WHAT ARE THE STATISTICAL LIMITS OF BATCH RL WITH LINEAR FUNCTION APPROXIMATION?

Anonymous authors
Paper under double-blind review

ABSTRACT

Function approximation methods coupled with batch reinforcement learning (or off-policy reinforcement learning) are providing an increasingly important framework to help alleviate the excessive sample complexity burden in modern reinforcement learning problems. However, the extent to which function approximation, when coupled with off-policy data, can be effective is not well understood, where the literature largely consists of *sufficient* conditions.

This work focuses on the basic question: what are *necessary* representational and distributional conditions that permit provable sample-efficient off-policy RL? Perhaps surprisingly, our main result shows even if 1) we have *realizability* in that the true value function of our target policy has a linear representation in a given set of features and 2) our off-policy data has good *coverage* over all these features (in a precisely defined and strong sense), any algorithm information-theoretically still requires an exponential number of off-policy samples to non-trivially estimate the value of the target policy. Our results highlight that sample-efficient, batch RL is not guaranteed unless significantly stronger conditions, such as the distribution shift is sufficiently mild (which we precisely characterize) or representation conditions that are far stronger than realizability, are met.

1 INTRODUCTION

Off-policy methods are a promising methodology to alleviate the sample complexity burden in challenging reinforcement learning (RL) settings, particularly those where sample efficiency is paramount (Mandel et al., 2014; Gottesman et al., 2018; Wang et al., 2018; Yu et al., 2019). Off-policy methods are often applied together with function approximation schemes; such methods take sample transition data and reward values as inputs, and approximate the value of a target policy or the value function of the optimal policy. Indeed, many practical deep RL algorithms find their prototypes in the literature of batch RL. For example, when running on off-policy data (sometimes termed as "experience replay"), deep Q-networks (DQN) (Mnih et al., 2015) can be viewed as an analog of Fitted Q-Iteration (Gordon, 1999) with neural networks being the function approximators. More recently, there are an increasing number of both model-free (Laroche et al., 2019; Fujimoto et al., 2019; Jaques et al., 2020; Kumar et al., 2019; Agarwal et al., 2020) and model-based (Ross & Bagnell, 2012; Kidambi et al., 2020) batch RL methods, with steady improvements in performance (Fujimoto et al., 2019; Kumar et al., 2019; Wu et al., 2020; Kidambi et al., 2020).

However, despite the importance of these methods, the extent to which data reuse is possible, especially when off-policy methods are combined with function approximation, is not well understood. For example, deep Q-network requires millions of samples to solve certain Atari games (Mnih et al., 2015). Also important is that in some safety-critical settings, we seek guarantees when offline-trained policies can be effective (Thomas, 2014; Thomas et al., 2019). A basic question here is that if there are fundamental statistical limits on such methods, where sample-efficient batch RL is simply not possible without further restrictions on the problem.

In the context of supervised learning, it is well-known that empirical risk minimization is sample-efficient if the hypothesis class has bounded complexity. For example, suppose the agent is given a d-dimensional feature extractor, and the ground truth labeling function is a (*realizable*) linear function with respect to the feature mapping. Here, it is well-known that a polynomial number of samples in d suffice for a given target accuracy. Furthermore, in this realizable case, provided

the training data has a good feature coverage, then we will have good accuracy against any test distribution.[1]

In the more challenging batch RL setting, it is unclear if sample-efficient methods are possible, even under analogous assumptions. This is our motivation to consider the following question:

What are the statistical limits for batch RL with linear function approximation?

Here, one may hope that value estimation for a given policy is possible in the batch RL setting under the analogous set of assumptions that enable sample-efficient supervised learning, i.e., 1) (realizability) the features can perfectly represent the value function of the given policy and 2) (good coverage) the feature covariance matrix of our off-policy data has lower bounded eigenvalues.

The extant body of provable methods on batch RL either make representational assumptions that are far stronger than realizability or assume distribution shift conditions that are far stronger than having coverage with regards to the spectrum of the feature covariance matrix of the data distribution. For example, Szepesvári & Munos (2005) analyze batch RL methods by assuming a representational condition where the features satisfy (approximate) closedness under Bellman updates, which is a far stronger representation condition than realizability. Recently, Xie & Jiang (2020a) propose a batch RL algorithm that only requires realizability as the representation condition. However, the algorithm in (Xie & Jiang, 2020a) requires a more stringent data distribution condition. Whether it is possible to design a sample-efficient batch RL method under the realizability assumption and a reasonable data coverage assumption — an open problem in (Chen & Jiang, 2019) — is the focus of this work.

Our Contributions. Perhaps surprisingly, our main result shows that, under only the above two assumptions, it is information-theoretically not possible to design a sample-efficient algorithm to non-trivially estimate the value of a given policy. The following theorem is an informal version of the results in Section 4 and Appendix B.

Theorem 1.1 (Informal)**.** *In the batch RL setting, suppose the data distributions have (polynomially) lower bounded eigenvalues. Given a policy π, where the Q-function of π is linear with respect to a given feature mapping, any algorithm requires an exponential number of samples in the horizon H to output a non-trivially accurate estimate of the value of π, with constant probability.*

Our hardness result formalizes a key issue in batch reinforcement learning with function approximation: geometric error amplification. To better illustrate the error amplification issue, in Section 5, we analyze the classical Least-Squares Value Iteration (LSVI) algorithm under the realizability assumption, which demonstrates how the error propagates as the algorithm proceeds. Here, our analysis shows that, if we only rely on the realizability assumption, then a far more stringent condition is required for sample-efficient off-policy evaluation: the off-policy, data distribution must be quite close to the distribution induced by the policy itself.

Our results highlight that sample-efficient batch RL is simply not possible unless either the distribution shift condition is sufficiently mild or we have stronger representation conditions that go well beyond realizability.[2]

Furthermore, our hardness result implies an exponential separation on the sample complexity between batch RL and supervised learning, since supervised learning (which is equivalent to batch RL with $H = 1$) is possible with polynomial number of samples under the same set of assumptions.

There are a few additional points worth emphasizing with regards to our lower bound construction:

- As a corollary, our results show that Least-Squares Value Iteration (LSVI, i.e. using Bellman backups with linear regression) will fail. Interestingly, while LSVI will provide an unbiased estimator, our results imply that it will have exponential variance (in the problem horizon).
- Our construction is simple and does not rely on having a large state or action space: the size of the state space is only $O(d \cdot H)$ where d is the feature dimension and H is the planning horizon,

[1] Specifically, if the features have a uniformly bounded norm and if the minimum eigenvalue of the feature covariance matrix of our data is bounded away from 0, say by $1/\text{poly}(d)$, then we have good accuracy on any test distribution. See Assumption 2 and the comments thereafter.

[2] For an upper bound that assumes closedness under Bellman updates (which is stronger than realizability), see e.g. (Chen & Jiang, 2019).

and the size of the action space is only is 2. This stands in contrast to other RL lower bounds, which typically require state spaces that are exponential in the problem horizon (e.g. see (Du et al., 2020)).
- We provide two hard instances, one with a sparse reward (and stochastic transitions) and another with deterministic dynamics (and stochastic rewards). These two hard instances jointly imply that both the estimation error on reward values and the estimation error on the transition probabilities could be geometrically amplified in batch RL.
- Of possibly broader interest is that our hard instances are, to our knowledge, the first concrete examples showing that geometric error amplification is real in RL problems (even with realizability). While this is a known concern in the analysis of RL algorithms, there have been no concrete examples exhibiting such behavior under only a realizability assumption.

2 Related Work

We now survey prior work batch reinforcement learning, largely focusing on theoretical results. We also discuss results on the error amplification issue in RL. Concurrent to this work, Xie & Jiang (2020a) propose a batch RL algorithm under the realizability assumption, which requires stronger distribution shift conditions. We will discuss this work shortly.

Existing Algorithms and Analysis. Batch RL with value function approximation is closely related to Approximate Dynamic Programming (Bertsekas & Tsitsiklis, 1995). Existing works (Munos, 2003; Szepesvári & Munos, 2005; Antos et al., 2008; Munos & Szepesvári, 2008; Tosatto et al., 2017; Xie & Jiang, 2020b) that analyze the sample complexity of approximate dynamic programming-based approaches usually make the following two categories of assumptions: (i) representation conditions that assume the function class approximates the value functions well and (ii) distribution shift conditions that assume the given data distribution has sufficient coverage over the state-action space. As mentioned in the introduction, the desired representation condition would be realizability, which only assumes the value function of the policy to be evaluated lies in the function class (for the case of off-policy evaluation) or the optimal value function lies in the function class (for the case of finding near-optimal policies), and existing works usually make stronger assumptions. For example, Szepesvári & Munos (2005) assume (approximate) closedness under Bellman updates, which is much stronger than realizability. Whether it is possible to design a sample-efficient batch RL method under the realizability assumption and reasonable data coverage assumption, is left as an open problem in (Chen & Jiang, 2019).

To measure the coverage over the state-action space of the given data distribution, existing works assume the concentratability coefficient (introduced by Munos (2003)) to be bounded. The concentratability coefficient, informally speaking, is the largest possible ratio between the probability for a state-action pair (s, a) to be visited by a policy, and the probability that (s, a) appears on the data distribution. Since we work with linear function approximation in this work, we measure the distribution shift in terms of the spectrum of the feature covariance matrices (see Assumption 2), which is a well-known sufficient condition in the context of supervised learning and is much more natural for the case of linear function approximation.

Concurrent to this work, Xie & Jiang (2020a) propose an algorithm that works under the realizability assumption instead of other stronger representation conditions used in prior work. However, the algorithm in (Xie & Jiang, 2020a) requires a much stronger data distribution condition which assumes a stringent version of concentrability coefficient introduced by (Munos, 2003) to be bounded. In contrast, in this work we measure the distribution shift in terms of the spectrum of the feature covariance matrix of the data distribution, which is more natural than the concentrability coefficient for the case of linear function approximation.

