

UCLA

UCLA Previously Published Works

Title

Near-Optimal Time and Sample Complexities for Solving Discounted Markov Decision Process with a Generative Model

Permalink

<https://escholarship.org/uc/item/1kb5z47v>

Authors

Sidford, Aaron
Wang, Mengdi
Wu, Xian
et al.

Publication Date

2018-06-04

Peer reviewed

Near-Optimal Time and Sample Complexities for Solving Discounted Markov Decision Process with a Generative Model

Aaron Sidford
Stanford University
sidford@stanford.edu

Mengdi Wang
Princeton University
mengdiw@princeton.edu

Xian Wu
Stanford University
xwu20@stanford.edu

Lin F. Yang
Princeton University
lin.yang@princeton.edu

Yinyu Ye
Stanford University
yyye@stanford.edu

June 7, 2019

Abstract

In this paper we consider the problem of computing an ϵ -optimal policy of a discounted Markov Decision Process (DMDP) provided we can only access its transition function through a generative sampling model that given any state-action pair samples from the transition function in $O(1)$ time. Given such a DMDP with states \mathcal{S} , actions \mathcal{A} , discount factor $\gamma \in (0, 1)$, and rewards in range $[0, 1]$ we provide an algorithm which computes an ϵ -optimal policy with probability $1 - \delta$ where *both* the time spent and number of sample taken are upper bounded by

$$O \left[\frac{|\mathcal{S}||\mathcal{A}|}{(1-\gamma)^3 \epsilon^2} \log \left(\frac{|\mathcal{S}||\mathcal{A}|}{(1-\gamma)\delta\epsilon} \right) \log \left(\frac{1}{(1-\gamma)\epsilon} \right) \right].$$

For fixed values of $\epsilon \in (0, 1)$, this improves upon the previous best known bounds by a factor of $(1-\gamma)^{-1}$ and matches the sample complexity lower bounds proved in [AMK13] up to logarithmic factors. We also extend our method to computing ϵ -optimal policies for finite-horizon MDP with a generative model and provide a nearly matching sample complexity lower bound.

1 Introduction

Markov decision processes (MDPs) are a fundamental mathematical abstraction used to model sequential decision making under uncertainty and are a basic model of discrete-time stochastic control and reinforcement learning (RL). Particularly central to RL is the case of computing or learning an approximately optimal policy when the MDP itself is not fully known beforehand. One of the simplest such settings is when the states, rewards, and actions are all known but the transition between states when an action is taken is probabilistic, unknown, and can only be sampled from.

Computing an approximately optimal policy with high probability in this case is known as PAC RL with a generative model. It is a well studied problem with multiple existing results providing algorithms with improved the sample complexity (number of sample transitions taken) and running time (the total time of the algorithm) under various MDP reward structures, e.g. discounted infinite-horizon, finite-horizon, etc. (See Section 2 for a detailed review of the literature.)

In this work, we consider this well studied problem of computing approximately optimal policies of discounted infinite-horizon Markov Decision Processes (DMDP) under the assumption we can only access the DMDP by sampling state transitions. Formally, we suppose that we have a DMDP with a known set of states, \mathcal{S} , a known set of actions that can be taken at each states, \mathcal{A} , a known reward $r_{s,a} \in [0, 1]$ for taking action $a \in \mathcal{A}$ at state $s \in \mathcal{S}$, and a discount factor $\gamma \in (0, 1)$. We assume that taking action a at state s probabilistically transitions an agent to a new state based on a fixed, but unknown probability vector $\mathbf{P}_{s,a}$. The objective is to maximize the cumulative sum of discounted rewards in expectation. Throughout this paper, we assume that we have a *generative model*, a notion introduced by [Kak03], which allows us to draw random state transitions of the DMDP. In particular, we assume that we can sample from the distribution defined by $\mathbf{P}_{s,a}$ for all $(s, a) \in \mathcal{S} \times \mathcal{A}$ in $O(1)$ time. This is a natural assumption and can be achieved in expectation in certain computational models with linear time preprocessing of the DMDP.¹

The main result of this paper is that we provide the first algorithm that is sample-optimal and runtime-optimal (up to polylogarithmic factors) for computing an ϵ -optimal policy of a DMDP with a generative model (in the regime of $1/\sqrt{(1-\gamma)|\mathcal{S}|} \leq \epsilon \leq 1$). In particular, we develop a randomized Variance-Reduced Q-Value Iteration (vQVI) based algorithm that computes an ϵ -optimal policy with probability $1 - \delta$ with a number of samples, i.e. queries to the generative model, bound by

$$O \left[\frac{|\mathcal{S}||\mathcal{A}|}{(1-\gamma)^3 \epsilon^2} \log \left(\frac{|\mathcal{S}||\mathcal{A}|}{(1-\gamma)\delta\epsilon} \right) \log \left(\frac{1}{(1-\gamma)\epsilon} \right) \right].$$

This result matches (up to polylogarithmic factors) the following sample complexity lower bound established in [AMK13] for finding ϵ -optimal policies with probability $1 - \delta$ (see Appendix D):

$$\Omega \left[\frac{|\mathcal{S}||\mathcal{A}|}{(1-\gamma)^3 \epsilon^2} \log \left(\frac{|\mathcal{S}||\mathcal{A}|}{\delta} \right) \right].$$

Furthermore, we show that the algorithm can be implemented using sparse updates such that the overall run-time complexity is equal to its sample complexity up to constant factors, as long as each sample transition can be generated in $O(1)$ time. Consequently, up to logarithmic factors our run time complexity is optimal as well. In addition, the algorithm's space complexity is $\Theta(|\mathcal{S}||\mathcal{A}|)$.

Our method and analysis builds upon a number of prior works. (See Section 2 for an in-depth comparison.) The paper [AMK13] provided the first algorithm that achieves the optimal

¹If instead the oracle needed time τ , every running time result in this paper should be multiplied by τ .

sample complexity for finding ϵ -optimal value functions (rather than ϵ -optimal policy), as well as the matching lower bound. Unfortunately an ϵ -optimal value function does not imply an ϵ -optimal policy and if we directly use the method of [AMK13] to get an ϵ -optimal policy for constant ϵ , the best known sample complexity is $\tilde{O}(|\mathcal{S}||\mathcal{A}|(1-\gamma)^{-5}\epsilon^{-2})$.² This bound is known to be improvable through related work of [SWWY18] which provides a method for computing an ϵ -optimal policy using $\tilde{O}(|\mathcal{S}||\mathcal{A}|(1-\gamma)^{-4}\epsilon^{-2})$ samples and total runtime and the work of [AMK13] which in the regime of small approximation error, i.e. where $\epsilon = O((1-\gamma)^{-1/2}|\mathcal{S}|^{-1/2})$, already provides a method that achieves the optimal sample complexity. However, when the approximation error takes fixed values, e.g. $\epsilon \geq \Omega((1-\gamma)^{-1/2}|\mathcal{S}|^{-1/2})$, there remains a gap between the best known runtime and sample complexity for computing an ϵ -optimal policy and the theoretical lower bounds. For fixed values of ϵ , which mostly occur in real applications, our algorithm improves upon the previous best sample and time complexity bounds by a factor of $(1-\gamma)^{-1}$ where $\gamma \in (0, 1)$, the discount factor, is typically close to 1.

We achieve our results by combining and strengthening techniques from both [AMK13] and [SWWY18]. On the one hand, in [AMK13] the authors showed that simply constructing a “sparsified” MDP model by taking samples and then solving this model to high precision yields a sample optimal algorithm in our setting for computing the approximate value of every state. On the other hand, [SWWY18] provided faster algorithms for solving explicit DMDPs and improved sample and time complexities given a sampling oracle. In fact, as we show in Appendix B.1, simply combining these two results yields the first nearly optimal runtime for approximately learning the value function with a generative model. Unfortunately, it is known that an approximate-optimal value function does not immediately yield an approximate-optimal policy of comparable quality (see e.g. [Ber13]) and it is was previously unclear how to combine these methods to improve upon previous known bounds for computing an approximate policy. To achieve our policy computation algorithm we therefore open up both the algorithms and the analysis in [AMK13] and [SWWY18], combining them in nontrivial ways. Our proofs leverage techniques ranging from standard probabilistic analysis tools such as Hoeffding and Bernstein inequalities, to optimization techniques such as variance reduction, to properties specific to MDPs such as the Bellman fixed-point recursion for expectation and variance of the optimal value vector, and monotonicity of value iteration.

Finally, we extend our method to finite-horizon MDPs, which are also occurred frequently in real applications. We show that the number of samples needed by this algorithm is $\tilde{O}(H^3|\mathcal{S}||\mathcal{A}|\epsilon^{-2})$, in order to obtain an ϵ -optimal policy for H -horizon MDP (see Appendix F). We also show that the preceding sample complexity is optimal up to logarithmic factors by providing a matching lower bound. We hope this work ultimately opens the door for future practical and theoretical work on solving MDPs and efficient RL more broadly.

2 Comparison to Previous Work

There exists a large body of literature on MDPs and RL (see e.g. [Kak03, SLL09, KBJ14, DB15] and reference therein). The classical MDP problem is to compute an optimal policy exactly or

²[AMK13] showed that one can obtain ϵ -optimal *value* v (instead of ϵ -optimal policy) using sample size $\propto (1-\gamma)^{-3}\epsilon^{-2}$. By using this ϵ -optimal value v , one can get a greedy policy that is $[(1-\gamma)^{-1}\epsilon]$ -optimal. By setting $\epsilon \rightarrow (1-\gamma)\epsilon$, one can obtain an ϵ -optimal policy, using the number of samples $\propto (1-\gamma)^{-5}\epsilon^{-2}$.

³Although not explicitly stated, an immediate derivation shows that obtaining an ϵ -optimal policy in [AMK13] requires $O(|\mathcal{S}||\mathcal{A}|(1-\gamma)^{-5}\epsilon^{-2})$ samples.

Algorithm	Sample Complexity	References
Phased Q-Learning	$\tilde{O}(C \frac{ \mathcal{S} \mathcal{A} }{(1-\gamma)^7 \epsilon^2})$	[KS99]
Empirical QVI	$\tilde{O}(\frac{ \mathcal{S} \mathcal{A} }{(1-\gamma)^3 \epsilon^2})^3$	[AMK13]
Empirical QVI	$\tilde{O}(\frac{ \mathcal{S} \mathcal{A} }{(1-\gamma)^3 \epsilon^2})$ if $\epsilon = \tilde{O}(\frac{1}{\sqrt{(1-\gamma) \mathcal{S} }})$	[AMK13]
Randomized Primal-Dual Method	$\tilde{O}(C \frac{ \mathcal{S} \mathcal{A} }{(1-\gamma)^4 \epsilon^2})$	[Wan17]
Sublinear Randomized Value Iteration	$\tilde{O}(\frac{ \mathcal{S} \mathcal{A} }{(1-\gamma)^4 \epsilon^2})$	[SWWY18]
Sublinear Randomized QVI	$\tilde{O}(\frac{ \mathcal{S} \mathcal{A} }{(1-\gamma)^3 \epsilon^2})$	This Paper

Table 1: **Sample Complexity to Compute ϵ -Approximate Policies Using the Generative Sampling Model:** Here $|\mathcal{S}|$ is the number of states, $|\mathcal{A}|$ is the number of actions per state, $\gamma \in (0, 1)$ is the discount factor, and C is an upper bound on the ergodicity. Rewards are bounded between 0 and 1.

approximately, when the full MDP model is given as input. For a survey on existing complexity results when the full MDP model is given, see Appendix A.

Despite the aforementioned results of [Kak03, AMK13, SWWY18], there exists only a handful of additional RL methods that achieve a small sample complexity and a small run-time complexity at the same time for computing an ϵ -optimal policy. A classical result is the phased Q-learning method by [KS99], which takes samples from the generative model and runs a randomized value iteration. The phased Q-learning method finds an ϵ -optimal policy using $\mathcal{O}(|\mathcal{S}||\mathcal{A}|\epsilon^{-2}/\text{poly}(1-\gamma))$ samples/updates, where each update uses $\tilde{O}(1)$ run time.⁴ Another work [Wan17] gave a randomized mirror-prox method that applies to a special Bellman saddle point formulation of the DMDP. They achieve a total runtime of $\tilde{O}(|\mathcal{S}|^3|\mathcal{A}|\epsilon^{-2}(1-\gamma)^{-6})$ for the general DMDP and $\tilde{O}(C|\mathcal{S}||\mathcal{A}|\epsilon^{-2}(1-\gamma)^{-4})$ for DMDPs that are ergodic under all possible policies, where C is a problem-specific ergodicity measure. A recent closely related work is [SWWY18] which gave a variance-reduced randomized value iteration that works with the generative model and finds an ϵ -approximate policy in sample size/run time $\tilde{O}(|\mathcal{S}||\mathcal{A}|\epsilon^{-2}(1-\gamma)^{-4})$, without requiring any ergodicity assumption.

Finally, in the case where $\epsilon = O(1/\sqrt{(1-\gamma)^{-1}|\mathcal{S}|})$, [AMK13] showed that the solution obtained by performing exact PI on the empirical MDP model provides not only an ϵ -optimal value but also an ϵ -optimal policy. In this case, the number of samples is $\tilde{O}(|\mathcal{S}||\mathcal{A}|(1-\gamma)^{-3}\epsilon^{-2})$ and matches the sample complexity lower bound. Although this sample complexity is optimal, it requires solving the empirical MDP exactly (see Appendix B), and is no longer sublinear in the size of the MDP model because of the very small approximation error $\epsilon = O(1/\sqrt{(1-\gamma)|\mathcal{S}|})$. See Table 1 for a list of comparable sample complexity results for solving MDP based on the generative model.

3 Preliminaries

We use calligraphy upper case letters for sets or operators, e.g., \mathcal{S} , \mathcal{A} and \mathcal{T} . We use bold small case letters for vectors, e.g., \mathbf{v} , \mathbf{r} . We denote \mathbf{v}_s or $\mathbf{v}(s)$ as the s -th entry of vector \mathbf{v} . We denote matrix as bold upper case letters, e.g., \mathbf{P} . We denote constants as normal upper case letters, e.g.,

⁴The dependence on $(1-\gamma)$ in [KS99] is not stated explicitly but we believe basic calculations yield $O(1/(1-\gamma)^7)$.

M . For a vector $\mathbf{v} \in \mathbb{R}^{\mathcal{N}}$ for index set \mathcal{N} , we denote $\sqrt{\mathbf{v}}$, $|\mathbf{v}|$, and \mathbf{v}^2 vectors in $\mathbb{R}^{\mathcal{N}}$ with $\sqrt{\cdot}$, $|\cdot|$, and $(\cdot)^2$ acting coordinate-wise. For two vectors $\mathbf{v}, \mathbf{u} \in \mathbb{R}^{\mathcal{N}}$, we denote by $\mathbf{v} \leq \mathbf{u}$ as coordinate-wise comparison, i.e., $\forall i \in \mathcal{N} : \mathbf{v}(i) \leq \mathbf{u}(i)$. The same definition are defined to relations \leq , $<$ and $>$.

We describe a DMDP by the tuple $(\mathcal{S}, \mathcal{A}, \mathbf{P}, \mathbf{r}, \gamma)$, where \mathcal{S} is a finite state space, \mathcal{A} is a finite action space, $\mathbf{P} \in \mathbb{R}^{\mathcal{S} \times \mathcal{A} \times \mathcal{S}}$ is the state-action-state transition matrix, $\mathbf{r} \in \mathbb{R}^{\mathcal{S} \times \mathcal{A}}$ is the state-action reward vector, and $\gamma \in (0, 1)$ is a discount factor. We use $\mathbf{P}_{s,a}(s')$ to denote the probability of going to state s' from state s when taking action a . We also identify each $\mathbf{P}_{s,a}$ as a vector in $\mathbb{R}^{\mathcal{S}}$. We use $\mathbf{r}_{s,a}$ to denote the reward obtained from taking action $a \in \mathcal{A}$ at state $s \in \mathcal{S}$ and assume $\mathbf{r} \in [0, 1]^{\mathcal{S} \times \mathcal{A}}$.⁵ For a vector $\mathbf{v} \in \mathbb{R}^{\mathcal{S}}$, we denote $\mathbf{P}\mathbf{v} \in \mathbb{R}^{\mathcal{S} \times \mathcal{A}}$ as $(\mathbf{P}\mathbf{v})_{s,a} = \mathbf{P}_{s,a}^\top \mathbf{v}$. A policy $\pi : \mathcal{S} \rightarrow \mathcal{A}$ maps each state to an action. The objective of MDP is to find the optimal policy π^* that maximizes the expectation of the cumulative sum of discounted rewards.

In the remainder of this section we give definitions for several prominent concepts in MDP analysis that we use throughout the paper.

Definition 3.1 (Bellman Value Operator). For a given DMDP the *value operator* $\mathcal{T} : \mathbb{R}^{\mathcal{S}} \mapsto \mathbb{R}^{\mathcal{S}}$ is defined for all $\mathbf{u} \in \mathbb{R}^{\mathcal{S}}$ and $s \in \mathcal{S}$ by $\mathcal{T}(\mathbf{u})_s = \max_{a \in \mathcal{A}} [\mathbf{r}_a(s) + \gamma \cdot \mathbf{P}_{s,a}^\top \mathbf{u}]$, and we let \mathbf{v}^* denote the *value of the optimal policy* π^* , which is the unique vector such that $\mathcal{T}(\mathbf{v}^*) = \mathbf{v}^*$.

Definition 3.2 (Policy). We call any vector $\pi \in \mathcal{A}^{\mathcal{S}}$ a *policy* and say that the action prescribed by policy π to be taken at state $s \in \mathcal{S}$ is π_s . We let $\mathcal{T}_\pi : \mathbb{R}^{\mathcal{S}} \mapsto \mathbb{R}^{\mathcal{S}}$ denote the *value operator associated with* π defined for all $\mathbf{u} \in \mathbb{R}^{\mathcal{S}}$ and $s \in \mathcal{S}$ by $\mathcal{T}_\pi(\mathbf{u})_s = \mathbf{r}_{s,\pi(s)} + \gamma \cdot \mathbf{P}_{s,\pi(s)}^\top \mathbf{u}$, and we let \mathbf{v}^π denote the *values of policy* π , which is the unique vector such that $\mathcal{T}_\pi(\mathbf{v}^\pi) = \mathbf{v}^\pi$.

Note that \mathcal{T}_π can be viewed as the value operator for the modified MDP where the only available action from each state is given by the policy π . Note that this modified MDP is essentially just an uncontrolled Markov Chain, i.e. there are no action choices that can be made.

