Adaptive Resolving Methods for Reinforcement Learning with Function Approximations

Jiashuo Jiang
Department of Industrial Engineering & Decision Analytics, HKUST
jsjiang@ust.hk

Yiming Zong
Department of Industrial Engineering & Decision Analytics, HKUST
yzongac@connect.ust.hk

Yinyu Ye
Department of Management Science & Engineering, Stanford University
yyye@stanford.edu

Abstract

Reinforcement learning (RL) problems are fundamental in online decision-making and have been instrumental for finding an optimal policy for Markov decision processes (MDPs). Function approximations are usually deployed to handle large or infinite state-action space. In our work, we consider the RL problems with function approximation and we develop a new algorithm to solve it efficiently. Our algorithm is based on the linear programming (LP) reformulation and it resolves the LP at each iteration improved with new data arrival. Such a resolving scheme enables our algorithm to achieve an instance-dependent sample complexity guarantee, more precisely, when we have N data, the output of our algorithm enjoys an instance-dependent $\tilde{O}(1/N)$ suboptimality gap. In comparison to the $O(1/\sqrt{N})$ worst-case guarantee established in the previous literature, our instance-dependent guarantee is tighter when the underlying instance is favorable, and the numerical experiments also reveal the efficient empirical performances of our algorithms.

1 Introduction

Reinforcement learning (RL) plays a crucial role in navigating uncertain environments, aiming to maximize rewards by iteratively interacting with and learning from the unknown surroundings. Markov Decision Processes (MDPs) serve as a widely adopted framework for modeling environmental dynamics. They have been pivotal in diverse fields like inventory management [3], video gaming [42], robotics [32], recommender systems [54], and more. One of the key to the successful deployment of RL is the use of *function approximator*, which can handle large or infinite state-action space of the underlying MDPs.

In this paper, we focus on developing reinforcement learning (RL) algorithms with function approximations that are provably efficient in handling large or infinite state-action spaces. One common approach is to frame the problem as linear programming (LP) (e.g. [14]), which is closely tied to approximate dynamic programming. Previous research (e.g. [46]) has established a worst-case $O(1/\sqrt{N})$ suboptimality gap, which translates into $O(1/\epsilon^2)$ sample complexity for this LP-based method, representing the best possible guarantee achievable for the most difficult problems in this problem class. However, these minimax bounds and worst-case analyses can often be overly cautious,

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: MLxOR: Mathematical Foundations and Operational Integration of Machine Learning for Uncertainty-Aware Decision-Making.

leading to a gap between theoretical assurances and practical performance in specific instances. We focus on a more tailored approach - one that guarantees outstanding performance for each unique problem and offers problem-specific assurances. Our study progresses by introducing RL algorithms that provide problem-specific guarantees.

2 Approximate Dynamic Programming and Linear Programming

We consider a MDP problem with the state space denoted by $\mathcal{S}$ and the action space denoted by $\mathcal{A}$. We denote by $\gamma \in (0, 1)$ a discount factor. We also denote by $P : \mathcal{S} \times \mathcal{A} \to \mathcal{D}(\mathcal{S})$ the probability transition kernel of the MDP, where $\mathcal{D}(\mathcal{S})$ denotes a probability measure over the state space $\mathcal{S}$. Then, $P(s'|s, a)$ denotes the probability of transiting from state $s \in \mathcal{S}$ to state $s' \in \mathcal{S}$ when the action $a \in \mathcal{A}$ is executed. The initial distribution over the states of the MDP is denoted by μ_1. There is also a cost function $c : \mathcal{S} \times \mathcal{A} \to \mathcal{D}[0, 1]$.

One can directly compute the optimal policy π^* by solving the dynamic programming according to the Bellman equation. However, when the state space $\mathcal{S}$ or the action space $\mathcal{A}$ is very large or infinite, solving the DP becomes computationally intractable. Various methods have been developed in the literature to tackle this issue (e.g. [47, 34, 57]), and we consider the function approximations. To be specific, we make the following assumption.

Assumption 2.1 (Linear Approximation)**.** The value function V^{π^*} can be well approximated in the following formulation:

$$V^{\pi^*}(s) \approx \sum_{i=1}^{d_1} \phi_i(s) \cdot w_i, \ \ \forall s \in \mathcal{S}. \tag{1}$$

where $\phi_1, \dots, \phi_{d_1}$ are some basis functions that are determined based on the particular problem instances, and $\boldsymbol{w} = (w_1, \dots, w_{d_1}) \in \mathbb{R}^{d_1}$ are the corresponding weights.

The approximate linear programming (ALP) has been developed in the previous literature (e.g. [48]) to find the optimal weights $\boldsymbol{w}$ that well approximates the optimal value function. The ALP is formulated as follows where the weights $\boldsymbol{w}$ are regarded as the decision variables.

$$V^{\text{ALP}} = \ \max \sum_{s\in\mathcal{S}} \mu(s) \cdot \sum_{i=1}^{d_1} \phi_i(s) w_i \tag{2a}$$

$$\text{s.t.} \sum_{i=1}^{d_1} \phi_i(s) w_i - \gamma \cdot \sum_{s'\in\mathcal{S}} P(s'|s,a) \cdot \sum_{i=1}^{d_1} \phi_i(s') w_i \leq c(s,a), \ \ \forall (s,a) \in \mathcal{S} \times \mathcal{A} \tag{2b}$$

$$\boldsymbol{w} \in \mathbb{R}^{d_1}. \tag{2c}$$

where μ is a *state-relevance weights* with positive elements. The ALP has been widely studied in the literature and numerous methods have been developed to solve it efficiently. In what follows, we focus on solving the ALP (2) in a data-driven way, where we aim to obtain a near-optimal solution with as few samples as possible. In Appendix B, we discuss the generative model assumption and the RLP formulation (13) for large problems.

3 Variables and Constraints Reduction

In this section, we describe a procedure that can further reduce the required number of constraints to a quantity d_2 that is guaranteed to be no larger than the number of basis functions d_1. For convenience, we denote by the ALP (or RLP for large problems) in the following standard matrix form. We set K to be the number of constraints in the LP, with $K = |\mathcal{S} \times \mathcal{A}|$ for ALP and $K = |\mathcal{K}|$ for RLP.

$$V^{\text{LP}} = \ \max \ \boldsymbol{r}^\top \boldsymbol{x} \ \ \text{s.t.} \ A\boldsymbol{x} + \boldsymbol{s} = \boldsymbol{c}, \ \ \boldsymbol{x} \in \mathbb{R}^{d_1}, \boldsymbol{s} \geq 0. \tag{3}$$

Here, we use $\boldsymbol{r} = \boldsymbol{\mu}^\top \Phi$ and the matrix $A \in \mathbb{R}^{d_1 \times K}$ denotes the constraint matrix. We set $\boldsymbol{s} \in \mathbb{R}^K$ to be the slackness variables. We now show that when solving (3), instead of focusing on all the constraints (the number of which may be large), we can focus on only a small set of constraints that is no larger than the number of decision variables $\boldsymbol{x}$.

3.1 Optimal Basis Identification

Following standard LP theory, when solving the LP (3), it is sufficient for us to focus only on the *corner points* of the feasible set to obtain an optimal solution. Such a solution is also called *basic solutions*. Therefore, we can simply focus on obtaining the optimal basic solution of LP (3). The optimal basic solution of LP (3) enjoys the following further characterizations. Note that in LP theory, the corner point can be represented by *LP basis*, which involves the set of basic variables that are allowed to be non-zero, and the set of active constraints that are binding under the corresponding basic solution. Then, we have the following result, where the formal proof is relegated to the appendix.

Lemma 3.1. *There exists an index set $\mathcal{I}^* \subset [d_1]$ and an index set $\mathcal{J}^* \subset \mathcal{K}$ such that $|\mathcal{I}^*| = |\mathcal{J}^*| = d_2$, for some integer $d_2 \leq d_1$. Also, for the given $\mathcal{I}^*$ and $\mathcal{J}^*$, there exists an optimal solution $\boldsymbol{x}^*$ to LP* (3) *that satisfies*

$$A_{\mathcal{J}^*,\mathcal{I}^*} \cdot \boldsymbol{x}^*_{\mathcal{I}^*} = \boldsymbol{c}_{\mathcal{J}^*} \tag{4}$$

and $\boldsymbol{x}^*_{\mathcal{I}^{*c}} = 0$ where $\mathcal{I}^{*c} = [d_1] \setminus \mathcal{I}^*$ denotes the complementary set of $\mathcal{I}^*$.

Therefore, characterizing the optimal corner point is equivalent to identifying the optimal index sets $\mathcal{I}^*$ and $\mathcal{J}^*$. We now describe how to identify one optimal basis of LP (3). Note that this procedure involves the exact formulation of A. However, since the transition kernel P is unknown, we can only use the historical dataset, denoted by $\mathcal{H}$, to construct an estimate of A, which we denote by $\hat{A}(\mathcal{H})$. We describe in the following paragraphs the exact procedure to construct estimate $\hat{A}(\mathcal{H})$. With $\hat{A}(\mathcal{H})$, we can consider the following LP, which serves as an estimate of LP (3).

$$\hat{V}^{\mathrm{LP}}(\mathcal{H}) = \ \max\ \boldsymbol{r}^\top \boldsymbol{x} \ \text{ s.t. } \hat{A}(\mathcal{H}) \cdot \boldsymbol{x} + \boldsymbol{s} = \boldsymbol{c}, \quad \boldsymbol{x} \in \mathbb{R}^{d_1}, \boldsymbol{s} \geq 0. \tag{5}$$

A key step in our approach is to identify one optimal basis of $\hat{V}^{\mathrm{LP}}(\mathcal{H})$ and then show that it is a good approximation of the optimal basis of V^{LP}.

Note that identifying an optimal basis is a classical topic in the study of linear programming and there has been multiple algorithms developed in the previous literature. One well-known algorithm is the simplex method, which identifies one optimal basic solution, as well as the optimal basis. Though the simplex method is developed more than half a century ago, it is still to date one of the most efficient methods for solving large-scale linear programming (LP) problems, and has been demonstrated to be computationally efficient for solving MDP problems with LP formulations. The simplex algorithm to identify an optimal basis of $\hat{V}^{\mathrm{LP}}(\mathcal{H})$ is formally presented in Appendix C.1. In Appendix C.2, we also develop a new algorithm that can identify the optimal basis of $\hat{V}^{\mathrm{LP}}(\mathcal{H})$, which enjoys a polynomial-time worst-case computational complexity. We leave the discussion of the high probability bound of optimality for the variables and constraint reduction in Appendix D.

