t-1}\big[Z_t^h\big]\big| + \mathbb{E}_{t-1}\Big[\big(Z_t^h\big)^2\Big]\Big) + \frac{\log(2/\delta)}{\lambda} \\ &\le 32\lambda R^2 \sum_{t=1}^T \mathbb{E}_{t-1}\Big[\big(h(x_t) - h^*(x_t)\big)^2\Big] + \frac{\log(2/\delta)}{\lambda},\end{aligned}$$

where the last inequality uses (5) and (6). Setting $\lambda = 1/64R^2$ in the above, and taking a union bound over h, we get that for any $h \in \mathcal{H}$ and $\delta > 0$, with probability at least $1-\delta$,

$$\left|\sum_{t=1}^T Z_t^h - \mathbb{E}_{t-1}\big[Z_t^h\big]\right| \le \frac{1}{2}\sum_{t=1}^T \mathbb{E}_{t-1}\Big[\big(h(x_t) - h^*(x_t)\big)^2\Big] + 64R^2 \log(2|\mathcal{H}|/\delta).$$

Rearranging the terms and using (5) in the above implies that,

$$\sum_{t=1}^T Z_t^h \le \frac{3}{2}\sum_{t=1}^T \mathbb{E}_{t-1}\Big[\big(h(x_t) - h^*(x_t)\big)^2\Big] + 64R^2 \log(2|\mathcal{H}|/\delta)$$

and

$$\sum_{t=1}^T \mathbb{E}_{t-1}\Big[\big(h(x_t) - h^*(x_t)\big)^2\Big] \le 2\sum_{t=1}^T Z_t^h + 128R^2 \log(2|\mathcal{H}|/\delta). \quad (7)$$

For the rest of the proof, we condition on the event that (7) holds for all $h \in \mathcal{H}$.

Define the function $\widetilde{h} := \operatorname{argmin}_{h\in\mathcal{H}} \sum_{t=1}^T \mathbb{E}_{t-1}\big[(h(x_t) - h^*(x_t))^2\big]$. Using (7), we get that

$$\begin{aligned}\sum_{t=1}^T Z_t^{\widetilde{h}} &\le \frac{3}{2}\sum_{t=1}^T \mathbb{E}_{t-1}\Big[\big(\widetilde{h}(x_t) - h^*(x_t)\big)^2\Big] + 64R^2 \log(2|\mathcal{H}|/\delta) \\ &\le \frac{3}{2}\gamma T + 64R^2 \log(2|\mathcal{H}|/\delta),\end{aligned}$$

where the last inequality follows from the approximate realizability assumption. Let $\widehat{h}$ denote the least squares solution on dataset $\{(x_t, y_t)\}_{t \le T}$. By definition, we have that

$$\sum_{t=1}^{T} Z_t^{\widehat{h}} = (\widehat{h}(x_t) - y_t)^2 - (h^*(x_t) - y_t)^2 \le (\widetilde{h}(x_t) - y_t)^2 - (h^*(x_t) - y_t)^2 = \sum_{t=1}^{T} Z_t^{\widetilde{h}}.$$

Combining the above two relations, we get that

$$\sum_{t=1}^{T} Z_t^{\widehat{h}} \le \frac{3}{2}\gamma T + 64R^2 \log(2|\mathcal{H}|/\delta). \tag{8}$$

Finally, using (7) for the function $\widehat{h}$, we get that

$$\begin{aligned}\sum_{t=1}^{T} \mathbb{E}_{t-1}\left[(\widehat{h}(x_t) - h^*(x_t))^2\right] &\le 2\sum_{t=1}^{T} Z_t^{\widehat{h}} + 128R^2 \log(2|\mathcal{H}|/\delta) \\ &\le 3\gamma T + 256R^2 \log(2|\mathcal{H}|/\delta),\end{aligned}$$

where the last inequality uses the relation (8). □

Lemma 4 (Performance difference lemma)**.** *For any function $f = (f_0, \ldots, f_{H-1})$ where $f_h : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}$ and $h \in [H-1]$, we have*

$$\mathbb{E}_{s\sim d_0}[\max_a f_0(s,a) - V_0^{\pi^f}(s)] \le \sum_{h=0}^{H-1} \left|\mathbb{E}_{s,a\sim d_h^{\pi^f}}[f_h(s,a) - \mathcal{T}f_{h+1}(s,a)]\right|,$$

where we define $f_H(s,a) = 0$ for all s, a.

Proof. We will prove using induction that for all $h \in [H]$,

$$\mathbb{E}_{s\sim d_h^{\pi^f}}[\max_a f_h(s,a) - V_h^{\pi^f}(s)] = \sum_{j=h}^{H-1} \mathbb{E}_{s,a\sim d_j^{\pi^f}}[f_j(s,a) - \mathcal{T}f_{j+1}(s,a)], \tag{9}$$

where we additionally define $V_H^{\pi^f}(s) = 0$ for any $s \in \mathcal{S}$.

For the base case ($h = H$), we note that $f_H(s,a) = 0$ and $V_H^{\pi^f} = 0$. Thus, $\mathbb{E}_{s\sim d_H^{\pi^f}}[\max_a f_H(s,a) - V_H^{\pi^f}(s)] = 0$ and (9) holds.

For the induction step, suppose (9) holds for step $h+1$. We note that

$$\begin{aligned}\mathbb{E}_{s\sim d_h^{\pi^f}}[\max_a f_h(s,a) - V_h^{\pi^f}(s)] &= \mathbb{E}_{s\sim d_h^{\pi^f}, a\sim\pi^f(s)}[f_h(s,a) - V_h^{\pi^f}(s)] \\ &= \mathbb{E}_{s,a\sim d_h^{\pi^f}}[f_h(s,a) - \mathcal{T}f_{h+1}(s,a)] \\ &\quad + \mathbb{E}_{s,a\sim d_h^{\pi^f}}[\mathcal{T}f_{h+1}(s,a) - V_h^{\pi^f}(s)],\end{aligned} \tag{10}$$

where the above equalities follows from the fact that policy π^f is defined such that at step h, $\pi_f(s) = \text{argmax}_a f_h(s,a)$ for any $s \in \mathcal{S}$. We further note that

$$\begin{aligned}\mathbb{E}_{s,a\sim d_h^{\pi^f}}[\mathcal{T}f_{h+1}(s,a) - V_h^{\pi^f}(s)] &= \mathbb{E}_{s,a\sim d_h^{\pi^f}, s'\sim P(s,a)}[r(s,a) + \max_{a'} f_{h+1}(s',a') - V_h^{\pi^f}(s)] \\ &= \mathbb{E}_{s,a\sim d_h^{\pi^f}, s'\sim P(s,a)}[r(s,a) + \max_{a'} f_{h+1}(s',a') - r(s,a) - V_{h+1}^{\pi^f}(s')] \\ &= \mathbb{E}_{s,a\sim d_h^{\pi^f}, s'\sim P(s,a)}[\max_{a'} f_{h+1}(s',a') - V_{h+1}^{\pi^f}(s')] \\ &= \mathbb{E}_{s'\sim d_h^{\pi^f}}[\max_{a'} f_{h+1}(s',a') - V_{h+1}^{\pi^f}(s')] \\ &= \mathbb{E}_{s'\sim d_{h+1}^{\pi^f}}[\max_a f_{h+1}(s,a) - V_{h+1}^{\pi^f}(s)]\end{aligned} \tag{11}$$

where the first line simply follows by expanding $\mathcal{T} f_{h+1}(s,a)$, and the second equality follows by expanding $V_h^{\pi^f}(s)$ and noting that $a = \pi_f(s) = \operatorname{argmax}_a f_h(s,a)$.

Plugging in (11) in (10), we get

$$\begin{aligned}\mathbb{E}_{s\sim d_h^{\pi^f}}[\max_a f_h(s,a) - V_h^{\pi^f}(s)] &= \mathbb{E}_{s,a\sim d_h^{\pi^f}}[f_h(s,a) - \mathcal{T} f_{h+1}(s,a)] \\ &\quad + \mathbb{E}_{s\sim s'\sim d_{h+1}^{\pi^d}}[\max_a f_{h+1}(s,a) - V_{h+1}^{\pi^f}(s)] \\ &= \sum_{j=h}^{H-1} \mathbb{E}_{s,a\sim d_j^{\pi^f}}[f_j(s,a) - \mathcal{T} f_{j+1}(s,a)],\end{aligned}$$

where the last line holds due to induction hypothesis for step $h+1$.

Thus, (9) holds for all $h \in [H]$, and in particular for $h = 0$. An application of Triangle inequality gives the desired result. □

Lemma 5 (Pseudo-optimism). *Let $\pi^e = (\pi_0^e, \ldots, \pi_{H-1}^e)$ be a comparator policy, and consider any value function $f = (f_0, \ldots, f_{H-1})$ where $f_h : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}$. Then,*

$$\mathbb{E}_{s\sim d_0}\left[V_0^{\pi^e}(s) - \max_a f_0(s,a)\right] \le \sum_{i=0}^{H-1} \mathbb{E}_{s,a\sim d_i^{\pi_e}}[\mathcal{T} f_{i+1}(s,a) - f_i(s,a)],$$

where we defined $f_H(s,a) = 0$ for all s, a.

Proof. The proof is similar to the proof of Lemma 4, and we will prove via induction that for all $h \in [H]$,

$$\mathbb{E}_{s\sim d_h^{\pi^e}}\left[V_h^{\pi^e}(s) - \max_a f_h(s,a)\right] \le \sum_{i=h}^{H-1} \mathbb{E}_{s,a\sim d_i^{\pi_e}}[\mathcal{T} f_{i+1}(s,a) - f_i(s,a)]. \tag{12}$$

The base case for $h = H$ follows by definition as $V_H^{\pi_e}(s) = 0$ and $f_H(s,a) = 0$ for all s, a. We next show the induction step.