Definition 3.3 (ϵ -optimal value and policy). We say values $\mathbf{u} \in \mathbb{R}^{\mathcal{S}}$ are ϵ -*optimal* if $\|\mathbf{v}^* - \mathbf{u}\|_\infty \leq \epsilon$ and policy $\pi \in \mathcal{A}^{\mathcal{S}}$ is ϵ -*optimal* if $\|\mathbf{v}^* - \mathbf{v}^\pi\|_\infty \leq \epsilon$, i.e. the values of π are ϵ -optimal.

Definition 3.4 (Q-function). For any policy π , we define the Q-function of a MDP with respect to π as a vector $\mathbf{Q} \in \mathbb{R}^{\mathcal{S} \times \mathcal{A}}$ such that $\mathbf{Q}^\pi(s, a) = \mathbf{r}(s, a) + \gamma \mathbf{P}_{s,a}^\top \mathbf{v}^\pi$. The optimal Q-function is defined as $\mathbf{Q}^* = \mathbf{Q}^{\pi^*}$. We call any vector $\mathbf{Q} \in \mathbb{R}^{\mathcal{S} \times \mathcal{A}}$ a Q-function even though it may not relate to a policy or a value vector and define $\mathbf{v}(\mathbf{Q}) \in \mathbb{R}^{\mathcal{S}}$ and $\pi(\mathbf{Q}) \in \mathcal{A}^{\mathcal{S}}$ as the value and policy implied by \mathbf{Q} , by

$$\forall s \in \mathcal{S} : \mathbf{v}(\mathbf{Q})(s) = \max_{a \in \mathcal{A}} \mathbf{Q}(s, a) \quad \text{and} \quad \pi(\mathbf{Q})(s) = \arg \max_{a \in \mathcal{A}} \mathbf{Q}(s, a).$$

For a policy π , let $\mathbf{P}^\pi \mathbf{Q} \in \mathbb{R}^{\mathcal{S} \times \mathcal{A}}$ be defined as $(\mathbf{P}^\pi \mathbf{Q})(s, a) = \sum_{s' \in \mathcal{S}} \mathbf{P}_{s,a}(s') \mathbf{Q}(s', \pi(s'))$.

4 Technique Overview

In this section we provide a more detailed and technical overview of our approach. At a high level, our algorithm shares a similar framework as the variance reduction algorithm presented in [SWWY18]. This algorithm used two crucial algorithmic techniques, which are also critical in this paper. We call these techniques as the *monotonicity* technique and the *variance reduction* technique. Our algorithm and the results of this paper can be viewed as an advanced, non-trivial integration of these two methods, augmented with a third technique which we refer to as a *total-variation*

⁵A general $\mathbf{r} \in \mathbb{R}^{\mathcal{S} \times \mathcal{A}}$ can always be reduced to this case by shifting and scaling.

technique which was discovered in several papers [MM99, LH12, AMK13]. In the remainder of this section we give an overview of these techniques and through this, explain our algorithm.

The Monotonicity Technique Recall that the classic value iteration algorithm for solving a MDP repeatedly applies the following rule

$$\mathbf{v}^{(i)}(s) \leftarrow \max_a (r(s, a) + \gamma \mathbf{P}_{s,a}^\top \mathbf{v}^{(i-1)}). \quad (4.1)$$

A greedy policy $\pi^{(i)}$ can be obtained at each iteration i by

$$\forall s : \pi^{(i)}(s) \leftarrow \operatorname{argmax}_a (r(s, a) + \gamma \mathbf{P}_{s,a}^\top \mathbf{v}^{(i)}). \quad (4.2)$$

For any $u > 0$, it can be shown that if one can approximate $\mathbf{v}^{(i)}(s)$ with $\widehat{\mathbf{v}}^{(i)}(s)$ such that $\|\widehat{\mathbf{v}}^{(i)} - \mathbf{v}^{(i)}\|_\infty \leq (1 - \gamma)u$ and run the above value iteration algorithm using these approximated values, then after $\Theta((1 - \gamma)^{-1} \log[u^{-1}(1 - \gamma)^{-1}])$ iterations, the final iteration gives an value function that is u -optimal ([Ber13]). However, a u -optimal value function only yields a $u/(1 - \gamma)$ -optimal greedy policy (in the worst case), even if (4.2) is precisely computed. To get around this additional loss, a monotone-VI algorithm was proposed in [SWWY18] as follows. At each iteration, this algorithm maintains not only an approximated value $\mathbf{v}^{(i)}$ but also a policy $\pi^{(i)}$. The key for improvement is to keep values as a lower bound of the value of the policy on a set of sample paths with high probability. In particular, the following *monotonicity condition* was maintained with high probability

$$\mathbf{v}^{(i)} \leq \mathcal{T}_{\pi^{(i)}}(\mathbf{v}^{(i)}).$$

By the monotonicity of the Bellman's operator, the above equation guarantees that $\mathbf{v}^{(i)} \leq \mathbf{v}^{\pi^{(i)}}$. If this condition is satisfied, then, if after R iterations of approximate value iteration we obtain an value $\widehat{\mathbf{v}}^{(R)}$ that is u -optimal then we also obtain a policy $\pi^{(R)}$ which by the monotonicity condition and the monotonicity of the Bellman operator $\mathcal{T}_{\pi^{(R)}}$ yields

$$\mathbf{v}^{(R)} \leq \mathcal{T}_{\pi^{(R)}}(\mathbf{v}^{(R)}) \leq \mathcal{T}_{\pi^{(R)}}^2(\mathbf{v}^{(R)}) \leq \dots \leq \mathcal{T}_{\pi^{(R)}}^\infty(\mathbf{v}^{(R)}) = \mathbf{v}^{\pi^{(R)}} \leq \mathbf{v}^*.$$

and therefore this $\pi^{(R)}$ is an u -optimal policy. Ultimately, this technique avoids the standard loss of a $(1 - \gamma)^{-1}$ factor when converting values to policies.

The Variance Reduction Technique Suppose now that we provide an algorithm that maintains the monotonicity condition using random samples from $\mathbf{P}_{s,a}$ to approximately compute (4.1). Further, suppose we want to obtain a new value function and policy that is at least $(u/2)$ -optimal. In order to obtain the desired accuracy, we need to approximate $\mathbf{P}_{s,a}^\top \mathbf{v}^{(i)}$ up to error at most $(1 - \gamma)u/2$. Since $\|\mathbf{v}^{(i)}\|_\infty \leq (1 - \gamma)^{-1}$, by Hoeffding bound, $\tilde{O}((1 - \gamma)^{-4}u^{-2})$ samples suffices. Note that the number of samples also determines the computation time and therefore each iteration takes $\tilde{O}((1 - \gamma)^{-4}u^{-2}|\mathcal{S}||\mathcal{A}|)$ samples/computation time and $\tilde{O}((1 - \gamma)^{-1})$ iterations for the value iteration to converge. Overall, this yields a sample/computation complexity of $\tilde{O}((1 - \gamma)^{-5}u^{-2}|\mathcal{S}||\mathcal{A}|)$. To reduce the $(1 - \gamma)^{-5}$ dependence, [SWWY18] uses properties of the input (and the initialization) vectors: $\|\mathbf{v}^{(0)} - \mathbf{v}^*\|_\infty \leq u$ and rewrites value iteration (4.1) as follows

$$\mathbf{v}^{(i)}(s) \leftarrow \max_a [r(s, a) + \mathbf{P}_{s,a}^\top (\mathbf{v}^{(i-1)} - \mathbf{v}^{(0)}) + \mathbf{P}_{s,a}^\top \mathbf{v}^{(0)}], \quad (4.3)$$

Notice that $\mathbf{P}_{s,a}^\top \mathbf{v}^{(0)}$ is shared over all iterations and we can approximate it up to error $(1-\gamma)u/4$ using only $\tilde{O}((1-\gamma)^{-4}u^{-2})$ samples. For every iteration, we have $\|\mathbf{v}^{(i-1)} - \mathbf{v}^{(0)}\|_\infty \leq u$ (recall that we demand the monotonicity is satisfied at each iteration). Hence $\mathbf{P}_{s,a}^\top (\mathbf{v}^{(i-1)} - \mathbf{v}^{(0)})$ can be approximated up to error $(1-\gamma)u/4$ using only $\tilde{O}((1-\gamma)^{-2})$ samples (note that there is no u -dependence here). By this technique, over $\tilde{O}((1-\gamma)^{-1})$ iterations only $\tilde{O}((1-\gamma)^{-4}u^{-2} + (1-\gamma)^{-3})$ samples/computation per state action pair are needed, i.e. there is a $(1-\gamma)$ improvement.

The Total-Variance Technique By combining the monotonicity technique and variance reduction technique, one can obtain a $\tilde{O}((1-\gamma)^{-4})$ sample/running time complexity (per state-action pair) on computing a policy; this was one of the results [SWWY18]. However, there is a gap between this bound and the best known lower bound of $\tilde{\Omega}(|S||A|\epsilon^{-2}(1-\gamma)^{-3})$ [AMK13]. Here we show how to remove the last $(1-\gamma)$ factor by better exploiting the structure of the MDP. In [SWWY18] the update error in each iteration was set to be at most $(1-\gamma)u/2$ to compensate for error accumulation through a horizon of length $(1-\gamma)^{-1}$ (i.e., the accumulated error is sum of the estimation error at each iteration). To improve we show how to leverage previous work to show that the true error accumulation is much less. To see this, let us now switch to Bernstein inequality. Suppose we would like to estimate the value function of some policy π . The estimation error vector of the value function is upper bounded by $\tilde{O}(\sqrt{\sigma_\pi/m})$, where $\sigma_\pi(s) = \text{Var}_{s' \sim \mathbf{P}_{s,\pi(s)}}(\mathbf{v}^\pi(s'))$ denotes the variance of the value of the next state if starting from state s by playing policy π , and m is the number of samples collected per state-action pair. The accumulated error due to estimating value functions can be shown to obey the following inequality (upper to logarithmic factors)

$$\text{accumulated error} \propto \sum_{i=0}^{\infty} \gamma^i \mathbf{P}_\pi^i \sqrt{\sigma_\pi/m} \leq c_1 \left(\frac{1}{1-\gamma} \sum_{i=0}^{\infty} \gamma^{2i} \mathbf{P}_\pi^i \sigma_\pi/m \right)^{1/2},$$

where c_1 is a constant and the inequality follows from a Cauchy-Swartz-like inequality. According to the *law of total variance*, for any given policy π (in particular, the optimal policy π^*) and initial state s , the expected sum of variance of the tail sums of rewards, $\sum \gamma^{2i} \mathbf{P}_\pi^i \sigma_\pi$, is exactly the variance of the total return by playing the policy π . This observation was previously used in the analysis of [MM99, LH12, AMK13]. Since the upper bound on the total return is $(1-\gamma)^{-1}$, it can be shown that $\sum_i \gamma^{2i} \mathbf{P}_\pi^i \sigma_\pi \leq (1-\gamma)^{-2} \cdot \mathbf{1}$ and therefore the total error accumulation is $\sqrt{(1-\gamma)^{-3}/m}$. Thus picking $m \approx (1-\gamma)^{-3}\epsilon^{-2}$ is sufficient to control the accumulated error (instead of $(1-\gamma)^{-4}$). To analyze our algorithm, we will apply the above inequality to the optimal policy π^* to obtain our final error bound.

Putting it All Together In the next section we show how to combine these three techniques into one algorithm and make them work seamlessly. In particular, we provide and analyze how to combine these techniques into an Algorithm 1 which can be used to at least halve the error of a current policy. Applying this routine a logarithmic number of time then yields our desired bounds. In the input of the algorithm, we demand the input value $\mathbf{v}^{(0)}$ and $\pi^{(0)}$ satisfies the required monotonicity requirement, i.e., $\mathbf{v}^{(0)} \leq \mathcal{T}_{\pi^{(0)}}(\mathbf{v}^{(0)})$ (in the first iteration, the zero vector $\mathbf{0}$ and an arbitrary policy π satisfies the requirement). We then pick a set of samples to estimate $\mathbf{P}\mathbf{v}^{(0)}$ accurately with $\tilde{O}((1-\gamma)^{-3}\epsilon^{-2})$ samples per state-action pair. The same set of samples is used to estimate the variance vector $\sigma_{\mathbf{v}^*}$. These estimates serve as the initialization of the algorithm. In each iteration i , we draw fresh new samples to compute estimate of $\mathbf{P}(\mathbf{v}^{(i)} - \mathbf{v}^{(0)})$. The sum of the

estimate of $\mathbf{P}\mathbf{v}^{(0)}$ and $\mathbf{P}(\mathbf{v}^{(i)} - \mathbf{v}^{(0)})$ gives an estimate of $\mathbf{P}\mathbf{v}^{(i)}$. We then make the above estimates have one-sided error by shifting them according to their estimation errors (which is estimated from the Bernstein inequality). These one-side error estimates allow us to preserve monotonicity, i.e., guarantees the new value is always improving on the entire sample path with high probability. The estimate of $\mathbf{P}\mathbf{v}^{(i)}$ is plugged in to the Bellman's operator and gives us new value function, $\mathbf{v}^{(i+1)}$ and policy $\pi^{(i+1)}$, satisfying the monotonicity and advancing accuracy. Repeating the above procedure for the desired number of iterations completes the algorithm.

Algorithm 1 Variance-Reduced QVI

- 1: **Input:** A sampling oracle for DMDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathbf{r}, \mathbf{P}, \gamma)$
 - 2: **Input:** Upper bound on error $u \in [0, (1 - \gamma)^{-1}]$ and error probability $\delta \in (0, 1)$
 - 3: **Input:** Initial values $\mathbf{v}^{(0)}$ and policy $\pi^{(0)}$ such that $\mathbf{v}^{(0)} \leq \mathcal{T}_{\pi^{(0)}}\mathbf{v}^{(0)}$, and $\mathbf{v}^* - \mathbf{v}^{(0)} \leq u\mathbf{1}$;
 - 4: **Output:** \mathbf{v}, π such that $\mathbf{v} \leq \mathcal{T}_{\pi}(\mathbf{v})$ and $\mathbf{v}^* - \mathbf{v} \leq (u/2) \cdot \mathbf{1}$.
 - 5:
 - 6: **INITIALIZATION:**
 - 7: Let $\beta \leftarrow (1 - \gamma)^{-1}$, and $R \leftarrow \lceil c_1\beta \ln[\beta u^{-1}] \rceil$ for constant c_1 ;
 - 8: Let $m_1 \leftarrow c_2\beta^3 u^{-2} \log(8|\mathcal{S}||\mathcal{A}|\delta^{-1})$ for constant c_2 ;
 - 9: Let $m_2 \leftarrow c_3\beta^2 \log[2R|\mathcal{S}||\mathcal{A}|\delta^{-1}]$ for constant c_3 ;
 - 10: Let $\alpha_1 \leftarrow m_1^{-1} \log(8|\mathcal{S}||\mathcal{A}|\delta^{-1})$;
 - 11: For each $(s, a) \in \mathcal{S} \times \mathcal{A}$, sample independent samples $s_{s,a}^{(1)}, s_{s,a}^{(2)}, \dots, s_{s,a}^{(m_1)}$ from $\mathbf{P}_{s,a}$;
 - 12: Initialize $\mathbf{w} = \tilde{\mathbf{w}} = \hat{\boldsymbol{\sigma}} = \mathbf{Q}^{(0)} \leftarrow \mathbf{0}_{\mathcal{S} \times \mathcal{A}}$, and $i \leftarrow 0$;
 - 13: **for** each $(s, a) \in \mathcal{S} \times \mathcal{A}$ **do**
 - 14: *Compute empirical estimates of $\mathbf{P}_{s,a}^\top \mathbf{v}^{(0)}$ and $\boldsymbol{\sigma}_{\mathbf{v}^{(0)}}(s, a)$*
 - 15: Let $\tilde{\mathbf{w}}(s, a) \leftarrow \frac{1}{m_1} \sum_{j=1}^{m_1} \mathbf{v}^{(0)}(s_{s,a}^{(j)})$
 - 16: Let $\hat{\boldsymbol{\sigma}}(s, a) \leftarrow \frac{1}{m_1} \sum_{j=1}^{m_1} (\mathbf{v}^{(0)}(s_{s,a}^{(j)}))^2 - \tilde{\mathbf{w}}^2(s, a)$
 - 17:
 - 18: *Shift the empirical estimate to have one-sided error and guarantee monotonicity*
 - 19: $\mathbf{w}(s, a) \leftarrow \tilde{\mathbf{w}}(s, a) - \sqrt{2\alpha_1 \hat{\boldsymbol{\sigma}}(s, a)} - 4\alpha_1^{3/4} \|\mathbf{v}^{(0)}\|_\infty - (2/3)\alpha_1 \|\mathbf{v}^{(0)}\|_\infty$
 - 20:
 - 21: *Compute coarse estimate of the Q-function*
 - 22: $\mathbf{Q}^{(0)}(s, a) \leftarrow \mathbf{r}(s, a) + \gamma \mathbf{w}(s, a)$
 - 23:
 - 24: **REPEAT:** *successively improve*
 - 25: **for** $i = 1$ to R **do**
 - 26: *Compute $\mathbf{g}^{(i)}$ the estimate of $\mathbf{P}[\mathbf{v}^{(i)} - \mathbf{v}^{(0)}]$ with one-sided error*
 - 27: Let $\mathbf{v}^{(i)} \leftarrow \mathbf{v}(\mathbf{Q}^{(i-1)})$, $\pi^{(i)} \leftarrow \pi(\mathbf{Q}^{(i-1)})$; *let $\tilde{\mathbf{v}}^{(i)} \leftarrow \mathbf{v}^{(i)}$, $\tilde{\pi}^{(i)} \leftarrow \pi^{(i)}$ (for analysis);*
 - 28: For each $s \in \mathcal{S}$, if $\mathbf{v}^{(i)}(s) \leq \mathbf{v}^{(i-1)}(s)$, then $\mathbf{v}^{(i)}(s) \leftarrow \mathbf{v}^{(i-1)}(s)$ and $\pi^{(i)}(s) \leftarrow \pi^{(i-1)}(s)$;
 - 29: For each $(s, a) \in \mathcal{S} \times \mathcal{A}$, draw independent samples $\tilde{s}_{s,a}^{(1)}, \tilde{s}_{s,a}^{(2)}, \dots, \tilde{s}_{s,a}^{(m_2)}$ from $\mathbf{P}_{s,a}$;
 - 30: Let $\mathbf{g}^{(i)}(s, a) \leftarrow \frac{1}{m_2} \sum_{j=1}^{m_2} [\mathbf{v}^{(i)}(\tilde{s}_{s,a}^{(j)}) - \mathbf{v}^{(0)}(\tilde{s}_{s,a}^{(j)})] - (1 - \gamma)u/8$;
 - 31:
 - 32: *Improve $\mathbf{Q}^{(i)}$*
 - 33: $\mathbf{Q}^{(i)} \leftarrow \mathbf{r} + \gamma \cdot [\mathbf{w} + \mathbf{g}^{(i)}]$;
 - 34: **return** $\mathbf{v}^{(R)}, \pi^{(R)}$.
-

5 Algorithm and Analysis

In this section we provide and analyze our near sample/time optimal ϵ -policy computation algorithm. As discussed in Section 4 our algorithm combines three main ideas: variance reduction, the monotone value/policy iteration, and the reduction of accumulated error via Bernstein inequality. These ingredients are used in the Algorithm 1 to provide a routine which halves the error of a given policy. We analyze this procedure in Section 5.1 and use it to obtain our main result in Section 5.2.