4 Our Formal Algorithm

In the previous section, we described how to identify one optimal basis $\mathcal{I}^*$ and $\mathcal{J}^*$. We now describe how to approximate the optimal weights $\boldsymbol{x}^*$ that corresponds to the optimal basis $\mathcal{I}^*$ and $\mathcal{J}^*$. Note that the optimal solution $\boldsymbol{x}^*$ enjoys the following structure: $\boldsymbol{x}^*_{\mathcal{I}^{*c}} = 0$ and the basic elements $\boldsymbol{x}^*_{\mathcal{I}^*}$ can be given as the solution to

$$A_{\mathcal{J}^*,\mathcal{I}^*} \cdot \boldsymbol{x}_{\mathcal{I}^*} = \boldsymbol{c}_{\mathcal{J}^*}. \tag{6}$$

However, in practice, we do not know the optimal basis $\mathcal{I}^*$ and $\mathcal{J}^*$, as well as the matrix A, beforehand. Therefore, on one hand, we use the simplex method to learn the optimal basis $\mathcal{I}^*$ and $\mathcal{J}^*$. On the other hand, we construct estimates of the matrix A and approximate $\boldsymbol{x}^*$, using the historical dataset. We denote by C an upper bound on $2 \cdot \|\boldsymbol{x}^*\|$. Our formal algorithm is given in Algorithm 1.

Note that in our algorithm, for each iteration $n \in N$, we need to query the generative model $\mathcal{M}$ to obtain an unbiased estimator of the matrix A to represent how the constraints are satisfied under the current action $\boldsymbol{x}^n$. Such an estimator can be constructed in the following way. For any state-action pair $(s, a) \in \mathcal{J}^*$, we query the generative model $\mathcal{M}$ to obtain a state transition, where we denote the new state by s'. Then, the element of A^n, at the row $(s, a) \in \mathcal{J}^*$ and column $i \in [d_2]$ can be represented by $\gamma \cdot \phi_i(s') - \phi_i(s)$. It is clear to see that

$$\mathbb{E}_{s'}[A^n_{(s,a),i}] = \mathbb{E}_{s'}[\gamma \cdot \phi_i(s') - \phi_i(s)] = \gamma \cdot \sum_{s' \in \mathcal{S}} P(s'|s, a) \cdot \phi_i(s') - \phi_i(s) = A_{(s,a),i}.$$

Algorithm 1 The Algorithm for Optimal Weights

```
1: **Input:** the number of samples $N$ and basis $\mathcal{I}^*$ and $\mathcal{J}^*$.
2: Initialize $\mathcal{H}^1 = \emptyset$ and $\boldsymbol{c}^1 = N \cdot \boldsymbol{c}$.
3: **for** $n = 1, \dots, N$ **do**
4:     Construct estimates $\hat{A}(\mathcal{H}^n)$ using the dataset $\mathcal{H}^n$.
5:     Construct a solution $\tilde{\boldsymbol{x}}^n$ such that $\tilde{\boldsymbol{x}}^n_{\mathcal{I}^{*c}} = 0$ and $\tilde{\boldsymbol{x}}^n_{\mathcal{I}^*}$ is the solution to
           $$\hat{A}_{\mathcal{J}^*,\mathcal{I}^*}(\mathcal{H}^n) \cdot \tilde{\boldsymbol{x}}^n_{\mathcal{I}^*} = \frac{\boldsymbol{c}^n_{\mathcal{J}^*}}{N - n + 1}. \quad (7)$$
6:     Project $\tilde{\boldsymbol{x}}^n$ to the set $\{\boldsymbol{x} : \|\boldsymbol{x}\| \le C\}$ to obtain $\boldsymbol{x}^n$.
7:     For each $(s, a) \in \mathcal{J}^*$, query the generative model $\mathcal{M}$ to obtain the new state transition $s'(s, a)$.
8:     Update $\mathcal{H}^{n+1} = \mathcal{H}^n \cup \{s'(s, a), \forall (s, a)\}$.
9:     Construct a matrix $A^n \in \mathbb{R}^{d_2 \times d_2}$ with the element of $A^n$, at the row $(s, a) \in \mathcal{J}^*$ and column
       $i \in [d_2]$, is given by $\gamma \cdot \phi_i(s'(s, a)) - \phi_i(s)$.
10:    Do the update:
           $$\boldsymbol{c}^{n+1}_{\mathcal{J}^*} = \boldsymbol{c}^n_{\mathcal{J}^*} - A^n \cdot \boldsymbol{x}^n_{\mathcal{I}^*}. \quad (8)$$
11: **end for**
12: Define
           $$\bar{\boldsymbol{x}}^N = \frac{1}{N} \cdot \sum_{n=1}^{N} \boldsymbol{x}^n. \quad (9)$$
13: **Output:** $\bar{\boldsymbol{x}}^N$.
```

Therefore, we know that for each iteration n, A^n is an unbiased estimator of $A_{\mathcal{J}^*,\mathcal{I}^*}$ and the distribution of A^n is also independent of the dataset $\mathcal{H}^n$.

Another crucial element in Algorithm 1 (step 10) is that we adaptively update the value of $\boldsymbol{c}^n$ as in (8). We then use the updated $\boldsymbol{c}^n$ to obtain the value of $\tilde{\boldsymbol{x}}^n_{\mathcal{I}^*}$ as in (7). Such a resolving algorithmic design has been developed in [26] for constrained MDP under the tabular setting and we further develop here for RL under the linear approximation setting. The resolving procedure has a natural interpretation that $\boldsymbol{c}^n$ will concentrate around $\boldsymbol{c}$, with a gap bounded at the order of $1/(N - n + 1)$, which is the key to achieving an instance-dependent guarantee. To be specific, if for one binding constraint $j \in \mathcal{J}^*$, the constraint value under the action $\sum_{n'=1}^{n} \boldsymbol{x}^{n'}/n$ is below the target $\boldsymbol{c}_j$, we know that $\boldsymbol{c}^n_j/(N - n + 1)$ is greater than $\boldsymbol{c}_j$, which results in a greater constraint value for $\tilde{\boldsymbol{x}}^{n+1}$. In this way, any gap between the real-time constraint value and the target will be self-corrected in the next period as we adaptively obtain $\tilde{\boldsymbol{x}}^{n+1}$. Therefore, such an adaptive design can re-adjust the possible constraint violation by itself, which results in a regret bound of a lower order at $O(1/(N - n + 1))$.

Here we also present the final sample complexity bound of our algorithm and leave the detailed analysis to the Appendix E. Note that we have a bound on the constraint violation for each $(s, a) \in \mathcal{J}^*$. For the constraints not in the index set $\mathcal{J}^*$, we can also bound its violation using the non-singularity of the matrix $A_{\mathcal{J}^*,\mathcal{I}^*}$. This is another benefit of identifying the optimal basis $\mathcal{I}^*$ and $\mathcal{J}^*$.

Theorem 4.1. *With a sample complexity bound of*

$$O\left(K \cdot \frac{\log(K/\varepsilon)}{\Delta^2} + \frac{d_2^2(1 + \|A_{\mathcal{J}^*,\mathcal{I}^*}\|_\infty)}{\sigma^2} \cdot \frac{\log(1/\varepsilon)}{\varepsilon}\right),$$

where K refers to the number of constraints in V^{LP}, the parameters Δ defined in (18)*, σ given in* (24)*, we obtain a solution $\bar{\boldsymbol{x}}^N$ from Algorithm 1 (defined in* (9)*) such that*

$$V^{\mathrm{LP}} - \boldsymbol{r}^\top \bar{\boldsymbol{x}}^N \le \varepsilon \ \textit{and}\ A \cdot \bar{\boldsymbol{x}}^N - \boldsymbol{c} \le \varepsilon,\ \forall k \in [K].$$

5 Concluding Remarks

In this work, we develop a new algorithm for RL with function approximation. Our algorithm is LP-based and enjoys an instance-dependent theoretical guarantee. The experimental results (detailed in Appendix F) show the efficient performance of our algorithm. Our work considers a generative model and we leave the extension to offline and online setting as future work to explore.

References

[1] S. Agrawal, Z. Wang, and Y. Ye. A dynamic near-optimal algorithm for online linear programming. *Operations Research*, 62(4):876–890, 2014.

[2] A. Al-Marjani, A. Tirinzoni, and E. Kaufmann. Towards instance-optimality in online pac reinforcement learning. *arXiv preprint arXiv:2311.05638*, 2023.

[3] M. Alvo, D. Russo, and Y. Kanoria. Neural inventory control in networks via hindsight differentiable policy optimization. *arXiv preprint arXiv:2306.11246*, 2023.

[4] R. Ao, J. Jiang, and D. Simchi-Levi. Learning to price with resource constraints: From full information to machine-learned prices. *arXiv preprint arXiv:2501.14155*, 2025.

[5] A. Arlotto and X. Xie. Logarithmic regret in the dynamic and stochastic knapsack problem with equal rewards. *Stochastic Systems*, 10(2):170–191, 2020.

[6] K. P. Badrinath and D. Kalathil. Robust reinforcement learning using least squares policy iteration with provable performance guarantees. In *International Conference on Machine Learning*, pages 511–520. PMLR, 2021.

[7] S. Banerjee and D. Freund. Good prophets know when the end is near. *Management Science*, 2024.

[8] N. Buchbinder and J. Naor. Online primal-dual algorithms for covering and packing. *Mathematics of Operations Research*, 34(2):270–286, 2009.

[9] P. Bumpensanti and H. Wang. A re-solving heuristic with uniformly bounded loss for network revenue management. *Management Science*, 66(7):2993–3009, 2020.

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

[11] J. Chen and N. Jiang. Offline reinforcement learning under value and density-ratio realizability: the power of gaps. In *Uncertainty in Artificial Intelligence*, pages 378–388. PMLR, 2022.