Recently, there has been great interest in applying importance sampling to approach off-policy evaluation (Precup, 2000). For a list of works on this topic, see (Dudík et al., 2011; Mandel et al., 2014; Thomas et al., 2015; Li et al., 2015; Jiang & Li, 2016; Thomas & Brunskill, 2016; Guo et al., 2017; Wang et al., 2017; Liu et al., 2018; Farajtabar et al., 2018; Xie et al., 2019; Kallus & Uehara, 2019; Liu et al., 2019; Uehara & Jiang, 2019; Kallus & Uehara, 2020; Jiang & Huang, 2020; Feng et al., 2020). Off-policy evaluation with importance sampling incurs exponential variance in the planning horizon when the behavior policy is significantly different from the policy to be evaluated. Bypass-

ing such exponential dependency requires non-trivial function approximation assumptions (Jiang & Huang, 2020; Feng et al., 2020; Liu et al., 2018). Finally, Kidambi et al. (2020) provides a model-based batch RL algorithm, with a theoretical analysis based on hitting times.

Hardness Results. Historically, algorithm-specific hardness results have been known for a long time in the literature of Approximate Dynamic Programming. See Chapter 4 in (Van Roy, 1994) and also (Gordon, 1995; Tsitsiklis & Van Roy, 1996). These works demonstrate that certain approximate dynamic programming-based methods will diverge on hard cases. However, such hardness results only hold for a restricted class of algorithms, and to demonstrate the fundamental difficulty of batch RL, it is more desirable to obtain information-theoretic lower bounds, as recently initiated by Chen & Jiang (2019).

Existing (information-theoretic) exponential lower bounds (Krishnamurthy et al., 2016; Sun et al., 2017; Chen & Jiang, 2019) usually construct unstructured MDPs with an exponentially large state space. Du et al. (2020) prove an exponential lower bound for planning under the assumption that the optimal Q-function is approximately linear. The condition that the optimal Q-function is only *approximately* linear is crucial for the correctness of the hardness result in Du et al. (2020) The techniques in (Du et al., 2020) are later generalized to other settings, including (Kumar et al., 2020; Wang et al., 2020; Mou et al., 2020).

Error Amplification In RL. Error amplification induced by distribution shift and long planning horizon is a known issue in the theoretical analysis of RL algorithms. See (Gordon, 1995; 1996; Munos & Moore, 1999; Ormoneit & Sen, 2002; Kakade, 2003; Zanette et al., 2019) for papers on this topic and additional assumptions that mitigate this issue. Error amplification in batch RL is also observed in empirical works (see e.g. (Fujimoto et al., 2019)). In this work, we provide the first information-theoretic lower bound showing that geometric error amplification is real in batch RL.

3 The Off-Policy Evaluation Problem

Throughout this paper, for a given integer H, we use $[H]$ to denote the set $\{1, 2, \ldots, H\}$.

Episodic Reinforcement Learning. Let $M = (\mathcal{S}, \mathcal{A}, P, R, H)$ be a *Markov Decision Process* (MDP) where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $P : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ is the transition operator which takes a state-action pair and returns a distribution over states, $R : \mathcal{S} \times \mathcal{A} \to \Delta(\mathbb{R})$ is the reward distribution, $H \in \mathbb{Z}_+$ is the planning horizon. For simplicity, we assume a fixed initial state $s_1 \in \mathcal{S}$.

A policy $\pi : \mathcal{S} \to \mathcal{A}$ chooses an action a based on the current state s. The policy π induces a (random) trajectory $s_1, a_1, r_1, s_2, a_2, r_2, \ldots, s_H, a_H, r_H$, where $a_1 = \pi_1(s_1)$, $r_1 \sim R(s_1, a_1)$, $s_2 \sim P(s_1, a_1)$, $a_2 = \pi_2(s_2)$, etc. To streamline our analysis, for each $h \in [H]$, we use $\mathcal{S}_h \subseteq \mathcal{S}$ to denote the set of states at level h, and we assume $\mathcal{S}_h$ do not intersect with each other. We assume, almost surely, that $r_h \in [-1, 1]$ for all $h \in [H]$.

Value Functions. Given a policy π, $h \in [H]$ and $(s, a) \in \mathcal{S}_h \times \mathcal{A}$, the Q-function is defined as $Q_h^\pi(s, a) = \mathbb{E}\left[\sum_{h'=h}^{H} r_{h'} \mid s_h = s, a_h = a, \pi\right]$. The value function of a state $s \in \mathcal{S}_h$ is defined as $V_h^\pi(s) = \mathbb{E}\left[\sum_{h'=h}^{H} r_{h'} \mid s_h = s, \pi\right]$. For a policy π, we define $V^\pi = V_1^\pi(s_1)$ to be the value of π from the fixed initial state s_1.

Batch Reinforcement Learning. This paper is concerned with the batch RL setting. In this setting, the agent does not have direct access to the MDP and instead is given access to data distributions $\{\mu_h\}_{h=1}^H$ where for each $h \in [H]$, $\mu_h \in \Delta(\mathcal{S}_h \times \mathcal{A})$. The inputs of the agent are H datasets $\{D_h\}_{h=1}^H$, and for each $h \in [H]$, D_h consists i.i.d. samples of the form $(s, a, r, s') \in \mathcal{S}_h \times \mathcal{A} \times \mathbb{R} \times \mathcal{S}_{h+1}$ tuples, where $(s, a) \sim \mu_h$, $r \sim r(s, a)$, $s' \sim P(s, a)$.

This paper is mainly concerned with the *off-policy evaluation problem*: Given a policy $\pi : \mathcal{S} \to \mathcal{A}$, the goal is to output an accurate estimate of the value of π (i.e., V^π) approximately, using the collected datasets $\{D_h\}_{h=1}^H$, with as few samples as possible.

Linear Function Approximation. When applying linear function approximation schemes, it is commonly assumed that the agent is given a feature extractor $\phi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$ which can either be hand-crafted or a pre-trained neural network that transforms a state-action pair to a d-dimensional embedding, and the Q-function can be predicted by linear functions of the features. We assume that for all $(s, a) \in \mathcal{S} \times \mathcal{A}$, we have $\|\phi(s, a)\|_2 \leq 1$.

We are interested in the off-policy evaluation problem, under the following *realizability* assumption:

Assumption 1 (Realizability)**.** *Suppose there exists $\theta_1, \ldots \theta_H \in \mathbb{R}^d$ such that, for the given policy $\pi : \mathcal{S} \to \mathcal{A}$ and for each $h \in [H]$, $Q_h^\pi(s, a) = \theta_h^\top \phi(s, a)$. Without loss of generality, we can assume that our coordinate system is such that $\|\theta_h\|_2 \leq \sqrt{d}$ and $\|\phi(s, a)\|_2 \leq 1$ for all $(s, a) \in \mathcal{S} \times \mathcal{A}$.*

In particular, we are interested in avoiding imposing stronger structural assumptions on the MDP, until we address the fundamental question of if realizability alone is sufficient for accurate off-policy evaluation, provided that our data distributions μ_h provide us with good coverage over the features.

Notation. For a vector $x \in \mathbb{R}^d$, we use $\|x\|_2$ to denote its ℓ_2 norm. For a positive semidefinite matrix A, we use $\|A\|_2$ to denote its operator norm, and $\sigma_{\min}(A)$ to denote its smallest eigenvalue. For two positive semidefinite matrices A and B, we write $A \succeq B$ to denote the Löwner partial ordering of matrices, i.e, $A \succeq B$ if and only if $A - B$ is positive semidefinite.

4 The Lower Bound: Realizability and Coverage are Insufficient

We now present our main hardness result for off-policy evaluation in batch RL with linear function approximation. It should be evident that without feature coverage in our dataset, then realizability alone is clearly not sufficient for sample-efficient estimation. Here, we will make the strongest possible assumption, with regards to good feature coverage of the feature covariance matrix.

Assumption 2 (Feature Coverage)**.** *Suppose for each $h \in [H]$, the data distributions μ_h satisfies the following minimum eigenvalue condition: $\sigma_{\min}(\mathbb{E}_{(s,a)\sim\mu_h}[\phi(s, a)\phi(s, a)^\top]) = 1/d$. Note that $1/d$ is the largest possible minimum eigenvalue due to that, for any data distribution $\widetilde{\mu}_h$, $\sigma_{\min}(\mathbb{E}_{(s,a)\sim\widetilde{\mu}_h}[\phi(s, a)\phi(s, a)^\top]) \leq \frac{1}{d}$ since we assume $\|\phi(s, a)\|_2 \leq 1$ for all $(s, a) \in \mathcal{S} \times \mathcal{A}$.*

It should be evident that for the case where $H = 1$, both our realizability assumption (Assumption 1) and feature coverage assumption (Assumption 2) will imply that the ordinary least squares estimator will accurately estimate θ_1.[3] Our main result now shows that these assumptions are not sufficient for batch RL for long horizon problems.

Theorem 4.1. *Suppose Assumption 1 and Assumption 2 hold. For any algorithm which takes as input both the policy π and the feature mapping ϕ, there exists a (deterministic) MDP such that the algorithm requires $\Omega((d/2)^H)$ samples to output the value of the policy π up to constant additive approximation error with probability at least 0.9.*

Remark 1. (The sparse reward case) As stated, the theorem uses a deterministic MDP (with stochastic rewards). See Appendix B for another hard case where the transition is stochastic and the reward is deterministic and sparse (only occurring at two states at $h = H$).

Remark 2. (Least-Squares Value Iteration (LSVI) has exponential variance) The most naïve algorithm here would be to use ordinary least squares (OLS) to estimate θ, starting at level $h = H$ and then proceeding backwards to level $h = 1$, using the plug-in estimator from the previous level. Here, LSVI will provide an unbiased estimate (provided the feature covariance matrices are full rank, which will occur with high probability). Interestingly, as a direct corollary, the above theorem implies that LSVI has exponential variance in H. More generally, our theorem implies that there is no estimator that can avoid such exponential dependence. See Section 5 for a more detailed discussion on LSVI.

Remark 3. (Policy improvement) Although we focus on off-policy evaluation in this work, our hardness result also holds for *finding* near-optimal policies under the realizability assumption[4] in the batch RL setting with linear function approximation. See Appendix C for more details.