5.1 The Analysis of the Variance Reduced Algorithm

In this section we analyze Algorithm 1, showing that each iteration of the algorithm approximately contracts towards the optimal value and policy and that ultimately the algorithm halves the error of the input value and policy with high probability. All proofs in this section are deferred to Appendix E.1.

We start with bounding the error of $\tilde{\mathbf{w}}$ and $\hat{\boldsymbol{\sigma}}$ defined in Line 15 and 16 of Algorithm 1. Notice that these are the empirical estimations of $\mathbf{P}_{s,a}^\top \mathbf{v}^{(0)}$ and $\boldsymbol{\sigma}_{\mathbf{v}^{(0)}}(s, a)$.

Lemma 5.1 (Empirical Estimation Error). *Let $\tilde{\mathbf{w}}$ and $\hat{\boldsymbol{\sigma}}$ be computed in Line 15 and 16 of Algorithm 1. Recall that $\tilde{\mathbf{w}}$ and $\hat{\boldsymbol{\sigma}}$ are empirical estimates of $\mathbf{P}\mathbf{v}$ and $\boldsymbol{\sigma}_{\mathbf{v}} = \mathbf{P}\mathbf{v}^2 - (\mathbf{P}\mathbf{v})^2$ using m_1 samples per (s, a) pair. With probability at least $1 - \delta$, for $L \stackrel{\text{def}}{=} \log(8|\mathcal{S}||\mathcal{A}|\delta^{-1})$, we have*

$$|\tilde{\mathbf{w}} - \mathbf{P}^\top \mathbf{v}^{(0)}| \leq \sqrt{2m_1^{-1}\boldsymbol{\sigma}_{\mathbf{v}^{(0)}} \cdot L} + 2(3m_1)^{-1}\|\mathbf{v}^{(0)}\|_\infty L \quad (5.1)$$

and

$$\forall (s, a) \in \mathcal{S} \times \mathcal{A}: |\hat{\boldsymbol{\sigma}}(s, a) - \boldsymbol{\sigma}_{\mathbf{v}^{(0)}}(s, a)| \leq 4\|\mathbf{v}^{(0)}\|_\infty^2 \cdot \sqrt{2m_1^{-1}L}. \quad (5.2)$$

The proof is a straightforward application of Bernstein's inequality and Hoeffding's inequality. Next we show that the difference between $\boldsymbol{\sigma}_{\mathbf{v}^{(0)}}$ and $\boldsymbol{\sigma}_{\mathbf{v}^*}$ is also bounded.

Lemma 5.2. *Suppose $\|\mathbf{v} - \mathbf{v}^*\|_\infty \leq \epsilon$ for some $\epsilon > 0$, then $\sqrt{\boldsymbol{\sigma}_{\mathbf{v}}} \leq \sqrt{\boldsymbol{\sigma}_{\mathbf{v}^*}} + \epsilon \cdot \mathbf{1}$.*

Next we show that in Line 30, the computed $\mathbf{g}^{(i)}$ concentrates to and is an overestimate of $\mathbf{P}[\mathbf{v}^{(i)} - \mathbf{v}^{(0)}]$ with high probability.

Lemma 5.3. *Let $\mathbf{g}^{(i)}$ be the estimate of $\mathbf{P}[\mathbf{v}^{(i)} - \mathbf{v}^{(0)}]$ defined in Line 30 of Algorithm 1. Then conditioning on the event that $\|\mathbf{v}^{(i)} - \mathbf{v}^{(0)}\|_\infty \leq 2u$, with probability at least $1 - \delta/R$,*

$$\mathbf{P}[\mathbf{v}^{(i)} - \mathbf{v}^{(0)}] - \frac{(1 - \gamma)u}{4} \cdot \mathbf{1} \leq \mathbf{g}^{(i)} \leq \mathbf{P}[\mathbf{v}^{(i)} - \mathbf{v}^{(0)}]$$

provided appropriately chosen constants c_1, c_2 , and c_3 in Algorithm 1.

Now we present the key contraction lemma, in which we set the constants, c_1, c_2, c_3 , in Algorithm 1 to be sufficiently large (e.g., $c_1 \geq 4, c_2 \geq 8192, c_3 \geq 128$). Note that these constants only need to be sufficiently large so that the concentration inequalities hold.

Lemma 5.4. Let $\mathbf{Q}^{(i)}$ be the estimated Q -function of $\mathbf{v}^{(i)}$ in Line 33 of Algorithm 1. Let $\pi^{(i)}$ and $\mathbf{v}^{(i)}$ be estimated in iteration i , as defined in Line 27 and 28. Then, with probability at least $1 - 2\delta$, for all $1 \leq i \leq R$,

$$\mathbf{v}^{(i-1)} \leq \mathbf{v}^{(i)} \leq \mathcal{T}_{\pi^{(i)}}[\mathbf{v}^{(i)}], \quad \mathbf{Q}^{(i)} \leq \mathbf{r} + \gamma \mathbf{P} \mathbf{v}^{(i)}, \quad \text{and} \quad \mathbf{Q}^* - \mathbf{Q}^{(i)} \leq \gamma \mathbf{P}^{\pi^*} [\mathbf{Q}^* - \mathbf{Q}^{(i-1)}] + \boldsymbol{\xi},$$

where for $\alpha_1 = m_1^{-1}L < 1$ the error vector $\boldsymbol{\xi}$ satisfies

$$\mathbf{0} \leq \boldsymbol{\xi} \leq C\sqrt{\alpha_1 \boldsymbol{\sigma}_{\mathbf{v}^*}} + \left[(1 - \gamma)u/C + C\alpha_1^{3/4} \|\mathbf{v}^{(0)}\|_{\infty} \right] \cdot \mathbf{1},$$

for some sufficiently large constant $C \geq 8$.

Using the previous lemmas we can prove the guarantees of Algorithm 1.

Proposition 5.4.1. On an input value vector $\mathbf{v}^{(0)}$, policy $\pi^{(0)}$, and parameters $u \in (0, (1 - \gamma)^{-1}]$, $\delta \in (0, 1)$ such that $\mathbf{v}^{(0)} \leq \mathcal{T}_{\pi^{(0)}}[\mathbf{v}^{(0)}]$, and $\mathbf{v}^* - \mathbf{v}^{(0)} \leq u\mathbf{1}$, Algorithm 1 halts in time

$$O \left[\left(u^{-2} + \log \frac{1}{(1 - \gamma)u} \right) \cdot \frac{|\mathcal{S}||\mathcal{A}| \cdot \log(|\mathcal{S}||\mathcal{A}|\delta^{-1})}{(1 - \gamma)^3} \right]$$

and outputs values \mathbf{v} and policy π such that $\mathbf{v} \leq \mathcal{T}_{\pi}(\mathbf{v})$ and $\mathbf{v}^* - \mathbf{v} \leq (u/2)\mathbf{1}$ with probability at least $1 - \delta$, provided appropriately chosen constants, c_1, c_2, c_3 .

We prove this proposition by iteratively applying Lemma 5.4. Suppose $\mathbf{v}^{(R)}$ is the output of the algorithm, after R iterations. We show $\mathbf{v}^* - \mathbf{v}^{(R)} \leq \gamma^{R-1} \mathbf{P}^{\pi^*} [\mathbf{Q}^* - \mathbf{Q}^0] + (\mathbf{I} - \gamma \mathbf{P}^{\pi^*})^{-1} \boldsymbol{\xi}$. Notice that $(\mathbf{I} - \gamma \mathbf{P}^{\pi^*})^{-1} \boldsymbol{\xi}$ is related to $(\mathbf{I} - \gamma \mathbf{P}^{\pi^*})^{-1} \sqrt{\boldsymbol{\sigma}_{\mathbf{v}^*}}$. We then apply the variance analytical tools presented in Section C to show that $(\mathbf{I} - \gamma \mathbf{P}^{\pi^*})^{-1} \boldsymbol{\xi} \leq (u/4)\mathbf{1}$ when setting the constants properly in Algorithm 1. We refer this technique as the *total-variance technique*, since $\|(\mathbf{I} - \gamma \mathbf{P}^{\pi^*})^{-1} \sqrt{\boldsymbol{\sigma}_{\mathbf{v}^*}\|_{\infty}^2 \leq O[(1 - \gamma)^{-3}]$ instead of a naïve bound of $(1 - \gamma)^{-4}$. We complete the proof by choosing $R = \Theta((1 - \gamma)^{-1} \log(u^{-1}))$ and showing that $\gamma^{R-1} \mathbf{P}^{\pi^*} [\mathbf{Q}^* - \mathbf{Q}^0] \leq (u/4)\mathbf{1}$.

5.2 From Halving the Error to Arbitrary Precision

In the previous section, we provided an algorithm that on an input policy, outputs a policy with value vector that has ℓ_{∞} distance to the optimal value vector only half of that of the input one. In this section, we give a complete policy computation algorithm by showing that it is possible to apply this error “halving” procedure iteratively. We summarize our meta algorithm in Algorithm 2. Note that in the algorithm, each call of HALFERR draws new samples from the sampling oracle. We refer in this section to Algorithm 1 as a subroutine HALFERR, which given an input MDP \mathcal{M} with a sampling oracle, an input value function $\mathbf{v}^{(i)}$, and an input policy $\pi^{(i)}$, outputs an value function $\mathbf{v}^{(i+1)}$ and a policy $\pi^{(i+1)}$.

Combining Algorithm 2 and Algorithm 1, we are ready to present main result.

Theorem 5.5. Let $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathbf{P}, \mathbf{r}, \gamma)$ be a DMDP with a generative model. Suppose we can sample a state from each probability vector $\mathbf{P}_{s,a}$ within time $O(1)$. Then for any $\epsilon, \delta \in (0, 1)$, there exists an algorithm that halts in time

$$T := O \left[\frac{|\mathcal{S}||\mathcal{A}|}{(1 - \gamma)^3 \epsilon^2} \log \left(\frac{|\mathcal{S}||\mathcal{A}|}{(1 - \gamma) \delta \epsilon} \right) \log \left(\frac{1}{(1 - \gamma) \epsilon} \right) \right]$$

Algorithm 2 Meta Algorithm

- 1: **Input:** A sampling oracle of some $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathbf{r}, \mathbf{P}, \gamma)$, $\epsilon > 0, \delta \in (0, 1)$
 - 2: **Initialize:** $\mathbf{v}^{(0)} \leftarrow \mathbf{0}$, $\pi^{(0)} \leftarrow$ arbitrary policy, $R \leftarrow \Theta[\log(\epsilon^{-1}(1 - \gamma)^{-1})]$
 - 3: **for** $i = \{1, 2, \dots, R\}$ **do**
 - 4: //HALFERR is initialized with $QVI(u = 2^{-i+1}(1 - \gamma)^{-1}, \delta, \mathbf{v}^{(0)} = \mathbf{v}^{(i-1)}, \pi^{(0)} = \pi^{(i-1)})$
 - 5: $\mathbf{v}^{(i)}, \pi^{(i)} \leftarrow \text{HALFERR} \leftarrow \mathbf{v}^{(i-1)}, \pi^{(i-1)}$
 - 6: **Output:** $\mathbf{v}^{(R)}, \pi^{(R)}$.
-

and obtains a policy π such that $\mathbf{v}^* - \epsilon \mathbf{1} \leq \mathbf{v}^\pi \leq \mathbf{v}^*$, with probability at least $1 - \delta$ where \mathbf{v}^* is the optimal value of \mathcal{M} . The algorithm uses space $O(|\mathcal{S}||\mathcal{A}|)$ and queries the generative model for at most $O(T)$ fresh samples.

Remark 5.6. In the above theorem, we require $\epsilon \in (0, 1)$. For $\epsilon \geq 1$, our sample complexity may fail to be optimal. We leave this for a future project.

Remark 5.7. The full analysis of the halving algorithm is presented in Section E.2. Our algorithm can be implemented in space $O(|\mathcal{S}||\mathcal{A}|)$ since in Algorithm 1, the initialization phase can be done for each (s, a) and compute $\mathbf{w}(s, a), \tilde{\mathbf{w}}(s, a), \hat{\boldsymbol{\sigma}}(s, a), \mathbf{Q}^{(0)}(s, a)$ without storing the samples. The updates can be computed in space $O(|\mathcal{S}||\mathcal{A}|)$ as well.

6 Concluding Remark

In summary, for a discounted Markov Decision Process (DMDP) $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathbf{P}, \mathbf{r}, \gamma)$ provided we can only access the transition function of the DMDP through a generative sampling model, we provide an algorithm which computes an ϵ -approximate optimal (for $\epsilon \in (0, 1)$) policy with probability $1 - \delta$ where both the time spent and number of sample taken is upper bounded by $\tilde{O}((1 - \gamma)^{-3} \epsilon^{-2} |\mathcal{S}||\mathcal{A}|)$. This improves upon the previous best known bounds by a factor of $1/(1 - \gamma)$ and matches the the lower bounds proved in [AMK13] up to logarithmic factors.

The appendix is structured as follows. Section A surveys the existing runtime results for solving the DMDP when a full model is given. Section B provides an runtime optimal algorithm for computing approximate value functions (by directly combining [AMK13] and [SWWY18]). Section C gives technical analysis and variance upper bounds for the total-variance technique. Section D discusses sample complexity lower bounds for obtaining approximate policies with a generative sampling model. Section E provides proofs to lemmas, propositions and theorems in the main text of the paper. Section F extends our method and results to the finite-horizon MDP and provides a nearly matching sample complexity lower bound.

References

- [AMK13] Mohammad Gheshlaghi Azar, Rémi Munos, and Hilbert J Kappen. Minimax pac bounds on the sample complexity of reinforcement learning with a generative model. *Machine learning*, 91(3):325–349, 2013.
- [Bel57] Richard Bellman. *Dynamic Programming*. Princeton University Press, Princeton, NJ, 1957.
- [Ber13] Dimitri P Bertsekas. *Abstract dynamic programming*. Athena Scientific, Belmont, MA, 2013.
- [Dan16] George Dantzig. *Linear Programming and Extensions*. Princeton University Press, Princeton, NJ, 2016.
- [DB15] Christoph Dann and Emma Brunskill. Sample complexity of episodic fixed-horizon reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 2818–2826, 2015.
- [d’E63] F d’Epenoux. A probabilistic production and inventory problem. *Management Science*, 10(1):98–108, 1963.
- [DG60] Guy De Ghellinck. Les problemes de decisions sequentielles. *Cahiers du Centre dEtudes de Recherche Opérationnelle*, 2(2):161–179, 1960.
- [HMZ13] Thomas Dueholm Hansen, Peter Bro Miltersen, and Uri Zwick. Strategy iteration is strongly polynomial for 2-player turn-based stochastic games with a constant discount factor. *J. ACM*, 60(1):1:1–1:16, February 2013.
- [How60] Ronald A. Howard. *Dynamic programming and Markov processes*. The MIT press, Cambridge, MA, 1960.
- [Kak03] Sham M Kakade. *On the sample complexity of reinforcement learning*. PhD thesis, University of London London, England, 2003.
- [KBJ14] Dileep Kalathil, Vivek S Borkar, and Rahul Jain. Empirical q-value iteration. *arXiv preprint arXiv:1412.0180*, 2014.
- [KS99] Michael J Kearns and Satinder P Singh. Finite-sample convergence rates for q-learning and indirect algorithms. In *Advances in neural information processing systems*, pages 996–1002, 1999.
- [LDK95] Michael L Littman, Thomas L Dean, and Leslie Pack Kaelbling. On the complexity of solving Markov decision problems. In *Proceedings of the Eleventh conference on Uncertainty in artificial intelligence*, pages 394–402. Morgan Kaufmann Publishers Inc., 1995.
- [LH12] Tor Lattimore and Marcus Hutter. Pac bounds for discounted mdps. In *International Conference on Algorithmic Learning Theory*, pages 320–334. Springer, 2012.

- [LS14] Yin Tat Lee and Aaron Sidford. Path finding methods for linear programming: Solving linear programs in $\tilde{O}(n^2)$ iterations and faster algorithms for maximum flow. In *Foundations of Computer Science (FOCS), 2014 IEEE 55th Annual Symposium on*, pages 424–433. IEEE, 2014.
- [LS15] Yin Tat Lee and Aaron Sidford. Efficient inverse maintenance and faster algorithms for linear programming. In *Foundations of Computer Science (FOCS), 2015 IEEE 56th Annual Symposium on*, pages 230–249. IEEE, 2015.
- [MM99] Remi Munos and Andrew W Moore. Variable resolution discretization for high-accuracy solutions of optimal control problems. *Robotics Institute*, page 256, 1999.
- [MS99] Yishay Mansour and Satinder Singh. On the complexity of policy iteration. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, pages 401–408. Morgan Kaufmann Publishers Inc., 1999.
- [Sch13] Bruno Scherrer. Improved and generalized upper bounds on the complexity of policy iteration. In *Advances in Neural Information Processing Systems*, pages 386–394, 2013.
- [SLL09] Alexander L Strehl, Lihong Li, and Michael L Littman. Reinforcement learning in finite mdps: Pac analysis. *Journal of Machine Learning Research*, 10(Nov):2413–2444, 2009.
- [SWWY18] Aaron Sidford, Mengdi Wang, Xian Wu, and Yinyu Ye. Variance reduced value iteration and faster algorithms for solving markov decision processes. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 770–787. SIAM, 2018.
- [Tse90] Paul Tseng. Solving h-horizon, stationary markov decision problems in time proportional to $\log(h)$. *Operations Research Letters*, 9(5):287–297, 1990.
- [Wan17] Mengdi Wang. Randomized linear programming solves the discounted Markov decision problem in nearly-linear running time. *arXiv preprint arXiv:1704.01869*, 2017.
- [Ye05] Yinyu Ye. A new complexity result on solving the Markov decision problem. *Mathematics of Operations Research*, 30(3):733–749, 2005.
- [Ye11] Yinyu Ye. The simplex and policy-iteration methods are strongly polynomial for the Markov decision problem with a fixed discount rate. *Mathematics of Operations Research*, 36(4):593–603, 2011.