[12] C.-A. Cheng, T. Xie, N. Jiang, and A. Agarwal. Adversarially trained actor critic for offline reinforcement learning. In *International Conference on Machine Learning*, pages 3852–3878. PMLR, 2022.

[13] C. Dann, L. Li, W. Wei, and E. Brunskill. Policy certificates: Towards accountable reinforcement learning. In *International Conference on Machine Learning*, pages 1507–1516. PMLR, 2019.

[14] D. P. De Farias and B. Van Roy. The linear programming approach to approximate dynamic programming. *Operations research*, 51(6):850–865, 2003.

[15] D. P. De Farias and B. Van Roy. On constraint sampling in the linear programming approach to approximate dynamic programming. *Mathematics of operations research*, 29(3):462–478, 2004.

[16] T. S. Ferguson. Who solved the secretary problem? *Statistical science*, 4(3):282–289, 1989.

[17] G. Gallego and G. Van Ryzin. A multiproduct dynamic pricing problem and its applications to network yield management. *Operations research*, 45(1):24–41, 1997.

[18] Z. D. Guo, S. Doroudi, and E. Brunskill. A pac rl algorithm for episodic pomdps. In *Artificial Intelligence and Statistics*, pages 510–518. PMLR, 2016.

[19] A. Gupta and M. Molinaro. How experts can solve lps online. In *European Symposium on Algorithms*, pages 517–529. Springer, 2014.

[20] J. He, D. Zhou, and Q. Gu. Logarithmic regret for reinforcement learning with linear function approximation. In *International Conference on Machine Learning*, pages 4171–4180. PMLR, 2021.

[21] J. He, H. Zhao, D. Zhou, and Q. Gu. Nearly minimax optimal reinforcement learning for linear markov decision processes. In *International Conference on Machine Learning*, pages 12790–12822. PMLR, 2023.

[22] N. J. Higham. *Accuracy and stability of numerical algorithms*. SIAM, 2002.

[23] J. Huang, H. Zhong, L. Wang, and L. Yang. Tackling heavy-tailed rewards in reinforcement learning with function approximation: Minimax optimal and instance-dependent regret bounds. *Advances in Neural Information Processing Systems*, 36, 2024.

[24] S. Jasin. Reoptimization and self-adjusting price control for network revenue management. *Operations Research*, 62(5):1168–1178, 2014.

[25] S. Jasin and S. Kumar. A re-solving heuristic with bounded revenue loss for network revenue management with customer choice. *Mathematics of Operations Research*, 37(2):313–345, 2012.

[26] J. Jiang and Y. Ye. Achieving $\tilde{O}(1/\epsilon)$ sample complexity for constrained markov decision process. *Advances in Neural Information Processing Systems*, 37:78679–78714, 2024.

[27] J. Jiang, W. Ma, and J. Zhang. Degeneracy is ok: Logarithmic regret for network revenue management with indiscrete distributions. *Operations Research*, 2025.

[28] C. Jin, Z. Allen-Zhu, S. Bubeck, and M. I. Jordan. Is q-learning provably efficient? *Advances in neural information processing systems*, 31, 2018.

[29] Y. Jin, Z. Yang, and Z. Wang. Is pessimism provably efficient for offline rl? In *International Conference on Machine Learning*, pages 5084–5096. PMLR, 2021.

[30] T. Kesselheim, A. Tönnis, K. Radke, and B. Vöcking. Primal beats dual on online packing lps in the random-order model. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, pages 303–312, 2014.

[31] Y. Kim, I. Yang, and K.-S. Jun. Improved regret analysis for variance-adaptive linear bandits and horizon-free linear mixture mdps. *arXiv preprint arXiv:2111.03289*, 2021.

[32] J. Kober, J. A. Bagnell, and J. Peters. Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research*, 32(11):1238–1274, 2013.

[33] G. Li, L. Shi, Y. Chen, Y. Chi, and Y. Wei. Settling the sample complexity of model-based offline reinforcement learning. *The Annals of Statistics*, 52(1):233–260, 2024.

[34] L. Li, T. J. Walsh, and M. L. Littman. Towards a unified theory of state abstraction for mdps. *AI&M*, 1(2):3, 2006.

[35] X. Li and Q. Sun. Variance-aware robust reinforcement learning with linear function approximation under heavy-tailed rewards. *arXiv preprint arXiv:2303.05606*, 2023.

[36] X. Li and Y. Ye. Online linear programming: Dual convergence, new algorithms, and regret bounds. *Operations Research*, 70(5):2948–2966, 2022.

[37] X. Li, C. Sun, and Y. Ye. The symmetry between arms and knapsacks: A primal-dual approach for bandits with knapsacks. In *International Conference on Machine Learning*, pages 6483–6492. PMLR, 2021.

[38] Y. Liu, A. Swaminathan, A. Agarwal, and E. Brunskill. Provably good batch off-policy reinforcement learning without great exploration. *Advances in neural information processing systems*, 33:1264–1274, 2020.

[39] W. Ma, Y. Cao, D. H. Tsang, and D. Xia. Optimal regularized online convex allocation by adaptive re-solving. *arXiv preprint arXiv:2209.00399*, 2022.

[40] A. Mehta, A. Saberi, U. Vazirani, and V. Vazirani. Adwords and generalized online matching. *Journal of the ACM (JACM)*, 54(5):22–es, 2007.

[41] P. Ménard, O. D. Domingues, A. Jonsson, E. Kaufmann, E. Leurent, and M. Valko. Fast active learning for pure exploration in reinforcement learning. *arXiv preprint arXiv:2007.13442*, 2020.

[42] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

[43] M. Molinaro and R. Ravi. The geometry of online packing linear programs. *Mathematics of Operations Research*, 39(1):46–59, 2014.

[44] R. Munos. Error bounds for approximate policy iteration. In *ICML*, volume 3, pages 560–567. Citeseer, 2003.

[45] R. Munos and C. Szepesvári. Finite-time bounds for fitted value iteration. *Journal of Machine Learning Research*, 9(5), 2008.

[46] A. E. Ozdaglar, S. Pattathil, J. Zhang, and K. Zhang. Revisiting the linear-programming framework for offline rl with general function approximation. In *International Conference on Machine Learning*, pages 26769–26791. PMLR, 2023.

[47] W. B. Powell. *Approximate Dynamic Programming: Solving the curses of dimensionality*, volume 703. John Wiley & Sons, 2007.

[48] M. L. Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 1994.

[49] P. Rashidinejad, B. Zhu, C. Ma, J. Jiao, and S. Russell. Bridging offline reinforcement learning and imitation learning: A tale of pessimism. *Advances in Neural Information Processing Systems*, 34:11702–11716, 2021.

[50] P. Rashidinejad, H. Zhu, K. Yang, S. Russell, and J. Jiao. Optimal conservative offline rl with general function approximation via augmented lagrangian. *arXiv preprint arXiv:2211.00716*, 2022.

[51] R. Reemtsen. Semi-infinite programming: discretization methods. 2001.

[52] B. Scherrer. Performance bounds for λ policy iteration and application to the game of tetris. *Journal of Machine Learning Research*, 14(4), 2013.

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

[54] G. Shani, D. Heckerman, R. I. Brafman, and C. Boutilier. An mdp-based recommender system. *Journal of Machine Learning Research*, 6(9), 2005.

[55] R. Shariff and C. Szepesvári. Efficient planning in large mdps with weak linear function approximation. *Advances in Neural Information Processing Systems*, 33:19163–19174, 2020.

[56] M. Simchowitz and K. G. Jamieson. Non-asymptotic gap-dependent regret bounds for tabular mdps. *Advances in Neural Information Processing Systems*, 32, 2019.

[57] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 2 edition, 2018. URL `http://incompleteideas.net/book/the-book-2nd.html`.

[58] A. Swietanowski. Simplex v. 2.17: an implementation of the simplex algorithm for large scale linear problems. user's guide. 1994.

[59] M. Uehara and W. Sun. Pessimistic model-based offline reinforcement learning under partial coverage. *arXiv preprint arXiv:2107.06226*, 2021.

[60] S. A. Vavasis and Y. Ye. Identifying an optimal basis in linear programming. *Annals of Operations Research*, 62(1):565–572, 1996.

[61] A. Vera and S. Banerjee. The bayesian prophet: A low-regret framework for online decision making. *Management Science*, 67(3):1368–1391, 2021.

[62] A. Wagenmaker and K. G. Jamieson. Instance-dependent near-optimal policy identification in linear mdps via online experiment design. *Advances in Neural Information Processing Systems*, 35:5968–5981, 2022.

[63] A. J. Wagenmaker, Y. Chen, M. Simchowitz, S. Du, and K. Jamieson. First-order regret in reinforcement learning with linear function approximation: A robust estimation approach. In *International Conference on Machine Learning*, pages 22384–22429. PMLR, 2022.

[64] A. J. Wagenmaker, M. Simchowitz, and K. Jamieson. Beyond no regret: Instance-dependent pac reinforcement learning. In *Conference on Learning Theory*, pages 358–418. PMLR, 2022.

[65] Y. Wei, J. Xu, and S. H. Yu. Constant regret primal-dual policy for multi-way dynamic matching. In *Abstract Proceedings of the 2023 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, pages 79–80, 2023.

[66] T. Xie and N. Jiang. Batch value-function approximation with only realizability. In *International Conference on Machine Learning*, pages 11404–11413. PMLR, 2021.

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

[68] Y. 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.

[69] A. Zanette, D. Brandfonbrener, E. Brunskill, M. Pirotta, and A. Lazaric. Frequentist regret bounds for randomized least-squares value iteration. In *International Conference on Artificial Intelligence and Statistics*, pages 1954–1964. PMLR, 2020.

[70] A. Zanette, M. J. Wainwright, and E. Brunskill. Provable benefits of actor-critic methods for offline reinforcement learning. *Advances in neural information processing systems*, 34: 13626–13640, 2021.

[71] W. Zhan, B. Huang, A. Huang, N. Jiang, and J. Lee. Offline reinforcement learning with realizability and single-policy concentrability. In *Conference on Learning Theory*, pages 2730–2775. PMLR, 2022.

[72] Z. Zhang, J. Yang, X. Ji, and S. S. Du. Improved variance-aware confidence sets for linear bandits and linear mixture mdp. *Advances in Neural Information Processing Systems*, 34: 4342–4355, 2021.