Suppose (12) holds for time step $h+1 \le H$. In the following, we will show that (12) also holds for time step h. Note that

$$\begin{aligned}\mathbb{E}_{s\sim d_h^{\pi^e}}[V_h^{\pi^e}(s) - \max_a f_h(s,a)] &\le \mathbb{E}_{s,a\sim d_h^{\pi^e}}[V_h^{\pi^e}(s) - f_h(s,a)] \\ &= \mathbb{E}_{s,a\sim d_h^{\pi^e}}[V_h^{\pi^e}(s) - \mathcal{T} f_{h+1}(s,a)] \\ &\quad + \mathbb{E}_{s,a\sim d_h^{\pi^e}}[\mathcal{T} f_{h+1}(s,a) - f_h(s,a)]. \end{aligned} \tag{13}$$

where the inequality in the first line above follows by replacing $\max_a$ by $a \sim \pi_h^e(s)$. For the first term in the above, note that

$$\begin{aligned}\mathbb{E}_{s,a\sim d_h^{\pi^e}}[V_h^{\pi^e}(s) - \mathcal{T} f_{h+1}(s,a)] &= \mathbb{E}_{s,a\sim d_h^{\pi^e}, s'\sim P(s,a)}[r(s,a) + V_{h+1}^{\pi^e}(s')] \\ &\quad - \mathbb{E}_{s,a\sim d_h^{\pi^e}, s'\sim P(s,a)}[r(s,a) + \max_{a'} f_{h+1}(s',a')] \\ &= \mathbb{E}_{s,a\sim d_h^{\pi^e}, s'\sim P(s,a)}[V_{h+1}^{\pi^e}(s') - \max_{a'} f_{h+1}(s',a')] \\ &= \mathbb{E}_{s'\sim d_{h+1}^{\pi^e}}[V_{h+1}^{\pi^e}(s') - \max_{a'} f_{h+1}(s',a')] \\ &= \mathbb{E}_{s\sim d_{h+1}^{\pi^e}}[V_{h+1}^{\pi^e}(s) - \max_a f_{h+1}(s,a)], \end{aligned} \tag{14}$$

where the first equality follows by expanding $V_h^{\pi^e}(s)$ and $\mathcal{T} f_{h+1}(s,a)$. The second equality is due to linearity of expectations. Using (14) in (13), we get

$$\begin{aligned}\mathbb{E}_{s\sim d_h^{\pi^e}}[V_h^{\pi^e}(s) - \max_a f_h(s,a)] &\le \mathbb{E}_{s\sim d_{h+1}^{\pi^e}}[V_{h+1}^{\pi^e}(s) - \max_a f_{h+1}(s,a)] \\ &\quad + \mathbb{E}_{s,a\sim d_h^{\pi^e}}[\mathcal{T} f_{h+1}(s,a) - f_h(s,a)]\end{aligned}$$

$$\leq \sum_{i=h}^{H-1} \mathbb{E}_{s,a\sim d_i^{\pi_e}}[\mathcal{T} f_{i+1}(s,a) - f_i(s,a)],$$

where the last line holds due to induction hypothesis for time step $h+1$. This completes the induction step showing the (12) holds for all $h \in [H]$.

The claimed bound follows by invoking (12) with $h = 0$ and noting that $d_0^{\pi_e} = d_0$. □

The following result directly follows from the elliptical potential lemma (Lattimore & Szepesvári, 2020, Lemma 19.4).

Lemma 6. *Let $X_h(f^1), \ldots, X_h(f^T) \in \mathbb{R}^d$ be a sequence of vectors with $\|X_h(f^t)\| \leq B_X < \infty$ for all $t \leq T$. Then,*

$$\sum_{t=1}^{T} \|X_h(f^t)\|_{\Sigma_{t-1;h}^{-1}} \leq \sqrt{2dT \log\left(1 + \frac{TB_X^2}{\lambda d}\right)},$$

where the matrix $\Sigma_{t;h} := \sum_{\tau=1}^{t} X_h(f^\tau) X_h(f^\tau)^\top + \lambda \mathbb{I}$ for $t \in [T]$ and $\lambda \geq B_X^2$.

Proof. Since $\lambda \geq B_X^2$, we have that

$$\|X_h(f^t)\|^2_{\Sigma_{t-1;h}^{-1}} \leq \frac{1}{\lambda}\|X_h(f^t)\|^2 \leq 1.$$

Thus, using elliptical potential lemma (Lattimore & Szepesvári, 2020, Lemma 19.4), we get that

$$\sum_{t=1}^{T} \|X_h(f^t)\|^2_{\Sigma_{t-1;h}^{-1}} \leq 2d \log\left(1 + \frac{TB_X^2}{\lambda d}\right).$$

The desired bound follows from Jensen's inequality which implies that

$$\sum_{t=1}^{T} \|X_h(f^t)\|_{\Sigma_{t-1;h}^{-1}} \leq \sqrt{T \cdot \sum_{t=1}^{T} \|X_h(f^t)\|^2_{\Sigma_{t-1;h}^{-1}}} \leq \sqrt{2Td \log\left(1 + \frac{TB_X^2}{\lambda d}\right)}.$$

□

A.2 Proof of Theorem 1

Before delving into the proof, we first state that following generalization bound for FQI.

Lemma 7 (Bellman error bound for FQI)**.** *Let $\delta \in (0,1)$ and let for $h \in [H-1]$ and $t \in [T]$, f_h^{t+1} be the estimated value function for time step h computed via least square regression using samples in the dataset $\left(\mathcal{D}_h^\nu, \mathcal{D}_h^1, \ldots, \mathcal{D}_h^t\right)$ in (1) in the iteration t of Algorithm 1. Then, with probability at least $1-\delta$, for any $h \in [H-1]$ and $t \in [T]$,*

$$\left\|f_h^{t+1} - \mathcal{T} f_{h+1}^{t+1}\right\|^2_{2,\nu_h} \leq \frac{1}{m_{\text{off}}}\left(2\gamma T + 256 V_{\max}^2 \log(2HT|\mathcal{F}|/\delta)\right) =: \Delta_{\text{off}},$$

and

$$\sum_{\tau=1}^{t}\left\|f_h^{t+1} - \mathcal{T} f_{h+1}^{t+1}\right\|^2_{2,\mu_h^\tau} \leq \frac{1}{m_{\text{on}}}\left(2\gamma T + 256 V_{\max}^2 \log(2HT|\mathcal{F}|/\delta)\right) =: \Delta_{\text{on}},$$

where ν_h denotes the offline data distribution at time h, and the distribution $\mu_h^\tau \in \Delta(s,a)$ is defined such that $s \sim d_h^{\pi^\tau}$ and $a \sim \pi_{\text{est}}(s; f^t)$.

Proof. Fix $t \in [T]$, $h \in [H-1]$ and $f_{h+1}^{t+1} \in \mathcal{F}_{h+1}$ and consider the regression problem

$$f_h^{t+1} \leftarrow \operatorname*{argmin}_{f \in \mathcal{F}_h}\left\{\widehat{\mathbb{E}}_{\mathcal{D}_h^\nu}(f(s,a) - r - \max_{a'} f_{h+1}^{t+1}(s',a'))^2 + \sum_{\tau=1}^{t} \widehat{\mathbb{E}}_{\mathcal{D}_h^\tau}(f(s,a) - r - \max_{a'} f_{h+1}^{t+1}(s',a'))^2\right\}$$

The above can be thought of as regression problem on dataset $\mathcal{D}$ consisting of $n = m_{\text{off}} + t \cdot m_{\text{on}}$ samples $\{(x_i, y_i)\}_{i \le n}$ where

$$x_i = (s_h^i, a_h^i) \qquad \text{and} \qquad y^i = r^i + \max_a f_{h+1}^{t+1}(s_{h+1}^i, a).$$

In particular, we define $\mathcal{D}$ such that the first m_{off} samples $\{(x_i, y_i)\}_{i \le m_{\text{off}}} = \mathcal{D}_h^\nu$, the next m_{on} samples $\{(x_i, y_i)\}_{i=m_{\text{off}}+1}^{m_{\text{off}}+m_{\text{on}}} = \mathcal{D}_h^1$, and so on where the samples $\{(x_i, y_i)\}_{i=m_{\text{off}}+\tau m_{\text{on}}}^{m_{\text{off}}+\tau m_{\text{on}}} = \mathcal{D}_h^\tau$. Note that: (a) for any sample $(x = (s_h, a_h), y = (r + \max_a f_{h+1}^{t+1}(s_{h+1}, a)))$ in $\mathcal{D}$, we have that

$$\begin{aligned}\mathbb{E}[y \mid x] &= \mathbb{E}_{s_{h+1} \sim P(s_h, a_h), r \sim R(s_h, a_h)}\left[r + \max_a f_{h+1}^{t+1}(s_{h+1}, a)\right] \\ &= \mathcal{T} f_{h+1}^{t+1}(s_h, a_h) \le g(s_h, a_h) + \gamma.\end{aligned}$$

where the last line holds since the approximate bellman completeness assumption implies existence of such a function g, (b) for any sample, $|y| \le B$ and $f(s, a) \le B$ for all s, a, (c) our construction of $\mathcal{D}$ implies that the samples $\{x_t, y_t\}_{t \le T}$ are adapted to an increasing filtration $\mathcal{G}_t$ (given by the natural filtration induced by the previous random variables), finally (d) the samples in $\mathcal{D}_h^\nu$ are drawn from the offline distribution ν_h, and the samples in $\mathcal{D}_h^\tau$ are drawn such that $s_h \sim d_h^{\pi^t}$ and $a_h \sim \pi_{\text{est}}(s_h; f^t)$. Thus, using Lemma 3, we get that the least square regression solution f_h^{t+1} satisfies

$$\sum_{i=1}^n \mathbb{E}_{i-1}\left[(f_h^{t+1}(s^i, a^i) - \mathcal{T} f_{h+1}^{t+1}(s^i, a^i))^2\right] \le 3\gamma T + 256 V_{\max}^2 \log(2|\mathcal{F}|/\delta).$$

Using the property-(d) in the above, we get that

$$m_{\text{off}} \cdot \left\|f_h^{t+1} - \mathcal{T} f_{h+1}^{t+1}\right\|_{2,\nu_h}^2 + m_{\text{on}} \cdot \sum_{\tau=1}^t \left\|f_h^{t+1} - \mathcal{T} f_{h+1}^{t+1}\right\|_{2,\mu_h^\tau}^2 \le 3\gamma T + 256 V_{\max}^2 \log(2|\mathcal{F}|/\delta),$$

where the distribution $\mu_h^\tau \in \Delta(s, a)$ is defined by sampling $s \sim d_h^{\pi^\tau}$ and $a \sim \pi_{\text{est}}(s; f^t)$. Taking a union bound over $h \in [H-1]$ and $t \in [T]$, and bounding each term separately, gives the desired statement. □

We next note a change in distribution lemma which allows us to bound expected bellman error under the (s, a) distribution generated by f^t in terms of the expected square bellman error w.r.t. the previous policies data distribution, which is further controlled using regression.