A Previous Work on Solving DMDP with a Full Model

Value iteration was proposed by [Bel57] to compute an exact optimal policy of a given DMDP in time $\mathcal{O}((1-\gamma)^{-1}|\mathcal{S}|^2|\mathcal{A}|L\log((1-\gamma)^{-1}))$, where L is the total number of bits needed to represent the input; and it can find an approximate ϵ -approximate solution in time $\mathcal{O}(|\mathcal{S}|^2|\mathcal{A}|(1-\gamma)^{-1}\log(1/\epsilon(1-\gamma)))$; see e.g. [Tse90, LDK95]. The policy iteration was introduced by [How60] shortly after, where the policy is monotonically improved according to its associated value function. Its complexity has also been analyzed extensively; see e.g. [MS99, Ye11, Sch13]. Ye [Ye11] showed that policy iteration and the simplex method are strongly polynomial for DMDP and terminates in $\mathcal{O}(|\mathcal{S}|^2|\mathcal{A}|(1-\gamma)^{-1}\log(|\mathcal{S}|(1-\gamma)^{-1}))$ number of iterations. Later [HMZ13] and [Sch13] improved the iteration bound to $O(|\mathcal{S}||\mathcal{A}|(1-\gamma)^{-1}\log((1-\gamma)^{-1}))$ for Howard’s policy iteration method. A third approach is to formulate the nonlinear Bellman equation into a linear program [d’E63, DG60], and solve it using standard linear program solvers, such as the simplex method by Dantzig [Dan16] and the combinatorial interior-point algorithm by [Ye05]. [LS14, LS15] showed that one can solve linear programs in $\tilde{O}(\sqrt{\text{rank}(A)})$ number of linear system solves, which, applied to DMDP, yields to a running time of $\tilde{O}(|\mathcal{S}|^{2.5}|\mathcal{A}|L)$ for computing the exact policy and $\tilde{O}(|\mathcal{S}|^{2.5}|\mathcal{A}|\log(1/\epsilon))$ for computing an ϵ -optimal policy. [SWWY18] further improved the complexity of value iteration by using randomization and variance reduction. See Table 2 for comparable run-time results or computing the optimal policy when the MDP model is fully given.

Algorithm	Complexity	References
Value Iteration (exact)	$ \mathcal{S} ^2 \mathcal{A} L\frac{\log(1/(1-\gamma))}{1-\gamma}$	[Tse90, LDK95]
Value Iteration	$ \mathcal{S} ^2 \mathcal{A} \frac{\log(1/(1-\gamma)\epsilon)}{1-\gamma}$	[Tse90, LDK95]
Policy Iteration (Block Simplex)	$\frac{ \mathcal{S} ^4 \mathcal{A} ^2}{1-\gamma}\log(\frac{1}{1-\gamma})$	[Ye11],[Sch13]
Recent Interior Point Methods	$\tilde{O}(\mathcal{S} ^{2.5} \mathcal{A} L)$ $\tilde{O}(\mathcal{S} ^{2.5} \mathcal{A} \log(1/\epsilon))$	[LS14]
Combinatorial Interior Point Algorithm	$ \mathcal{S} ^4 \mathcal{A} ^4\log\frac{ \mathcal{S} }{1-\gamma}$	[Ye05]
High Precision Randomized Value Iteration	$\tilde{O}\left[\left(\text{nnz}(P) + \frac{ \mathcal{S} \mathcal{A} }{(1-\gamma)^3}\right)\log\left(\frac{1}{\epsilon\delta}\right)\right]$	[SWWY18]

Table 2: **Running Times to Solve DMDPs Given the Full MDP Model:** In this table, $|\mathcal{S}|$ is the number of states, $|\mathcal{A}|$ is the number of actions per state, $\gamma \in (0, 1)$ is the discount factor, and L is a complexity measure of the linear program formulation that is at most the total bit size to present the DMDP input. Rewards are bounded between 0 and 1.

B Sample and Time Efficient Value Computation

In this section, we describe an algorithm that obtains an ϵ -optimal values in time $\tilde{O}(\epsilon^{-2}(1-\gamma)^{-3}|\mathcal{S}||\mathcal{A}|)$. Note that the time and number of samples of this algorithm is optimal (up to logarithmic factors) due to the lower bound in [AMK13] which also established this upper bound on the sample complexity (but not time complexity) of the problem.

We achieve this by combining the algorithms in [AMK13] and [SWWY18]. First, we use the

ideas and analysis of [AMK13] to construct a sparse MDP where the optimal value function of this MDP approximates the optimal value function of the original MDP and then we run the high precision algorithm in [SWWY18] on this sparsified MDP. We show that [SWWY18] runs in nearly linear time on sparsified MDP. Since the number of samples taken to construct the sparsified MDP was the the optimal number of samples, to solve the problem, the ultimate running time we thereby achieve is nearly optimal as any algorithm needs spend time at least the number of samples to obtain these samples.

We include this for completeness but note that the approximate value function we show how to compute here does not suffice to compute policy of the MDP of comparable quality. The greedy policy of an ϵ -optimal value function is an $\epsilon/(1-\gamma)$ -optimal policy in the worst case. It has been shown in [AMK13] that the greedy policy of their value function is ϵ -optimal if $\epsilon \leq (1-\gamma)^{1/2}|\mathcal{S}|^{-1/2}$. However, when ϵ is so small, the seemingly sublinear runtime $\tilde{O}((1-\gamma)^{-3}|\mathcal{S}||\mathcal{A}|/\epsilon^2)$ essentially means a linear running time and sample complexity as $O((1-\gamma)^{-3}|\mathcal{S}|^2|\mathcal{A}|)$. The running time can be obtained by merely applying the result in [SWWY18] (although with a slightly different computation model).

B.1 The Sparsified DMDP

Suppose we are given a DMDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathbf{r}, \mathbf{P}, \gamma)$ with a sampling oracle. To approximate the optimal value of this MDP, we perform a sparsification procedure as in [AMK13]. Sparsification of DMDP is conducted as follows. Let $\delta > 0, \epsilon > 0$ be arbitrary. First we pick a number

$$m = \Theta \left[\frac{1}{(1-\gamma)^3 \epsilon^2} \log \left(\frac{|\mathcal{S}||\mathcal{A}|}{\delta} \right) \right]. \quad (\text{B.1})$$

For each $s \in \mathcal{S}$ and each $a \in \mathcal{A}$, we generate a sequence of independent samples from \mathcal{S} using the probability vector $\mathbf{P}_{s,a}$

$$s_{s,a}^{(1)}, s_{s,a}^{(2)}, \dots, s_{s,a}^{(m)}.$$

Next we construct a new and sparse probability vector $\hat{\mathbf{P}}_{s,a} \in \Delta_{|\mathcal{S}|}$ as

$$\forall s' \in \mathcal{S} : \hat{\mathbf{P}}_{s,a}(s') = \frac{1}{m} \cdot \sum_{i=1}^m \mathbf{1}(s_{s,a}^{(i)} = s').$$

Combining these $|\mathcal{S}||\mathcal{A}|$ new probability vectors, we obtain a new probability transition matrix $\hat{\mathbf{P}} \in \mathbb{R}^{\mathcal{S} \times \mathcal{A} \times \mathcal{S}}$ with number of non-zeros

$$\text{nnz}(\hat{\mathbf{P}}) = O \left[\frac{|\mathcal{S}||\mathcal{A}|}{(1-\gamma)^3 \epsilon^2} \log \left(\frac{|\mathcal{S}||\mathcal{A}|}{\delta} \right) \right].$$

Denote $\hat{\mathcal{M}} = (\mathcal{S}, \mathcal{A}, \mathbf{r}, \hat{\mathbf{P}}, \gamma)$ as the sparsified DMDP. In the rest of this section, we use $\hat{\cdot}$ to represent the quantities corresponding to DMDP $\hat{\mathcal{M}}$, e.g., $\hat{\mathbf{v}}^*$ for the optimal value function, $\hat{\pi}^*$ for a optimal policy, and $\hat{\mathbf{Q}}^*$ for the optimal Q -function. There is a strong approximation guarantee of the optimal Q -function of the sparsified MDP, presented as follows.

Theorem B.1 ([AMK13]). *Let \mathcal{M} be the original DMDP and $\hat{\mathcal{M}}$ be the corresponding sparsified version. Let \mathbf{Q}^* be the optimal Q -function vector of the original DMDP and $\hat{\mathbf{Q}}^*$ be the optimal Q -function of $\hat{\mathcal{M}}$. Then with probability at least $1 - \delta$ (over the randomness of the samples),*

$$\|\hat{\mathbf{Q}}^* - \mathbf{Q}^*\|_\infty \leq \epsilon.$$

Recall that \mathbf{v}^* and $\widehat{\mathbf{v}}^*$ are the optimal value functions of \mathcal{M} and $\widehat{\mathcal{M}}$. From Theorem B.1, we immediately have

$$\forall s \in \mathcal{S} : |\mathbf{v}^*(s) - \widehat{\mathbf{v}}^*(s)| = \left| \max_{a \in \mathcal{A}} \mathbf{Q}^*(s, a) - \max_{a \in \mathcal{A}} \widehat{\mathbf{Q}}^*(s, a) \right| \leq \max_{a \in \mathcal{A}} |\mathbf{Q}^*(s, a) - \widehat{\mathbf{Q}}^*(s, a)| \leq \epsilon,$$

with probability at least $1 - \delta$.

B.2 High Precision Algorithm in the Sparsified MDP

Next we shall use the high precision algorithm of the [SWWY18] which has the following guarantee.

Theorem B.2 ([SWWY18]). *There is an algorithm which given an input DMDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathbf{r}, \mathbf{P}, \gamma)$ in time⁶*

$$\tilde{O} \left[\left(\text{nnz}(\mathbf{P}) + \frac{|\mathcal{S}||\mathcal{A}|}{(1-\gamma)^3} \right) \cdot \log \epsilon^{-1} \cdot \log \delta^{-1} \right]$$

and outputs a vector $\tilde{\mathbf{v}}^*$ such that with probability at least $1 - \delta$,

$$\|\tilde{\mathbf{v}}^* - \mathbf{v}^*\|_\infty \leq \epsilon.$$

where \mathbf{v}^* is the optimal value of \mathcal{M} .

Combining the above two theorems, we immediately obtain an algorithm for finding ϵ -optimal value functions. It works by first generating enough samples for each state-action pair and then call the high-precision MDP solver by [SWWY18]. It does not sample transitions adaptively. We show that it achieves an optimal running time guarantee (up to poly log factors) of obtaining the value function under the sampling oracle model.

Theorem B.3. *Given an input DMDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathbf{r}, \mathbf{P}, \gamma)$ with a sampling oracle and optimal value function \mathbf{v}^* , there exists an algorithm, that runs in time*

$$\tilde{O} \left(\frac{|\mathcal{S}||\mathcal{A}|}{(1-\gamma)^3} \cdot \frac{1}{\epsilon^2} \cdot \log^2 \left(\frac{1}{\delta} \right) \right)$$

and outputs a vector $\widehat{\mathbf{v}}^*$ such that $\|\widehat{\mathbf{v}}^* - \mathbf{v}^*\|_\infty \leq O(\epsilon)$ with probability at least $1 - O(\delta)$.

Proof. We first obtain a sparsified MDP $\widehat{\mathcal{M}} = (\mathcal{S}, \mathcal{A}, \mathbf{r}, \widehat{\mathbf{P}}, \gamma)$ using the procedure described in Section B.1. This procedure runs in time $O(|\mathcal{S}||\mathcal{A}|m)$, recalling that m is the number of samples per (s, a) , defined in (B.1). Let $\widehat{\mathbf{u}}^*$ be the optimal value function of $\widehat{\mathcal{M}}$. By Theorem B.1, with probability at least $1 - \delta$, $\|\widehat{\mathbf{u}}^* - \mathbf{v}^*\| \leq \epsilon$, which we condition on for the rest of the proof. Calling the algorithm in Theorem B.2, we obtain a vector $\tilde{\mathbf{u}}^*$ in time

$$\tilde{O} \left[\left(\text{nnz}(\widehat{\mathbf{P}}) + \frac{|\mathcal{S}||\mathcal{A}|}{(1-\gamma)^3} \right) \cdot \log \epsilon^{-1} \cdot \log \delta^{-1} \right] = \tilde{O} \left(\frac{|\mathcal{S}||\mathcal{A}|}{(1-\gamma)^3} \cdot \frac{1}{\epsilon^2} \cdot \log^2 \frac{1}{\delta} \right)$$

and that with probability at least $1 - \delta$, $\|\tilde{\mathbf{u}}^* - \widehat{\mathbf{u}}^*\| \leq \epsilon$, which we condition on. By triangle inequality, we have

$$\|\tilde{\mathbf{u}}^* - \mathbf{v}^*\|_\infty \leq \|\tilde{\mathbf{u}}^* - \widehat{\mathbf{u}}^*\|_\infty + \|\widehat{\mathbf{u}}^* - \mathbf{v}^*\|_\infty \leq 2\epsilon.$$

This concludes the proof. □

⁶ $\tilde{O}(f)$ denotes $O(f \cdot \log^{O(1)} f)$.

C Variance Bounds

In this section, we study some properties of a DMDP. Most of the content in this section is similar to [AMK13]. We provide slight modifications and improvement to make the results fit to our application. The main result of this section is to show the following lemma.

Lemma C.1 (Upper Bound on Variance). *For any π , we have*

$$\|(\mathbf{I} - \gamma \mathbf{P}^\pi)^{-1} \sqrt{\boldsymbol{\sigma}_{\mathbf{v}^\pi}}\|_\infty^2 \leq \frac{1 + \gamma}{\gamma^2(1 - \gamma)^3},$$

where $\boldsymbol{\sigma}_{\mathbf{v}^\pi} = \mathbf{P}^\pi(\mathbf{v}^\pi)^2 - (\mathbf{P}^\pi \mathbf{v}^\pi)^2$ is the “one-step” variance of playing policy π .

Before we prove this lemma, we introduce another notation. We define $\boldsymbol{\Sigma}^\pi \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$ for all $(s, a) \in \mathcal{S} \times \mathcal{A}$ by

$$\boldsymbol{\Sigma}^\pi(s, a) := \mathbb{E} \left[\left(\mathbf{r}(s, a) + \sum_{t \geq 1} \gamma^t \mathbf{r}(s^t, a^t) - \mathbf{Q}^\pi(s, a) \right)^2 \middle| s^0 = s, a^0 = a, a^t = \pi(s^t) \right]$$

where $a^t = \pi(s^t)$. Thus $\boldsymbol{\Sigma}^\pi$ is the variance of the reward of starting with (s, a) and play π for infinite steps. The crucial observation of obtaining the near-optimal sample complexity is the following “Bellman Equation” for variance. It is a consequence of “the law of total variance”.

Lemma C.2 (Bellman Equation for variance). $\boldsymbol{\Sigma}^\pi$ satisfies the Bellman equation

$$\boldsymbol{\Sigma}^\pi = \gamma^2 \boldsymbol{\sigma}_{\mathbf{v}^\pi} + \gamma^2 \cdot \mathbf{P}^\pi \boldsymbol{\Sigma}^\pi.$$

Proof. By direct expansion,

$$\boldsymbol{\Sigma}^\pi(s, a) = \mathbb{E} \left[\left(\mathbf{r}(s, a) + \sum_{t \geq 1} \gamma^t \mathbf{r}(s^t, a^t) \right)^2 \middle| s^0 = s, a^0 = a, a^t = \pi(s^t) \right] - (\mathbf{Q}^\pi(s, a))^2. \quad (\text{C.1})$$

The first term in RHS can be written as

$$\begin{aligned} & \mathbb{E} \left[\left(\mathbf{r}(s, a) + \sum_{t \geq 1} \gamma^t \mathbf{r}(s^t, a^t) \right)^2 \middle| s^0 = s, a^0 = a, a^t = \pi(s^t) \right] \\ &= \sum_{s' \in \mathcal{S}} \mathbf{P}_{s,a}(s') \mathbb{E} \left[\left(\mathbf{r}(s, a) + \gamma \mathbf{r}(s', \pi(s')) + \sum_{t \geq 1} \gamma^t \mathbf{r}(s^t, a^t) \right)^2 \middle| s^0 = s', a^0 = \pi(s'), a^t = \pi(s^t) \right] \\ &= \mathbf{r}(s, a)^2 + 2\gamma \mathbf{r}(s, a) \cdot \sum_{s' \in \mathcal{S}} \mathbf{P}_{s,a}(s') \mathbf{Q}^\pi(s', \pi(s')) \\ & \quad + \gamma^2 \sum_{s' \in \mathcal{S}} \mathbf{P}_{s,a}(s') \mathbb{E} \left[\left(\mathbf{r}(s', \pi(s')) + \sum_{t \geq 1} \gamma^t \mathbf{r}(s^t, a^t) \right)^2 \middle| s^0 = s', a^0 = \pi(s'), a^t = \pi(s^t) \right] \\ &= \mathbf{r}(s, a)^2 + 2\gamma \mathbf{r}(s, a) \cdot \sum_{s' \in \mathcal{S}} \mathbf{P}_{s,a}(s') \mathbf{Q}^\pi(s', \pi(s')) + \gamma^2 (\mathbf{P}^\pi \boldsymbol{\Sigma}^\pi)(s, a) + \gamma^2 \sum_{s' \in \mathcal{S}} \mathbf{P}_{s,a}(s') (\mathbf{Q}^\pi(s', \pi(s')))^2 \\ &= \mathbf{Q}^\pi(s, a)^2 + \gamma^2 (\mathbf{P}^\pi \boldsymbol{\Sigma}^\pi)(s, a) + \gamma^2 \sum_{s' \in \mathcal{S}} \mathbf{P}_{s,a}(s') (\mathbf{Q}^\pi(s', \pi(s')))^2 - \gamma^2 \left(\sum_{s' \in \mathcal{S}} \mathbf{P}_{s,a}(s') \mathbf{Q}^\pi(s', \pi(s')) \right)^2 \\ &= \mathbf{Q}^\pi(s, a)^2 + \gamma^2 (\mathbf{P}^\pi \boldsymbol{\Sigma}^\pi)(s, a) + \gamma^2 \boldsymbol{\sigma}_{\mathbf{v}^\pi}(s, a). \end{aligned}$$

Combining the above two equations, we conclude the proof. \square

As a remark, we note that

$$\Sigma^\pi = \gamma^2(\mathbf{I} - \gamma^2\mathbf{P}^\pi)^{-1}\boldsymbol{\sigma}_{\mathbf{v}^\pi}.$$

Furthermore, by definition, we have

$$\max_{(s,a) \in \mathcal{S}} \Sigma^\pi(s,a) \leq (1-\gamma)^{-2},$$

The next lemma is crucial in proving the error bounds.