[73] D. Zhou and Q. Gu. Computationally efficient horizon-free reinforcement learning for linear mixture mdps. *Advances in neural information processing systems*, 35:36337–36349, 2022.

[74] D. Zhou, Q. Gu, and C. Szepesvari. Nearly minimax optimal reinforcement learning for linear mixture markov decision processes. In *Conference on Learning Theory*, pages 4532–4576. PMLR, 2021.

[75] H. Zhu, P. Rashidinejad, and J. Jiao. Importance weighted actor-critic for optimal conservative offline reinforcement learning. *Advances in Neural Information Processing Systems*, 36, 2024.

A More Detailed Literature Review

We now provide a more detailed discussion over the related work.

RL with function approximation. There are two main research lines in RL with function approximation. In the first research line, a lot of works focused on the theoretical basis and guarantee of offline RL. Early works mainly discussed algorithms' theoretical performance guarantee [45, 52, 53], but neglected the analysis of their underlying assumptions. Chen and Jiang [10] concluded and proposed two fundamental requirements of offline RL: Concentrability coefficient (also called 'full-data coverage' in some literature) and Bellman completeness. Later on, many works tried to develop offline RL algorithms under weaker assumptions. Most of works focused on weakening the 'Concentrability coefficient' assumption and utilized the pessimism principle to develop algorithms for tabular MDP settings [49, 33] and linear MDP settings [29, 59] under the assumption of partial data coverage. Besides, different from using the pessimism principle to discover good policy, Xie et al. [67] implemented pessimism with Bellman consistence, while Zanette et al. [70] incorporated it to the offline actor-critic algorithm. Other recent works under partial data coverage assumption are Shariff and Szepesvári [55], Zhan et al. [71], Rashidinejad et al. [50], Ozdaglar et al. [46] with general function approximation and Zhu et al. [75], Cheng et al. [12] with actor-critic algorithm.

The common assumption on function class is Bellman completeness [45, 10, 38, 67, 70]. Xie and Jiang [66] successfully weakened this assumption to only realizability, but used a stronger assumption than 'concentrability coefficient' [44]. Recently, many works have made progress by utilizing function approximation for density ratio instead of value function [11, 71, 46].

In the second, a line of works proposed more practical and provable efficient algorithms. For linear MDPs, **?**] proposed the LSVI-UCB with $\tilde{O}(\sqrt{d^3H^3T})$ regret bound and He et al. [21] further achieved the minimax optimality with $\tilde{O}(d\sqrt{H^3T})$ regret bound, which was also the lower bound Zhou et al. [74] provided. Zanette et al. [69] considered the undiscounted finite-horizon MDP and proposed a Thompson sampling-based algorithm, which achieves an upper bound $\tilde{O}(d^2H^2\sqrt{T})$. All of these algorithms are based on the LSVI, while Badrinath and Kalathil [6] proposed an algorithm based on the LSPI. There are also many works on linear mixture MDPs [74, 72, 73]. From another perspective, we consider using the LP framework for general MDP to build a provable and efficient algorithm compared to Ozdaglar et al. [46].

Instance dependent bound for RL. There are emerging works of instance dependent bound for RL under different conditions. Many recent work focused on the linear MDP and linear mixure MDP settings [63, 20, 73, 72, 31]. He et al. [20] shown that standard optimistic algorithms can achieve $O(\frac{d^3H^5}{\Delta_{min}\cdot log(T)})$ in the setting of linear MDP and $O(\frac{d^2H^5}{\Delta_{min}\cdot log^3(T)})$ in the setting of linear mixure MDP, and proved an $\Omega(\frac{dH}{\Delta_{min}})$ lower bound in both settings, where Δ_{min} is the minimum value-function gap. Zhou and Gu [73] further proposed the first computationally efficient horizon-free algorithm and achieved the optimal $\tilde{O}(d\sqrt{K}+d^2)$ regret. Li and Sun [35] and Huang et al. [23] both discussed heavy-tail rewards and achieved a variance-aware regret bound and the first computationally efficient instance-dependent K-episode regret bound separately. There is a lot of work on PAC RL [18, 28, 13, 41], but very little about instant-dependent bounds. Wagenmaker et al. [64] proposed an algorithm in tabular RL, whose instance-dependent sample complexity attains significant improvements over worst-case bounds. Al-Marjani et al. [2] proposed the first instance-dependent lower bound on the sample complexity and the PEDEL algorithm [62] is quite close to this lower bound. Our algorithm differs from above literatures by developing new primal-dual algorithm and achieving new instance-dependent sample complexity bound.

Near-optimal algorithms for online resource allocation. The online resource allocation problem has been extensively studied and encompasses a wide range of applications, each characterized by different formulations of the underlying LP. Examples include the secretary problem [16], online knapsack problem [5], network revenue management [17], network routing problem [8], and matching problem [40], among others [43, 1, 19]. Research on the online LP problem typically considers two main models: (i) the stochastic input model, where each column of the constraint matrix and the corresponding objective coefficient are independently drawn from an unknown distribution $\mathcal{P}$, and (ii) the random permutation model, where inputs arrive in a uniformly random order [43, 1, 30, 19]. Under an additional non-degeneracy assumption, logarithmic regret bounds have been established for the quantity-based network revenue management problem [25, 24], the general online LP problem

[36], and more broadly for the convex allocation problem [39]. More recently, this non-degeneracy assumption has been relaxed in several works [9, 61, 27, 65, 4, 7], leading to improved theoretical guarantees under broader settings.

B Preliminaries

We consider a MDP problem with the state space denoted by $\mathcal{S}$ and the action space denoted by $\mathcal{A}$. We denote by $\gamma \in (0, 1)$ a discount factor. We also denote by $P : \mathcal{S} \times \mathcal{A} \to \mathcal{D}(\mathcal{S})$ the probability transition kernel of the MDP, where $\mathcal{D}(\mathcal{S})$ denotes a probability measure over the state space $\mathcal{S}$. Then, $P(s'|s, a)$ denotes the probability of transiting from state $s \in \mathcal{S}$ to state $s' \in \mathcal{S}$ when the action $a \in \mathcal{A}$ is executed. The initial distribution over the states of the MDP is denoted by μ_1.

There is a cost function $c : \mathcal{S} \times \mathcal{A} \to \mathcal{D}[0, 1]$. We focus on the Markovian policy, which takes the action only based on the current state of the MDP. To be specific, any Markovian policy π can be denoted as a function $\pi : \mathcal{S} \to \mathcal{A}$. For any Markovian policy π, we denote by $V^{\pi}(\mu_1)$ the infinite horizon discounted cost of the policy π, with the formulation of $V^{\pi}(\mu_1)$ given below:

$$V^{\pi}(\mu_1) = \mathbb{E}\left[\sum_{t=1}^{\infty} \gamma^{t-1} \cdot c(s_t, a_t) \mid \mu_1\right], \tag{10}$$

where (s_t, a_t) is generated according to the policy π and the transition kernel P with the initial state distribution μ_1. To solve the MDP problem, we aim to find an optimal Markovian policy, denoted by π^*, that minimizes the cost in (10). Importantly, we assume that the transition kernel P is *unknown* to the decision maker. We obtain samples to learn the transition kernel. The sampling procedure can be described as follows.

Assumption B.1 (Generative Model)**.** For each state and action pair (s, a), we can query the model $\mathcal{M}$ to obtain an observation of the new state $s' \in \mathcal{S}$, where the transition from s to s' follows the probability kernel $P(s'|s, a)$ independently.

Querying the generative model $\mathcal{M}$ can be costly, and it is desirable to approximate the optimal policy π^* well with as few samples as possible. Therefore, we measure the performance of a policy π by the *sample complexity* bound. That is, for any ε, we compute a bound on the number of samples that we need to construct a policy π such that

$$V^{\pi}(\mu_1) - V^{\pi^*}(\mu_1) \leq \varepsilon. \tag{11}$$

The Bellman optimality equation can be written as

$$V^{\pi^*}(s) = \min_{a \in \mathcal{A}} c(s, a) + \gamma \cdot \sum_{s' \in \mathcal{S}} P(s'|s, a) \cdot V^{\pi^*}(s'). \tag{12}$$

Note that (12) implies that the optimal policy π^* is the greedy policy with respect to the optimal value function V^{π^*} through the Bellman equation (12). Thus, in order to approximate the optimal policy π^*, it is sufficient to approximate the value function V^{π^*}.

B.1 Reduced LP for Large or Infinite State Space

When the underlying state-action space $\mathcal{S} \times \mathcal{A}$ is large or infinite, the ALP (2) will have a large number of or infinite constraints and thus intractable to solve. A common approach to deal with this issue in the literature is through constraint sampling. To be specific, following [51, 15], we can sample a finite subset $\mathcal{K} \subset \mathcal{S} \times \mathcal{A}$, and we consider the reduced LP (RLP) given as follows,

$$V^{\mathrm{RLP}} = \max \sum_{s \in \mathcal{S}} \mu(s) \cdot \sum_{i=1}^{d_1} \phi_i(s) w_i \tag{13a}$$

$$\text{s.t.} \sum_{i=1}^{d_1} \phi_i(s) w_i - \gamma \cdot \sum_{s' \in \mathcal{S}} P(s'|s, a) \cdot \sum_{i=1}^{d_1} \phi_i(s') w_i \leq c(s, a), \ \forall (s, a) \in \mathcal{K} \tag{13b}$$

$$\boldsymbol{w} \in \mathbb{R}^{d_1}. \tag{13c}$$

We have the following result regarding the size of V^{RLP}.

Theorem B.2 (Theorem 3.1 of [15])**.** *Let the elements in the set $\mathcal{K}$ be sampled independently from $\mathcal{S} \times \mathcal{A}$. Then, for any $\epsilon, \delta > 0$, when $|\mathcal{K}| = O\left(\frac{\log(1/\epsilon)}{\epsilon} \cdot \log(1/\delta)\right)$, it holds that $P\left(|V^{\mathrm{ALP}} - V^{\mathrm{RLP}}| \le \varepsilon\right) \ge 1 - \delta$.*

C Two algorithms for Basis Identification

C.1 The Simplex Method

In this section, we formally present the simplex method in Algorithm 2.