Lemma 8. *For any $t \ge 0$ and $h \in [H-1]$, we have*

$$\left|\left\langle W_h(f^t), X_h(f^t)\right\rangle\right| \le \|X_h(f^t)\|_{\Sigma_{t-1;h}^{-1}} \sqrt{\sum_{i=1}^{t-1} \mathbb{E}_{s,a \sim d_h^{f^i}}\left[\left(f_h^t - \mathcal{T} f_{h+1}^t\right)\right]^2 + \lambda B_W^2}.$$

where Σ_{t-1}^{-1} is defined in (3).

Proof. Using Cauchy-Schwarz inequality, we get that

$$\begin{aligned}\left|\left\langle W_h(f^t), X_h(f^t)\right\rangle\right| &\le \|X_h(f^t)\|_{\Sigma_{t-1;h}^{-1}} \|W_h(f^t)\|_{\Sigma_{t-1}} \\ &= \|X_h(f^t)\|_{\Sigma_{t-1;h}^{-1}} \sqrt{(W_h(f^t))^\top \Sigma_{t-1} W_h(f^t)} \\ &= \|X_h(f^t)\|_{\Sigma_{t-1;h}^{-1}} \sqrt{(W_h(f^t))^\top \left(\sum_{i=1}^{t-1} X_h(f^i) X_h(f^i)^\top + \lambda \mathbb{I}\right) W_h(f^t)} \\ &= \|X_h(f^t)\|_{\Sigma_{t-1;h}^{-1}} \sqrt{\sum_{i=1}^{t-1} \left|\langle W_h(f^t), X_h(f^i)\rangle\right|^2 + \lambda \|W_h(f^t)\|^2} \\ &\le \|X_h(f^t)\|_{\Sigma_{t-1;h}^{-1}} \sqrt{\sum_{i=1}^{t-1} \left|\langle W_h(f^t), X_h(f^i)\rangle\right|^2 + \lambda B_W^2}\end{aligned}$$

$$\leq \|X_h(f^t)\|_{\Sigma_{t-1;h}^{-1}} \sqrt{\sum_{i=1}^{t-1} \mathbb{E}_{s,a\sim d_h^{f^i}} \left[\left(f_h^t - \mathcal{T} f_{h+1}^t\right)\right]^2 + \lambda B_W^2}$$

where the inequality in the second last line holds by plugging in the bound on $\|W_h(f^t)\|$, and the last line holds by using Definition 2 which implies that

$$\left|\langle W_h(f^t), X_h(f^i)\rangle\right|^2 = \left(\mathbb{E}_{s,a\sim d_h^{f^i}}\left[f_h^t - \mathcal{T} f_{h+1}^t\right]\right)^2 \leq \mathbb{E}_{s,a\sim d_h^{f^i}}\left[\left(f_h^t - \mathcal{T} f_{h+1}^t\right)^2\right],$$

where the last inequality is due to Jensen's inequality. □

We now have all the tools to prove Theorem 1. We first restate the bound with the exact dependence on problem parameters.

Theorem (Theorem 1 restated)**.** *Let $m_{\text{off}} = HT/d$ and $m_{\text{on}} = 2H^2$. Then, with probability at least $1-\delta$, the total complexity of Algorithm 1 is bounded as*

$$\sum_{t=1}^{T} V_0^{\pi^e} - V_0^{\pi^{f^t}} = O\left(B \max\{C_{\pi^e}, 1\} \sqrt{dT \cdot \log\left(1 + \frac{T}{d}\right) \log\left(\frac{2HT|\mathcal{F}|}{\delta}\right)}\right).$$

Proof of Theorem 1. Let π^e be any comparator policy with bounded transfer coefficient i.e.

$$C_{\pi^e} := \max\left\{0,\ \max_{f\in\mathcal{F}} \frac{\sum_{h=0}^{H-1} \mathbb{E}_{s,a\sim d_h^{\pi^e}}\left[f_h(s,a) - \mathcal{T} f_{h+1}(s,a)\right]}{\sqrt{\sum_{h=0}^{H-1} \mathbb{E}_{s,a\sim \nu_h}\left[\left(f_h(s,a) - \mathcal{T} f_{h+1}(s,a)\right)^2\right]}}\right\} < \infty. \tag{15}$$

We start by noting that

$$\begin{aligned}
\sum_{t=1}^{T} V_0^{\pi^e} - V_0^{\pi^{f^t}} &= \sum_{t=1}^{T} \mathbb{E}_{s\sim d_0}\left[V_0^{\pi^e}(s) - V_0^{\pi^{f^t}}(s)\right] \\
&= \sum_{t=1}^{T} \mathbb{E}_{s\sim d_0}\left[V_0^{\pi^e}(s) - \max_a f_0^t(s,a)\right] + \sum_{t=1}^{T} \mathbb{E}_{s\sim d_0}\left[\max_a f_0^t(s,a) - V_0^{\pi^{f^t}}(s)\right].
\end{aligned} \tag{16}$$

For the first term in the right hand side of (16), note that using Lemma 5 for each f_t for $1 \leq t \leq T$, we get

$$\begin{aligned}
\sum_{t=1}^{T} \mathbb{E}_{s\sim d_0}\left[V_0^{\pi^e}(s) - \max_a f_0^t(s,a)\right] &\leq \sum_{t=1}^{T} \sum_{h=0}^{H-1} \mathbb{E}_{s,a\sim d_h^{\pi^e}}[\mathcal{T} f_{h+1}^t(s,a) - f_h^t(s,a)] \\
&\leq \sum_{t=1}^{T} C_{\pi^e} \cdot \sqrt{\sum_{h=0}^{H-1} \mathbb{E}_{s,a\sim\nu_h}\left[\left(f_h^t(s,a) - \mathcal{T} f_{h+1}^t(s,a)\right)^2\right]} \\
&= TC_{\pi^e} \cdot \sqrt{H \cdot \Delta_{\text{off}}},
\end{aligned} \tag{17}$$

where the second inequality follows from plugging in the definition of C_{π_e} in (15). The last line follows from Lemma 7.

For the second term in (16), using Lemma 4 for each f_t for $1 \leq t \leq T$, we get

$$\begin{aligned}
\sum_{t=1}^{T} \mathbb{E}_{s\sim d_0}\left[\max_a f_0^t(s,a) - V_0^{\pi^{f^t}}(s)\right] &\leq \sum_{t=1}^{T} \sum_{h=0}^{H-1} \left|\mathbb{E}_{s,a\sim d_h^{\pi^{f^t}}}\left[f_h^t(s,a) - \mathcal{T}_h f_{h+1}^t(s,a)\right]\right| \\
&= \sum_{t=1}^{T} \sum_{h=0}^{H-1} \left|\langle X_h(f^t), W_h(f^t)\rangle\right|
\end{aligned} \tag{18}$$

$$\leq \sum_{t=1}^{T}\sum_{h=0}^{H-1} \|X_h(f^t)\|_{\Sigma_{t-1;h}^{-1}} \sqrt{\Delta_{\text{on}} + \lambda B_W^2},$$

where the second line follows from Definition 2, the third line follows from Lemma 8 and by plugging in the bound in Lemma 7. Using the bound in Lemma 6 in the above, we get that

$$\sum_{t=1}^{T} \mathbb{E}_{s\sim d_0}\left[\max_a f_0^t(s,a) - V_0^{\pi^{f^t}}(s)\right] \leq \sqrt{2dH^2 \log\left(1 + \frac{TB_X^2}{\lambda d}\right) \cdot (\Delta_{\text{on}} + \lambda B_W^2) \cdot T}$$
$$\leq \sqrt{2dH^2 \log\left(1 + \frac{T}{d}\right) \cdot (\Delta_{\text{on}} + B_X^2 B_W^2) \cdot T}, \tag{19}$$

where the second line follows by plugging in $\lambda = B_X^2$.