Lemma C.3. *Let $\mathbf{P} \in \mathbb{R}^{n \times n}$ be a non-negative matrix in which every row has ℓ_1 norm at most 1, i.e. ℓ_∞ operator norm at most 1. Then for all $\gamma \in (0,1)$ and $\mathbf{v} \in \mathbb{R}_{\geq 0}^n$ we have*

$$\|(\mathbf{I} - \gamma\mathbf{P})^{-1}\sqrt{\mathbf{v}}\|_\infty \leq \sqrt{\frac{1}{1-\gamma} \|(\mathbf{I} - \gamma\mathbf{P})^{-1}\mathbf{v}\|_\infty} \leq \sqrt{\frac{1+\gamma}{1-\gamma} \|(\mathbf{I} - \gamma^2\mathbf{P})^{-1}\mathbf{v}\|_\infty}.$$

Proof. Since, every row of \mathbf{P} has ℓ_1 norm at most 1, by Cauchy-Schwarz for $i \in [n]$ we have

$$[\mathbf{P}\sqrt{\mathbf{v}}]_i = \sum_{j \in [n]} P_{ij}\sqrt{v_j} \leq \sqrt{\sum_{j \in [n]} P_{ij} \cdot \sum_{j \in [n]} P_{ij}v_j} \leq \sqrt{P\mathbf{v}}.$$

Since \mathbf{v} is non-negative and applying \mathbf{P} preserves non-negativity, applying this inequality repeatedly yields that $\mathbf{P}^k\sqrt{\mathbf{v}} \leq \sqrt{\mathbf{P}^k\mathbf{v}}$ entrywise for all $k > 0$. Consequently, Cauchy-Schwarz again yields

$$\begin{aligned} (\mathbf{I} - \gamma\mathbf{P})^{-1}\sqrt{\mathbf{v}} &= \sum_{i=0}^{\infty} [\gamma\mathbf{P}]^i \sqrt{\mathbf{v}} \leq \sum_{i=0}^{\infty} \gamma^i \sqrt{\mathbf{P}^i\mathbf{v}} \leq \sqrt{\sum_{i=0}^{\infty} \gamma^i \cdot \sum_{i=0}^{\infty} \gamma^i \mathbf{P}^i\mathbf{v}} \\ &\leq \sqrt{\frac{1}{1-\gamma} \|(\mathbf{I} - \gamma\mathbf{P})^{-1}\mathbf{v}\|_\infty}. \end{aligned}$$

Next, as $(\mathbf{I} - \gamma\mathbf{P})(\mathbf{I} + \gamma\mathbf{P}) = (\mathbf{I} - \gamma\mathbf{P}^2)$ we see that $(\mathbf{I} - \gamma\mathbf{P})^{-1} = (\mathbf{I} + \gamma\mathbf{P})(\mathbf{I} - \gamma^2\mathbf{P})^{-1}$. Furthermore, as $\|\mathbf{P}x\|_\infty \leq \|x\|_\infty$ for all x we have $\|(\mathbf{I} + \gamma\mathbf{P})x\|_\infty \leq (1+\gamma)\|x\|_\infty$ for all x and therefore $\|(\mathbf{I} - \gamma\mathbf{P})^{-1}\mathbf{v}\|_\infty \leq (1+\gamma)\|(\mathbf{I} - \gamma^2\mathbf{P})^{-1}\mathbf{v}\|_\infty$ as desired. \square

We are now ready to prove Lemma C.1.

Proof of Lemma C.1. The lemma follows directly from the application of Lemma C.3. This proof is slightly simpler, tighter, and more general than the one in [AMK13]. \square

D Lower Bounds on Policy

Lemma D.1. *Suppose $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, \gamma, \mathbf{r})$ is a DMDDP with an sampling oracle. Suppose π is a given policy. Then there is an algorithm, halts in $\tilde{O}((1-\gamma)^{-3}\epsilon^{-2}|\mathcal{S}|)$ time, outputs a vector \mathbf{v} such that, with high probability, $\|\mathbf{v}^\pi - \mathbf{v}\|_\infty \leq \epsilon$.*

Proof. The lemma follows from a direct application of Theorem B.2. \square

Remark D.2. Suppose $|\mathcal{A}| = \tilde{\Omega}(1)$. Suppose there is an algorithm that obtains an ϵ -optimal policy with Z samples, then the above lemma implies an algorithm for obtaining an ϵ -optimal value function with $Z + \tilde{O}((1-\gamma)^{-3}\epsilon^{-2}|\mathcal{S}|)$ samples. By the $\Omega((1-\gamma)^{-3}\epsilon^{-2}|\mathcal{S}||\mathcal{A}|)$ sample bound on obtaining approximate value functions given in [AMK13], the above lemma implies a

$$Z = \Omega((1-\gamma)^{-3}\epsilon^{-2}|\mathcal{S}||\mathcal{A}|) - \tilde{O}((1-\gamma)^{-3}\epsilon^{-2}|\mathcal{S}|) = \Omega((1-\gamma)^{-3}\epsilon^{-2}|\mathcal{S}||\mathcal{A}|)$$

sample lower bound for obtaining an ϵ -optimal policy.

E Missing Proofs

Here are several standard properties of the Bellman value operator (see, e.g., [Ber13]).

Fact 1. Let $\mathbf{v}_1, \mathbf{v}_2 \in \mathbb{R}^{\mathcal{S}}$ be two vectors. Let \mathcal{T} be a value operator of a DMDP with discount factor γ . Let $\pi \in \mathcal{A}^{\mathcal{S}}$ be an arbitrary policy. Then the follows hold.

- **Monotonicity:** If $\mathbf{v}_1 \leq \mathbf{v}_2$ then $\mathcal{T}(\mathbf{v}_1) \leq \mathcal{T}(\mathbf{v}_2)$;
- **Contraction:** $\|\mathcal{T}(\mathbf{v}_1) - \mathcal{T}(\mathbf{v}_2)\|_{\infty} \leq \gamma \|\mathbf{v}_1 - \mathbf{v}_2\|_{\infty}$ and $\|\mathcal{T}_{\pi}(\mathbf{v}_1) - \mathcal{T}_{\pi}(\mathbf{v}_2)\|_{\infty} \leq \gamma \|\mathbf{v}_1 - \mathbf{v}_2\|_{\infty}$.

E.1 Missing Proofs from Section 5

To begin, we introduce two standard concentration results. Let $\mathbf{p} \in \Delta_{\mathcal{S}}$ be a probability vector, and $\mathbf{v} \in \mathbb{R}^{\mathcal{S}}$ be a vector. Let $\mathbf{p}_m \in \Delta_{\mathcal{S}}$ be empirical estimations of \mathbf{p} using m i.i.d. samples from the distribution \mathbf{p} . For instance, let these samples be $s_1, s_2, \dots, s_m \in \mathcal{S}$, then $\forall s \in \mathcal{S} : \mathbf{p}_m(s) = \sum_{j=1}^m \mathbf{1}(s_j = s)/m$.

Theorem E.1 (Hoeffding Inequality). Let $\delta \in (0, 1)$ be a parameter, vectors \mathbf{p}, \mathbf{p}_m and \mathbf{v} defined above. Then with probability at least $1 - \delta$,

$$|\mathbf{p}^{\top} \mathbf{v} - \mathbf{p}_m^{\top} \mathbf{v}| \leq \|\mathbf{v}\|_{\infty} \cdot \sqrt{2m^{-1} \log(2\delta^{-1})}.$$

Theorem E.2 (Bernstein Inequality). Let $\delta \in (0, 1)$ be a parameter, vectors \mathbf{p}, \mathbf{p}_m and \mathbf{v} defined as in Theorem E.1. Then with probability at least $1 - \delta$

$$|\mathbf{p}^{\top} \mathbf{v} - \mathbf{p}_m^{\top} \mathbf{v}| \leq \sqrt{2m^{-1} \text{Var}_{s' \sim \mathbf{p}}(\mathbf{v}(s')) \cdot \log(2\delta^{-1})} + (2/3)m^{-1} \|\mathbf{v}\|_{\infty} \cdot \log(2\delta^{-1}),$$

where $\text{Var}_{s' \sim \mathbf{p}}(\mathbf{v}(s')) = \mathbf{p}^{\top} \mathbf{v}^2 - (\mathbf{p}^{\top} \mathbf{v})^2$.

Proof of Lemma 5.1. By Theorem E.2 and a union bound over all (s, a) pairs, with probability at least $1 - \delta/4$, for every (s, a) , we have

$$|\tilde{\mathbf{w}}(s, a) - \mathbf{P}_{s,a}^{\top} \mathbf{v}^{(0)}| \leq \sqrt{2\sigma_{\mathbf{v}^{(0)}} \cdot m_1^{-1} \cdot L} + 2 \cdot (3m_1)^{-1} \cdot \|\mathbf{v}^{(0)}\|_{\infty} \cdot L, \quad (\text{E.1})$$

which is the first inequality.

Next, by Theorem E.1 and a union bound over all (s, a) pairs, with probability at least $1 - \delta/4$, for every (s, a) , we have

$$|\tilde{\mathbf{w}}(s, a) - \mathbf{P}_{s,a}^{\top} \mathbf{v}^{(0)}| \leq \|\mathbf{v}^{(0)}\|_{\infty} \cdot \sqrt{2m_1^{-1} L},$$

which we condition on. Thus

$$\begin{aligned} | \tilde{\mathbf{w}}(s, a)^2 - (\mathbf{P}_{s,a}^{\top} \mathbf{v}^{(0)})^2 | &= (\tilde{\mathbf{w}}(s, a) + \mathbf{P}_{s,a}^{\top} \mathbf{v}^{(0)}) \cdot | \tilde{\mathbf{w}}(s, a) - \mathbf{P}_{s,a}^{\top} \mathbf{v}^{(0)} | \\ &\leq \left[2\mathbf{P}_{s,a}^{\top} \mathbf{v}^{(0)} + \|\mathbf{v}^{(0)}\|_{\infty} \cdot \sqrt{2m_1^{-1} L} \right] \cdot | \tilde{\mathbf{w}}(s, a) - \mathbf{P}_{s,a}^{\top} \mathbf{v}^{(0)} | \\ &\leq 2(\mathbf{P}_{s,a}^{\top} \mathbf{v}^{(0)}) \cdot \|\mathbf{v}^{(0)}\|_{\infty} \cdot \sqrt{2m_1^{-1} L} + \|\mathbf{v}^{(0)}\|_{\infty}^2 \cdot 2m_1^{-1} L. \end{aligned}$$

Since $\mathbf{P}_{s,a}^\top \mathbf{v}^{(0)} \leq \|\mathbf{v}^{(0)}\|_\infty$, we obtain

$$|\tilde{\mathbf{w}}(s, a)^2 - (\mathbf{P}_{s,a}^\top \mathbf{v}^{(0)})^2| \leq 3\|\mathbf{v}^{(0)}\|_\infty^2 \cdot \sqrt{2m_1^{-1}L},$$

provided $2m_1^{-1}L \leq 1$. Next by Lemma E.1 and a union bound over all (s, a) pairs, with probability at least $1 - \delta/4$, for every (s, a) , we have

$$\left| \frac{1}{m_1} \sum_{j=1}^{m_1} \mathbf{v}^2(s_{s,a}^{(j)}) - \mathbf{P}_{s,a}^\top \mathbf{v}^2 \right| \leq \|\mathbf{v}^{(0)}\|_\infty^2 \cdot \sqrt{2L/m_1}.$$

By a union bound, we obtain, with probability at least $1 - \delta/2$,

$$\begin{aligned} |\hat{\boldsymbol{\sigma}}(s, a) - \boldsymbol{\sigma}_{\mathbf{v}^{(0)}}(s, a)| &\leq |\tilde{\mathbf{w}}(s, a)^2 - (\mathbf{P}_{s,a}^\top \mathbf{v}^{(0)})^2| + |m_1^{-1} \sum_{j=1}^{m_1} \mathbf{v}^2(s_{s,a}^{(j)}) - \mathbf{P}_{s,a}^\top \mathbf{v}^2| \\ &\leq 4\|\mathbf{v}^{(0)}\|_\infty^2 \cdot \sqrt{2m_1^{-1}L}. \end{aligned} \quad (\text{E.2})$$

By a union bound, with probability at least $1 - \delta$, both (E.1) and (E.2) hold, concluding the proof. \square

Proof of Lemma 5.2. Since for each (s, a) , $\boldsymbol{\sigma}_{\mathbf{v}}(s, a)$ is a variance, then we have triangle inequality,

$$\sqrt{\boldsymbol{\sigma}_{\mathbf{v}}} \leq \sqrt{\boldsymbol{\sigma}_{\mathbf{v}^*}} + \sqrt{\boldsymbol{\sigma}_{\mathbf{v}-\mathbf{v}^*}}.$$

Observing that

$$\boldsymbol{\sigma}_{\mathbf{v}-\mathbf{v}^*}(s, a) \leq \mathbf{P}_{s,a}^\top (\mathbf{v} - \mathbf{v}^*)^2 \leq \epsilon^2 \cdot \mathbf{1}.$$

We conclude the proof by taking a square root of all three sides of the above inequality. \square

Proof of Lemma 5.3. Recall that for each $(s, a) \in \mathcal{S} \times \mathcal{A}$,

$$\mathbf{g}^{(i)}(s, a) = \frac{1}{m_2} \sum_{j=1}^{m_2} [\mathbf{v}^{(i)}(s_{s,a}^{(j)}) - \mathbf{v}^{(0)}(s_{s,a}^{(j)})] - (1 - \gamma) \frac{u}{8},$$

where $m_2 = 128(1 - \gamma)^{-2} \cdot \log(2|\mathcal{S}||\mathcal{A}|R/\delta)$ and $s_{s,a}^{(1)}, s_{s,a}^{(2)}, \dots, s_{s,a}^{(m_2)}$ is a sequence of independent samples from $\mathbf{P}_{s,a}$. Thus by Theorem E.1 and a union bound over $\mathcal{S} \times \mathcal{A}$, with probability at least $1 - \delta/R$, we have

$$\begin{aligned} \forall (s, a) \in \mathcal{S} \times \mathcal{A} : \left| \sum_{j=1}^{m_2} [\mathbf{v}^{(i)}(s_{s,a}^{(j)}) - \mathbf{v}^{(0)}(s_{s,a}^{(j)})] - \mathbf{P}_{s,a}^\top [\mathbf{v}^{(i)} - \mathbf{v}^{(0)}] \right| \\ \leq \|\mathbf{v}^{(i)} - \mathbf{v}^{(0)}\|_\infty \sqrt{2m_2^{-1} \log(2|\mathcal{S}||\mathcal{A}|\delta^{r-1})} \leq (1 - \gamma)u/8. \end{aligned}$$

Finally by shifting the estimate to have one-sided error, we obtain the one-side error $(1 - \gamma)u/4$ in the statement of this lemma. \square

Proof of Lemma 5.4. For $i = 0$, $\mathbf{Q}^{(0)} = \mathbf{r} + \gamma\mathbf{w}$. By Lemma 5.1, with probability at least $1 - \delta$,

$$|\tilde{\mathbf{w}} - \mathbf{P}\mathbf{v}^{(0)}| \leq \sqrt{2\alpha_1\sigma_{\mathbf{v}^{(0)}}} + \frac{2}{3} \cdot \alpha_1 \cdot \|\mathbf{v}^{(0)}\|_\infty \mathbf{1},$$

and

$$|\hat{\sigma} - \sigma_{\mathbf{v}^{(0)}}| \leq 4\|\mathbf{v}^{(0)}\|_\infty^2 \cdot \sqrt{2\alpha_1} \mathbf{1}, \quad (\text{E.3})$$

which we condition on. We have

$$|\tilde{\mathbf{w}} - \mathbf{P}\mathbf{v}^{(0)}| \leq \sqrt{2\alpha_1\hat{\sigma}} + (4\alpha_1^{3/4}\|\mathbf{v}^{(0)}\|_\infty + \frac{2}{3} \cdot \alpha_1 \cdot \|\mathbf{v}^{(0)}\|_\infty) \mathbf{1}.$$

Thus

$$\mathbf{w} = \tilde{\mathbf{w}} - \sqrt{2\alpha_1\hat{\sigma}} - 4\alpha_1^{3/4}\|\mathbf{v}^{(0)}\|_\infty \mathbf{1} - \frac{2}{3} \cdot \alpha_1 \cdot \|\mathbf{v}^{(0)}\|_\infty \mathbf{1} \leq \mathbf{P}\mathbf{v}^{(0)}, \quad (\text{E.4})$$

and

$$\mathbf{w} \geq \mathbf{P}\mathbf{v}^{(0)} - 2\sqrt{2\alpha_1\hat{\sigma}} - (8\alpha_1^{3/4}\|\mathbf{v}^{(0)}\|_\infty + \frac{4}{3} \cdot \alpha_1 \cdot \|\mathbf{v}^{(0)}\|_\infty) \mathbf{1}.$$

By (E.3) and Lemma 5.2, we have

$$\sqrt{\hat{\sigma}} \leq \sqrt{\sigma_{\mathbf{v}^{(0)}}} + 2\|\mathbf{v}^{(0)}\|_\infty(2\alpha)^{1/4} \mathbf{1} \leq \sqrt{\sigma_{\mathbf{v}^*}} + u\mathbf{1} + 2\|\mathbf{v}^{(0)}\|_\infty(2\alpha)^{1/4} \mathbf{1}.$$

we have

$$\mathbf{w} \geq \mathbf{P}\mathbf{v}^{(0)} - 2\sqrt{2\alpha_1\sigma_{\mathbf{v}^*}} - 2\sqrt{2\alpha_1}u\mathbf{1} - 16\alpha_1^{3/4}\|\mathbf{v}^{(0)}\|_\infty \mathbf{1} - \frac{4}{3} \cdot \alpha_1 \cdot \|\mathbf{v}^{(0)}\|_\infty \mathbf{1} \quad (\text{E.5})$$