Algorithm 2 Simplex algorithm for optimal basis identification

```
Input: the historical sample set $\mathcal{H}$.
Construct the $\hat{V}^{\mathrm{LP}}(\mathcal{H})$ with the historical sample set $\mathcal{H}$ as in (5).
Initialize Basis:
        • Initial basis: $B \leftarrow$ indices of slack variables $\mathbf{s}$.
        • Non-basis: $N \leftarrow$ indices of original variables $\mathbf{x}$.
        • Initial Basic Feasible Solutions: $\mathbf{s} = \mathbf{c},\ \mathbf{x} = \mathbf{0}$.
Convert to canonical form: Express $\mathbf{s}$ as identity matrix columns.
while there exists $\sigma_j = r_j - \mathbf{r}_B^\top \hat{A}_{:,j}(\mathcal{H}) > 0$ for $j \in N$ do
    Select $x_k \in N$ with $\sigma_k = \max\{\sigma_j > 0\}$.
    if $\hat{A}_{:,k}(\mathcal{H}) \le \mathbf{0}$ then
        Problem unbounded. Terminate.
    else
        Compute $\theta_i = \frac{c_i}{\hat{A}_{i,k}(\mathcal{H})}$ for $\hat{A}_{i,k}(\mathcal{H}) > 0$.
        Select leaving variable $s_r \in B$ with $\theta_r = \min\{\theta_i\}$ .
        Update basis $B \leftarrow (B \setminus \{r\}) \cup \{k\}$ and non-basis $N \leftarrow (N \setminus \{k\}) \cup \{r\}$.
        Perform Gaussian elimination to make $\hat{A}_{:,k}(\mathcal{H})$ an identity column.
        Recompute $\mathbf{x}_B = \mathbf{c} - \hat{A}_{:,N}(\mathcal{H})\mathbf{x}_N$ and $\sigma_j$.
    end if
end while
Set $\mathcal{I}^* = B \cap \{$indices of original variables $\mathbf{x}\}$ and $\mathcal{J}^* = B \cap \{$indices of slack variables $\mathbf{s}\}$.
Output: the sets of indices $\mathcal{I}^*$ and $\mathcal{J}^*$.
```

Note that the computation complexity of the simplex method is equivalent to solving the LP $\hat{V}^{\mathrm{LP}}(\mathcal{H})$ by one time using the simplex method. The simplex method is well-known to be practically efficient, especially for large-scale problems (see an early report [58] on simplex solver) and has been demonstrated to enjoy a polynomial-time average-case complexity. Moreover, the simplex method has been shown to be particularly efficient when solving the LPs that results from MDP problems (e.g. [68]). Besides the simplex method, there has been other algorithms developed in the literature which can be applied here to identify the optimal basis, see for example [60].

C.2 Additional algorithm for basis identification

Here we introduce an additional method to identify the basis of LP variables and constraints, which is also based on the Lemma 3.1. This algorithm solves the LPs by $K + d$ times and thus enjoys a polynomial-time worst-case computational complexity.

Note that if a variable is not a basic variable, we can restrict its value to 0 without changing the LP values. To detect whether we can restrict one variable to be 0 without changing the LP value, we can add the constraint $x_i = 0$ to LP (3) and compare its value to the original LP. If the objective value has not changed, then we know that the i-th variable is not a basic variable and does not belong to the basis $\mathcal{I}^*$. We repeat the above procedure for each variable i. Note that during the repeating procedure, if we can restrict one variable $x_i = 0$ without changing the LP value, we will remain this restriction when we test the remaining variables. In this way, we identify one optimal basis from possibly many. To this end, for an index set $\mathcal{I}$, we define the following LP with the variables not in $\mathcal{I}$ restricted to be

0, as well as its estimate,

$$\begin{array}{rlrl} V_{\mathcal{I}}^{\mathrm{LP}} = \max & \boldsymbol{r}^\top \boldsymbol{x} & \hat{V}_{\mathcal{I}}^{\mathrm{LP}}(\mathcal{H}) = \quad \max & \boldsymbol{r}^\top \boldsymbol{x} \\ \text{s.t.} & A\boldsymbol{x} \le \boldsymbol{c} & \text{s.t.} & \hat{A}(\mathcal{H})\boldsymbol{x} \le \boldsymbol{c} \\ & \boldsymbol{x}_{\mathcal{I}^c} = 0 & & \boldsymbol{x}_{\mathcal{I}^c} = 0 \\ & \boldsymbol{x} \ge 0, & & \boldsymbol{x} \ge 0. \end{array} \tag{14}$$

Algorithm 3 Additional algorithm for optimal basis identification

```
Input: the historical sample set $\mathcal{H}$ with $N$ transition data for each $(s,a)$.
Compute the value of $\hat{V}^{\mathrm{LP}}(\mathcal{H})$ as in (14).
Initialize $\mathcal{I} = [d_1]$ to be the whole index set that contains every variable of LP (3) and $\mathcal{J} = \mathcal{K}$.
for $i \in \mathcal{I}$ do
    Let $\mathcal{I}' = \mathcal{I} \backslash \{i\}$.
    Compute the value of $\hat{V}_{\mathcal{I}'}^{\mathrm{LP}}(\mathcal{H})$ as in (14).
    If $|\hat{V}^{\mathrm{LP}}(\mathcal{H}) - \hat{V}_{\mathcal{I}'}^{\mathrm{LP}}(\mathcal{H})| \le \sqrt{\mathrm{Rad}(N,\varepsilon)}$ with $\mathrm{Rad}(N,\varepsilon)$ given in (19), then we set $\mathcal{I} = \mathcal{I}'$.
end for
for $(s,a) \in \mathcal{J}$ do
    Let $\mathcal{J}' = \mathcal{J} \backslash \{(s,a)\}$.
    Compute the value of $\hat{D}_{\mathcal{I},\mathcal{J}'}^{\mathrm{LP}}(\mathcal{H})$ as in (15).
    If $|\hat{V}^{\mathrm{LP}}(\mathcal{H}) - \hat{D}_{\mathcal{I},\mathcal{J}'}^{\mathrm{LP}}(\mathcal{H})| \le \sqrt{\mathrm{Rad}(N,\varepsilon)}$, then we set $\mathcal{J} = \mathcal{J}'$.
end for
Output: the sets of indices $\mathcal{I}$ and $\mathcal{J}$.
```

We also need to detect whether a constraint is binding under the optimal basic solution corresponding to the basis $\mathcal{I}^*$. In order to tell whether a constraint is redundant and can be removed (not in the index set $\mathcal{J}^*$), we can consider the dual of LP (3), with the additional constraints $\boldsymbol{x}_{\mathcal{I}^*} = 0$. If a dual variable can be restricted to 0 without influencing the LP value, we know that the corresponding constraint is redundant. Denote by $\mathcal{J}$ the index set of the dual variables that could be restricted to 0. The dual LP with the restriction set $\mathcal{J}$ can be written as follows, as well as its empirical estimation. Denote by $\mathcal{J}$ the index set of the constraints that could be binding. We consider the following LP with the constraints not in $\mathcal{J}$ removed, as well as its estimate,

$$\begin{array}{rlrl} D_{\mathcal{I},\mathcal{J}}^{\mathrm{LP}} = \min & \boldsymbol{c}^\top \boldsymbol{y} & \hat{D}_{\mathcal{I},\mathcal{J}}^{\mathrm{LP}}(\mathcal{H}) = \min & \boldsymbol{c}^\top \boldsymbol{y} \\ \text{s.t.} & A_{:,\mathcal{I}}^\top \boldsymbol{y} = \boldsymbol{r}_{\mathcal{I}} & \text{s.t.} & \hat{A}_{:,\mathcal{I}}^\top(\mathcal{H}) \boldsymbol{y} = \boldsymbol{r}_{\mathcal{I}} \\ & \boldsymbol{y}_{\mathcal{J}^c} = 0 & & \boldsymbol{y}_{\mathcal{J}^c} = 0 \\ & \boldsymbol{y} \ge 0, & & \boldsymbol{y} \ge 0. \end{array} \tag{15}$$

As we can see, Algorithm 3 only requires us to compute the LP values by a finite number of time. More specifically, we need to compute the LP values for $K + d_1$ number of times. The size of the LP is also polynomial in the dimension of the problem (the size is $K \times d_1$). Therefore, we know that Algorithm 3 can be conducted in polynomial time.

D High Probability Bound of Optimality for Variables and Constraints Reduction

We know that Algorithm 2 identifies an optimal basis of $\hat{V}^{\mathrm{LP}}(\mathcal{H})$. In this section, we further show that the output of Algorithm 2, denoted by $\mathcal{I}^*$ and $\mathcal{J}^*$, forms an optimal basis to V^{LP} with a high probability, where the randomness comes from the randomness of the dataset $\mathcal{H}$ that we use to construct estimate of A.

There are two key quantities that we need to specify in order to derive the high probability bound. For an arbitrary basis $\mathcal{I}$ and $\mathcal{J}$ to V^{LP}, satisfying $|\mathcal{I}| = |\mathcal{J}|$, a solution $\boldsymbol{x}(\mathcal{I},\mathcal{J})$ satisfying the characterizations in Lemma 3.1 is called a *basic solution* if $A_{\mathcal{J},\mathcal{I}}$ is a non-singular sub-matrix such that the solution to linear equations (4) is uniquely determined. We now denote by $\mathcal{F}_1$ the collection of all basis $(\mathcal{I}, \mathcal{J})$ such that $\boldsymbol{x}(\mathcal{I}, \mathcal{J})$ is a basic solution. Note that not all basic solutions will be feasible. Therefore, we define δ_1 as the *feasibility gap*, specified below.

$$\delta_1 = \min_{(\mathcal{I},\mathcal{J})\in\mathcal{F}_1} \left\{ \max_{k\in[K]} \left\{ [A_{k,:} \cdot \boldsymbol{x}(\mathcal{I}, \mathcal{J}) - c_k]^+ \right\} : \max_{k\in[K]} \left\{ [A_{k,:} \cdot \boldsymbol{x}(\mathcal{I}, \mathcal{J}) - c_k]^+ \right\} > 0 \right\} \tag{16}$$

Note that if a basic solution $\boldsymbol{x}(\mathcal{I}, \mathcal{J})$ is feasible, then $\max_{k\in[K]} \left\{ [A_{k,:} \cdot \boldsymbol{x}(\mathcal{I}, \mathcal{J}) - c_k]^+ \right\} = 0$. However, if $\boldsymbol{x}(\mathcal{I}, \mathcal{J})$ is infeasible, then there must exists a $k \in [K]$ such that $[A_{k,:} \cdot \boldsymbol{x}(\mathcal{I}, \mathcal{J}) - c_k]^+ > 0$ becomes an infeasibility gap over the constraints. The parameter δ_1 specifies the minimum of such a gap. Since the number of basis is always finite, we know that it must hold $\delta_1 > 0$ (the situation where all basic solutions are feasible is discussed later).