[3] For $H = 1$, the ordinary least squares estimator will satisfy that $\|\theta_1 - \widehat{\theta}_{\text{OLS}}\|^2 \leq O(d/n)$ with high probability. See e.g. (Hsu et al., 2012b).

[4] In the case of finding near-optimal policies, the realizability assumption would be $Q_h^*(s, a) = \theta_h^\top \phi(s, a)$ for all $h \in [H]$. Here Q^* is the Q-function of the optimal policy (see e.g. (Chen & Jiang, 2019)).

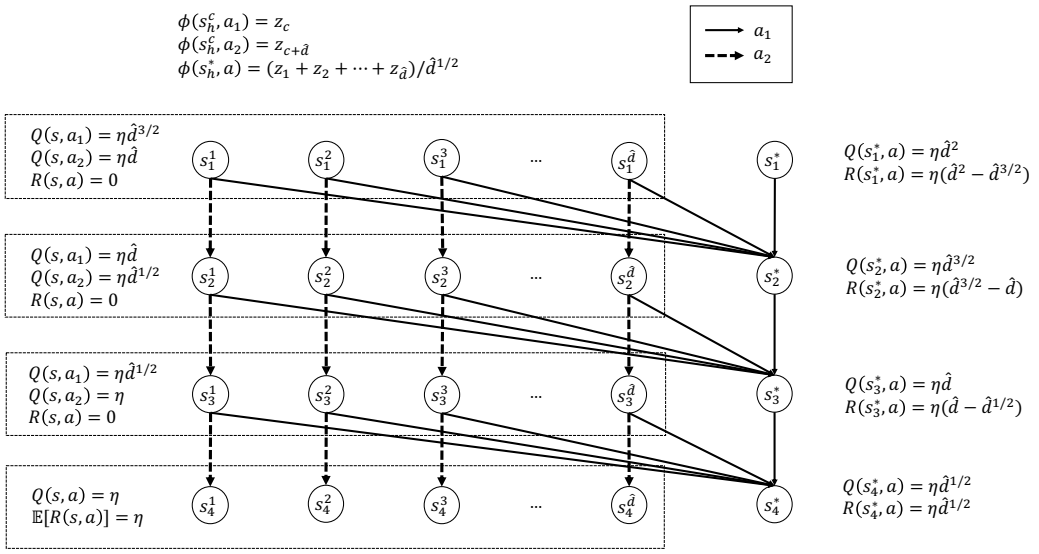

Figure 1: An illustration of the hard instance with $H = 4$. Recall that $\hat{d} = d/2$. States on the top are those in the first level ($h = 1$), while states at the bottom are those in the last level ($h = 4$). Solid line (with arrow) corresponds to transitions associated with action a_1, while dotted line (with arrow) corresponds to transitions associated with action a_2. For each level $h \in [H]$, reward values and Q-values associated with $s_h^1, s_h^2, \ldots, s_h^{\hat{d}}$ are marked on the left, while reward values and Q-values associated with s_h^* are mark on the right. Rewards and transitions are all deterministic, except for the reward distributions associated with $s_4^1, s_4^2, \ldots, s_4^{\hat{d}}$. We mark the expectation of the reward value when it is stochastic. For each level $h \in [H]$, for the data distribution μ_h, the state is chosen uniformly at random from those states in the dashed rectangle, i.e., $\{s_h^1, s_h^2, \ldots, s_h^{\hat{d}}\}$, while the action is chosen uniformly at random from $\{a_1, a_2\}$. Suppose the initial state is s_1^*. When $\eta = 0$, the value of the policy is 0. When $\eta = \hat{d}^{-H/2} = \hat{d}^{-2}$, the value of the policy is $Q(s_1^*, a) = \eta\hat{d}^2 = 1$.

In the rest part of this section, we give the hard instance construction and the proof of Theorem 4.1. We use d the denote the feature dimension, and we assume d is even for simplicity. We use $\hat{d}$ to denote $d/2$ for convenience. We also provide an illustration of the construction in Figure 1.

State Space, Action Space and Transition Operator. The action space $\mathcal{A} = \{a_1, a_2\}$. For each $h \in [H]$, $\mathcal{S}_h$ contains $\hat{d} + 1$ states $s_h^1, s_h^2, \ldots, s_h^{\hat{d}}$ and s_h^*. For each $h \in [H-1]$, for each $c \in \{1, 2, \ldots, \hat{d}, *\}$, we have $P(s_h^c, a_1) = s_{h+1}^*$ and $P(s_h^c, a_2) = s_{h+1}^c$

Reward Distributions. Let $0 \leq \eta \leq \hat{d}^{-H/2}$ be a parameter to be determined. For each $(h, c) \in [H-1] \times [\hat{d}]$ and $a \in \mathcal{A}$, we set $R(s_h^c, a) = 0$ and $R(s_h^*, a) = \eta \cdot (\hat{d}^{1/2} - 1) \cdot \hat{d}^{(H-h)/2}$. For the last level, for each $c \in [\hat{d}]$ and $a \in \mathcal{A}$, we set $R(s_H^c, a) = \begin{cases} 1 & \text{with probability } (1+\eta)/2 \\ -1 & \text{with probability } (1-\eta)/2 \end{cases}$ so that $\mathbb{E}[R(s_H^c, a)] = \eta$. Moreover, for all actions $a \in \mathcal{A}$, $R(s_H^*, a) = \eta \cdot \hat{d}^{1/2}$.

Feature Mapping. Let $z_1, z_2, \ldots, z_d$ be a set of orthonormal vectors in $\mathbb{R}^d$. For each $(h, c) \in [H] \times [\hat{d}]$, we set $\phi(s_h^c, a_1) = z_c$, $\phi(s_h^c, a_2) = z_{c+\hat{d}}$, and $\phi(s_h^*, a) = \sum_{c \in \hat{d}} z_c / \hat{d}^{1/2}$ for all $a \in \mathcal{A}$.

Verifying Realizability. Now we consider the policy $\pi : \mathcal{S} \to \mathcal{A}$, which is defined to be $\pi(s) = a_1$ for all $s \in \mathcal{S}$. We show that Assumption 1 holds for π. The formal proof is given in Appendix A.

Lemma 4.2. *For each $h \in [H]$, for all $(s,a) \in \mathcal{S}_h \times \mathcal{A}$, we have $Q_h^\pi(s,a) = \theta_h^\top \phi(s,a)$ for some $\theta_h \in \mathbb{R}^d$ with $\|\theta\|_2 \leq \sqrt{d}$.*

The Data Distributions. For each level $h \in [H]$, the data distribution μ_h is a uniform distribution over $\{(s_h^1, a_1), (s_h^1, a_2), (s_h^2, a_1), (s_h^2, a_2), \ldots, (s_h^{\hat{d}}, a_1), (s_h^{\hat{d}}, a_2)\}$. Notice that (s_h^*, a) is *not* in the support of μ_h for all $a \in \mathcal{A}$. It can be seen that, $\mathbb{E}_{(s,a)\sim\mu_h}[\phi(s,a)\phi(s,a)^\top] = \frac{1}{d}\sum_{c=1}^d z_c z_c^\top = \frac{1}{d} I$.

The Lower Bound. We show that it is information-theoretically hard for any algorithm to distinguish the case $\eta = 0$ and $\eta = \hat{d}^{-H/2}$. We fix the initial state to be s_1^*, and consider the policy π defined above which returns action a_1 for all input states. When $\eta = 0$, all reward values will be zero, and thus the value of π would be zero. On the other hand, when $\eta = \hat{d}^{-H/2}$, the value of π would be $Q(s_1^*, \pi(s_1^*)) = Q(s_1^*, a_1) = \eta \cdot \hat{d}^{H/2} = 1$. Thus, if the algorithm approximates the value of the policy up to an error of $1/2$, then it must distinguish the case that $\eta = 0$ and $\eta = \hat{d}^{-H/2}$.

We first notice that for the case $\eta = 0$ and $\eta = \hat{d}^{-H/2}$, the data distributions $\{\mu_h\}_{h=1}^H$, the feature mapping $\phi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$, the policy π to be evaluated and the transition operator P are the same. Thus, in order to distinguish the case $\eta = 0$ and $\eta = \hat{d}^{-H/2}$, the only way is to query the reward distribution by using sampling taken from the data distributions.

For all state-action pairs (s,a) in the support of the data distributions of the first $H-1$ levels, the reward distributions will be identical. This is because for all $s \in \mathcal{S}_h \setminus \{s_h^*\}$ and $a \in \mathcal{A}$, we have $R(s,a) = 0$. For the case $\eta = 0$ and $\eta = \hat{d}^{-H/2}$, for all state-action pairs (s,a) in the support of the data distribution of the last level, $R(s,a) = \begin{cases} 1 & \text{with probability } (1+\eta)/2 \\ -1 & \text{with probability } (1-\eta)/2 \end{cases}$. Therefore, to distinguish the case that $\eta = 0$ and $\eta = \hat{d}^{-H/2}$, the agent needs to distinguish two reward distributions $r_1 = \begin{cases} 1 & \text{with probability } 1/2 \\ -1 & \text{with probability } 1/2 \end{cases}$ and $r_2 = \begin{cases} 1 & \text{with probability } (1+\hat{d}^{-H/2})/2 \\ -1 & \text{with probability } (1-\hat{d}^{-H/2})/2 \end{cases}$. It is well known that in order to distinguish r_1 and r_2 with probability at least 0.9, any algorithm requires $\Omega(\hat{d}^H)$ samples. See e.g. Lemma 5.1 in (Anthony & Bartlett, 2009). See also (Chernoff, 1972; Mannor & Tsitsiklis, 2004).

Remark 4. The key in our construction is the state s_h^* in each level, whose feature vector is defined to be $\sum_{c\in\hat{d}} z_c/\hat{d}^{1/2}$. In each level, s_h^* amplifies the Q-values by a $\hat{d}^{1/2}$ factor, due to the linearity of the Q-function. After all the H levels, the value will be amplified by a $\hat{d}^{H/2}$ factor. Since s_h^* is not in the support of the data distribution, the only way for the agent to estimate the value of the policy is to estimate the expected reward value in the last level. Our construction forces the estimation error of the last level to be amplified exponentially and thus implies an exponential lower bound.