Combining the bound (17) and (19), we get that

$$\sum_{t=1}^{T} V_0^{\pi^e} - V_0^{\pi^{f^t}} \leq TC_{\pi^e} \cdot \sqrt{H \cdot \Delta_{\text{off}}} + \sqrt{2dH^2 \log\left(1 + \frac{T}{d}\right) \cdot (\Delta_{\text{on}} + B_X^2 B_W^2) \cdot T}$$

Plugging in the values of Δ_{on} and Δ_{off} in the above, we get that

$$\sum_{t=1}^{T} V_0^{\pi^e} - V_0^{\pi^{f^t}} \leq TC_{\pi^e} \cdot \sqrt{H \cdot \left(\frac{1}{m_{\text{off}}}\left(2\gamma T + 256 V_{\max}^2 \log\left(\frac{2HT|\mathcal{F}|}{\delta}\right)\right)\right)}$$
$$+$$
$$\sqrt{2dH^2 \log\left(1 + \frac{T}{d}\right) \cdot \left(\frac{1}{m_{\text{on}}}\left(2\gamma T + 256 V_{\max}^2 \log\left(\frac{2HT|\mathcal{F}|}{\delta}\right)\right) + B_X^2 B_W^2\right) \cdot T}.$$

Setting $\gamma = 0$ (under exact Bellman completeness), we get that

$$\sum_{t=1}^{T} V_0^{\pi^e} - V_0^{\pi^{f^t}} \leq 16BC_{\pi^e} T\sqrt{\frac{H}{m_{\text{off}}} \log\left(\frac{2HT|\mathcal{F}|}{\delta}\right)} + 16B\sqrt{\frac{2dH^2}{m_{\text{on}}} \log\left(1 + \frac{T}{d}\right) \log\left(\frac{2HT|\mathcal{F}|}{\delta}\right)}$$
$$+ HB_X B_W \sqrt{2dT \log\left(1 + \frac{T}{d}\right)}.$$

Setting $m_{\text{off}} = HT/d$ and $m_{\text{on}} = 2H^2$ in the above gives the total complexity bound

$$\sum_{t=1}^{T} V_0^{\pi^e} - V_0^{\pi^{f^t}} = O\left(B \max\{C_{\pi^e}, 1\} \sqrt{dT \cdot \log\left(1 + \frac{T}{d}\right) \log\left(\frac{2HT|\mathcal{F}|}{\delta}\right)}\right). \tag{20}$$

□

Proof of Corollary 1. We next convert the above total complexity bound into sample complexity via a standard online-to-batch conversion. Setting $\pi^e = \pi^*$ in (20) and defining the policy $\widehat{\pi} = \text{Uniform}\left(\{\pi^1, \ldots, \pi^T\}\right)$, we get that

$$\mathbb{E}\left[V^{\pi^*} - V^{\widehat{\pi}}\right] = \frac{1}{T}\left(\sum_{t=1}^{T} V_0^{\pi^*} - V_0^{\pi^t}\right)$$
$$= O\left(B \max\{C_{\pi^*}, 1\} \sqrt{\frac{d}{T} \cdot \log\left(1 + \frac{T}{d}\right) \log\left(\frac{2HT|\mathcal{F}|}{\delta}\right)}\right).$$

Thus, we get that for $T \geq \widetilde{O}\left(\frac{V_{\max}^2 \max\left\{C_{\pi^*}^2, 1\right\} d \log(2HT|\mathcal{F}|/\delta)}{\epsilon^2}\right)$, we get that

$$\mathbb{E}\left[V^{\pi^*} - V^{\widehat{\pi}}\right] \leq \epsilon.$$

In these T iterations, the total number of offline samples used is

$$m_{\text{off}} = \frac{HT}{d} = \widetilde{O}\left(\frac{V_{\max}^2 \max\{C_{\pi^*}^2, 1\} H \log(2HT|\mathcal{F}|/\delta)}{\epsilon^2}\right).$$

Furthermore, the number of offline samples collected is

$$m_{\text{on}} \cdot H \cdot T = \widetilde{O}\left(\frac{V_{\max}^2 \max\{C_{\pi^*}^2, 1\} H^3 d \log(2HT|\mathcal{F}|/\delta)}{\epsilon^2}\right).$$

□

A.3 V-type vs Q-type Bilinear Rank

Our previous result focus on the Q-type bilinear model. Here we provide the V-type Bilinear rank definition. This V-type Bilinear rank definition is basically the same as the low Bellman rank model proposed by Jiang et al. (2017).

Definition 4 (V-type Bilinear model)**.** *Consider any pair of functions (f, g) with $f, g \in \mathcal{F}$. Denote the greedy policy of f as $\pi^f = \{\pi_h^f := \text{argmax}_a f_h(s,a), \forall h\}$. We say that the MDP together with the function $\mathcal{F}$ admits a bilinear structure of rank d if for any $h \in [H-1]$, there exist two (unknown) mappings $X_h : \mathcal{F} \mapsto \mathbb{R}^d$ and $W_h : \mathcal{F} \mapsto \mathbb{R}^d$ with $\max_f \|X_h(f)\|_2 \leq B_X$ and $\max_f \|W_h(f)\|_2 \leq B_W$, such that:*

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

Note that different from the Q-type definition, here the action a is taken from the greedy policy with respect to g. This way $\max_a g(s,a)$ can serve as an approximation of $V^\star$ – thus the name of V-type.

To make Hy-Q work for the V-type Bilinear model, we only need to make slight change on the data collection process, i.e., when we collect online batch $\mathcal{D}_h$, we sample $s \sim d_h^{\pi^t}, a \sim \text{Uniform}(\mathcal{A}), s' \sim P(\cdot|s,a)$. Namely the action is taken uniformly randomly here. We skip the detailed discussion of the V-type model and its sample complexity here. We refer readers to Du et al. (2021); Jin et al. (2021a) for more detailed discussion.

A.4 Low-rank MDP

In this section, we briefly introduce the low-rank MDP model which is captured by the V-type bilinear model. (Du et al., 2021). Unlike the linear MDP model discussed in Section 5.1, low-rank MDP does not assume the feature ϕ is known a priori.

Definition 5 (Low-rank MDP)**.** *A MDP is called low-rank MDP if there exists $\mu^\star : \mathcal{S} \mapsto \mathbb{R}^d, \phi^\star : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}^d$, such that the transition dynamics $P(s'|s,a) = \mu^\star(s')^\top \phi^\star(s,a)$ for all s, a, s'. We additionally assume that we are given a realizable representation class Φ such that $\phi^\star \in \Phi$, and that $\sup_{s,a} \|\phi^\star(s,a)\|_2 \leq 1$, and $\|f^\top \mu^\star\|_2 \leq \sqrt{d}$ for any $f : \mathcal{S} \mapsto [-1,1]$.*

Consider the function class $\mathcal{F}_h = \{w^\top \phi(s,a) : \phi \in \Phi, w \in \mathbb{B}_d(B_W)\}$, and through the bilinear decomposition we have that $B_W \leq 2\sqrt{d}$. By inspection, we know that this function class satisfies Assumption 1. Furthermore, it is well known that the low rank MDP model has V-type bilinear rank of at most d (Du et al., 2021).

For low-rank MDP, the transfer coefficient C_π is upper bounded by a relative condition number style quantity defined using the unknown ground truth feature $\phi^\star$ (see Lemma 11). On the computational side, Algorithm 1 (with the modification of $a \sim \text{Uniform}(\mathcal{A})$ in the online data collection step) requires to solve a least squares regression problem at every round. The objective of this regression problem is a convex functional of the hypothesis f over the constraint set $\mathcal{F}$. While this is not fully efficiently implementable due to the non-convex constraint set $\mathcal{F}$, our regression problem is still much simpler than the oracle models considered in the prior works for this model (Agarwal et al., 2020a; Uehara et al., 2021; Modi et al., 2021).

A.5 Bounds on transfer coefficient

Note that C_π takes both the distribution shift and the function class into consideration, and is smaller than the existing density ratio based concentrability coefficient (Kakade & Langford, 2002; Munos & Szepesvári, 2008; Chen & Jiang, 2019) and also existing Bellman error based concentrability coefficient Xie et al. (2021a). We formalize this in the following lemma.

Lemma 9. *For any π and offline distribution ν,*

$$C_\pi \le \sqrt{\max_{f,h} \frac{\|f_h - \mathcal{T} f_{h+1}\|_{d_h^\pi}^2}{\|f_h - \mathcal{T} f_{h+1}\|_{\nu_h}^2}} \le \sup_{h,s,a} \frac{d_h^\pi(s,a)}{\nu_h(s,a)}.$$

Proof. Using Jensen's inequality, we get that

$$\begin{aligned} C_\pi &\le \sqrt{\max_{f} \frac{\sum_{h=0}^{H-1} \|f_h - \mathcal{T} f_{h+1}\|_{d_h^\pi}^2}{\sum_{h=0}^{H-1} \|f_h - \mathcal{T} f_{h+1}\|_{\nu_h}^2}} \\ &\le \sqrt{\max_{f,h} \frac{\|f_h - \mathcal{T} f_{h+1}\|_{d_h^\pi}^2}{\|f_h - \mathcal{T} f_{h+1}\|_{\nu_h}^2}} \\ &\le \sqrt{\sup_{h,s,a} \frac{d_h^\pi(s,a)}{\nu_h(s,a)}} \\ &\le \sup_{h,s,a} \frac{d_h^\pi(s,a)}{\nu_h(s,a)}, \end{aligned}$$

where the second line follows from the Mediant inequality and the last line holds whenever $\sup_{h,s,a} \frac{d_h^\pi(s,a)}{\nu_h(s,a)} \ge 1$. □

Next we show that in the linear Bellman complete setting, C_π is bounded by the relative condition number.