We also need a parameter that characterizes the suboptimality gap of feasible basic solutions. Denote by $\mathcal{F}_2$ the collection of all feasible basis such that $\boldsymbol{x}(\mathcal{I}, \mathcal{J})$ is feasible to V^{LP} for all $(\mathcal{I}, \mathcal{J}) \in \mathcal{F}_2$. Then, we define

$$\delta_2 = \min_{(\mathcal{I},\mathcal{J})\in\mathcal{F}_2} \left\{ V^{\mathrm{LP}} - \boldsymbol{r}^\top \boldsymbol{x}(\mathcal{I}, \mathcal{J}) : V^{\mathrm{LP}} - \boldsymbol{r}^\top \boldsymbol{x}(\mathcal{I}, \mathcal{J}) > 0 \right\}. \tag{17}$$

Briefly speaking, δ_2 specifies the minimum suboptimality gap between the optimal solution and the best sub-optimal basic feasible solution of V^{LP}. We now define

$$\Delta = \min\{\delta_1, \delta_2\}. \tag{18}$$

Note that for the situation when all basic solutions are feasible, we define $\Delta = \delta_2$. And when all feasible basic solutions are optimal, we define $\Delta = \delta_1$. It won't happen when all basic solutions are feasible and optimal, otherwise, the LP would be trivial to solve. In general, the parameter Δ specifies the distance between an optimal basis solution and other non-optimal or non-feasible basic solutions.

One crucial part of our analysis is to show that when the dataset $\mathcal{H}$ is large enough such that the estimation gap is smaller than $\Delta/2$, the output of Algorithm 2 is an optimal basis to V^{LP}. To bound the estimation gap, we define the following quantity:

$$\mathrm{Rad}(N, \varepsilon) = \sqrt{\frac{\log(2/\varepsilon)}{2N}}. \tag{19}$$

Suppose that the dataset $\mathcal{H}$ contains N transition data of each state-action pair $(s, a) \in \mathcal{K}$. Denote by $\{s_1, \ldots, s_N\}$ the state transition. Then, the element of $\hat{A}(\mathcal{H})$ at the row (s, a) and column i can be represented by $\frac{\gamma}{N} \cdot \sum_{n=1}^{N} \phi_i(s_n) - \phi_i(s)$. Following the standard Hoeffding's inequality, we know that the gap between $\frac{\gamma}{N} \cdot \sum_{n=1}^{N} \phi_i(s_n) - \phi_i(s)$ and $A_{(s,a),i}$ is upper bounded by $\mathrm{Rad}(N, \varepsilon)$ with probability at least $1 - \varepsilon$. We now present the theorem showing that Algorithm 2 indeed helps us identify one optimal basis with a high probability.

Theorem D.1. *For any $\varepsilon > 0$, as long as $N \geq N_0$ with N_0 satisfying the condition*

$$\mathrm{Rad}(N_0, \varepsilon/K) \leq O(\Delta), \tag{20}$$

the outputs $\mathcal{I}$ and $\mathcal{J}$ of Algorithm 2 satisfy the conditions described in Lemma 3.1 for the true LP V^{LP} in (3) *with probability at least* $1 - \varepsilon$.

The formal proof of Theorem D.1 has been relegated to the appendix. Note that Theorem D.1 shows that in order to identify the optimal basis with a probability at least $1 - \epsilon$, the total number of samples we need can be bounded as

$$O\left(K \cdot \frac{\log(K/\epsilon)}{\Delta^2}\right),$$

where K refers to the number of constraints in V^{LP}. Since we have shown that the basis identified by Algorithm 2 is an optimal basis to V^{LP} with a high probability, from now on, we denote by $\mathcal{I}^*$ and $\mathcal{J}^*$ the output of Algorithm 2.

E Instance-dependent Sample Complexity

In this section, we analyze the sample complexity of our Algorithm 1. We aim to solve the LP (3), with the constraint matrix A unknown and need to be estimated from the data. Our Algorithm 1 is developed to solve the LP near-optimally in a data-driven way. Note that after Algorithm 2, we identify the optimal basis $\mathcal{I}^*$ and $\mathcal{J}^*$. Our next lemma shows that in order to bound the gap of solving the LP (3), it is sufficient to analyze the term $\boldsymbol{c}_{\mathcal{J}^*}^N$, which is defined in (8).

Lemma E.1. *Denote by $\mathcal{I}^*$ and $\mathcal{J}^*$ the optimal basis identified by Algorithm 2. We also denote by $\boldsymbol{x}^*$ the corresponding optimal solution and $\boldsymbol{y}^*$ the corresponding optimal dual solution. Then, it holds that*

$$N \cdot V^{\mathrm{LP}} - \sum_{n=1}^{N} \boldsymbol{r}^\top \mathbb{E}[\boldsymbol{x}^n] \leq \sum_{(s,a)\in\mathcal{J}^*} y^*_{(s,a)} \cdot \mathbb{E}[\boldsymbol{c}^N_{(s,a)}]. \tag{21}$$

Therefore, it suffices to analyze how $\boldsymbol{c}^n_{\mathcal{J}^*}$ behave. We now define

$$\tilde{c}_{(s,a)}(n) = \frac{c^n_{(s,a)}}{N-n}, \ \forall (s,a) \in \mathcal{J}^*. \tag{22}$$

The key is to show that the stochastic process $\tilde{c}_{(s,a)}(n)$ possesses some concentration properties such that they will stay within a small neighborhood of their initial value $c(s,a)$ for a sufficiently long time. We denote by τ the time that one of $\tilde{c}_{(s,a)}(n)$ for each $(s,a) \in \mathcal{J}^*$ escape this neighborhood. Then, both the gap over the objective value and the gap over the constraint satisfaction can be upper bounded by $\mathbb{E}[N-\tau]$. From the update rule (8), we know that

$$\tilde{c}_{(s,a)}(n+1) = \tilde{c}_{(s,a)}(n) - \frac{A^n_{(s,a),:} \cdot \boldsymbol{x}^n_{\mathcal{I}^*} - \tilde{c}_{(s,a)}(n)}{N-n}, \ \forall (s,a) \in \mathcal{J}^*. \tag{23}$$

Ideally, $\tilde{c}_{(s,a)}(n+1)$ will have the same expectation as $\tilde{c}_{(s,a)}(n)$ such that it becomes a martingale, for each $(s,a) \in \mathcal{J}^*$. However, this is not true since we have estimation error over $A_{\mathcal{J}^*,\mathcal{I}^*}$, and we only use their estimates to compute $\boldsymbol{x}^n$. Nevertheless, we can show that $\tilde{c}_{(s,a)}(n)$ for each $(s,a) \in \mathcal{J}^*$ behaves as a sub-martingale. Then, from the concentration property of the sub-martingale, we upper bound $\mathbb{E}[c^N_{(s,a)}]$ for each $(s,a) \in \mathcal{J}^*$. The term $|\mathbb{E}[c^N_{(s,a)}]|$ for each $k \in [K]\backslash\mathcal{J}^*$ can be upper bounded as well.

There will be an additional important problem parameter showing up in our bound, which is related to $\mathcal{I}^*$ and $\mathcal{J}^*$, and can be described as follows. Denote by $\{\sigma_1(A_{\mathcal{J}^*,\mathcal{I}^*}), \ldots, \sigma_{d_2}(A_{\mathcal{J}^*,\mathcal{I}^*})\}$ the eigenvalues of the matrix $A_{\mathcal{J}^*,\mathcal{I}^*}$. We define σ as

$$\sigma = \min \left\{ |\sigma_1(A_{\mathcal{J}^*,\mathcal{I}^*})|, \ldots, |\sigma_{d_2}(A_{\mathcal{J}^*,\mathcal{I}^*})| \right\}. \tag{24}$$

From the optimality of $\mathcal{I}^*, \mathcal{J}^*$ and thus the non-singularity of the matrix $A_{\mathcal{J}^*,\mathcal{I}^*}$, we know that $\sigma > 0$. Our bounds are presented in the following theorem. The formal proof is relegated to Appendix J and we provide a brief sketch of the proof for illustration.

Theorem E.2. *Denote by $\bar{\pi}^N$ the output policy of Algorithm 1 and denote by N the number of rounds. Then, it holds that*

$$N \cdot V^{\mathrm{LP}} - \sum_{n=1}^{N} \boldsymbol{r}^\top \mathbb{E}[\boldsymbol{x}^n] \leq O\left(\frac{d_2^2(1 + \|A_{\mathcal{J}^*,\mathcal{I}^*}\|_\infty)}{\sigma^2} \cdot \log(N) \right) \tag{25}$$

where the parameter σ defined in (24). *Also, in terms of the constraint violation, for any $(s,a) \in \mathcal{J}^*$, we have*

$$\left| N \cdot c_{(s,a)} - \sum_{n=1}^{N} A_{(s,a),:} \mathbb{E}\left[\boldsymbol{x}^n\right] \right| \leq O\left(\frac{d_2^2(1 + \|A_{\mathcal{J}^*,\mathcal{I}^*}\|_\infty)}{\sigma^2} \cdot \log(N) \right). \tag{26}$$

The value of σ shows up in our final bounds as it characterizes how hard it is to learn the matrix $A_{\mathcal{J}^*,\mathcal{I}^*}$ and how sensitive the solution to the linear system (6) will be when some perturbation is introduced by replacing $A_{\mathcal{J}^*,\mathcal{I}^*}$ with its estimation and replacing $\boldsymbol{c}_{\mathcal{J}^*}$ with $\boldsymbol{c}^n_{\mathcal{J}^*}/(N-n+1)$. Also, σ gives a natural upper bound of the range of the optimal dual variable corresponding to the optimal basis $\mathcal{I}^*$ and $\mathcal{J}^*$. Note that in previous literature that establishes logarithmic regret for online LP/resource allocation (e.g. [61, 37, 27]), the conditional number of the constraint matrix will also show up in the final bounds. Therefore, we regard the existence of σ as the consequence of adopting the resolving algorithms to solve the LP (3).