We would like to remark that the design of the feature mapping in our construction could be flexible. It suffices if $z_1, z_2, \ldots, z_d$ are only nearly orthogonal. Moreover, the feature of s_h^* can be changed to $\sum_{c=1}^{\hat{d}} w_c z_c$ for a general set of coefficients $w_1, w_2, \ldots, w_{\hat{d}}$ so long as $\sum_{c=1}^{\hat{d}} w_c$ is sufficiently large.

5 Upper Bounds: Least-Squares Value Iteration Under Realizability and Low Distribution Shift

In order to illustrate the error amplification issue and discuss conditions that permit sample-efficient batch RL, in this section, we analyze Least-Squares Value Iteration when applied to off-policy evaluation under the realizability assumption. The algorithm is presented in Algorithm 1.

Measuring the Distribution Shift. To measure the distribution shift between the data distributions $\{\mu_h\}_{h=1}^H$ and the distribution induced by π, define $\overline{\Lambda}_1 = \mathbb{E}_{s_1}[\phi(s_1, \pi(s_1))\phi(s_1, \pi(s_1))^\top]$, and for each $h \in [H]$, define $\Lambda_h = \mathbb{E}_{(s,a)\sim\mu_h}[\phi(s,a)\phi(s,a)^\top]$ to be the covariance matrix of the data distribution, and $\overline{\Lambda}_{h+1} = \mathbb{E}_{(s,a)\sim\mu_h, \overline{s}\sim P(\cdot|s,a)}[\phi(\overline{s}, \pi(\overline{s}))\phi(\overline{s}, \pi(\overline{s}))^\top]$ to be the covariance matrix of the one-step lookahead distribution induced by the data distribution and π. To measure the distribution shift, our main assumption is as follows.

Algorithm 1 Least-Squares Value Iteration

Input: Policy $\pi : \mathcal{S} \to \mathcal{A}$, number of samples N, regularization parameter $\lambda > 0$
Let $Q_{H+1}(\cdot,\cdot) = 0$ and $V_{H+1}(\cdot) = 0$
for $h = H, H-1, \ldots, 1$ **do**
  Take samples $(s_h^i, a_h^i) \sim \mu_h$, $r_h^i \sim r(s_h^i, a_h^i)$ and $\overline{s}_h^i \sim P(s_h^i, a_h^i)$ for each $i \in [N]$
  Let $\hat{\Lambda}_h = \sum_{i\in[N]} \phi(s_h^i, a_h^i)\phi(s_h^i, a_h^i)^\top + \lambda I$
  Let $\hat{\theta}^h = \hat{\Lambda}_h^{-1}\left(\sum_{i=1}^N \phi(s_h^i, a_h^i)\cdot(r_h^i + \hat{V}_{h+1}(\overline{s}_h^i))\right)$
  Let $\hat{Q}_h(\cdot,\cdot) = \phi(\cdot,\cdot)^\top\hat{\theta}_h$ and $\hat{V}_h(\cdot) = \hat{Q}(\cdot, \pi(\cdot))$

Assumption 3. *We assume that for each $h \in [H]$, there exists $C_h \geq 1$ such that $\overline{\Lambda_h} \preceq C_h\Lambda_h$.*

Remark 5. For each $h \in [H]$, if $\sigma_{\min}(\Lambda_h) \succeq \frac{1}{C_h} I$ for some $C_h \geq 1$, then we have $\overline{\Lambda}_h \preceq I \preceq C_h\Lambda_h$. Therefore, Assumption 3 can be replaced with the assumption that $C_h\Lambda_h \succeq I$. However, we stick to the original version of Assumption 3, since it gives a tighter characterization of the distribution shift when applying Algorithm 1 to off-policy evaluation under the realizability assumption, which will be made clearer in the formal analysis in the appendix.

Now we state the theoretical guarantee of Algorithm 1. The proof can be found in Appendix D.

Theorem 5.1. *Suppose the given policy π satisfies Assumption 1. Let $\lambda = CH\sqrt{d\log(dH/\delta)N}$ for some $C > 0$. With probability at least $1-\delta$, for some $c > 0$,*

$$\mathbb{E}_{s_1}[(Q_1^\pi(s_1,\pi(s_1)) - \hat{Q}_1(s_1,\pi(s_1)))^2] \leq c\cdot\left(\prod_{h=1}^H C_h\right)\cdot dH^3\cdot\left(\sqrt{\frac{d\log(dH/\delta)}{N}} + \frac{H\log(1/\delta)}{N}\right).^5$$

Error Amplification. The factor $\prod_{h=1}^H C_h$ in Theorem 5.1 implies that the estimation error will be amplified *geometrically* as the algorithm proceeds. Now we briefly discuss how the error is amplified when running Algorithm 1 on the instance in Section 4 to better illustrate the issue.

If we run Algorithm 1 on the hard instance in Section 4, when $h = H$, the estimation error on $V(s_H^c)$ would be roughly $N^{-1/2}$ for each $c \in [\hat{d}]$. When using the linear predictor at level H to predict the value of s_H^*, the error will be amplified by $\hat{d}^{1/2}$. When $h = H-1$, the dataset contains only s_{H-1}^c for $c \in [\hat{d}]$, and the estimation error on the value of s_{H-1}^c will be the same as that of s_H^*, which is roughly $(\hat{d}/N)^{1/2}$. Again, the estimation error on the value of s_{H-1}^* will be $(\hat{d}^2/N)^{1/2}$ when using the linear predictor at level $H-1$. As the algorithm proceeds, the error will eventually be amplified by a factor of $\hat{d}^{H/2}$, which corresponds to the factor $\prod_{h=1}^H C_h$ in Theorem 5.1.

Implications. The above analysis again implies that geometric error amplification is a real issue in batch RL, and sample-efficient batch RL is impossible unless the distribution shift is sufficiently mild, i.e., $\prod_{h=1}^H C_h$ is bounded, or stronger representation condition such as closedness under Bellman updates is assumed as in prior works (Szepesvári & Munos, 2005; Chen & Jiang, 2019).

6 Conclusion

While the extant body of provable results in the literature largely focus on *sufficient* conditions for sample-efficient batch RL, this work focuses on obtaining a better understanding of the *necessary* conditions, where we seek to understand to what extent mild assumptions can imply sample-efficient batch RL. This work shows that for off-policy evaluation, even if we are given a representation that can perfectly represent the value function of the given policy and the data distribution has good coverage over the features, any provable algorithm still requires an exponential number of samples to non-trivially approximate the value of the given policy. These results highlight that provable sample-efficient batch RL is simply not possible unless either the distribution shift condition is sufficiently mild or we have stronger representation conditions that go well beyond realizability.

[5] We remark that better dependency on N is possible by relying on conditions on higher moments of the data distributions. We omit such discussion for simplicity. See e.g. (Hsu et al., 2012b).

REFERENCES

Rishabh Agarwal, Dale Schuurmans, and Mohammad Norouzi. An optimistic perspective on offline reinforcement learning. In *International Conference on Machine Learning*, 2020.

Martin Anthony and Peter L Bartlett. *Neural network learning: Theoretical foundations*. cambridge university press, 2009.

András Antos, Csaba Szepesvári, and Rémi Munos. Learning near-optimal policies with bellman-residual minimization based fitted policy iteration and a single sample path. *Machine Learning*, 71(1):89–129, 2008.

Dimitri P Bertsekas and John N Tsitsiklis. Neuro-dynamic programming: an overview. In *Proceedings of 1995 34th IEEE Conference on Decision and Control*, volume 1, pp. 560–564. IEEE, 1995.

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

Herman Chernoff. *Sequential analysis and optimal design*. SIAM, 1972.

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

Miroslav Dudík, John Langford, and Lihong Li. Doubly robust policy evaluation and learning. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, pp. 1097–1104, 2011.

Mehrdad Farajtabar, Yinlam Chow, and Mohammad Ghavamzadeh. More robust doubly robust off-policy evaluation. In *International Conference on Machine Learning*, pp. 1447–1456, 2018.

Yihao Feng, Tongzheng Ren, Ziyang Tang, and Qiang Liu. Accountable off-policy evaluation with kernel bellman statistics. *arXiv preprint arXiv:2008.06668*, 2020.

Scott Fujimoto, David Meger, and Doina Precup. Off-policy deep reinforcement learning without exploration. In *International Conference on Machine Learning*, pp. 2052–2062, 2019.

Geoffrey J Gordon. Stable function approximation in dynamic programming. In *Machine Learning Proceedings 1995*, pp. 261–268. Elsevier, 1995.

Geoffrey J Gordon. Stable fitted reinforcement learning. In *Advances in neural information processing systems*, pp. 1052–1058, 1996.

Geoffrey J Gordon. Approximate solutions to markov decision processes. Technical report, CARNEGIE-MELLON UNIV PITTSBURGH PA SCHOOL OF COMPUTER SCIENCE, 1999.

Omer Gottesman, Fredrik Johansson, Joshua Meier, Jack Dent, Donghun Lee, Srivatsan Srinivasan, Linying Zhang, Yi Ding, David Wihl, Xuefeng Peng, Jiayu Yao, Isaac Lage, Christopher Mosch, Li wei H. Lehman, Matthieu Komorowski, Matthieu Komorowski, Aldo Faisal, Leo Anthony Celi, David Sontag, and Finale Doshi-Velez. Evaluating reinforcement learning algorithms in observational health settings, 2018.

Zhaohan Guo, Philip S Thomas, and Emma Brunskill. Using options and covariance testing for long horizon off-policy policy evaluation. In *Advances in Neural Information Processing Systems*, pp. 2492–2501, 2017.

Daniel Hsu, Sham Kakade, Tong Zhang, et al. A tail inequality for quadratic forms of subgaussian random vectors. *Electronic Communications in Probability*, 17, 2012a.

Daniel Hsu, Sham M Kakade, and Tong Zhang. Random design analysis of ridge regression. In *Conference on learning theory*, pp. 9–1, 2012b.