Lemma 10. *Consider the linear Bellman complete setting (Definition 3) with known feature ϕ. Suppose that the feature covariance matrix induced by offline distribution ν: $\Sigma_{\nu_h} := \mathbb{E}_{s,a\sim\nu_h}[\phi^\star(s,a)\phi^\star(s,a)^\top]$ is invertible. Then for any policy π, we have*

$$C_\pi \le \sqrt{\sum_{h=1}^{H} \mathbb{E}_{s,a\sim d_{h-1}^\pi} \|\phi(s,a)\|_{\Sigma_{\nu_{h-1}}^{-1}}^2}.$$

Proof. Repeat the argument in Lemma 9, we have

$$\begin{aligned} C_\pi &\le \sqrt{\max_{f,h} \frac{\|f_h - \mathcal{T} f_{h+1}\|_{d_h^\pi}^2}{\|f_h - \mathcal{T} f_{h+1}\|_{\nu_h}^2}} \\ &\le \sqrt{\max_{w,h} \frac{\|w_h^\top \phi - {w'}_h^\top \phi\|_{d_h^\pi}^2}{\|w_h^\top \phi - {w'}_h^\top \phi\|_{\nu_h}^2}} \\ &\le \sqrt{\max_{w,h} \frac{\|(w_h - w'_h)\|_{\Sigma_{\nu_h}}^2 \mathbb{E}_{d_h^\pi} \|\phi\|_{\Sigma_{\nu_h}^{-1}}^2}{\|(w_h - w'_h)^\top \phi\|_{\nu_h}^2}} \\ &\le \sqrt{\max_{w,h} \frac{\|(w_h - w'_h)\|_{\Sigma_{\nu_h}}^2 \mathbb{E}_{d_h^\pi} \|\phi\|_{\Sigma_{\nu_h}^{-1}}^2}{\|(w_h - w'_h)^\top \phi\|_{\nu_h}^2}} \\ &= \sqrt{\max_{h} \mathbb{E}_{s,a\sim d_{h-1}^\pi} \|\phi(s,a)\|_{\Sigma_{\nu_{h-1}}^{-1}}^2}. \end{aligned}$$

Recall that in linear Bellman complete setting, we can write f as $w^\top \phi$, and for any w that defines f, there exists w' such that $\mathcal{T} f = {w'}^\top \phi$. □

Now we proceed to low-rank MDPs. We show that for low-rank MDPs, C_π is bounded by the partial feature coverage.

Lemma 11. *Consider the low-rank MDP setting (Definition 5) where the transition dynamics P is given by $P(s' \mid s, a) = \langle \mu^\star(s'), \phi^\star(s,a) \rangle$ for some $\mu^\star, \phi^\star \in \mathbb{R}^d$. Suppose that the offline distribution $\nu = (\nu_0, \ldots, \nu_{H-1})$ is such that $\max_h \max_{s,a} \frac{\nu_h(s)}{\nu_h(s,a)} \le \alpha$ for any s, a. Furthermore, suppose that ν is induced via trajectories i.e. $\nu_0(s) = d_0(s)$ and $\nu_h(s) = \mathbb{E}_{\bar{s},\bar{a} \sim \nu_{h-1}} P(s|\bar{s},\bar{a})$ for any $h \ge 1$, and that the feature covariance matrix $\Sigma_{\nu_{h-1},\phi^\star} := \mathbb{E}_{s,a\sim\nu_{h-1}}[\phi^\star(s,a)\phi^\star(s,a)^\top]$ is invertible. Then for any policy π, we have*

$$C_\pi \le \sqrt{\alpha} \sum_{h=1}^{H} \mathbb{E}_{s,a\sim d^\pi_{h-1}} \left[\|\phi^\star(s,a)\|_{\Sigma^{-1}_{\nu_{h-1},\phi^\star}} \right] + \sqrt{\alpha}.$$

Proof. We first upper bound the numerator separately. First note that for $h = 0$,

$$\begin{aligned}
\mathbb{E}_{s,a\sim d^\pi_0} \left[\mathcal{T} f_1(s,a) - f_0(s,a) \right] &\le \sqrt{\mathbb{E}_{s\sim d_0, a\sim\pi(\cdot|s)} \left[\left(\mathcal{T} f_1(s,a) - f_0(s,a) \right)^2 \right]} \\
&\le \sqrt{\max_{s,a} \frac{d^\pi_0(s,a)}{\nu_0(s,a)} \cdot \mathbb{E}_{s,a\sim\nu_0} \left[\left(\mathcal{T} f_1(s,a) - f_0(s,a) \right)^2 \right]} \\
&\le \sqrt{\alpha \cdot \mathbb{E}_{s,a\sim\nu_0} \left[\left(\mathcal{T} f_1(s,a) - f_0(s,a) \right)^2 \right]}, \qquad (21)
\end{aligned}$$

where the last inequality follows from our assumption since $\max_{s,a} \frac{d^\pi_0(s,a)}{\nu_0(s,a)} \le \max_s \frac{d^\pi_0(s)}{\nu_0(s,a)} \le \alpha$.

Next, for any $h \ge 1$, we note that backing up one step and looking at the pair $\bar{s}, \bar{a}$ that lead to the state s, we get that

$$\begin{aligned}
&\mathbb{E}_{s,a\sim d^\pi_h} \left[\mathcal{T} f_{h+1}(s,a) - f_h(s,a) \right] \\
&\quad = \mathbb{E}_{\bar{s},\bar{a}\sim d^\pi_{h-1}, s\sim P_{h-1}(\bar{s},\bar{a}), a\sim\pi(s)} \left[\mathcal{T} f_{h+1}(s,a) - f_h(s,a) \right] \\
&\quad = \mathbb{E}_{\bar{s},\bar{a}\sim d^\pi_{h-1}} \left[\int \left(\phi^\star(\bar{s},\bar{a})^\top \mu^\star(s) \right) \sum_a \pi(a|s) \left[\mathcal{T} f_{h+1}(s,a) - f_h(s,a) \right] \mathrm{d}s \right] \\
&\quad = \mathbb{E}_{\bar{s},\bar{a}\sim d^\pi_{h-1}} \left[\phi^\star(\bar{s},\bar{a})^\top \int \sum_a \mu^\star(s)\pi(a|s) \left[\mathcal{T} f_{h+1}(s,a) - f_h(s,a) \right] \mathrm{d}s \right] \\
&\quad \le \mathbb{E}_{\bar{s},\bar{a}\sim d^\pi_{h-1}} \left[\|\phi^\star(\bar{s},\bar{a})\|_{\Sigma^{-1}_{\nu_{h-1},\phi^\star}} \left\| \int \sum_a \mu^\star(s)\pi(a|s) \left[\mathcal{T} f_{h+1}(s,a) - f_h(s,a) \right] \mathrm{d}s \right\|_{\Sigma_{\nu_{h-1},\phi^\star}} \right], \qquad (22)
\end{aligned}$$

where the last line follows from an application of Cauchy-Schwarz inequality. For the term inside the expectation in the right hand side above, we note that for any s, a,

$$\begin{aligned}
&\left\| \int \sum_a \mu^\star(s)\pi(a|s) \left[\mathcal{T} f_{h+1}(s,a) - f_h(s,a) \right] \mathrm{d}s \right\|^2_{\Sigma_{\nu_{h-1},\phi^\star}} \\
&\quad \overset{(i)}{\le} \mathbb{E}_{\bar{s},\bar{a}\sim\nu_{h-1}} \left[\left(\int \sum_a \left(\mu^\star(s)^\top \phi^*(\bar{s},\bar{a}) \right) \pi(a|s) (\mathcal{T} f_{h+1}(s,a) - f_h(s,a)) \mathrm{d}s \right)^2 \right] \\
&\quad = \mathbb{E}_{\bar{s},\bar{a}\sim\nu_{h-1}} \left[\left(\mathbb{E}_{s\sim P(\bar{s},\bar{a}), a\sim\pi(s)} [\mathcal{T} f_{h+1}(s,a) - f_h(s,a)] \right)^2 \right] \\
&\quad \overset{(ii)}{\le} \mathbb{E}_{\bar{s},\bar{a}\sim\nu_{h-1}, s\sim P(\bar{s},\bar{a}), a\sim\pi(s)} \left[\left(\mathcal{T} f_{h+1}(s,a) - f_h(s,a) \right)^2 \right] \\
&\quad \overset{(iii)}{=} \mathbb{E}_{s\sim\nu_h, a\sim\pi(s)} \left[\left(\mathcal{T} f_{h+1}(s,a) - f_h(s,a) \right)^2 \right]
\end{aligned}$$

$$\overset{(iv)}{\leq} \alpha \cdot \mathbb{E}_{s,a\sim\nu_h}\left[\left(\mathcal{T}f_{h+1}(s,a) - f_h(s,a)\right)^2\right] \tag{23}$$

where (i) follows by expanding the norm , (ii) follows an application of Jensen's inequality, (iii) is due to our assumption that the offline dataset is generated using trajectories such that $\nu_h(s) = \mathbb{E}_{\bar{s},\bar{s}\sim\nu_{h-1}}[P(s \mid \bar{s},\bar{a})]$. Finally, (iv) follows from the definition of α. Plugging (23) in (22), we get that for $h \geq 1$,

$$\begin{aligned}
&\mathbb{E}_{s,a\sim d_h^\pi}\left[\mathcal{T}f_{h+1}(s,a) - f_h(s,a)\right] \\
&\quad\leq \mathbb{E}_{\bar{s},\bar{a}\sim d_{h-1}^\pi}\left[\|\phi^\star(\bar{s},\bar{a})\|_{\Sigma^{-1}_{\nu_{h-1},\phi^\star}} \sqrt{\alpha \cdot \mathbb{E}_{s,a\sim\nu_h}\left[\left(\mathcal{T}f_{h+1}(s,a) - f_h(s,a)\right)^2\right]}\right]
\end{aligned} \tag{24}$$