For the rest of the proof, we condition on the event that (E.4) and (E.5) hold, which happens with probability at least $1 - \delta$. Denote $\mathbf{v}^{(-1)} = \mathbf{0}$. Thus we have $\mathbf{v}^{(-1)} \leq \mathbf{v}^{(0)} \leq \mathcal{T}_{\pi^{(0)}}(\mathbf{v}^{(0)})$. Next we prove the lemma by induction on i . Assume for some $i \geq 1$, with probability at least $1 - (i-1)\delta'$ the following holds,

$$\forall 0 \leq k \leq i-1 : \quad \mathbf{v}^{(k-1)} \leq \mathbf{v}^{(k)} \leq \mathcal{T}_{\pi^{(k)}}(\mathbf{v}^{(k)}),$$

which we condition on. Next we show that the lemma statement holds for $k = i$. By definition of $\mathbf{v}^{(i)}$ (Line 27 and 28),

$$\mathbf{v}^{(i-1)} \leq \mathbf{v}^{(i)} \quad \text{and} \quad \mathbf{v}(\mathbf{Q}^{(i-1)}) \leq \mathbf{v}^{(i)}.$$

Furthermore, since $\mathbf{v}^{(0)} \leq \mathbf{v}^{(1)} \leq \dots \leq \mathbf{v}^{(i-1)} \leq \mathcal{T}_{\pi^{(i-1)}}\mathbf{v}^{(i-1)} \leq \mathcal{T}\mathbf{v}^{(i-1)} \leq \mathcal{T}^\infty\mathbf{v}^{(i-1)} = \mathbf{v}^*$, we have

$$\mathbf{v}^{(i)} - \mathbf{v}^{(0)} \leq \mathbf{v}^* - \mathbf{v}^{(0)} \leq u\mathbf{1}.$$

By Lemma 5.3, we have, with probability at least $1 - \delta'$

$$\mathbf{P}[\mathbf{v}^{(i)} - \mathbf{v}^{(0)}] - \frac{(1-\gamma)u}{8} \cdot \mathbf{1} \leq \mathbf{g}^{(i)} \leq \mathbf{P}[\mathbf{v}^{(i)} - \mathbf{v}^{(0)}], \quad (\text{E.6})$$

which we condition on for the rest of the proof. Thus we have

$$\mathbf{Q}^{(i)} = \mathbf{r} + \gamma(\mathbf{w} + \mathbf{g}^{(i)}) \leq \mathbf{r} + \gamma(\mathbf{P}\mathbf{v}^{(0)} + \mathbf{P}\mathbf{v}^{(i)} - \mathbf{P}\mathbf{v}^{(0)}) = \mathbf{r} + \gamma\mathbf{P}\mathbf{v}^{(i)}.$$

To show $\mathbf{v}^{(i)} \leq \mathcal{T}_{\pi^{(i)}} \mathbf{v}^{(i)}$, we notice that if for some s , $\pi^{(i)}(s) \neq \pi^{(i-1)}(s)$, then

$$\mathbf{v}^{(i)}(s) \leq [\mathcal{T}_{\pi^{(i)}} \mathbf{v}^{(i-1)}](s) \leq [\mathcal{T}_{\pi^{(i)}} \mathbf{v}^{(i)}](s),$$

where the first inequality follows from $\mathbf{v}^{(i)}(s) \leq \mathbf{r}(s, \pi^{(i)}(s)) + \gamma \mathbf{P}_{s, \pi^{(i)}(s)}^\top \mathbf{v}^{(i-1)} = \mathcal{T}_{\pi^{(i)}} \mathbf{v}^{(i-1)}$. On the other hand, if $\pi^{(i)}(s) = \pi^{(i-1)}(s)$, then

$$\mathbf{v}^{(i)}(s) = \mathbf{v}^{(i-1)}(s) \leq (\mathcal{T}_{\pi^{(i-1)}} \mathbf{v}^{(i-1)})(s) \leq (\mathcal{T}_{\pi^{(i-1)}} \mathbf{v}^{(i)})(s) = (\mathcal{T}_{\pi^{(i)}} \mathbf{v}^{(i)})(s).$$

This completes the induction step. Lastly, combining (E.5) and (E.6), we have

$$\begin{aligned} \mathbf{Q}^* - \mathbf{Q}^{(i)} &= \mathbf{Q}^* - \mathbf{r} - \gamma(\mathbf{w} + \mathbf{g}^{(i)}) = \gamma \mathbf{P} \mathbf{v}(\mathbf{Q}^*) - \gamma(\mathbf{w} + \mathbf{g}^{(i)}) \\ &= \gamma \mathbf{P} \mathbf{v}(\mathbf{Q}^*) - \gamma \mathbf{P}(\mathbf{v}^{(i)} - \mathbf{v}^{(0)}) - \gamma \mathbf{P} \mathbf{v}^{(0)} + \boldsymbol{\xi}^{(i)} \\ &= \gamma \mathbf{P} \mathbf{v}(\mathbf{Q}^*) - \gamma \mathbf{P} \mathbf{v}^{(i)} + \boldsymbol{\xi}^{(i)}, \end{aligned}$$

where

$$\boldsymbol{\xi}^{(i)} \leq (1 - \gamma)u/8 \cdot \mathbf{1} + 2\sqrt{2\alpha_1 \boldsymbol{\sigma}_{\mathbf{v}^*}} + 2\sqrt{2\alpha_1}u \cdot \mathbf{1} + 16\alpha_1^{3/4} \|\mathbf{v}^{(0)}\|_\infty \cdot \mathbf{1} + (4/3) \cdot \alpha_1 \cdot \|\mathbf{v}^{(0)}\|_\infty \cdot \mathbf{1},$$

where $\alpha_1 = \log(8|\mathcal{S}||\mathcal{A}|\delta^{-1})/m_1 \leq 1$. Moreover, since $\mathbf{v}(\mathbf{Q}^{(i-1)}) \leq \mathbf{v}^{(i)}$, we obtain

$$\mathbf{Q}^* - \mathbf{Q}^{(i)} \leq \gamma \mathbf{P} \mathbf{v}(\mathbf{Q}^*) - \gamma \mathbf{P} \mathbf{v}(\mathbf{Q}^{(i-1)}) + \boldsymbol{\xi}^{(i)} \leq \gamma \mathbf{P}^{\pi^*} \mathbf{Q}^* - \gamma \mathbf{P}^{\pi^*} \mathbf{Q}^{(i-1)} + \boldsymbol{\xi}^{(i)},$$

where π^* is an arbitrary optimal policy and we use the fact that $\max_a \mathbf{Q}^*(s, a) = \mathbf{Q}^*(s, \pi^*(s))$. This completes the proof of the lemma. \square

Proof of Proposition 5.4.1. Recall that we are able to sample a state from each $\mathbf{P}_{s,a}$ with time $O(1)$. Let $\beta = (1 - \gamma)^{-1}$, $R = \lceil c_1 \beta \ln[\beta u^{-1}] \rceil$, $m_1 = c_2 \beta^3 u^{-2} \cdot \log(8|\mathcal{S}||\mathcal{A}|\delta^{-1})$ and $m_2 = c_3 \beta^2 \cdot \log[2R|\mathcal{S}||\mathcal{A}|\delta^{-1}]$ for some constants c_1, c_2 and c_3 required in Algorithm 1. In the following proof, we set c_1, c_2, c_3 to be sufficiently large but otherwise arbitrary absolute constants (e.g., $c_1 \geq 4, c_2 \geq 8192, c_3 \geq 128$). By Lemma 5.4, with probability at least $1 - 2\delta$ for each $1 \leq i \leq R$, we have $\mathbf{v}^{(i-1)} \leq \mathbf{v}^{(i)} \leq \mathcal{T}_{\pi^{(i)}} \mathbf{v}^{(i)}$, and $\mathbf{Q}^{(i)} \leq \mathbf{r} + \gamma \mathbf{P} \mathbf{v}^{(i)}$,

$$\mathbf{Q}^* - \mathbf{Q}^{(i)} \leq \gamma \mathbf{P}^{\pi^*} [\mathbf{Q}^* - \mathbf{Q}^{(i-1)}] + \boldsymbol{\xi},$$

where

$$\boldsymbol{\xi} \leq (1 - \gamma)u/C \cdot \mathbf{1} + C\sqrt{\alpha_1 \boldsymbol{\sigma}_{\mathbf{v}^*}} + C\alpha_1^{3/4} \|\mathbf{v}^{(0)}\|_\infty \cdot \mathbf{1}$$

for $\alpha_1 = \log(8|\mathcal{S}||\mathcal{A}|\delta^{-1})/m_1$ and sufficiently large constant C . Solving the recursion, we obtain

$$\begin{aligned} \mathbf{Q}^* - \mathbf{Q}^{(R-1)} &\leq \gamma^{R-1} \mathbf{P}^{\pi^*} [\mathbf{Q}^* - \mathbf{Q}^0] + \sum_{i=0}^{R-1} \gamma^i (\mathbf{P}^{\pi^*})^i \boldsymbol{\xi} \\ &\leq \gamma^{R-1} \mathbf{P}^{\pi^*} [\mathbf{Q}^* - \mathbf{Q}^0] + (I - \gamma \mathbf{P}^{\pi^*})^{-1} \boldsymbol{\xi}. \end{aligned}$$

We first apply a naïve bound $\|\mathbf{P}^{\pi^*} [\mathbf{Q}^* - \mathbf{Q}^0]\|_\infty \leq (1 - \gamma)^{-1}$. Hence

$$\gamma^{R-1} \mathbf{P}^{\pi^*} [\mathbf{Q}^* - \mathbf{Q}^0] \leq \frac{u}{4} \cdot \mathbf{1},$$

where $R = \lceil (1 - \gamma)^{-1} \ln[4(1 - \gamma)^{-1}u^{-1}] \rceil + 1$. The next step is the key to the improvement in our analysis. We further apply the bound in Lemma C.1, given by

$$(\mathbf{I} - \gamma \mathbf{P}^{\pi^*})^{-1} \sqrt{\sigma_{\mathbf{v}^*}} \leq \min(2\gamma^{-1}(1 - \gamma)^{-1.5}, (1 - \gamma)^{-2}) \cdot \mathbf{1} \leq 3(1 - \gamma)^{-1.5} \cdot \mathbf{1},$$

where the last inequality follows since $\min(2\gamma^{-1}, (1 - \gamma)^{-1/2}) \leq 3$. With $\|(\mathbf{I} - \gamma \mathbf{P}^{\pi^*})^{-1} \mathbf{1}\|_\infty \leq (1 - \gamma)^{-1}$ and $\|\mathbf{v}^{(0)}\|_\infty \leq (1 - \gamma)^{-1}$, we have,

$$\begin{aligned} (\mathbf{I} - \gamma \mathbf{P}^{\pi^*})^{-1} \boldsymbol{\xi} &\leq \left[\frac{u}{8} + C' \sqrt{\frac{2\alpha_1}{\gamma^2(1 - \gamma)^3}} + C' \frac{\alpha_1^{3/4}}{(1 - \gamma)^2} \right] \cdot \mathbf{1} \\ &\leq \left[\frac{u}{8} + \frac{u}{16} + \left(\frac{(1 - \gamma)^3 u^2}{C'' \cdot (1 - \gamma)^{8/3}} \right)^{3/4} \right] \cdot \mathbf{1} \\ &\leq \frac{u}{4} \cdot \mathbf{1}, \end{aligned}$$

for some sufficiently large C' and C'' , which depend on c_1, c_2 and c_3 . Since $\mathbf{v}(\mathbf{Q}^{(R-1)}) \leq \mathbf{v}^{(R)}$, we have

$$\mathbf{v}^* - \mathbf{v}^{(R)} \leq \mathbf{v}^* - \mathbf{v}(\mathbf{Q}^{(R-1)}) \leq \gamma^{R-1} \mathbf{P}^{\pi^*} [\mathbf{Q}^* - \mathbf{Q}^0] + (\mathbf{I} - \gamma \mathbf{P}^{\pi^*})^{-1} \boldsymbol{\xi} \leq \frac{u}{2} \cdot \mathbf{1}.$$

This completes the proof of the correctness. It remains to bound the time complexity. The initialization stage costs $O(m_1)$ time per (s, a) . Each iteration costs $O(m_2)$ time per (s, a) . We thus have the total time complexity as

$$O(m_1 + Rm_2)|\mathcal{S}||\mathcal{A}| = O\left[\frac{|\mathcal{S}||\mathcal{A}|}{(1 - \gamma)^3} \cdot \log \frac{|\mathcal{S}||\mathcal{A}|}{\delta \cdot (1 - \gamma) \cdot u} \cdot \left(\frac{1}{u^2} + \log \frac{1}{(1 - \gamma) \cdot u} \right) \right].$$

Since $\log[(1 - \gamma)^{-1}u^{-1}] = O(\log[(1 - \gamma)^{-1}]u^{-2})$, we conclude the proof. \square

E.2 Missing Analysis of Halving Errors

We refer in this section to Algorithm 1 as a subroutine HALFERR, which given an input MDP \mathcal{M} with a sampling oracle, an input value function $\mathbf{v}^{(i)}$ and an input policy $\pi^{(i)}$, outputs an value function $\mathbf{v}^{(i+1)}$ and a policy $\pi^{(i+1)}$ such that, with high probability (over the new samples of the sampling oracle),

$$\|\mathbf{Q}^{(i+1)} - \mathbf{Q}^*\|_\infty \leq \|\mathbf{Q}^{(i)} - \mathbf{Q}^*\|_\infty / 2 \quad \text{and} \quad \|\mathbf{v}^{\pi^{(i+1)}} - \mathbf{v}^*\|_\infty \leq \|\mathbf{v}^{\pi^{(i)}} - \mathbf{v}^*\|_\infty / 2.$$

After $\log[\epsilon^{-1}(1 - \gamma)^{-1}]$ calls of the subroutine HALFERR, the final output policy and value functions are ϵ -close to the optimal ones with high probability.

We summarize our meta algorithm in Algorithm 2. Note that in the algorithm, each call of HALFERR will draw new samples from the sampling oracle. These new samples guarantee the independence of successive improvements and also save space of the algorithm. For instance, the algorithm HALFERR only needs to use $O(|\mathcal{S}||\mathcal{A}|)$ words of memory instead of storing all the samples. The guarantee of the algorithm is summarized in Proposition E.2.1.

Proposition E.2.1. Let $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathbf{r}, \mathbf{P}, \gamma)$ with a sampling oracle. Suppose HALFERR is an algorithm that takes an input $\mathbf{v}^{(i)}$ and an input policy $\pi^{(i)}$ and a number $u \in [0, (1 - \gamma)^{-1}]$ satisfying $\mathbf{v}^* - u\mathbf{1} \leq \mathbf{v}^{(i)} \leq \mathbf{v}^{\pi^{(i)}}$, halts in time τ and outputs a $\mathbf{v}^{(i+1)}$ and a policy $\pi^{(i+1)}$ satisfying,

$$\mathbf{v}^* - \frac{u}{2} \cdot \mathbf{1} \leq \mathbf{v}^{(i+1)} \leq \mathbf{v}^{\pi^{(i+1)}} \leq \mathbf{v}^*.$$

with probability at least $1 - (1 - \gamma) \cdot \epsilon \cdot \delta$ (over the randomness of the new samples given by the sampling oracle), then the meta algorithm described in Algorithm 2, given input \mathcal{M} and the sampling oracle, halts in $\tau \cdot \log(\epsilon^{-1} \cdot (1 - \gamma)^{-1})$ and outputs an policy $\pi^{(R)}$ such that

$$\mathbf{v}^* - \epsilon \cdot \mathbf{1} \leq \mathbf{v}^{(R)} \leq \mathbf{v}^{\pi^{(R)}} \leq \mathbf{v}^*$$

with probability at least $1 - \delta$ (over the randomness of all samples drawn from the sampling oracle). Moreover, if HALFERR uses space s , then the meta algorithm uses space $s + O(|\mathcal{S}||\mathcal{A}|)$. If each call of HALFERR takes m samples from the oracle, then the overall samples taken by Algorithm 2 is $m \cdot \log(\epsilon^{-1} \cdot (1 - \gamma)^{-1})$.

The proof of this proposition is a straightforward application of conditional probability.

Proof of Proposition E.2.1. The proof follows from a straightforward induction. For simplicity, denote $\beta = (1 - \gamma)^{-1}$. In the meta-algorithm, the initialization is $\mathbf{v}^{(0)} = \mathbf{0}$ and $\pi^{(0)}$ is an arbitrary policy. Thus $\mathbf{v}^* - \beta \cdot \mathbf{1} \leq \mathbf{v}^{(0)} \leq \mathbf{v}^{\pi^{(0)}}$. By running the meta-algorithm, we obtain a sequence of value functions and policies: $\{\mathbf{v}^{(i)}\}_{i=0}^R$ and $\{\pi^{(i)}\}_{i=0}^R$. Since each call of the HALFERR uses new samples from the oracle, the sequence of value functions and policies satisfies strong Markov property (given $(\mathbf{v}^{(i)}, \pi^{(i)})$, $(\mathbf{v}^{(i+1)}, \pi^{(i+1)})$ is independent with $\{(\mathbf{v}^{(j)}, \pi^{(j)})\}_{j=0}^{i-1}$). Thus

$$\begin{aligned} & \Pr[\mathbf{v}^* - 2^{-R}\beta \cdot \mathbf{1} \leq \mathbf{v}^{(R)} \leq \mathbf{v}^{\pi^{(R)}}] \\ & \geq \prod_{i=1}^R \Pr[\mathbf{v}^* - 2^{-i}\beta \cdot \mathbf{1} \leq \mathbf{v}^{(i)} \leq \mathbf{v}^{\pi^{(i)}} \mid \mathbf{v}^* - 2^{-i+1}\beta \cdot \mathbf{1} \leq \mathbf{v}^{(i-1)} \leq \mathbf{v}^{\pi^{(i-1)}}] \\ & \geq 1 - \delta. \end{aligned}$$

Since $2^{-R}(1 - \gamma)^{-1} \leq \epsilon$, we conclude the proof. \square

Proof of Theorem 5.5. Our algorithm is simply plugging in Algorithm 1 as the HALFERR subroutine in Algorithm 2. The correctness is guaranteed by Proposition E.2.1 and Proposition 5.4.1. The running time guarantee follows from a straightforward calculation. \square

F Extension to Finite Horizon

In this section we show how to apply similar techniques to achieve improved sample complexities for solving finite Horizon MDPs given a generative model and we prove that the sample complexity we achieve is optimal up to logarithmic factors.