Natasha Jaques, Asma Ghandeharioun, Judy Hanwen Shen, Craig Ferguson, Agata Lapedriza, Noah Jones, Shixiang Gu, and Rosalind Picard. Way off-policy batch deep reinforcement learning of human preferences in dialog, 2020. URL https://openreview.net/forum?id=rJl5rRVFvH.

Nan Jiang and Jiawei Huang. Minimax confidence interval for off-policy evaluation and policy optimization. *arXiv preprint arXiv:2002.02081*, 2020.

Nan Jiang and Lihong Li. Doubly robust off-policy value evaluation for reinforcement learning. In *International Conference on Machine Learning*, pp. 652–661. PMLR, 2016.

Sham Machandranath Kakade. *On the sample complexity of reinforcement learning*. PhD thesis, University of London London, England, 2003.

Nathan Kallus and Masatoshi Uehara. Efficiently breaking the curse of horizon in off-policy evaluation with double reinforcement learning. *arXiv preprint arXiv:1909.05850*, 2019.

Nathan Kallus and Masatoshi Uehara. Double reinforcement learning for efficient off-policy evaluation in markov decision processes. *Journal of Machine Learning Research*, 21(167):1–63, 2020.

Rahul Kidambi, Aravind Rajeswaran, Praneeth Netrapalli, and Thorsten Joachims. Morel : Model-based offline reinforcement learning, 2020.

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

Aviral Kumar, Justin Fu, Matthew Soh, George Tucker, and Sergey Levine. Stabilizing off-policy q-learning via bootstrapping error reduction. *neural information processing systems*, pp. 11761–11771, 2019.

Aviral Kumar, Abhishek Gupta, and Sergey Levine. Discor: Corrective feedback in reinforcement learning via distribution correction. *arXiv preprint arXiv:2003.07305*, 2020.

Romain Laroche, Paul Trichelair, and Rémi Tachet des Combes. Safe policy improvement with baseline bootstrapping. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, 2019.

Lihong Li, Remi Munos, and Csaba Szepesvari. Toward minimax off-policy value estimation. In *Artificial Intelligence and Statistics*, pp. 608–616, 2015.

Qiang Liu, Lihong Li, Ziyang Tang, and Dengyong Zhou. Breaking the curse of horizon: Infinite-horizon off-policy estimation. In *Advances in Neural Information Processing Systems*, pp. 5356–5366, 2018.

Yao Liu, Pierre-Luc Bacon, and Emma Brunskill. Understanding the curse of horizon in off-policy evaluation via conditional importance sampling. *arXiv preprint arXiv:1910.06508*, 2019.

Travis Mandel, Yun-En Liu, Sergey Levine, Emma Brunskill, and Zoran Popovic. Offline policy evaluation across representations with applications to educational games. In *AAMAS*, pp. 1077–1084, 2014.

Shie Mannor and John N Tsitsiklis. The sample complexity of exploration in the multi-armed bandit problem. *Journal of Machine Learning Research*, 5(Jun):623–648, 2004.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529, 2015.

Wenlong Mou, Zheng Wen, and Xi Chen. On the sample complexity of reinforcement learning with policy space generalization. *arXiv preprint arXiv:2008.07353*, 2020.

Rémi Munos. Error bounds for approximate policy iteration. In *ICML*, volume 3, pp. 560–567, 2003.

Remi Munos and Andrew W Moore. Barycentric interpolators for continuous space and time reinforcement learning. In *Advances in neural information processing systems*, pp. 1024–1030, 1999.

Rémi Munos and Csaba Szepesvári. Finite-time bounds for fitted value iteration. *Journal of Machine Learning Research*, 9(May):815–857, 2008.

Dirk Ormoneit and Śaunak Sen. Kernel-based reinforcement learning. *Machine learning*, 49(2-3): 161–178, 2002.

Doina Precup. Eligibility traces for off-policy policy evaluation. *Computer Science Department Faculty Publication Series*, pp. 80, 2000.

Stéphane Ross and Drew Bagnell. Agnostic system identification for model-based reinforcement learning. In *Proceedings of the 29th International Conference on Machine Learning, ICML 2012, Edinburgh, Scotland, UK, June 26 - July 1, 2012*. icml.cc / Omnipress, 2012.

Wen Sun, Arun Venkatraman, Geoffrey J Gordon, Byron Boots, and J Andrew Bagnell. Deeply aggrevated: Differentiable imitation learning for sequential prediction. In *International Conference on Machine Learning*, pp. 3309–3318, 2017.

Csaba Szepesvári and Rémi Munos. Finite time bounds for sampling based fitted value iteration. In *Proceedings of the 22nd international conference on Machine learning*, pp. 880–887, 2005.

Philip Thomas and Emma Brunskill. Data-efficient off-policy policy evaluation for reinforcement learning. In *International Conference on Machine Learning*, pp. 2139–2148, 2016.

Philip S. Thomas. *Safe reinforcement learning.* PhD thesis, University of Massachusetts, Amherst, 2014.

Philip S Thomas, Georgios Theocharous, and Mohammad Ghavamzadeh. High-confidence off-policy evaluation. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

Philip S. Thomas, Bruno Castro da Silva, Andrew G. Barto, Stephen Giguere, Yuriy Brun, and Emma Brunskill. Preventing undesirable behavior of intelligent machines. *Science*, 366(6468): 999–1004, 2019. ISSN 0036-8075. doi: 10.1126/science.aag3311.

Samuele Tosatto, Matteo Pirotta, Carlo d'Eramo, and Marcello Restelli. Boosted fitted q-iteration. In *International Conference on Machine Learning*, pp. 3434–3443. PMLR, 2017.

Joel A Tropp. An introduction to matrix concentration inequalities. *Foundations and Trends® in Machine Learning*, 8(1-2):1–230, 2015.

J Tsitsiklis and B Van Roy. An analysis of temporal-difference learning with function approximation (technical report lids-p-2322). *Laboratory for Information and Decision Systems*, 1996.

Masatoshi Uehara and Nan Jiang. Minimax weight and q-function learning for off-policy evaluation. *arXiv preprint arXiv:1910.12809*, 2019.

Benjamin Van Roy. *Feature-based methods for large scale dynamic programming*. PhD thesis, Massachusetts Institute of Technology, 1994.

L. Wang, Wei Zhang, Xiaofeng He, and H. Zha. Supervised reinforcement learning with recurrent neural network for dynamic treatment recommendation. *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018.

Ruosong Wang, Simon S Du, Lin F Yang, and Ruslan Salakhutdinov. On reward-free reinforcement learning with linear function approximation. *arXiv preprint arXiv:2006.11274*, 2020.

Yu-Xiang Wang, Alekh Agarwal, and Miroslav Dudık. Optimal and adaptive off-policy evaluation in contextual bandits. In *International Conference on Machine Learning*, pp. 3589–3597. PMLR, 2017.

Yifan Wu, George Tucker, and Ofir Nachum. Behavior regularized offline reinforcement learning, 2020. URL `https://openreview.net/forum?id=BJg9hTNKPH`.

Tengyang Xie and Nan Jiang. Batch value-function approximation with only realizability. *arXiv preprint arXiv:2008.04990*, 2020a.

Tengyang Xie and Nan Jiang. $Q^\star$ approximation schemes for batch reinforcement learning: A theoretical comparison. *arXiv preprint arXiv:2003.03924*, 2020b.

Tengyang Xie, Yifei Ma, and Yu-Xiang Wang. Towards optimal off-policy evaluation for reinforcement learning with marginalized importance sampling. In *Advances in Neural Information Processing Systems*, pp. 9668–9678, 2019.

C. Yu, G. Ren, and J. Liu. Deep inverse reinforcement learning for sepsis treatment. In *2019 IEEE International Conference on Healthcare Informatics (ICHI)*, pp. 1–3, 2019.

Andrea Zanette, Alessandro Lazaric, Mykel J Kochenderfer, and Emma Brunskill. Limiting extrapolation in linear approximate value iteration. In *Advances in Neural Information Processing Systems*, pp. 5616–5625, 2019.

A PROOF OF LEMMA 4.2

Proof. We first verify Q^π is linear for the first $H-1$ levels. For each $(h,c) \in [H-1] \times [\hat{d}]$, we have

$$Q_h^\pi(s_h^c, a_1) = R(s_h^c, a_1) + R(s_{h+1}^*, a_1) + R(s_{h+2}^*, a_1) + \ldots + R(s_H^*, a_1) = \eta \cdot \hat{d}^{(H-h)/2}$$

and

$$Q_h^\pi(s_h^c, a_2) = R(s_h^c, a_2) + R(s_{h+1}^c, a_1) + R(s_{h+2}^*, a_1) + \ldots + R(s_H^*, a_1) = \eta \cdot \hat{d}^{(H-h-1)/2}.$$

Moreover, for all $a \in \mathcal{A}$,

$$Q_h^\pi(s_h^*, a) = R(s_h^*, a) + R(s_{h+1}^*, a_1) + R(s_{h+2}^*, a_1) + \ldots + R(s_H^*, a_1) = \eta \cdot \hat{d}^{(H-h+1)/2}.$$

Therefore, if we define $\theta_h = \sum_{c=1}^{\hat{d}} \eta \cdot \hat{d}^{(H-h)/2} \cdot z_c + \sum_{c=1}^{\hat{d}} \eta \cdot \hat{d}^{(H-h-1)/2} \cdot z_{c+\hat{d}}$, then $Q_h^\pi(s,a) = \theta_h^\top \phi(s,a)$ for all $(s,a) \in \mathcal{S}_h \times \mathcal{A}$. Clearly, we have $\|\theta_h\|_2 \leq \sqrt{d}$ for all $h \in [H-1]$.

Now we verify that the Q-function is linear for the last level. Clearly, for all $c \in [\hat{d}]$ and $a \in \mathcal{A}$, $Q_H(s_H^c, a) = \eta$ and $Q_H(s_H^*, a) = \eta \cdot \sqrt{\hat{d}}$. Thus by defining $\theta_H = \sum_{c=1}^{d} \eta \cdot z_c$, we have $Q_H(s,a) = \theta_H^\top \phi(s,a)$ for all $(s,a) \in \mathcal{S}_H \times \mathcal{A}$.