We are now ready to bound the transfer coefficient. First note that using (21), for any f,

$$\begin{aligned}
\frac{\mathbb{E}_{s,a\sim d_0^\pi}\left[\mathcal{T}f_1(s,a) - f_0(s,a)\right]}{\sqrt{\sum_{h=0}^{H-1}\mathbb{E}_{s,a\sim\nu_h}\left[\left(\mathcal{T}f_{h+1}(s,a) - f_h(s,a)\right)^2\right]}} &\leq \frac{\sqrt{\alpha \cdot \mathbb{E}_{s,a\sim d_0^\pi}\left[\left(\mathcal{T}f_1(s,a) - f_0(s,a)\right)^2\right]}}{\sqrt{\sum_{h=0}^{H-1}\mathbb{E}_{s,a\sim\nu_h}\left[\left(\mathcal{T}f_{h+1}(s,a) - f_h(s,a)\right)^2\right]}} \\
&\leq \sqrt{\alpha}.
\end{aligned}$$

Furthermore, for any f, using (24), we get that

$$\begin{aligned}
&\frac{\sum_{h=1}^{H-1}\mathbb{E}_{s,a\sim d_h^\pi}\left[\mathcal{T}f_{h+1}(s,a) - f_h(s,a)\right]}{\sqrt{\sum_{h=0}^{H-1}\mathbb{E}_{s,a\sim\nu_h}\left[\left(\mathcal{T}f_{h+1}(s,a) - f_h(s,a)\right)^2\right]}} \\
&\quad\leq \sum_{h=1}^{H-1}\mathbb{E}_{\bar{s},\bar{a}\sim d_{h-1}^\pi}\left[\|\phi^\star(\bar{s},\bar{a})\|_{\Sigma^{-1}_{\nu_{h-1},\phi^\star}} \frac{\sqrt{\alpha \cdot \mathbb{E}_{s,a\sim\nu_h}\left[\left(\mathcal{T}f_{h+1}(s,a) - f_h(s,a)\right)^2\right]}}{\sqrt{\sum_{h=0}^{H-1}\mathbb{E}_{s,a\sim\nu_h}\left[\left(\mathcal{T}f_{h+1}(s,a) - f_h(s,a)\right)^2\right]}}\right] \\
&\quad\leq \sum_{h=1}^{H}\mathbb{E}_{\bar{s},\bar{a}\sim d_{h-1}^\pi}\left[\|\phi^\star(\bar{s},\bar{a})\|_{\Sigma^{-1}_{\nu_{h-1},\phi^\star}} \sqrt{\alpha}\right],
\end{aligned}$$

where the last line holds for an appropriate choice of λ (e.g. $\lambda = 0$). Combining the above two bounds in the definition of C_π we get that

$$C_\pi \leq \sqrt{\alpha}\sum_{h=1}^{H}\mathbb{E}_{\bar{s},\bar{a}\sim d_{h-1}^\pi}\left[\|\phi^\star(\bar{s},\bar{a})\|_{\Sigma^{-1}_{\nu_{h-1},\phi^\star}}\right] + \sqrt{\alpha}.$$

□

B Low Bellman Eluder Dimension problems

To capture the complexity of the MDP and the function class we use the notion of Bellman Eluder dimension introduced by Jin et al. (2021a). This complexity measure is a distributional version of the Eluder dimension applied to the class of Bellman residuals w.r.t. $\mathcal{F}$.

We first introduce the key definitions:

Definition 6 (ε-independence between distributions (Jin et al., 2021a)). *Let $\mathcal{G}$ be a class of functions defined on a space $\mathcal{X}$, and $\nu, \mu_1, \ldots, \mu_n$ be probability measures over $\mathcal{X}$. We say ν is ε-independent of $\{\mu_1, \mu_2, \ldots, \mu_n\}$ with respect to $\mathcal{G}$ if there exists $g \in \mathcal{G}$ such that $\sqrt{\sum_{i=1}^n(\mathbb{E}_{\mu_i}[g])^2} \leq \varepsilon$, but $|\mathbb{E}_\nu[g]| > \varepsilon$.*

Definition 7 (Distributional Eluder (DE) dimension). *Let $\mathcal{G}$ be a function class defined on $\mathcal{X}$, and $\mathcal{P}$ be a family of probability measures over $\mathcal{X}$. The distributional Eluder dimension $\dim_{\mathrm{DE}}(\mathcal{F}, \mathcal{P}, \varepsilon)$ is the length of the longest sequence $\{\rho_1, \ldots, \rho_n\} \subset \Pi$ such that there exists $\varepsilon' \geq \varepsilon$ where ρ_i is ε'-independent of $\{\rho_1, \ldots, \rho_{i-1}\}$ for all $i \in [n]$.*

Definition 8 (Bellman Eluder (BE) dimension (Jin et al., 2021a)). *Given a value function class $\mathcal{F}$, let $\mathcal{G}_h := (f_h - \mathcal{T} f_{h+1} \mid f \in \mathcal{F}_h, f_{h+1} \in \mathcal{F}_{h+1})$ be the set of Bellman residuals induced by $\mathcal{F}$ at step h, and $\mathcal{P} = \{\mathcal{P}_h\}_{h=1}^H$ be a collection of H probability measure families over $\mathcal{X} \times \mathcal{A}$. The ϵ-Bellman Eluder dimension of $\mathcal{F}$ with respect to $\mathcal{P}$ is defined as*

$$\dim_{\mathrm{BE}}(\mathcal{F}, \mathcal{P}, \varepsilon) := \max_{h \in [H]} \dim_{\mathrm{DE}}(\mathcal{G}_h, \mathcal{P}_h, \epsilon)\,.$$

We also note the following lemma that controls the rate at which Bellman error accummulates.

Lemma 12 (Lemma 41, (Jin et al., 2021a)). *Given a function class $\mathcal{G}$ defined on a space $\mathcal{X}$ with $\sup_{g \in \mathcal{G}, x \in \mathcal{X}} |g(x)| \le C$, and a set of probability measures $\mathcal{P}$ over $\mathcal{X}$. Suppose that the sequence $\{g_k\}_{k=1}^K \subset \mathcal{G}$ and $\{\mu_k\}_{k=1}^K \subset \mathcal{P}$ satisfy that $\sum_{t=1}^{k-1} (\mathbb{E}_{\mu_t}[g_k])^2 \le \beta$ for all $k \in [K]$. Then, for all $k \in [K]$ and $\gamma > 0$,*

$$\sum_{t=1}^{k} |\mathbb{E}_{\mu_t}[g_t]| \le O\Big(\sqrt{\dim_{\mathrm{DE}}(\mathcal{G}, \mathcal{P}, \gamma)\beta k} + \min\{k, \dim_{\mathrm{DE}}(\mathcal{G}, \mathcal{P}, \gamma) C\} + k\gamma\Big).$$

We next state our main theorem whose proof is similar to that of Theorem 1.

Theorem 2 (Cumulative subpotimality). *Fix $\delta \in (0,1)$, $m_{\mathrm{off}} = HT/d$ and $m_{\mathrm{on}} = H^2$, and suppose that the underlying MDP admits Bellman eluder dimention d, and the function class $\mathcal{F}$ satisfies Assumption 1. Then with probability at least $1-\delta$, Algorithm 1 obtains the following bound on cumulative subpotimality w.r.t. any comparator policy π^e,*

$$\sum_{t=1}^{T} V^{\pi^e} - V^{\pi^t} = \widetilde{O}\Big(V_{\max} \max\{C_{\pi^e}, 1\} \sqrt{dT \cdot \log(H|\mathcal{F}|/\delta)}\Big),$$

where $\pi^t = \pi^{f^t}$ is the greedy policy w.r.t. f^t at round t and $d = \dim_{\mathrm{BE}}(\mathcal{F}, \mathcal{P}_{\mathcal{F}}, 1/\sqrt{T})$. Here $\mathcal{P}_{\mathcal{F}}$ is the class of occupancy measures that can be be induced by greedy policies w.r.t. value functions in $\mathcal{F}$.

Proof. Repeating the analysis till (18) in the proof of Theorem 1, we get that

$$\sum_{t=1}^{T} V^{\pi^e} - V^{\pi^t} \le T C_{\pi^e} \cdot \sqrt{H \cdot \Delta_{\mathrm{off}}} + \sum_{t=1}^{T} \sum_{h=0}^{H-1} \Big| \mathbb{E}_{s,a \sim d_h^{\pi^t f^t}} \left[f_h^t(s,a) - \mathcal{T}_h f_{h+1}^t(s,a) \right] \Big|$$

Using the bound in Lemma 7 and Lemma 12 in the above, we get that

$$\begin{aligned}\sum_{t=1}^{T} V^{\pi^e} - V^{\pi^t} \lesssim T C_{\pi^e} \cdot \sqrt{H \cdot \Delta_{\mathrm{off}}} + \sum_{h=0}^{H-1} \sqrt{\dim_{\mathrm{DE}}(\mathcal{G}_h, \mathcal{P}_{\mathcal{F};h}, \gamma) \Delta_{\mathrm{on}} T} \\ + \min\{T, \dim_{\mathrm{DE}}(\mathcal{G}_h, \mathcal{P}_{\mathcal{F};h}, \gamma) C\} + T\gamma.\end{aligned}$$

where $\mathcal{G}_h := (f_h - \mathcal{T} f_{h+1} \mid f \in \mathcal{F}_h, f_{h+1} \in \mathcal{F}_{h+1})$ denotes the set of Bellman residuals induced by $\mathcal{F}$ at step h, and $\mathcal{P} = \{\mathcal{P}_{\mathcal{F};h}\}_{h=1}^H$ is the collection of occupancy measures at step h induced by greedy policies w.r.t. value functions in $\mathcal{F}$. We set $\gamma = 1/\sqrt{T}$ and define $d = \dim_{\mathrm{BE}}(\mathcal{F}, \mathcal{P}, \gamma) = \max_h \dim_{\mathrm{DE}}(\mathcal{G}_h, \mathcal{P}_{\mathcal{F};h}, \gamma)$. Ignoring the lower order terms, we get that

$$\begin{aligned}\sum_{t=1}^{T} V^{\pi^e} - V^{\pi^t} &\lesssim T C_{\pi^e} \cdot \sqrt{H \cdot \Delta_{\mathrm{off}}} + H\sqrt{d \Delta_{\mathrm{on}} T} \\ &\lesssim T C_{\pi^e} V_{\max} \cdot \sqrt{H \cdot \frac{\log(HT|\mathcal{F}|/\delta)}{m_{\mathrm{off}}}} + H V_{\max} \sqrt{dT \cdot \frac{\log(HT|\mathcal{F}|/\delta)}{m_{\mathrm{on}}}},\end{aligned}$$

where $\lesssim$ hides lower order terms, multiplying constants and log factors. Setting $m_{\mathrm{off}} = HT/d$ and $m_{\mathrm{on}} = H^2$, we get that

$$\sum_{t=1}^{T} V^{\pi^e} - V^{\pi^t} = \widetilde{O}\Big(C_{\pi^e} V_{\max} \sqrt{dT \log(HT|\mathcal{F}|/\delta)}\Big).$$