The finite horizon problem is to compute an optimal non-stationary policy over a fixed time horizon H , i.e. a policy of the form $\pi(s, h)$ for $s \in \mathcal{S}$ and $h \in \{0, \dots, H\}$, where the reward is the expected cumulative (un-discounted) reward for following this policy. In classic value iteration, this is typically done using a backward recursion from time $H, H - 1, \dots, 0$. We show how to use the ideas in this paper to solve for an ϵ -approximate policy. As we have shown in the discounted case, it is suffice to show an algorithm that decrease the error of the value at each stage by half. Our algorithm is presented in Algorithm 3.

To analyze the algorithm, we first provide an analogous lemma of Lemma 5.1,

Lemma F.1 (Empirical Estimation Error). *Let $\tilde{\mathbf{w}}_h$ and $\hat{\boldsymbol{\sigma}}_h$ be computed in Line 10 of Algorithm 3. Recall that $\tilde{\mathbf{w}}_h$ and $\hat{\boldsymbol{\sigma}}_h$ are empirical estimates of $\mathbf{P}\mathbf{v}_h$ and $\boldsymbol{\sigma}_{\mathbf{v}_h} = \mathbf{P}\mathbf{v}_h^2 - (\mathbf{P}\mathbf{v}_h)^2$ using m_1 samples per (s, a) pair. Then with probability at least $1 - \delta$, for $L \stackrel{\text{def}}{=} \log(8|\mathcal{S}||\mathcal{A}|\delta^{-1})$ and every $h = 1, 2, \dots, H$, we have*

$$|\tilde{\mathbf{w}}_h - \mathbf{P}^\top \mathbf{v}_h^{(0)}| \leq \sqrt{2m_1^{-1} \boldsymbol{\sigma}_{\mathbf{v}_h^{(0)}} \cdot L} + 2(3m_1)^{-1} \|\mathbf{v}_h^{(0)}\|_\infty L \quad (\text{F.1})$$

and

$$\forall (s, a) \in \mathcal{S} \times \mathcal{A} : |\hat{\boldsymbol{\sigma}}_h(s, a) - \boldsymbol{\sigma}_{\mathbf{v}_h^{(0)}}(s, a)| \leq 4\|\mathbf{v}_h^{(0)}\|_\infty^2 \cdot \sqrt{2m_1^{-1} L}. \quad (\text{F.2})$$

Proof. The proof of this lemma is identical to that of Lemma 5.1. \square

An analogous lemma to Lemma 5.3 is also presented here.

Lemma F.2. *Let $\mathbf{g}_h^{(i)}$ be the estimate of $\mathbf{P}[\mathbf{v}_h^{(i)} - \mathbf{v}_h^{(0)}]$ defined in Line 27 of Algorithm 3. Then conditioning on the event that $\|\mathbf{v}_h^{(i)} - \mathbf{v}_h^{(0)}\|_\infty \leq 2\epsilon$, with probability at least $1 - \delta/H$,*

$$\mathbf{P}[\mathbf{v}_h^{(i)} - \mathbf{v}_h^{(0)}] - \frac{\epsilon}{4H} \cdot \mathbf{1} \leq \mathbf{g}_h^{(i)} \leq \mathbf{P}[\mathbf{v}_h^{(i)} - \mathbf{v}_h^{(0)}]$$

provided appropriately chosen constants in Algorithm 3.

Proof. The proof of this lemma is identical to that of Lemma 5.3 except that $(1 - \gamma)^{-1}$ is replaced with H . \square

Similarly, we can show the following improvement lemma.

Lemma F.3. *Let \mathbf{Q}_h be the estimated Q -function of \mathbf{v}_{h+1} in Line 30 of Algorithm 3. Let $\mathbf{Q}_h^* = \mathbf{r} + \mathbf{P}_h \mathbf{v}_{h+1}^*$ be the optimal Q -function of the DMDP. Let $\pi(\cdot, h)$ and \mathbf{v}_h be estimated in iteration h , as defined in Line 24 and 25. Let π^* be an optimal policy for the DMDP. For a policy π , let $\mathbf{P}_h^\pi \mathbf{Q} \in \mathbb{R}^{\mathcal{S} \times \mathcal{A}}$ be defined as $(\mathbf{P}_h^\pi \mathbf{Q})(s, a) = \sum_{s' \in \mathcal{S}} \mathbf{P}_{s,a}(s') \mathbf{Q}(s', \pi(s', h))$. Suppose for all $h \in [H-1]$, $\mathbf{v}_h^{(0)} \leq \mathcal{T}_{\pi^{(0)}(\cdot, h)} \mathbf{v}_{h+1}^{(0)}$. Let $\mathbf{v}_{H+1} \stackrel{\text{def}}{=} \mathbf{0}$ and $\mathbf{Q}_{H+1} \stackrel{\text{def}}{=} \mathbf{0}$. Then, with probability at least $1 - 2\delta$, for all $1 \leq h \leq H$, $\mathbf{v}_h^{(0)} \leq \mathbf{v}_h \leq \mathcal{T}_{\pi(\cdot, h)} \mathbf{v}_{h+1} \leq \mathbf{v}_h^*$, $\mathbf{Q}_h \leq \mathbf{r} + \mathbf{P}_h \mathbf{v}_{h+1}$ and*

$$\mathbf{Q}_h^* - \mathbf{Q}_h \leq \mathbf{P}_h^{\pi^*} [\mathbf{Q}_{h+1}^* - \mathbf{Q}_{h+1}] + \boldsymbol{\xi}_h,$$

where the error vector $\boldsymbol{\xi}_h$ satisfies

$$\mathbf{0} \leq \boldsymbol{\xi}_h \leq 8H^{-1}u \cdot \mathbf{1} + 2\sqrt{2\alpha_1 \boldsymbol{\sigma}_{\mathbf{v}_{h+1}^*}} + 2\sqrt{2\alpha_1}u \cdot \mathbf{1} + 16\alpha_1^{3/4} \|\mathbf{v}_{h+1}^{(0)}\|_\infty \cdot \mathbf{1} + (4/3) \cdot \alpha_1 \cdot \|\mathbf{v}_{h+1}^{(0)}\|_\infty \cdot \mathbf{1},$$

and $\alpha_1 = \log(8|\mathcal{S}||\mathcal{A}|H\delta^{-1})/m_1$.

Proof of Lemma F.3. By Lemma 5.1, for any $h = 1, 2, \dots, H$, with probability at least $1 - \delta/H$,

$$|\tilde{\mathbf{w}}_h - \mathbf{P}\mathbf{v}_{h+1}| \leq \sqrt{2\alpha_1 \boldsymbol{\sigma}_{\mathbf{v}_{h+1}^{(0)}}} + \frac{2}{3} \cdot \alpha_1 \cdot \|\mathbf{v}_{h+1}^{(0)}\|_\infty \cdot \mathbf{1},$$

and

$$|\hat{\boldsymbol{\sigma}}_{h+1} - \boldsymbol{\sigma}_{\mathbf{v}_{h+1}^{(0)}}| \leq 4\|\mathbf{v}_{h+1}^{(0)}\|_\infty^2 \cdot \sqrt{2\alpha_1} \cdot \mathbf{1}, \quad (\text{F.3})$$

which we condition on. We have

$$|\tilde{\mathbf{w}}_h - \mathbf{P}\mathbf{v}_{h+1}^{(0)}| \leq \sqrt{2\alpha_1\widehat{\sigma}_{h+1}} + (4\alpha_1^{3/4}\|\mathbf{v}_{h+1}^{(0)}\|_\infty + \frac{2}{3} \cdot \alpha_1 \cdot \|\mathbf{v}_{h+1}^{(0)}\|_\infty)\mathbf{1}.$$

Thus

$$\mathbf{w}_h = \tilde{\mathbf{w}}_h - \sqrt{2\alpha_1\widehat{\sigma}_{h+1}} - 4\alpha_1^{3/4}\|\mathbf{v}_{h+1}^{(0)}\|_\infty\mathbf{1} - \frac{2}{3} \cdot \alpha_1 \cdot \|\mathbf{v}_{h+1}^{(0)}\|_\infty\mathbf{1} \leq \mathbf{P}\mathbf{v}_{h+1}^{(0)}, \quad (\text{F.4})$$

and

$$\mathbf{w}_h \geq \mathbf{P}\mathbf{v}_{h+1}^{(0)} - 2\sqrt{2\alpha_1\widehat{\sigma}_{h+1}} - (8\alpha_1^{3/4}\|\mathbf{v}_{h+1}^{(0)}\|_\infty + \frac{4}{3} \cdot \alpha_1 \cdot \|\mathbf{v}_{h+1}^{(0)}\|_\infty)\mathbf{1}.$$

By (E.3) and Lemma 5.2, we have

$$\sqrt{\widehat{\sigma}_{h+1}} \leq \sqrt{\sigma_{\mathbf{v}_{h+1}^{(0)}}} + 2\|\mathbf{v}_{h+1}^{(0)}\|_\infty(2\alpha)^{1/4}\mathbf{1} \leq \sqrt{\sigma_{\mathbf{v}_{h+1}^*}} + \epsilon\mathbf{1} + 2\|\mathbf{v}_{h+1}^{(0)}\|_\infty(2\alpha)^{1/4}\mathbf{1}.$$

we have

$$\mathbf{w}_h \geq \mathbf{P}\mathbf{v}_{h+1}^{(0)} - 2\sqrt{2\alpha_1\sigma_{\mathbf{v}_{h+1}^*}} - 2\sqrt{2\alpha_1}\epsilon\mathbf{1} - 16\alpha_1^{3/4}\|\mathbf{v}_{h+1}^{(0)}\|_\infty\mathbf{1} - \frac{4}{3} \cdot \alpha_1 \cdot \|\mathbf{v}_{h+1}^{(0)}\|_\infty\mathbf{1} \quad (\text{F.5})$$

For the rest of the proof, we condition on the event that (F.4) and (F.5) hold for all $h = 1, 2, \dots, H$, which happens with probability at least $1 - \delta$. Denote $\mathbf{v}_{H+1}^* = \mathbf{v}_{H+1} = \mathbf{v}_{H+1}^{(0)} = \mathbf{0}$. Thus we have $\mathbf{v}_{H+1}^{(0)} \leq \mathbf{v}_{H+1} \leq \mathbf{v}_{H+1}^*$. Next we prove the lemma by induction on h . Assume for some h , with probability at least $1 - (h-1)\delta/H$ the following holds, for all $h' = h+1, h+2, \dots, H$,

$$\mathbf{v}_{h'}^{(0)} \leq \mathbf{v}_{h'} \leq \mathbf{v}_{h'}^*,$$

which we condition on. Next we show that the lemma statement holds for h as well. By definition of \mathbf{v}_h (Line 27 and 28),

$$\mathbf{v}_h^{(0)} \leq \mathbf{v}_h \quad \text{and} \quad \mathbf{v}(\mathbf{Q}_h) \leq \mathbf{v}_h.$$

Furthermore, since $\mathbf{v}_{h+1}^{(0)} \leq \mathbf{v}_{h+1}^* \leq \mathbf{v}_{h+1}^{(0)} + u\mathbf{1}$ we have

$$\mathbf{v}_{h+1}^* - \mathbf{v}_{h+1} \leq \mathbf{v}_{h+1}^* - \mathbf{v}_{h+1}^{(0)} \leq u\mathbf{1}.$$

By Lemma 5.3, we have, with probability at least $1 - \delta'$

$$\mathbf{P}[\mathbf{v}_{h+1} - \mathbf{v}_{h+1}^{(0)}] - \frac{u}{8H} \cdot \mathbf{1} \leq \mathbf{g}_h \leq \mathbf{P}[\mathbf{v}_{h+1} - \mathbf{v}_{h+1}^{(0)}], \quad (\text{F.6})$$

which we condition on for the rest of the proof. Thus we have

$$\mathbf{Q}_h = \mathbf{r} + (\mathbf{w}_h + \mathbf{g}_h) \leq \mathbf{r} + \mathbf{P}\mathbf{v}_{h+1}^{(0)} + \mathbf{P}\mathbf{v}_{h+1} - \mathbf{P}\mathbf{v}_{h+1}^{(0)} = \mathbf{r} + \mathbf{P}\mathbf{v}_{h+1} \leq \mathbf{Q}_h^*.$$

To show $\mathbf{v}_h \leq \mathcal{T}_{\pi(\cdot, h)}\mathbf{v}_{h+1}$, we notice that if for some s , $\pi(s, h) \neq \pi^{(0)}(s, h)$, then,

$$\mathbf{v}_h(s) \leq \mathbf{r}(s, \pi(s, h)) + \mathbf{P}_{s, \pi(s, h)}^\top \mathbf{v}_{h+1} = \mathcal{T}_{\pi(\cdot, h)}\mathbf{v}_{h+1}.$$

On the other hand, if $\pi(s, h) = \pi^{(0)}(s, h)$, then

$$\forall s \in \mathcal{S} : \quad \mathbf{v}_h(s) = \mathbf{v}_h^{(0)}(s) \leq (\mathcal{T}_{\pi^{(0)}(\cdot, h)} \mathbf{v}_{h+1}^{(0)})(s) \leq (\mathcal{T}_{\pi^{(0)}(\cdot, h)} \mathbf{v}_{h+1})(s) = (\mathcal{T}_{\pi(\cdot, h)} \mathbf{v}_{h+1})(s).$$

This completes the induction step. Lastly, combining (F.5) and (F.6), we have

$$\begin{aligned} \mathbf{Q}_h^* - \mathbf{Q}_h &= \mathbf{Q}_h^* - \mathbf{r} - (\mathbf{w}_h + \mathbf{g}_h) = \mathbf{P}\mathbf{v}(\mathbf{Q}_{h+1}^*) - (\mathbf{w}_h + \mathbf{g}_h) \\ &= \mathbf{P}\mathbf{v}(\mathbf{Q}_{h+1}^*) - \mathbf{P}(\mathbf{v}_{h+1} - \mathbf{v}_{h+1}^{(0)}) - \mathbf{P}\mathbf{v}_{h+1} + \boldsymbol{\xi}_h \\ &= \mathbf{P}\mathbf{v}(\mathbf{Q}_{h+1}^*) - \mathbf{P}\mathbf{v}_{h+1} + \boldsymbol{\xi}_h, \end{aligned}$$

where

$$\boldsymbol{\xi}_h \leq H^{-1}u/8 \cdot \mathbf{1} + 2\sqrt{2\alpha_1\sigma_{\mathbf{v}_{h+1}^*}} + 2\sqrt{2\alpha_1}u \cdot \mathbf{1} + 16\alpha_1^{3/4}\|\mathbf{v}_{h+1}^{(0)}\|_\infty \cdot \mathbf{1} + (4/3) \cdot \alpha_1 \cdot \|\mathbf{v}_{h+1}^{(0)}\|_\infty \cdot \mathbf{1},$$

where $\alpha_1 = \log(8|\mathcal{S}||\mathcal{A}|\delta^{-1})/m_1$. Moreover, since $\mathbf{v}(\mathbf{Q}_{h+1}) \leq \mathbf{v}_{h+1}$, we obtain

$$\mathbf{Q}_h^* - \mathbf{Q}_h \leq \mathbf{P}\mathbf{v}(\mathbf{Q}_{h+1}^*) - \mathbf{P}\mathbf{v}(\mathbf{Q}_{h+1}) + \boldsymbol{\xi}_h \leq \mathbf{P}_h^{\pi^*} \mathbf{Q}_{h+1}^* - \mathbf{P}_h^{\pi^*} \mathbf{Q}_{h+1} + \boldsymbol{\xi}_h,$$

where π^* is an arbitrary optimal policy and we use the fact that $\max_a \mathbf{Q}_h^*(s, a) = \mathbf{Q}_h^*(s, \pi^*(s, h))$. This completes the proof of the lemma. \square

Furthermore, we show an analogous lemma of Lemma C.1.

Lemma F.4 (Upper Bound on Variance). *For any π , we have*

$$\left\| \sum_{h'=h}^{H-1} \left(\prod_{i=h+1}^{h'} \mathbf{P}_i^\pi \right) \sqrt{\sigma_{\mathbf{v}_{h'+1}^\pi}} \right\|_\infty^2 \leq H^{3/2}.$$

Proof. First, by Cauchy-Swartz inequality, we have

$$\sum_{h'=h}^{H-1} \left(\prod_{i=h+1}^{h'} \mathbf{P}_i^\pi \right) \sqrt{\sigma_{\mathbf{v}_{h'+1}^\pi}} \leq \sqrt{H \sum_{h'=h}^{H-1} \left(\prod_{i=h+1}^{h'} \mathbf{P}_i^\pi \right) \sigma_{\mathbf{v}_{h'+1}^\pi}}.$$

Next, by a similar argument of the proof of Lemma C.2, we can show that

$$\left[\sum_{h'=h}^{H-1} \left(\prod_{i=h+1}^{h'} \mathbf{P}_i^\pi \right) \sigma_{\mathbf{v}_{h'+1}^\pi} \right] (s) = \text{Var} \left[\sum_{t=h}^H r(s^t, \pi(s^t, t)) \middle| s^h = s \right] \leq H^2.$$

This completes the proof. \square

We are now ready to present the guarantee of the algorithm 3.

Proposition F.4.1. On an input value vectors $\mathbf{v}_1^{(0)}, \mathbf{v}_2^{(0)}, \dots, \mathbf{v}_H^{(0)}$, policy $\pi^{(0)}$, and parameters $u \in (0, \beta], \delta \in (0, 1)$ such that $\mathbf{v}_h^{(0)} \leq \mathcal{T}_{\pi^{(0)}(\cdot, h)} \mathbf{v}_{h+1}^{(0)}$ for all $h \in [H-1]$, and $\mathbf{v}_h^{(0)} \leq \mathbf{v}_h^* \leq \mathbf{v}_h^{(0)} + u\mathbf{1}$, Algorithm 3 halts in time $O[u^{-2} \cdot H^4 |\mathcal{S}| |\mathcal{A}| \cdot \log(|\mathcal{S}| |\mathcal{A}| \delta^{-1} H u^{-1})]$ and outputs $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_H$ and $\pi : \mathcal{S} \times [H] \rightarrow \mathcal{A}$ such that

$$\forall h \in [H] : \quad \mathbf{v}_h \leq \mathcal{T}_{\pi(\cdot, h)}(\mathbf{v}_{h+1}) \quad \text{and} \quad \mathbf{0} \leq \mathbf{v}_h^* - \mathbf{v}_h \leq (u/2) \cdot \mathbf{1}$$

with probability at least $1 - \delta$, provided appropriately chosen constants, c_1, c_2 and c_3 , in Algorithm 3. Moreover, the algorithm uses $O[u^{-2} \cdot H^3 |\mathcal{S}| |\mathcal{A}| \cdot \log(|\mathcal{S}| |\mathcal{A}| \delta^{-1} H u^{-1})]$ samples from the sampling oracle.