□

B HARD INSTANCE WITH STOCHASTIC TRANSITION AND SPARSE REWARD

In this section, we present another hard case where the transition operator is stochastic and the reward is deterministic and sparse, meaning that the reward value is non-zero only for the last level. We also illustrate the construction in Figure 2. Throughout this section, we use d the denote the feature dimension, and we assume d is an even integer for simplicity. We use $\hat{d}$ to denote $d/2-2$.

State Space, Action Space and Transition Operator. Similar to the hard instance in Section 4, in this hard case, the action space $\mathcal{A} = \{a_1, a_2\}$ contains two elements. For each $h \in [H]$, $\mathcal{S}_h$ contains $\hat{d}+3$ states $s_h^1, s_h^2, \ldots, s_h^{\hat{d}}, s_h^*, s_h^+$ and s_h^-.

Let $0 \leq \eta \leq \hat{d}^{-(H-1)/2}$ be a parameter to be determined. We first define the transition operator for states in the first $H-2$ levels. For each $h \in [H-2]$, for each $c \in [\hat{d}]$, we have

$$P(s_h^c, a) = \begin{cases} s_{h+1}^* & a = a_1 \\ s_{h+1}^c & a = a_2 \end{cases}.$$

For each $h \in [H-2]$ and $a \in \mathcal{A}$, we have $P(s_h^+, a) = s_{h+1}^+$ and $P(s_h^-, a) = s_{h+1}^-$. For all actions $a \in \mathcal{A}$, we have

$$P(s_h^*, a) = \begin{cases} s_{h+1}^+ & \text{with probability } (1 + \eta \cdot \hat{d}^{(H-h)/2})/2 \\ s_{h+1}^- & \text{with probability } (1 - \eta \cdot \hat{d}^{(H-h)/2})/2 \end{cases}.$$

Now we define the transition operator for the second last level. For all $c \in [\hat{d}]$, we have

$$P(s_{H-1}^c, a_1) = \begin{cases} s_H^+ & \text{with probability } (1+\eta)/2 \\ s_H^- & \text{with probability } (1-\eta)/2 \end{cases}$$

and $P(s_{H-1}^c, a_2) = s_H^c$. For all $a \in \mathcal{A}$, we have $P(s_{H-1}^+, a) = s_H^+$ and $P(s_{H-1}^-, a) = s_H^-$. For all actions $a \in \mathcal{A}$, we also have

$$P(s_{H-1}^*, a) = \begin{cases} s_H^+ & \text{with probability } (1 + \eta \cdot \sqrt{\hat{d}})/2 \\ s_H^- & \text{with probability } (1 - \eta \cdot \sqrt{\hat{d}})/2 \end{cases}.$$

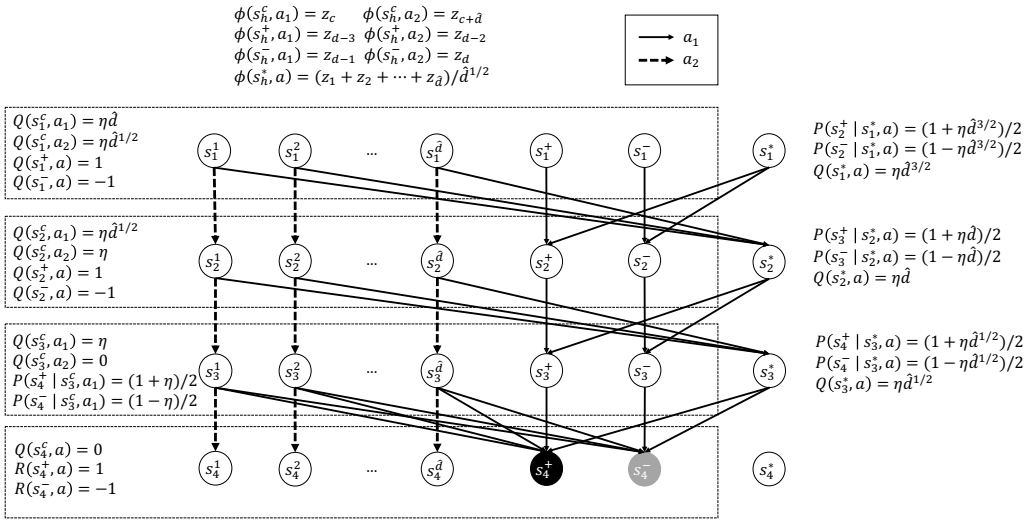

Figure 2: An illustration of the hard instance with $H = 4$. Recall that $\hat{d} = d/2 - 2$. States on the top are those in the first level ($h = 1$), while states at the bottom are those in the last level ($h = 4$). Solid line (with arrow) corresponds to transitions associated with action a_1, while dotted line (with arrow) corresponds to transitions associated with action a_2. We omit the transition associated with a_2 in the figure if a_1 and a_2 give the same transition. For each level $h \in [H]$, Q-values associated with $s_h^1, s_h^2, \ldots, s_h^{\hat{d}}, s_h^+, s_h^-$ are marked on the left, while transition distributions and Q-values associated with s_h^* are marked on the right. Rewards are all deterministic, and the only two states (s_4^+ and s_4^-) with non-zero reward values are marked in black and grey. For each level $h \in [H]$, for the data distribution μ_h, the state is chosen uniformly at random from those states in the dashed rectangle, i.e., $s_h^1, s_h^2, \ldots, s_h^{\hat{d}}, s_h^+, s_h^-$, while the action is chosen uniformly at random from $\{a_1, a_2\}$. Suppose the initial state is s_1^*. When $\eta = 0$, the value of the policy is 0. When $\eta = \hat{d}^{-(H-1)/2} = \hat{d}^{-3/2}$, the value of the policy is $Q(s_1^*, a) = \eta\hat{d}^{(H-1)/2} = 1$.

Reward Values. In this hard case, all reward values are deterministic, and reward values can be non-zero only for the last level. Formally, we have

$$R(s, a) = \begin{cases} 1 & s = s_H^+ \\ -1 & s = s_H^- \\ 0 & \text{otherwise} \end{cases}.$$

Feature Mapping. As in the in hard instance in Section 4, let $z_1, z_2, \ldots, z_d$ be a set of of orthonormal vectors in $\mathbb{R}^d$. For each $(h, c) \in [H] \times [\hat{d}]$,

$$\phi(s_h^c, a) = \begin{cases} z_c & a = a_1 \\ z_{c+\hat{d}} & a = a_2 \end{cases}.$$

Moreover, for all actions $a \in \mathcal{A}$,

$$\phi(s_h^*, a) = \frac{1}{\sqrt{\hat{d}}} \left(z_1 + z_2 + \ldots + z_{\hat{d}}\right).$$

Finally, for all $a \in \mathcal{A}$,

$$\phi(s_h^+, a) = \begin{cases} z_{d-3} & a = a_1 \\ z_{d-2} & a = a_2 \end{cases}.$$

and

$$\phi(s_h^-, a) = \begin{cases} z_{d-1} & a = a_1 \\ z_d & a = a_2 \end{cases}.$$

Clearly, for all $(s, a) \in \mathcal{S} \times \mathcal{A}$, $\|\phi(s, a)\|_2 \leq 1$.

Linearity of the Q-function. Now we consider the deterministic policy $\pi : \mathcal{S} \to \mathcal{A}$, which is defined to be $\pi(s) = a_1$ for all $s \in \mathcal{S}$. We show that Q^π is a linear function with respect to the feature mapping $\phi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$ defined above. Here we directly define the vectors $\{\theta_h\}_{h=1}^H$. For each $h \in [H-2]$, define

$$\theta_h = \sum_{c=1}^{\hat{d}} \eta \cdot \hat{d}^{(H-h-1)/2} \cdot z_c + \sum_{c=1}^{\hat{d}} \eta \cdot \hat{d}^{(H-h-2)/2} \cdot z_{c+\hat{d}} + z_{d-3} + z_{d-2} - z_{d-1} - z_d.$$

For the second last level $h = H-1$, define

$$\theta_{H-1} = \sum_{c=1}^{\hat{d}} \eta \cdot z_c + z_{d-3} + z_{d-2} - z_{d-1} - z_d.$$

Finally, for the last level $h = H$, define

$$\theta_H = z_{d-3} + z_{d-2} - z_{d-1} - z_d.$$

It can be verified (as we have done in Section 4) that for each $h \in [H]$, $Q_h^\pi(s,a) = \theta_h^\top \phi(s,a)$ for all $(s,a) \in \mathcal{S}_h \times \mathcal{A}$.

The Data Distributions. Similar to the construction in Section 4, for each level $h \in [H]$, the data distribution μ_h is a uniform distribution over

$$\{(s_h^1, a_1), (s_h^1, a_2), (s_h^2, a_1), (s_h^2, a_2), \ldots, (s_h^{\hat{d}}, a_1), (s_h^{\hat{d}}, a_2), (s_h^+, a_1), (s_h^+, a_2), (s_h^-, a_1), (s_h^-, a_2)\}.$$

Notice that again (s_h^*, a) is *not* in the support of μ_h for all actions $a \in \mathcal{A}$. It can be seen that

$$\mathbb{E}_{(s,a)\sim\mu_h}[\phi(s,a)\phi(s,a)^\top] = \frac{1}{d}\sum_{c=1}^{d} z_c z_c^\top = \frac{1}{d} I.$$

The Lower Bound. Now we show that it is information-theoretically hard for any algorithm to distinguish the case $\eta = 0$ and $\eta = \hat{d}^{-(H-1)/2}$ in the batch setting by taking samples from the data distributions $\{\mu_h\}_{h=1}^H$. Here we fix the initial state to be s_1^*, and consider the above policy π defined above which returns action a_1 for all input states. Notice that when $\eta = 0$, the value of the policy would be zero. On the other hand, when $\eta = \hat{d}^{-(H-1)/2}$, the value of the policy would be

$$Q(s_1^*, \pi(s_1^*)) = Q(s_1^*, a_1) = \eta \cdot \hat{d}^{(H-1)/2} = 1.$$

Therefore, if the algorithm approximates the value of the policy up to an approximation error of $1/2$, then it must distinguish the case that $\eta = 0$ and $\eta = \hat{d}^{-(H-1)/2}$.