□

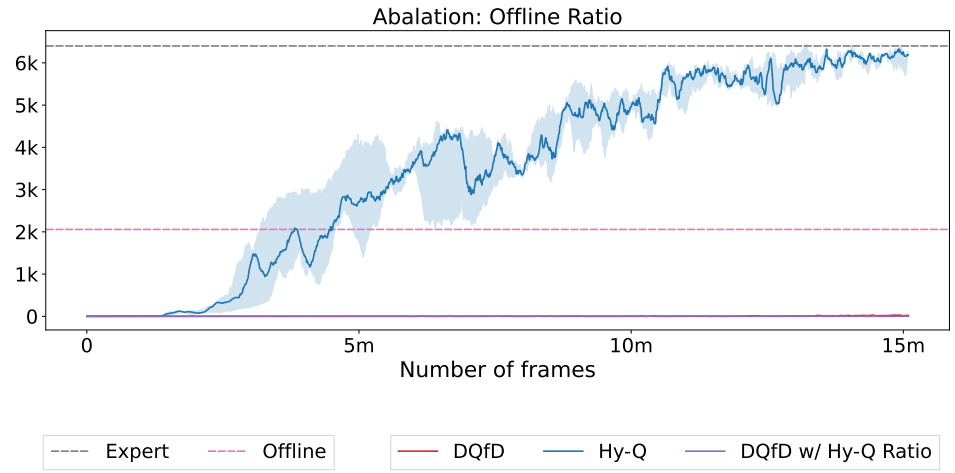

Figure 6: Ablation study: replacing DQFD's offline buffer sampling ratio schedule with Hy-Q's schedule. We use the hard offline dataset for this experiment. Similar to the previous experiments, the plots show the median and 80th/20th quantile for 5 replicates. "Expert" denotes V^{π^e} and "Offline" denotes the average trajectory reward in the offline dataset. The y-axis denotes the (moving) average of 100 episodes for the methods involving online interactions. Note that both variants of DQFD overlap at the bottom of the plot. Note that we focus on comparing the two variants of DQFD so we remove the offline RL and imitation learning baselines from the plot.

C Additional Ablation Experiments

C.1 Ablation on offline buffer ratio

In this section, we analyze how DQFD performs when the offline buffer sampling ratio schedule is replaced by Hy-Q's offline buffer sampling ratio schedule. The experiment is conducted on the hard dataset and all setups remain fixed as in the main experiment section. We report the training curves in Figure 6. We observe that switching the offline buffer ratio schedule does not improve the performance of DQFD on the hard dataset, and we believe this is because of the usage of the imitation learning loss in their offline data loss.

C.2 Ablation on pretraining

In this section, we analyze the performance difference once we allow Hy-Q to perform pretraining on the offline data before the online stage. For the pretraining, we use the same FQI/DQN style update as Hy-Q trains the offline data during the online stage. We use the same number of pretrain steps as DQFD. We present the result in Figure 7. We observe that there is no major performance difference with or without the pretraining stage for Hy-Q. We believe this is because FQI algorithm provably fails when the offline data does not have sufficient global coverage (in fact the hard dataset indeed does not contain global coverage).

D Comparison with previous works

As mentioned in the main text, many previous empirical works consider combining offline expert demonstrations with online interaction (Rajeswaran et al., 2017; Hester et al., 2018; Nair et al., 2018, 2020; Vecerik et al., 2017; Lee et al., 2022; Jia et al., 2022; Niu et al., 2022). Thus the idea of performing RL algorithm on both offline data (expert demonstrations) and online data is also explored in some of the previous works, for example, Vecerik et al. (2017) runs DDPG on both the online and expert data, and Hester et al. (2018) uses DQN on both data but with an additional supervised loss. Since we already compared with Hester et al. (2018) in the experiment, here we focus on our discussion with Vecerik et al. (2017).

We first emphasize that Vecerik et al. (2017) only focuses on expert demonstrations and their experiments entirely rely on using expert demonstrations, while we focus on more general offline

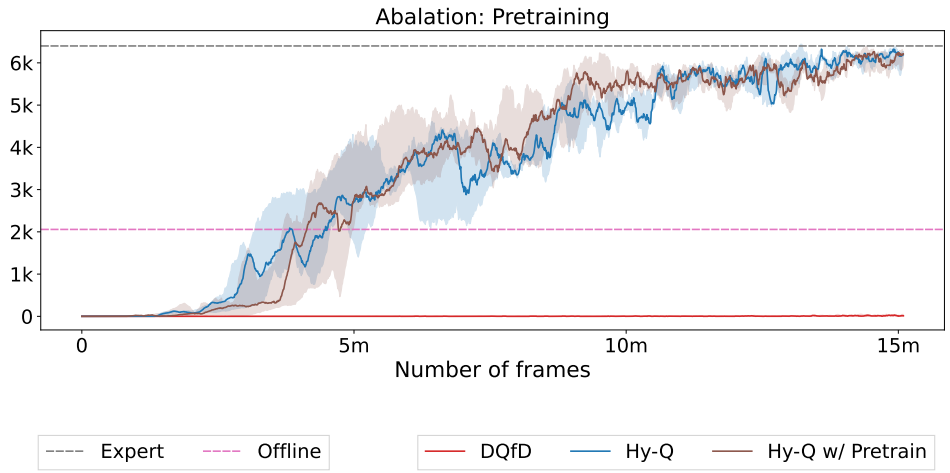

Figure 7: Ablation study: adding an additional pretraining stage for Hy-Q. We use the hard offline dataset for this experiment. The plots show the median and 80th/20th quantile for 5 replicates. "Expert" denotes V^{π^e} and "Offline" denotes the average trajectory reward in the offline dataset. The y-axis denotes the (moving) average of 100 episodes for the methods involving online interactions. Note that we focus on comparing the two variants of Hy-Q so we remove the offline RL and imitation learning baselines from the plot.

dataset that is not necessarily coming from experts. Said though, the DDPG-based algorithm from Vecerik et al. (2017) potentially can be used when offline data is not from experts. Although the algorithm from Vecerik et al. (2017) and Hy-Q share the same high-level intuition that one should perform RL on both the datasets, there are still a few differences : (1) Hy-Q uses Q-learning instead of deterministic policy gradients; note that deterministic policy gradient methods cannot be directly applied to discrete action setting; (2) Hy-Q does not require n-step TD style update, since in off-policy case, without proper importance weighting, n-step TD could incur strong bias. While proper tuning on n could balance bias and variance, one does not need to tune such n-step at all in Hy-Q; (3) The idea of keeping a non-zero ratio to sample offline dataset is also proposed in Vecerik et al. (2017). Our buffer ratio is derived from our theory analysis but meanwhile proves the advantage of the similar heuristic applied in Vecerik et al. (2017). (4) In their experiment, Vecerik et al. (2017) only considers expert demonstrations. In our experiment, we considered offline datasets with different amounts of transitions from very low-quality policies and showed Hy-Q is robust to low-quality transitions in offline data. Note that some of the differences may seem minor on the implementation level, but they may be important to the theory.

Regarding the experiments, our experimental evaluation adds the following insights over those in Vecerik et al. (2017): (i) hybrid methods can succeed without expert data, (ii) hybrid methods can succeed in hard exploration discrete-action tasks, (iii) the core algorithm (Q-learning vs DDPG) is not essential although some details may matter. Due to the similarity between the two methods, we believe some of these insights may also translate to Vecerik et al. (2017) and we expect that the choice between Hy-Q and Hy-DDPG will be environment specific, as it is with the purely online versions of these methods. In some situations, Q-learning works does not immediately imply Deterministic policy gradient methods work, nor vice versa. Nevertheless, it is beyond the scope of this paper to rigorously verify this claim and we deem the study of Actor-critic algorithms in Hybrid RL setting an interesting future direction.

E Experiment Details

E.1 Combination Lock

In this section we provide a detailed description of combination lock experiment. The combination lock environment has a horizon H and 10 actions at each state. There are three latent states $z_{i,h}, i \in \{0, 1, 2\}$ for each timestep h, where $z_{i,h}, i \in \{0, 1\}$ are good states and $z_{2,h}$ is the bad state. For each good state, we randomly pick a good action $a_{i,h}$, such that in latent state $z_{i,h}, i \in \{0, 1\}$,

Algorithm 2 V-type Hy-Q

Require: Value function class: $\mathcal{F}$, #iterations: T, Offline dataset $\mathcal{D}_h^\nu$ of size m_{off} for $h \in [H-1]$.