We present the final sample complexity bound of our algorithm. Note that we have a bound on the constraint violation for each $(s, a) \in \mathcal{J}^*$. For the constraints not in the index set $\mathcal{J}^*$, we can also bound its violation using the non-singularity of the matrix $A_{\mathcal{J}^*, \mathcal{I}^*}$. This is another benefit of identifying the optimal basis $\mathcal{I}^*$ and $\mathcal{J}^*$.

Theorem E.3. *With a sample complexity bound of*

$$O\left(K \cdot \frac{\log(K/\varepsilon)}{\Delta^2} + \frac{d_2^2(1 + \|A_{\mathcal{J}^*, \mathcal{I}^*}\|_\infty)}{\sigma^2} \cdot \frac{\log(1/\varepsilon)}{\varepsilon}\right),$$

where K refers to the number of constraints in V^{LP}, the parameters Δ defined in (18)*, σ given in* (24)*, we obtain a solution $\bar{\boldsymbol{x}}^N$ from Algorithm 1 (defined in* (9)*) such that*

$$V^{\mathrm{LP}} - \boldsymbol{r}^\top \bar{\boldsymbol{x}}^N \leq \varepsilon \;\; \textit{and} \;\; A \cdot \bar{\boldsymbol{x}}^N - \boldsymbol{c} \leq \varepsilon, \; \forall k \in [K].$$

When we are dealing with a RL problem with a large or infinite state-action space, we can consider the Reduced LP (V^{LP} refers to V^{RLP} in (13)) and we can further apply the results from Theorem B.2 to bound the number K. Then, we have the following sample complexity bound.

Corollary E.4. *For RL problems with large or infinite state-action pairs, we have the sample complexity bound*

$$O\left(\left(\frac{d_2^2(1 + \|A_{\mathcal{J}^*, \mathcal{I}^*}\|_\infty)}{\sigma^2} + \frac{1}{\Delta^2}\right) \cdot \frac{\log^2(1/\varepsilon)}{\varepsilon}\right)$$

for our algorithm.

Note that the above bound is independent of the original state-action space. One crucial part of our sample complexity bound is the dependency on a constant suboptimality gap Δ, where its definition distinguishes our work from the previous work. Notably, a prevalent way to define the suboptimality gap in previous works (see e.g. [56, 20]) is through the Q-value. To be specific, a parameter δ is defined as

$$\delta := \min_{(s,a) \in \mathcal{S} \times \mathcal{A}} \left\{V^*(s) - Q^*(s, a) : V^*(s) - Q^*(s, a) > 0\right\}. \tag{27}$$

Here, δ represents the minimal possible gap between the V-value and the Q-value for a sub-optimal action, where the gap is minimized over all possible states. Note that when the state space $\mathcal{S}$ is infinite, taking the minimum over $\mathcal{S}$ may result in $\delta = 0$. Therefore, it is usually assumed that $\delta > 0$ in previous literature to carry out their analysis. In contrast, the constant gap Δ defined in (18) is always greater than 0 and $\Delta > 0$ is automatically satisfied without having to assume it. In summary, the constant gap Δ in (18) represents the gap between the basic solutions of the LP formulation while the gap δ in (27) represents the gap between the Q-values of the RL problem. In general, the gap Δ defined in our paper takes a different perspective to measure the sub-optimality gap of the underlying problem instance.

F Numerical Experiments

We conduct numerical experiments to test the practical performance of our algorithms. We test on the "Mountain Car" problem, which is a well-known continuous control task, with additional random noise to represent the sampling uncertainty in real-life scenarios. The detailed experimental setup is reported below and the experimental results are also presented in the following sections.

F.1 Experimental Setup

We report the detailed experimental setup of our numerical experiments.

Mountain Car Problem The "Mountain Car" problem is a well-known continuous control task. The car is located in the middle of the valley and we need to control the acceleration of the car so that it can overcome its gravity and climb to the top of the mountain. It has a state space $S \subset R^2$ and an action space $A \subset R$, where the state s is made up of position p and velocity v. The domain of these variables are:

$$p \in [-1.20, 0.60], \ \ v \in [-0.07, 0.07], \ \ a \in [-1, 1].$$

The car intends to follow the following transition dynamics:

$$\begin{aligned} & v_{t+1} = v_t + a * 0.0015 - 0.0025 * cos(3 * p_t), \\ & p_{t+1} = p_t + v_{t+1}. \end{aligned}$$

However, instead of letting the car follow the transition dynamics in (28) deterministically, there are some additional random noises in our simulator, which represents the sampling uncertainty in real-life scenarios and makes our setting more challenging. To be specific, we add a random noise ϵ to the simulator to simulate various unexpected situations in reality. Define the function $K : S \times A \to S$ to represent the intended state transition given state $s = (p, v)$ and action a as described in (28). Then, the probability of the next state $s^{'}$ given current state s and action a is:

$$P(s^{'}|s, a) = \begin{cases} 0.9 + 0.1 * P_{random}(s^{'}), \text{ if} s^{'} == K(s, a) \\ 0.1 * P_{random}(s^{'}), \text{ otherwise} \end{cases} \tag{28}$$

$P_{random}(s^{'})$ is the probability that $s^{'}$ is uniformly distributed in the neighborhood of the real next state $K(s, a)$. The neighborhood can be described as an interval. Denote by i^{intend} the index of $K(s, a)$, then the neighborhood is given by $[i^{\text{intend}} - r_\epsilon, i^{\text{intend}} + r_\epsilon]$, where r_ϵ is a pre-specified range. The real transition procedure of our setting can be described as with probability 0.9, the next state is $K(s, a)$, and with probability 0.1, the next state is uniformly sampled from a small neighborhood of $K(s, a)$ with a radius r_ϵ.

Linear Function Approximation We approximate the V-function by defining base feature maps for position p and velocity v separately. Specifically, we use radius basic functions, where p_i and v_i are fifth equal parts of the domain of position p and velocity v:

$$\phi_p(p) = \sum_{i=1}^{5} exp(-(\frac{p - p_i}{0.2})^2), \quad \phi_v(v) = \sum_{i=1}^{5} exp(-(\frac{v - v_i}{0.2})^2).$$

Then we can derive the overall feature map for the V-function by taking the outer product of ϕ_p and ϕ_v:

$$\phi_V(p, v) := vec\{\phi_p(p) \otimes \phi_v(v)\} \in R^{25}. \tag{29}$$

Given any position p, velocity v and weight w, we can get the V-function: $f_w(p, v) := \langle w, \phi_V(p, v)\rangle$.

Experiment Configuration Note that "Mountain Car" problem is a continuous control problem with infinitely large state-action space, thus the ALP (3) is a semi-infinite LP with infinite number of constraints. In order to solve the ALP (3) approximately, we adopt the idea of constraint sampling described in Appendix B.1 to consider the reduced LP (RLP) as given in (13). The constraint sampling procedure can be regarded as discretizing the state-action space. To be specific, our constraint sampling procedure can be regarded as dividing position p, velocity v and action a into 40, 60 and 5 parts, respectively. Therefore, there are in total $12,000$ constraints in the reduced LP (RLP) in (13). For our resolving algorithm, we apply Algorithm 2 and Algorithm 1 to the RLP (13) to approximate the optimal weight. For the non-resolving algorithm, we use the same number of samples to construct an estimation of the RLP (13) and directly solve this estimated RLP to approximate the optimal weight.

We define that y_{real} refers to the optimal value of the benchmark RLP (13) and $y_{resolve}$ is the objective value of the RLP under our solution, then the relative optimal value gap is:

$$\frac{|y_{real} - y_{resolve}|}{y_{real}}.$$

We define that $\boldsymbol{x}_{real}$ refers to the solution of the benchmark RLP (13) corresponding to the optimal basis identified by Algorithm 2, and $\boldsymbol{x}_{resolve}$ is our solution, then the relative solution gap is:

$$\frac{\|\boldsymbol{x}_{real} - \boldsymbol{x}_{resolve}\|_2}{\|\boldsymbol{x}_{real}\|_2}.$$

After substituting our result $\boldsymbol{x}_{resolve}$ into constraints, constraint violation refers to the maximum constraint as they are expected to be equal or less than zero:

$$\max\left\{([A \cdot \boldsymbol{x}_{resolve} - \boldsymbol{c}]^{+})\right\}.$$

When constructing the RLP (13) and the dataset H^n in Algorithm 1, for each (s, a) pair, we sample L number of state transition $s^{'}$ from the simulator as its next state. Therefore, suppose $n_{s'}$ represents the sampling number of state $s^{'}$, the probability of each state is:

$$P(s^{'}|(s, a)) = \frac{n_{s'}}{L}$$

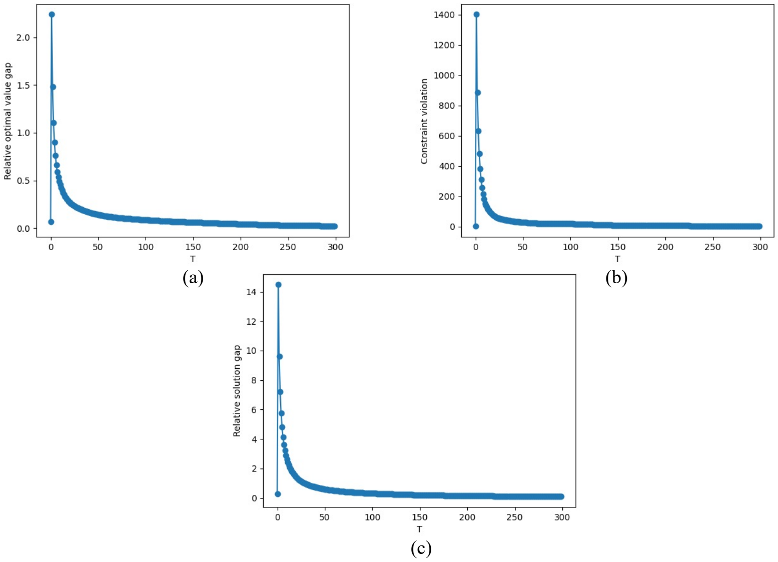
The true RLP is the benchmark and the output of our algorithm should be close to the true RLP. We evaluate this from three aspects: relative optimal value gap, constraints violations and the relative solution gap, with the definitions given above. In the implementation of our algorithm (Algorithm 2 and Algorithm 1), we use the first $120,000$ number of samples (10 samples for each constraint) to construct an estimation of the RLP and then carry out Algorithm 2 to identify the basis $\mathcal{I}^*$ and $\mathcal{J}^*$. We then use the rest of the samples to carry out the resolving steps in Algorithm 1. Note that the samples used to carry out Algorithm 2 can also be used again to construct an initial estimate of $A_{\mathcal{I}^*, \mathcal{J}^*}$ such that Algorithm 1 will have a warm start. In the implementation of the non-resolving algorithm, we use the same amount of samples to construct an estimate of the RLP, and directly solve this estimation to obtain a solution. Note that if a total $240,000$ samples are used in our algorithm ($120,000$ for Algorithm 2 and $120,000$ for Algorithm 1), we also use the same amount $240,000$ samples for the non-resovling algorithm, with 20 sample for each constraint as the original RLP has $12,000$ constraints.