We first notice that for the case $\eta = 0$ and $\eta = \hat{d}^{-(H-1)/2}$, the data distributions $\{\mu_h\}_{h=1}^H$, the feature mapping $\phi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$, the policy π to be evaluated and the reward distributions R are the same. Thus, in order to distinguish the case $\eta = 0$ and $\eta = \hat{d}^{-(H-1)/2}$, the only way is to query the transition operator P by using sampling taken from the data distributions.

Now, for all state-action pairs (s,a) in the support of the data distributions of the first $H-2$ levels (namely $\mu_1, \mu_2, \ldots, \mu_{H-2}$), the transition operator will be identical. This is because changing η only changes the transition probabilities of state-actions pairs of the form (s_h^*, a), and such state-actions are not in the support of μ_h for all $h \in [H]$. Moreover, for any $s \in \{s_{H-1}^+, s_{H-1}^-\}$ and $a \in \mathcal{A}$, $P(s,a)$ will also be identical no matter $\eta = 0$ or $\eta = \hat{d}^{-(H-1)/2}$. For those state-action pairs (s,a) in the support of μ_{H-1} with $s \notin \{s_{H-1}^+, s_{H-1}^-\}$, we have

$$P(s,a) = \begin{cases} s_H^+ & \text{with probability } (1+\eta)/2 \\ s_H^- & \text{with probability } (1-\eta)/2 \end{cases}.$$

Again, this is because (s_{H-1}^*, a) is not in the support of μ_{H-1} for all $a \in \mathcal{A}$.

Therefore, in order to distinguish the case $\eta = 0$ and $\eta = \hat{d}^{-(H-1)/2}$, the agent needs distinguish two transition distributions

$$p_1 = \begin{cases} s_H^+ & \text{with probability } 1/2 \\ s_H^- & \text{with probability } 1/2 \end{cases}$$

and

$$p_2 = \begin{cases} s_H^+ & \text{with probability } (1+\hat{d}^{-(H-1)/2})/2 \\ s_H^- & \text{with probability } (1-\hat{d}^{-(H-1)/2})/2 \end{cases}.$$

Again, by Lemma 5.1 in (Anthony & Bartlett, 2009), in order to distinguish p_1 and p_2 with probability at least 0.9, one needs $\Omega(\hat{d}^{H-1})$ samples. Formally, we have the following theorem.

Theorem B.1. *Suppose Assumption 1 and Assumption 2 hold. For any algorithm which takes as input both the policy π and the feature mapping ϕ, there exists an MDP such that the reward values are deterministic and could be none-zero only for state-action pairs in the last level, and the algorithm requires $\Omega((d/2-2)^{H-1})$ samples to output the value of the policy π up to constant constant approximation error with probability at least 0.9.*

C Hardness Result For Policy Improvement

Below we give a simple reduction to demonstrate the hardness of policy improvement. At the initial state, if the agent chooses action a_1, then the agent receives a fixed reward value (say 0.5) and terminates. If the agent chooses action a_2, then the agent transits to our hard instance constructed in Section 4. Note that for the hard instance constructed in Section 4, the given policy π that returns action a_1 for all input states is indeed the optimal policy. Therefore, in order to find a policy with suboptimality at most 0.5, the agent must evaluate the value of π up to an error of 0.5, and hence the hardness result holds. Moreover, since the given π is the optimal policy, and Q^π is linear with respect to the given feature mapping, the optimal Q-function is thus linear with respect to the feature mapping.

D Analysis of Algorithm 1

We define Φ_h to be a $N \times d$ matrix, whose i-th row is $\phi(s_h^i, a_h^i)$, and define $\overline{\Phi}_{h+1}$ to be another $N \times d$ matrix whose i-th row is $\phi(\overline{s}_h^i, \pi(\overline{s}_h^i))$. For each $h \in [H]$ and $i \in [N]$, define $\xi_h^i = r_h^i + V(\overline{s}_h^i) - Q(s_h^i, a_h^i)$. Clearly, $\mathbb{E}[\xi_h^i] = 0$ and $|\xi_h^i| \leq 2H$. We also use Ξ_h to denote a vector whose i-th entry is ξ_h^i.

By matrix concentration inequality (Tropp, 2015), we have the following lemma.

Lemma D.1. *For each $h \in [H]$, with probability $1-\delta/(4H)$, for some universal constant C, we have*

$$\left\| \frac{1}{N}\Phi_h^\top \Phi_h - \Lambda_h \right\|_2 \leq C\sqrt{d\log(dH/\delta)/N}.$$

and

$$\left\| \frac{1}{N}\overline{\Phi}_{h+1}\overline{\Phi}_{h+1} - \overline{\Lambda}_{h+1} \right\|_2 \leq C\sqrt{d\log(dH/\delta)/N}.$$

Therefore, since $\lambda = CH\sqrt{d\log(dH/\delta)N}$, with probability $1-\delta/(4H)$, we have

$$\hat{\Lambda}_h = \Phi_h^\top \Phi_h + \lambda I \succeq N\Lambda_h.$$

Clearly,

$$
\begin{aligned}
\hat{\theta}_h &= \hat{\Lambda}_h^{-1}\left(\sum_{i=1}^N \phi(s_h^i, a_h^i)\cdot(r_h^i + \hat{V}_{h+1}(\overline{s}_h^i))\right)\\
&= \hat{\Lambda}_h^{-1}\left(\sum_{i=1}^N \phi(s_h^i, a_h^i)\cdot(r_h^i + \hat{Q}_{h+1}(\overline{s}_h^i, \pi(\overline{s}_h^i)))\right)\\
&= \hat{\Lambda}_h^{-1}\left(\sum_{i=1}^N \phi(s_h^i, a_h^i)\cdot(r_h^i + \phi(\overline{s}_h^i, \pi(\overline{s}_h^i))^\top\hat{\theta}_{h+1})\right)\\
&= \hat{\Lambda}_h^{-1}\left(\sum_{i=1}^N \phi(s_h^i, a_h^i)\cdot(r_h^i + \phi(\overline{s}_h^i, \pi(\overline{s}_h^i))^\top\theta_{h+1}) + \sum_{i=1}^N \phi(s_h^i, a_h^i)\cdot\phi(\overline{s}_h^i, \pi(\overline{s}_h^i))^\top(\hat{\theta}_{h+1} - \theta_{h+1})\right)\\
&= \hat{\Lambda}_h^{-1}\left(\sum_{i=1}^N \phi(s_h^i, a_h^i)\cdot(r_h^i + \phi(\overline{s}_h^i, \pi(\overline{s}_h^i))^\top\theta_{h+1})\right) + \hat{\Lambda}_h^{-1}\left(\sum_{i=1}^N \phi(s_h^i, a_h^i)\cdot\phi(\overline{s}_h^i, \pi(\overline{s}_h^i))^\top(\hat{\theta}_{h+1} - \theta_{h+1})\right).
\end{aligned}
$$

For the first term, we have

$$
\begin{aligned}
&\hat{\Lambda}_h^{-1}\left(\sum_{i=1}^N \phi(s_h^i, a_h^i)\cdot(r_h^i + \phi(\overline{s}_h^i, \pi(\overline{s}_h^i))^\top\theta_{h+1})\right)\\
=&\hat{\Lambda}_h^{-1}\left(\sum_{i=1}^N \phi(s_h^i, a_h^i)\cdot(r_h^i + Q^\pi(\overline{s}_h^i, \pi(\overline{s}_h^i)))\right)\\
=&\hat{\Lambda}_h^{-1}\left(\sum_{i=1}^N \phi(s_h^i, a_h^i)\cdot(r_h^i + V^\pi(\overline{s}_h^i))\right)\\
=&\hat{\Lambda}_h^{-1}\left(\sum_{i=1}^N \phi(s_h^i, a_h^i)\cdot(Q^\pi(s_h^i, a_h^i) + \xi_h^i)\right)\\
=&\hat{\Lambda}_h^{-1}\sum_{i=1}^N \phi(s_h^i, a_h^i)\cdot\xi_h^i + \hat{\Lambda}_h^{-1}\sum_{i=1}^N \phi(s_h^i, a_h^i)\cdot\phi(s_h^i, a_h^i)^\top\theta_h\\
=&\hat{\Lambda}_h^{-1}\sum_{i=1}^N \phi(s_h^i, a_h^i)\cdot\xi_h^i + \hat{\Lambda}_h^{-1}(\Phi_h^\top\Phi_h)\theta_h\\
=&\hat{\Lambda}_h^{-1}\Phi_h\Xi_h + \theta_h - \lambda\hat{\Lambda}_h^{-1}\theta_h.
\end{aligned}
$$

Therefore,

$$
\begin{aligned}
\hat{\theta}_1 - \theta_1 &= (\hat{\Lambda}_1^{-1}\Phi_1\Xi_1 - \lambda\hat{\Lambda}_1^{-1}\theta_1) + \hat{\Lambda}_1^{-1}\Phi_1^\top\overline{\Phi}_2(\theta_2 - \hat{\theta}_2)\\
&= (\hat{\Lambda}_1^{-1}\Phi_1\Xi_1 - \lambda\hat{\Lambda}_1^{-1}\theta_1) + \hat{\Lambda}_1^{-1}\Phi_1^\top\overline{\Phi}_2(\hat{\Lambda}_2^{-1}\Phi_2^\top\Xi_2 - \lambda\hat{\Lambda}_2^{-1}\theta_2)\\
&+ \hat{\Lambda}_1^{-1}\Phi_1^\top\overline{\Phi}_2\hat{\Lambda}_2^{-1}\Phi_2^\top\overline{\Phi}_3(\theta_3 - \hat{\theta}_3)\\
&= \dots\\
&= \sum_{h=1}^H \hat{\Lambda}_1^{-1}\Phi_1^\top\overline{\Phi}_2\hat{\Lambda}_2^{-1}\Phi_2^\top\overline{\Phi}_3\cdots(\hat{\Lambda}_h^{-1}\Phi_h^\top\Xi_h - \lambda\hat{\Lambda}_h^{-1}\theta_h).
\end{aligned}
$$