1: Initialize $f_h^1(s,a) = 0$.
2: **for** $t = 1, \ldots, T$ **do**
3: Let π^t be the greedy policy w.r.t. f^t i.e., $\pi_h^t(s) = \text{argmax}_a f_h^t(s,a)$.
`// Online collection`
4: For each h, collect m_{on} online tuples $\mathcal{D}_h^t \sim d_h^{\pi^t} \circ \text{Uniform}(\mathcal{A})$.
`// FQI using both online and offline data`
5: Set $f_H^{t+1}(s,a) = 0$.
6: **for** $h = H-1, \ldots, 0$ **do**
7: Estimate f_h^{t+1} using least squares regression on the aggregated data:

$$f_h^{t+1} \leftarrow \underset{f \in \mathcal{F}_h}{\text{argmin}} \left\{ \widehat{\mathbb{E}}_{\mathcal{D}_h^\nu}(f(s,a) - r - \max_{a'} f_{h+1}^{t+1}(s',a'))^2 + \sum_{\tau=1}^{t} \widehat{\mathbb{E}}_{\mathcal{D}_h^\tau}(f(s,a) - r - \max_{a'} f_{h+1}^{t+1}(s',a'))^2 \right\}$$

8: **end for**
9: **end for**

taking the good action $a_{i,h}$ will result in 0.5 probability of transiting to $z_{0,h+1}$ and 0.5 probability of transiting to $z_{1,h+1}$ while taking all other actions will result in a 1 probability of transiting to $z_{2,h+1}$. At $z_{2,h}$, all actions will result in a deterministic transition to $z_{2,h+1}$. For the reward, we give an optimal reward of 1 for landing $z_{i,H}, i \in \{0,1\}$. We also give an anti-shaped reward of 0.1 for all transitions from a good state to a bad state. All other transitions have a reward of 0. The initial distribution is a uniform distribution over $z_{0,0}$ and $z_{1,0}$. The observation space has dimension $2^{\lceil \log(H+1) \rceil}$, created by concatenating a one-hot representation of the latent state and a one-hot representation of the horizon (appending 0 if necessary). Random noise from $\mathcal{N}(0, 0.1)$ is added to each dimension, and finally the observation is multiplied by a Hadamard matrix. Note that in this environment, the agent needs to perform optimally for all H timesteps to hit the final good state for an optimal reward of 1. Once the agent chooses a bad action, it will stay in the bad state until the end with at most 0.1 possible reward for the trajectory received while transitting from a good state to a bad state.

E.2 Implementation Details of Combination Lock experiment

We train H separate Q-functions for all H timesteps. Our function class consists of an encoder and a decoder. For the encoder, we feed the observation into one linear layer with 3 outputs, followed by a softmax layer to get a state-representation. This design of encoder is intended to learn a one-hot representation of the latent state. We take a Kronecker Product of the state-representation and the action, and feed the result to a linear layer with only one output, which will be our Q value. In order to stabilize the training, we warm-start the Q-function of timestep $h-1$ with the encoder from h Q-function of the current iteration and the decoder from the $h-1$ Q-function of the last iteration, for each iteration of training.

One remark is that since combination lock belongs to Block MDPs, we require a V-type algorithm instead of the Q-type algorithm as shown in the main text. The only difference lies in the online sampling process: instead of sampling from $d_h^{\pi^t}$, for each h, we sample from $d_h^{\pi^t} \circ \text{Uniform}(\mathcal{A})$, i.e., we first rollin with respect to π^t to timestep $h-1$, then take a random action, observe the transition and collect that tuple. We provide Algorithm 2 for completeness. Note that the only difference is in line 4.

For CQL, we implemented the variant of CQL-DQN and picked the peak in the learning curve to report in the main paper (so it should represent an upper bound of the performance of CQL).

E.3 Implementation Details of Montezuma's Revenge experiment

In this section we provide the detailed algorithm for the discounted setting. The overall algorithm is described in Algorithm 3. For the function approximation, we use a class of convolutional neural networks (parameterized by class Θ) as promoted by the original DQN paper. We include several

Algorithm 3 Discounted Hy-Q

Require: Value function class: $\mathcal{F}$ (induced by Θ), #iterations: T, Offline dataset $\mathcal{D}^\nu$ of size m_{off}, discounted factor γ, target update frequency n_{target}, learning rate α, offline sample ratio β, exploration rate ϵ, action space $\mathcal{A}$.

Randomly initialize value function f^θ.
Initialize target value function $\tilde{f} = f^\theta$.
Initialize online buffer $\mathcal{D} = \emptyset$.
Sample initial state $s \sim d_0$
for $t = 1, \ldots, T$ **do**
  Let π be the ϵ-greedy policy w.r.t. f^θ i.e., $\pi(s) = \text{argmax}_a f^\theta(s, a)$ with probability $1 - \epsilon$ and $\pi(s) = \mathcal{U}(\mathcal{A})$ with probability ϵ.
  // Online collection
  Interact with the environment for one step:

$$a = \pi(s), s' \sim P(s, a), r \sim R(s, a).$$

  Update online buffer: $\mathcal{D} = \mathcal{D} \cup \{s, a, r, s'\}$.
  // Discounted minibatch FQI using both online and offline data
  if $t \mod n_{\text{value}} = 0$ **then**
    With probability $1 - \beta$: Sample a minibatch D with size $n_{\text{minibatch}}$ from online buffer $\mathcal{D}$.
    Otherwise: Sample a minibatch D with size $n_{\text{minibatch}}$ from offline buffer $\mathcal{D}^\nu$.
    Perform one-step gradient descent on D:

$$\theta = \theta - \alpha \nabla_\theta \hat{\mathbb{E}}_D \left(f^\theta(s, a) - r_i - \gamma \max_{a'} \tilde{f}(s', a') \right)^2.$$

  end if
  // Delayed update of target function every n_{target} updates
  if $t \mod n_{\text{target}} = 0$ **then**
    Set target function to the current value function: $\tilde{f} = f^\theta$.
  end if
  Update $s \leftarrow s'$.
end for

standard empirical design choices that have been practically proven to stabilize the training: we use Prioritize Experience Replay (Schaul et al., 2015) for our buffer. We also add Double DQN (Van Hasselt et al., 2016) and Dueling DQN (Wang et al., 2016) during our Q-update. We also observe that a decaying schedule on the offline sample ratio β and the exploration rate ϵ also helps provide better performance. Note that an annealing β does not contradict to our comment in Section 4 on catastrophic forgetting because we set β to small after our online trajectory distribution covers d^{π^e}. In addition, we also perform per step update instead of per episode update since this has been the popular design choice and leads to better efficiency in practice.

E.4 Baseline implementation

E.4.1 Combination Lock

We use the open-sourced implementation https://github.com/BY571/CQL/tree/main/CQL-DQN for CQL. For BRIEE, we use the official code released by the authors: https://github.com/yudasong/briee, where we rely on the code there for the combination lock environment.

E.4.2 Montezuma's Revenge

We use the open-sourced implementation https://github.com/jcwleo/random-network-distillation-pytorch for RND. For CQL, we use https://github.com/takuseno/d3rlpy for their implementation of CQL for atari. We use https://github.com/felix-kerkhoff/DQfD for DQFD. For all baselines, we keep the

hyperparameters used in these public repositories. For CQL and DQFD, we provide the offline datasets as described in the main text instead of using the offline dataset provided in the public repositories.[7] All baselines are tested in the same stochastic environment setup as in Burda et al. (2018).

E.5 Hardware Infrastructure

We run our experiments on a cluster of computes with Nvidia RTX 3090 GPUs and various CPUs which do not incur any randomness to the results.

E.6 Hyperparameters

E.6.1 Combination Lock

We provide the hyperparameters of Hy-Q in Table. 1. In addition, we provide the hyperparameters we tried for CQL baseline in Table. 2.

Table 1: Hyperparameters for Hy-Q in combination lock

	Value Considered	Final Value
Learning rate	{1e-2, 2e-2, 1e-3}	2e-2
Buffer size	{1e8}	1e8
Optimizer	{Adam, SGD}	Adam
Number of updates per iteration	{30, 300, 500}	500
Batch size	{512}	512

Table 2: Hyperparameters for CQL(DQN) in combination lock

	Value Considered	Final Value
Learning rate	{1e-3}	1e-3
Optimizer	{Adam}	Adam
Buffer size	{1e8}	1e8
Batch size	{512}	512
Discount Factor	{0.99}	0.99
Moving Average Factor τ	{0.01, 0.1, 1}	0.01
Weight on CQL loss α	{0, 0.1, 0.01}	0.1

E.6.2 Montezuma's Revenge

We provide the hyperparameter of Hy-Q in Table. 3. We reuse many hyperparameter choices from DQFD. Note that $[a, b]$ denotes a decreasing/increasing schedule from a to b.

[7] We note that CQL also fails completely with the original offline dataset (with 1 million samples) provided in the public repository.

Table 3: Hyperparameter of Discounted Hy-Q in Montezuma's Revenge.

	Value Considered	Final Value
Learning rate	{6.25e-5, [1e-4,1e-5]}	[1e-4,1e-5]
Offline Schedule β	{0.5,0.2,[0.2,0.01]}	[0.2,0.01]
Exploration ϵ rate	{[0.25,0.001]}	[0.25,0.001]
Minibatch size $n_{\text{minibatch}}$	{32}	32
Weight decay (regularization) coefficient	{1e-5}	1e-5
Gradient Clipping	{10,20}	10
Discount factor γ	{0.99}	0.99
Value function update frequency n_{update}	{4}	4
Target function update frequency n_{target}	{1000,2000,5000,10000}	10000
Buffer size	$\{2^{20}\}$	2^{20}
PER Importance Sampling ratio	{[0.6,1]}	[0.6,1]
Online PER ϵ	{0.001}	0.001
Offline PER ϵ	{0.0001}	0.0001
Online PER Priority Coefficient	{0.4}	0.4
Offline PER Priority Coefficient	{1}	1