Proof of Proposition F.4.1. Recall that we are able to sample a state from each $\mathbf{P}_{s,a}$ with time $O(1)$. Let $R = \lceil c_1 H \ln[Hu^{-1}] \rceil$, $m_1 = c_2 H^3 u^{-2} \cdot \log(8|\mathcal{S}||\mathcal{A}|\delta^{-1})$ and $m_2 = c_3 H^2 \cdot \log[2R|\mathcal{S}||\mathcal{A}|\delta^{-1}]$ for some constants c_1, c_2 and c_3 required in Algorithm 1. In the following proof, we set $c_1 = 4$, $c_2 = 8192$, $c_3 = 128$. By Lemma 5.4, with probability at least $1 - 2\delta$ for each $1 \leq h \leq H$, we have $\mathbf{v}_h^{(0)} \leq \mathbf{v}_h \leq \mathcal{T}_{\pi(\cdot, h)} \mathbf{v}_h$, and $\mathbf{Q}_h \leq \mathbf{r} + \mathbf{P} \mathbf{v}_{h+1}$,

$$\mathbf{Q}_h^* - \mathbf{Q}_h \leq \mathbf{P}_h^{\pi^*} [\mathbf{Q}_{h+1}^* - \mathbf{Q}_{h+1}] + \boldsymbol{\xi}_h,$$

where

$$\boldsymbol{\xi}_h \leq H^{-1}u/8 \cdot \mathbf{1} + 2\sqrt{2\alpha_1 \boldsymbol{\sigma}_{\mathbf{v}_{h+1}^*}} + 2\sqrt{2\alpha_1}u \cdot \mathbf{1} + 16\alpha_1^{3/4} \|\mathbf{v}_{h+1}^{(0)}\|_\infty \cdot \mathbf{1} + (4/3) \cdot \alpha_1 \cdot \|\mathbf{v}_{h+1}^{(0)}\|_\infty \cdot \mathbf{1},$$

and $\alpha_1 = \log(8|\mathcal{S}||\mathcal{A}|\delta^{-1})/m_1$. Notice that $\mathbf{v}_H^{(0)} = \mathbf{v}_H^* = \mathbf{v}(\mathbf{r})$, thus the $\mathbf{v}_H - \mathbf{v}_H^* = \mathbf{0}$. Solving the recursion, we obtain

$$\mathbf{Q}_h^* - \mathbf{Q}_h \leq \sum_{h'=h}^{H-1} \left(\prod_{i=h+1}^{h'} \mathbf{P}_i^{\pi^*} \right) \boldsymbol{\xi}_{h'}.$$

The next step is the key to the improvement in our analysis. We further apply the bound in Lemma C.1, given by

$$\sum_{h'=h}^{H-1} \left(\prod_{i=h+1}^{h'} \mathbf{P}_i^{\pi^*} \right) \sqrt{\boldsymbol{\sigma}_{\mathbf{v}_{h'+1}^*}} \leq H^{3/2} \cdot \mathbf{1}.$$

With $\|\sum_{h'=h}^{H-1} \prod_{i=h+1}^{h'} \mathbf{P}_i^{\pi^*} \mathbf{1}\|_\infty \leq H - h + 1$ and $\|\mathbf{v}_h^{(0)}\|_\infty \leq H$, we have,

$$\begin{aligned} \sum_{h'=h}^{H-1} \left(\prod_{i=h+1}^{h'} \mathbf{P}_i^{\pi^*} \right) \boldsymbol{\xi}_{h'} &\leq \left[\frac{u}{8} + 4\sqrt{2\alpha_1 H^3} + 2H\sqrt{2\alpha_1}u + 16H^2\alpha_1^{3/4} + \frac{4\alpha_1 H^2}{3} \right] \mathbf{1} \\ &\leq \left[\frac{u}{8} + \frac{u}{16} + \frac{\sqrt{H^{-1}}u}{32} + 16 \left(\frac{H^{-3}u^2}{32 \cdot 256 \cdot (H)^{-8/3}} \right)^{3/4} + \frac{4H^{-1}u^2}{24 \cdot 256} \right] \cdot \mathbf{1} \\ &\leq \frac{u}{4} \cdot \mathbf{1}, \end{aligned}$$

provided

$$\alpha_1 = \frac{\log(8|\mathcal{S}||\mathcal{A}|\delta^{-1})}{m_1} = c_2^{-1} H^3 u^{-2} \leq \frac{H^{-3}u^2}{32 \cdot 256}.$$

Since $\mathbf{v}(\mathbf{Q}_h) \leq \mathbf{v}_h$, we have

$$\mathbf{v}_h^* - \mathbf{v}_h \leq \mathbf{v}^* - \mathbf{v}(\mathbf{Q}_h) \leq \sum_{h'=h}^{H-1} \left(\prod_{i=h+1}^{h'} \mathbf{P}_i^{\pi^*} \right) \boldsymbol{\xi}_{h'} \leq \frac{u}{2} \cdot \mathbf{1}.$$

This completes the proof of the correctness. It remains to bound the time complexity. The initialization stage costs $O(m_1)$ time per (s, a) per stage h . Each iteration costs $O(m_2)$ time per (s, a) . We thus have the total time complexity as

$$O(Hm_1 + Hm_2)|\mathcal{S}||\mathcal{A}| = O \left[H^4 \cdot |\mathcal{S}||\mathcal{A}| \cdot \log \frac{H|\mathcal{S}||\mathcal{A}|}{\delta \cdot u} \cdot \frac{1}{u^2} \right].$$

The total number of samples used is

$$O(m_1 + Hm_2)|\mathcal{S}||\mathcal{A}| = O\left[H^3 \cdot |\mathcal{S}||\mathcal{A}| \cdot \log \frac{H|\mathcal{S}||\mathcal{A}|}{\delta \cdot u} \cdot \frac{1}{u^2}\right].$$

This completes the proof. \square

We can then use our meta-algorithm and obtain the following theorem.

Theorem F.5. *Let $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathbf{P}, \mathbf{r}, H)$ be a H -MDP with a sampling oracle. Suppose we can sample a state from each probability vector $\mathbf{P}_{s,a}$ within time $O(1)$. Then there exists an algorithm that runs in time*

$$O\left[\frac{1}{\epsilon^2} \cdot H^4 |\mathcal{S}||\mathcal{A}| \cdot \log \frac{H|\mathcal{S}||\mathcal{A}|}{\delta \cdot \epsilon} \cdot \log \frac{H}{\epsilon}\right]$$

and obtains a policy π such that, with probability at least $1 - \delta$,

$$\forall h \in [H] : \mathbf{v}_h^* - \epsilon \mathbf{1} \leq \mathbf{v}_h^\pi \leq \mathbf{v}_h^*,$$

where \mathbf{v}_h^* is the optimal value of \mathcal{M} at stage h . Moreover, the number of samples used by the algorithm is

$$O\left[\frac{1}{\epsilon^2} \cdot H^3 |\mathcal{S}||\mathcal{A}| \cdot \log \frac{H|\mathcal{S}||\mathcal{A}|}{\delta \cdot \epsilon} \cdot \log \frac{H}{\epsilon}\right].$$

F.1 Sample Lower Bound On H -MDP

In this section we show that the sample complexity obtained by the algorithm in the last section is essentially tight. Our proof idea is simple, we will reduce the H -MDP problem to a discounted MDP problem. If there is an algorithm that solves an H -MDP to obtain an ϵ -optimal value, it also gives an value function to the discounted MDP. Therefore, the lower bound of solving H -MDP inherits from that of the discounted MDP. The formal guarantee is presented in the following theorem.

Theorem F.6. *Let \mathcal{S} and \mathcal{A} be finite sets of states and actions. Let $H > 0$ be a positive integer and $\epsilon \in (0, 1/2)$ be an error parameter. Let \mathcal{K} be an algorithm that, on input an H -MDP $\mathcal{M} \stackrel{\text{def}}{=} (\mathcal{S}, \mathcal{A}, \mathbf{P}, \mathbf{r})$ with a sampling oracle, outputs a value function \mathbf{v}_1 for the first stage, such that $\|\mathbf{v}_1 - \mathbf{v}_1^*\|_\infty \leq \epsilon$ with probability at least 0.9. Then \mathcal{K} calls the sampling oracle at least $\Omega(H^{-3}\epsilon^{-2}|\mathcal{S}||\mathcal{A}|/\log \epsilon^{-1})$ times on some input \mathbf{P} and $\mathbf{r} \in [0, 1]^{\mathcal{S}}$.*

Proof. Let $s_0 \in \mathcal{S}$ be a state. Denote $\mathcal{S}' = \mathcal{S} \setminus \{s_0\}$ be a subset of \mathcal{S} . Let $\gamma \in (0, 1)$ be such that $(1 - \gamma)^{-1} \log \epsilon^{-1} \leq H$. Suppose we have an DMDP $\mathcal{M}' = (\mathcal{S}', \mathcal{A}, \mathbf{P}', \gamma, \mathbf{r}')$ with a sampling oracle. Let $\mathbf{v}^{*'}$ be the optimal value function of \mathcal{M}' . Note that $\mathbf{v}^{*'} \in \mathbb{R}^{\mathcal{S}'}$. We will show, in the next paragraph, an H -MDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathbf{P}, H, \mathbf{r})$ with first stage value \mathbf{v}_1^* , such that $\|\mathbf{v}_1^*|_{\mathcal{S}'} - \mathbf{v}^{*'}\| \leq \epsilon$. Therefore, an ϵ -approximation of \mathbf{v}_1^* gives a 2ϵ -approximation to \mathbf{v}^* . We show that \mathcal{K} can be used to obtain an ϵ -approximate value \mathbf{v}_1 for \mathbf{v}_1^* of \mathcal{M} and thus \mathcal{K} inherits the lower bound for obtaining (2ϵ) -approximated value for γ -DMDPs.

For \mathcal{M} , in each state $s \in \mathcal{S}'$, for any action there is a $(1 - \gamma)$ probability transiting to s_0 and γ probability to do the original transitions in \mathcal{M}' ; for s_0 , no matter what action taken, it transits to itself with probability 1. Formally, for each state $s, s' \in \mathcal{S}', a \in \mathcal{A}$, $P(s'|s, a) = \gamma \cdot P'(\cdot|s, a)$ and $P(s_0|s, a) = (1 - \gamma)$; $P(s'|s_0, a) = 0$ and $P(s_0|s_0, a) = 1$. For \mathbf{r} , we set $\mathbf{r}(s_0, \cdot) = \mathbf{0}$ and

Algorithm 3 FiniteHorizonRandomQVI

- 1: **Input:** $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathbf{r}, \mathbf{P})$ with a sampling oracle, $\mathbf{v}_1^{(0)}, \mathbf{v}_2^{(0)}, \dots, \mathbf{v}_H^{(0)}, \pi^{(0)} : \mathcal{S} \times [H] \rightarrow \mathcal{A}, u, \delta \in (0, 1)$;
 - 2: $\backslash\backslash u$ is the initial error, $\pi^{(0)}$ is the input policy, and δ is the error probability
 - 3: **Output:** $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_H, \pi$
 - 4:
 - 5: **INITIALIZATION:**
 - 6: Let $m_1 \leftarrow c_1 H^3 u^{-2} \log(8|\mathcal{S}||\mathcal{A}|\delta^{-1})$ for constant c_1 ;
 - 7: Let $m_2 \leftarrow c_2 H^2 \log[2H|\mathcal{S}||\mathcal{A}|\delta^{-1}]$ for constant c_2 ;
 - 8: Let $\alpha_1 \leftarrow m_1^{-1} \log(8|\mathcal{S}||\mathcal{A}|\delta^{-1})$;
 - 9: For each $(s, a) \in \mathcal{S} \times \mathcal{A}$, sample independent samples $s_{s,a}^{(1)}, s_{s,a}^{(2)}, \dots, s_{s,a}^{(m_1)}$ from $\mathbf{P}_{s,a}$;
 - 10: Initialize $\mathbf{w}_h = \tilde{\mathbf{w}}_h = \hat{\boldsymbol{\sigma}}_h = \mathbf{Q}_h^{(0)} \leftarrow \mathbf{0}_{\mathcal{S} \times \mathcal{A}}$ for all $h \in [H]$, and $i \leftarrow 0$;
 - 11: Denote $\mathbf{v}_{H+1} \leftarrow \mathbf{0}$ and $\mathbf{Q}_{H+1} \leftarrow \mathbf{0}$
 - 12: **for** each $(s, a) \in \mathcal{S} \times \mathcal{A}, h \in [H]$ **do**
 - 13: $\backslash\backslash$ Compute empirical estimates of $\mathbf{P}_{s,a}^\top \mathbf{v}_h^{(0)}$ and $\boldsymbol{\sigma}_{\mathbf{v}_h^{(0)}}(s, a)$
 - 14: Let $\tilde{\mathbf{w}}_h(s, a) \leftarrow \frac{1}{m_1} \sum_{j=1}^{m_1} \mathbf{v}_h^{(0)}(s_{s,a}^{(j)})$
 - 15: Let $\hat{\boldsymbol{\sigma}}_h(s, a) \leftarrow \frac{1}{m_1} \sum_{j=1}^{m_1} (\mathbf{v}_h^{(0)}(s_{s,a}^{(j)}))^2 - \tilde{\mathbf{w}}_h^2(s, a)$
 - 16:
 - 17: $\backslash\backslash$ Shift the empirical estimate to have one-sided error
 - 18: $\mathbf{w}_h(s, a) \leftarrow \tilde{\mathbf{w}}_h(s, a) - \sqrt{2\alpha_1 \hat{\boldsymbol{\sigma}}_h(s, a)} - 4\alpha_1^{3/4} \|\mathbf{v}_h^{(0)}\|_\infty - (2/3)\alpha_1 \|\mathbf{v}_h^{(0)}\|_\infty$
 - 19: Let $\mathbf{v}_{H+1} \leftarrow \mathbf{0}$ and $\mathbf{Q}_{H+1} \leftarrow \mathbf{0}$.
 - 20:
 - 21: **REPEAT:** $\backslash\backslash$ successively improve
 - 22: **for** $h = H, H-1$ to 1 **do**
 - 23: $\backslash\backslash$ Compute $\mathbf{P}_{s,a}^\top [\mathbf{v}_h - \mathbf{v}_h^{(0)}]$ with one-sided error
 - 24: Let $\tilde{\mathbf{v}}_h \leftarrow \mathbf{v}_h \leftarrow \mathbf{v}(\mathbf{Q}_{h+1}), \tilde{\pi}(\cdot, h) \leftarrow \pi(\cdot, h) \leftarrow \pi(\mathbf{Q}_{h+1}), \mathbf{v}_h \leftarrow \tilde{\mathbf{v}}_h$;
 - 25: For each $s \in \mathcal{S}$, if $\tilde{\mathbf{v}}_h(s) \leq \mathbf{v}_h^{(0)}(s)$, then $\mathbf{v}_h(s) \leftarrow \mathbf{v}_h^{(0)}(s)$ and $\pi(s, h) \leftarrow \pi^{(0)}(s, h)$;
 - 26: For each $(s, a) \in \mathcal{S} \times \mathcal{A}$, sample independent samples $\tilde{s}_{s,a}^{(1)}, \tilde{s}_{s,a}^{(2)}, \dots, \tilde{s}_{s,a}^{(m_2)}$ from $\mathbf{P}_{s,a}$;
 - 27: Let $\mathbf{g}_h(s, a) \leftarrow m_2^{-1} \sum_{j=1}^{m_2} [\mathbf{v}_h(\tilde{s}_{s,a}^{(j)}) - \mathbf{v}_h^{(0)}(\tilde{s}_{s,a}^{(j)})] - H^{-1}u/8$;
 - 28:
 - 29: $\backslash\backslash$ Improve \mathbf{Q}_h :
 - 30: $\mathbf{Q}_h \leftarrow \mathbf{r} + \mathbf{w}_h + \mathbf{g}_h$;
 - 31: **return** $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_H, \pi$.
-

$\mathbf{r}(s, \cdot) = \mathbf{r}'(s, \cdot)$ for $s \in \mathcal{S}'$. It remains to show that $\|\mathbf{v}_1^*|_{\mathcal{S}'} - \mathbf{v}^*\|_\infty \leq \epsilon$. First we note that $\mathbf{v}(\mathbf{r}) = \mathbf{v}_H^* \leq \mathbf{v}^*$. Then, by monotonicity of the \mathcal{T} operator, we have, for all $h \in [H-1]$ and $s \in \mathcal{S}'$,

$$\mathbf{v}_h^*|_{\mathcal{S}'}(s) = \max_a [\mathbf{r}'(s, a) + \gamma \mathbf{P}_{s,a}'^\top \mathbf{v}_{h+1}^*] \leq \mathbf{v}^*.$$

In particular, $\mathbf{v}_1^*|_{\mathcal{S}'} \leq \mathbf{v}^*$. Since the optimal policy $\pi^{*'}$ of \mathcal{M}' can be used as a policy for the H -MDP as a non-optimal one, we have

$$\mathbf{v}^* - \epsilon \cdot \mathbf{1} \leq \left[1 + \gamma \mathbf{P}_{\pi^{*'}} + \gamma^2 \mathbf{P}_{\pi^{*'}}^2 + \dots + \gamma^H \cdot \mathbf{P}_{\pi^{*'}}^H \right] \mathbf{r}^{\pi^{*'}} \leq \mathbf{v}_1^*|_{\mathcal{S}'}.$$

This completes the proof. □

The above lower bound with our algorithm also implies a sample lower bound for an ϵ -policy.

Corollary F.7. *Let \mathcal{S} and \mathcal{A} be finite sets of states and actions. Let $H > 0$ be a positive integer and $\epsilon \in (0, 1/2)$ be an error parameter. Let \mathcal{K} be an algorithm that, on input an H -MDP $\mathcal{M} := (\mathcal{S}, \mathcal{A}, P, \mathbf{r})$ with a sampling oracle, outputs a policy $\pi : \mathcal{S} \times [H] \rightarrow \mathcal{A}$, such that $\forall h : \|\mathbf{v}_h^\pi - \mathbf{v}_h^*\|_\infty \leq \epsilon$ with probability at least 0.9. Then \mathcal{K} calls the sampling oracle at least $\Omega(H^{-3}\epsilon^{-2}|\mathcal{S}||\mathcal{A}|/\log \epsilon^{-1})$ times on the worst case input P and $\mathbf{r} \in [0, 1]^\mathcal{S}$.*