Notice that

$$
\mathbb{E}_{s_1}[(Q^\pi(s_1, \pi(s_1)) - \hat{Q}(s_1, \pi(s_1)))^2] = \mathbb{E}_{s_1}\left[\left(\phi(s_1, \pi(s_1))^\top(\theta_1 - \hat{\theta}_1)\right)^2\right] = \|\theta_1 - \hat{\theta}_1\|_{\Lambda_1}^2.
$$

Thus,

$$
\begin{aligned}
&\mathbb{E}_{s_1}[(Q^\pi(s_1,\pi(s_1)) - \hat{Q}(s_1,\pi(s_1)))^2] \\
\leq & H \cdot \left(\sum_{h=1}^{H} \left\| \hat{\Lambda}_1^{-1}\Phi_1^\top \overline{\Phi}_2 \hat{\Lambda}_2^{-1}\Phi_2^\top \overline{\Phi}_3 \cdots (\hat{\Lambda}_h^{-1}\Phi_h^\top \Xi_h - \lambda \hat{\Lambda}_h^{-1}\theta_h) \right\|_{\overline{\Lambda}_1}^2 \right) \\
\leq & 2H \cdot \left(\sum_{h=1}^{H} \left\| \hat{\Lambda}_1^{-1}\Phi_1^\top \overline{\Phi}_2 \hat{\Lambda}_2^{-1}\Phi_2^\top \overline{\Phi}_3 \cdots \hat{\Lambda}_h^{-1}\Phi_h^\top \Xi_h \right\|_{\overline{\Lambda}_1}^2 + \sum_{h=1}^{H} \left\| \hat{\Lambda}_1^{-1}\Phi_1^\top \overline{\Phi}_2 \hat{\Lambda}_2^{-1}\Phi_2^\top \overline{\Phi}_3 \cdots \lambda \hat{\Lambda}_h^{-1}\theta_h \right\|_{\overline{\Lambda}_1}^2 \right).
\end{aligned}
$$

For each $h \in [H]$,

$$
\begin{aligned}
&\|\hat{\Lambda}_1^{-1}\Phi_1^\top \overline{\Phi}_2 \hat{\Lambda}_2^{-1}\Phi_2^\top \overline{\Phi}_3 \cdots \hat{\Lambda}_h^{-1}\Phi_h^\top \Xi_h\|_{\overline{\Lambda}_1}^2 \\
\leq & \|\Phi_1 \hat{\Lambda}_1^{-1}\overline{\Lambda}_1 \hat{\Lambda}_1^{-1}\Phi_1^\top\|_2 \cdot \|\overline{\Phi}_2 \hat{\Lambda}_2^{-1}\Phi_2^\top \overline{\Phi}_3 \cdots \hat{\Lambda}_h^{-1}\Phi_h^\top \Xi_h\|_2^2 \\
\leq & \|\hat{\Lambda}_1^{-1/2}\overline{\Lambda}_1 \hat{\Lambda}_1^{-1/2}\|_2 \cdot \|\Phi_1 \hat{\Lambda}_1^{-1}\Phi_1^\top\|_2 \cdot \|\overline{\Phi}_2 \hat{\Lambda}_2^{-1}\Phi_2^\top \overline{\Phi}_2 \cdots \hat{\Lambda}_h^{-1}\Phi_h^\top \Xi_h\|_2^2 \\
\leq & \|\hat{\Lambda}_1^{-1/2}\overline{\Lambda}_1 \hat{\Lambda}_1^{-1/2}\|_2 \cdot \prod_{h'=1}^{h-1} \left(\|\Phi_{h'} \hat{\Lambda}_{h'}^{-1}\Phi_{h'}^\top\|_2 \cdot \|\hat{\Lambda}_{h'+1}^{-1/2}(\overline{\Phi}_{h'+1}^\top \overline{\Phi}_{h'+1})\hat{\Lambda}_{h'+1}^{-1/2}\|_2 \right) \cdot \|\Xi_h\|_{\Phi_h \hat{\Lambda}_h^{-1}\Phi_h^\top}^2.
\end{aligned}
$$

Similarly,

$$
\begin{aligned}
&\|\hat{\Lambda}_1^{-1}\Phi_1^\top \overline{\Phi}_2 \hat{\Lambda}_2^{-1}\Phi_2^\top \overline{\Phi}_3 \cdots \lambda \hat{\Lambda}_h^{-1}\theta_h\|_{\overline{\Lambda}_1}^2 \\
\leq & \|\hat{\Lambda}_1^{-1/2}\overline{\Lambda}_1 \hat{\Lambda}_1^{-1/2}\|_2 \cdot \prod_{h'=1}^{h-1} \left(\|\Phi_{h'} \hat{\Lambda}_{h'}^{-1}\Phi_{h'}^\top\|_2 \cdot \|\hat{\Lambda}_{h'+1}^{-1/2}(\overline{\Phi}_{h'+1}^\top \overline{\Phi}_{h'+1})\hat{\Lambda}_{h+1}^{-1/2}\|_2 \right) \cdot \lambda^2 \cdot \|\theta_h\|_{\hat{\Lambda}_h^{-1}}^2 \\
\leq & \|\hat{\Lambda}_1^{-1/2}\overline{\Lambda}_1 \hat{\Lambda}_1^{-1/2}\|_2 \cdot \prod_{h'=1}^{h-1} \left(\|\Phi_{h'} \hat{\Lambda}_{h'}^{-1}\Phi_{h'}^\top\|_2 \cdot \|\hat{\Lambda}_{h'+1}^{-1/2}(\overline{\Phi}_{h'+1}^\top \overline{\Phi}_{h'+1})\hat{\Lambda}_{h'+1}^{-1/2}\|_2 \right) \cdot \lambda \cdot d.
\end{aligned}
$$

For all $h \in [H]$, we have

$$\|\Phi_h \hat{\Lambda}_h^{-1}\Phi_h^\top\|_2 \leq 1$$

and

$$\|\hat{\Lambda}_h^{-1/2}(\overline{\Phi}_h^\top \overline{\Phi}_h)\hat{\Lambda}_h^{-1/2}\|_2 \leq \|N\hat{\Lambda}_h^{-1/2}\overline{\Lambda}_h \hat{\Lambda}_h^{-1/2}\|_2 + \|\hat{\Lambda}_h^{-1/2}(\overline{\Phi}_h^\top \overline{\Phi}_h - N\overline{\Lambda}_h)\hat{\Lambda}_h^{-1/2}\|_2.$$

Conditioned on the event in Lemma D.1,

$$\hat{\Lambda}_h \succeq N\Lambda_h \succeq \frac{N}{C_h}\overline{\Lambda}_h,$$

which implies $\|N\hat{\Lambda}_h^{-1/2}\overline{\Lambda}_h \hat{\Lambda}_h^{-1/2}\| \leq C_h$. Moreover, conditioned on the event in Lemma D.1,

$$\|\hat{\Lambda}_h^{-1/2}(\overline{\Phi}_h^\top \overline{\Phi}_h - N\overline{\Lambda}_h)\hat{\Lambda}_h^{-1/2}\|_2 \leq C\sqrt{d\log(dH/\delta)N}/\lambda.$$

Thus,

$$\|\hat{\Lambda}_1^{-1/2}\overline{\Lambda}_1 \hat{\Lambda}_1^{-1/2}\|_2 \leq C_1/N.$$

and

$$\|\hat{\Lambda}_h^{-1/2}(\overline{\Phi}_h^\top \overline{\Phi}_h)\hat{\Lambda}_h^{-1/2}\|_2 \leq C_h + C\sqrt{d\log(dH/\delta)N}/\lambda.$$

Finally, by Theorem 1.2 in (Hsu et al., 2012a), with probability $1 - \delta/(4H)$, for some constant C', we have

$$\|\Xi_h\|_{\Phi_h \hat{\Lambda}_h^{-1}\Phi_h^\top}^2 \leq C' H^2 d\log(1/\delta).$$

Therefore,

$$
\begin{aligned}
&\left\| \hat{\Lambda}_1^{-1}\Phi_1^\top \overline{\Phi}_2 \hat{\Lambda}_2^{-1}\Phi_2^\top \overline{\Phi}_3 \cdots \hat{\Lambda}_h^{-1}\Phi_h^\top \Xi_h \right\|_{\overline{\Lambda}_1}^2 + \left\| \hat{\Lambda}_1^{-1}\Phi_1^\top \overline{\Phi}_2 \hat{\Lambda}_2^{-1}\Phi_2^\top \overline{\Phi}_3 \cdots \lambda \hat{\Lambda}_h^{-1}\theta_h \right\|_{\overline{\Lambda}_1}^2 \\
\leq & \frac{C_1}{N}(C_2 + C\sqrt{d\log(d/\delta)N}/\lambda) \times \cdots \times (C_h + C\sqrt{d\log(d/\delta)N}/\lambda) \times (C' H^2 d\log(1/\delta) + \lambda) \\
\leq & \frac{C_1}{N}(C_2 + 1/H) \times \cdots \times (C_h + 1/H) \times (C' H^2 d\log(1/\delta) + \lambda d) \\
\leq & \frac{e}{N} C_1 \times C_2 \times \cdots \times C_h \times (C' H^2 d\log(1/\delta) + CdH\sqrt{d\log(dH/\delta)N}).
\end{aligned}
$$

Let $c > 0$ be a large enough constant. We now have

$$\mathbb{E}_{s_1}[(Q_1^\pi(s_1, \pi(s_1)) - \hat{Q}_1(s_1, \pi(s_1)))^2] \leq c \cdot \left(\prod_{h=1}^{H} C_h\right) \cdot dH^3 \cdot \left(\sqrt{\frac{d\log(dH/\delta)}{N}} + \frac{H\log(1/\delta)}{N}\right).$$