F.2 Experiment Results.

We use T to denote the number of resolving steps in our algorithm. We study how our algorithm converge with the number of resolving steps. At $T = 0$, our resolving algorithm has not started yet and the estimated LP is still far from the real LP. Figure 1 (a) and (c) show that the solution without resolving is still far from the benchmark, which reveals the necessity and importance of our resolving algorithm. As T increases with more and more resolving steps, as shown in Figure 1 (a), the relative optimal value gap converges to 0 as T increases, which means our resolving algorithm's performance is close to the benchmark. In Figure 1 (c), the relative error of our LP solution and the true LP solution also converges to zero. These two figures together confirm that our resolving algorithm obtains a result that is very close to the benchmark. Figure 1 (b) further tests whether our solution satisfies the constraints, and the maximum constraint violation error converges to 0, indicating that our solution can approximately satisfy all constraints. The exact definitions of the metrics in Figure 1 are presented in Appendix F.1.

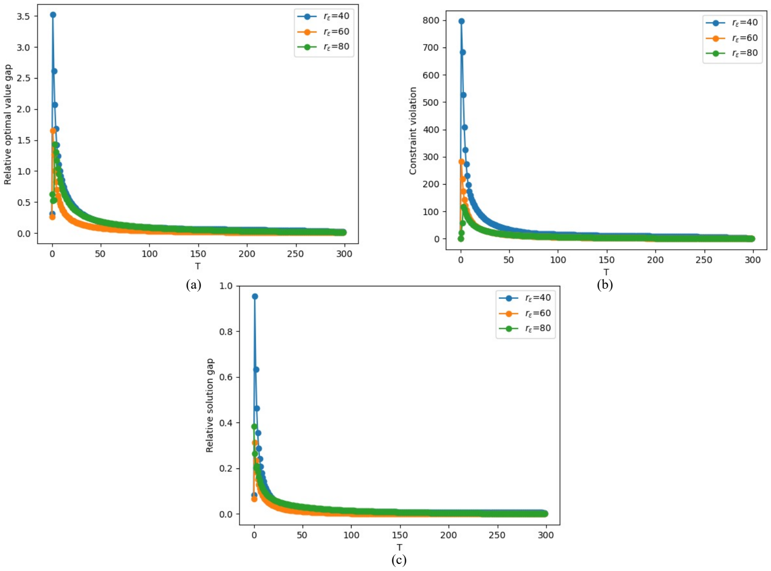
Figure 2 studies the performances of our algorithm with respect to different noise radius r_ϵ and all three figures share common trends. The setting with $r_\epsilon = 40$ has the smallest range of the random noise, while $r_\epsilon = 80$ has the largest range. When r_ϵ decreases, convergence rates in all three figures become faster. This is easy to interpret as that a smaller r_ϵ indicates a smaller range of random noise, which makes it easier for our algorithm to converge to the real LP solution.

Moreover, we implement our policy on the real Mountain Car problem to see whether our policy can succeed in real problems. Following classical settings, we restrict our policy to push the car to reach the top of the mountain within 1000 steps. We repeat the experiments by 1000 times and we compute the success rate, i.e., the percentage of times that our policy can push the car to reach the top of the mountain within 1000 steps. We compare the performances of our resolving algorithm (Algorithm 2 and Algorithm 1) with the Deep Q-learning Network that has the same number of parameters as our resolving algorithm and a non-resovling algorithm that directly uses the same amount of sample to construct the estimation of the RLP (13) to approximate the LP solution. Figure 3 shows a significant



(a) (b) (c)

Figure 1: Numerical performance of our resolving algorithm on the Mountain Car Problem. (a) The relative optimal value gap between our algorithm 1 and the benchmark RLP (13). (b) The maximum constraint violation after substituting our result into constraints. If the constraints are satisfied, then the violation is 0. (c) The relative gap between our LP solution and the real LP solution.



(a) (b) (c)

Figure 2: Numerical performance comparison between different random noise radius r_ϵ. (a) The relative optimal value gap between our algorithm 1 and the benchmark RLP (13). (b) The maximum constraint violation after substituting our result into constraints. If the constraints are satisfied, then the violation is 0. (c) The relative gap between our LP solution and the real LP solution.

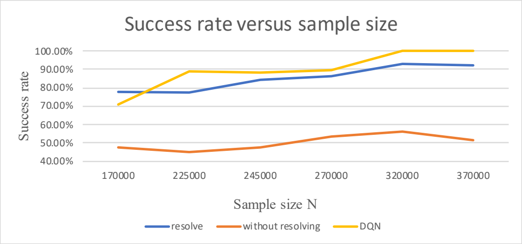


Figure 3: Success rates of our algorithm on the real Mountain Car Problem. We compare the performance of our algorithm with the Deep Q-learning Network and a non-resolving algorithm 1 and that directly solves the estimated LP, under the same sample size N.

in the success rate of our algorithm compared to the non-resolving algorithm. Figure 3 shows that with the same amount of sample N, the performance of our algorithm is significantly improved. The success rate of the non-resolving algorithm is about 50%, and the success rate is increased by about 40% with our algorithm. In addition, as the sample size increases, the success rate of our algorithm can reach up to 92.5%, which reveals the great empirical performances of our algorithm for solving real-life problems. Besides, Figure 3 also shows that the performance of our algorithm is comparable to DQN, where the difference between the success rate is relatively small, and our algorithm even outperforms DQN when the number of samples is relatively small (about 17000 samples). We believe these results demonstrate the effectiveness of our algorithm.

Discussions on computation cost. Note that our Algorithm 2 can be conducted very efficiently in practice, and the computational complexity is equivalent to solving the LP by one time, using the simplex method. In our numerical experiment, we use Gurobi (as well as COPT) to solve the reduced LP, which has 12,000 constraints, and it takes 0.3 second to finish running Algorithm 2. We run our code on a computer with Apple M1 chip, 8 GB memory, and it only takes about one hour to run the entire Algorithm 2 and Algorithm 1, which highlights the numerical efficiency of our approach. Moreover, we can also adopt Algorithm 3 in Appendix C to identify the basis. Note that Algorithm 3 only needs to solve an LP by a finite number of times and also can be carried out very efficiently due to the power of modern LP solvers. Moreover, it has been demonstrated that Algorithm 3 enjoys a polynomial-time computational complexity. In this way, we conclude that our algorithms are numerically efficient.

G Proof of Lemma 3.1

Our proof follows from the standard LP theory regarding the optimality of basic solutions, with mild modifications. Note that in the LP standard form

$$V^{\mathrm{LP}} = \max\ \boldsymbol{r}^\top \boldsymbol{x} \quad \text{s.t. } A\boldsymbol{x} + \boldsymbol{s} = \boldsymbol{c}, \quad \boldsymbol{x} \in \mathbb{R}^{d_1}, \boldsymbol{s} \geq 0, \tag{30}$$

we can sort the decision variable as $(\boldsymbol{x}, \boldsymbol{s}) \in \mathbb{R}^{d_1+K}$ and we let an index set $\mathcal{I}' \subset [d_1 + K]$ be an optimal basis set (there must exist an optimal basis set). Then, from the standard LP theory, the following condition holds for the optimal basis set $\mathcal{I}'$. It holds $|\mathcal{I}'| = K$. Also, we can divide $\mathcal{I}'$ as $\mathcal{I}' = \mathcal{I}_1' \cup \mathcal{I}_2'$ with $\mathcal{I}_1'$ being an index set for the $\boldsymbol{x}$ variable and $\mathcal{I}_2'$ being an index set for the $\boldsymbol{s}$ variable. Then, we have that the solution $(\boldsymbol{x}^*, \boldsymbol{s}^*)$ satisfying the conditions

$$\boldsymbol{x}^*_{\mathcal{I}_1'^c} = 0, \quad \boldsymbol{s}^*_{\mathcal{I}_2'^c} = 0 \tag{31}$$

and

$$A_{:,\mathcal{I}_1} \cdot \boldsymbol{x}^*_{\mathcal{I}_1} + \boldsymbol{s}^* = \boldsymbol{c}. \tag{32}$$

is an optimal solution to V^{LP}. Moreover, the linear system described in (31) and (32) is uniquely determined. We now write the linear system in (31) and (32) into the matrix form. Since $|\mathcal{I}'| = K = |\mathcal{I}_1'| + |\mathcal{I}_2'|$, we define the matrix

$$\bar{A} = \begin{bmatrix} A_{\mathcal{I}_2'^c,\, \mathcal{I}_1}, & 0, \ldots, 0 \\ A_{\mathcal{I}_2',\, \mathcal{I}_1}, & I_{|\mathcal{I}_2'|} \end{bmatrix}$$

where $I_{|\mathcal{I}'_2|}$ denotes an identify matrix with size $|\mathcal{I}'_2|$. Then, the linear system described in (31) and (32) can be written as

$$\boldsymbol{x}^*_{\mathcal{I}'^c_1} = 0, \quad \boldsymbol{s}^*_{\mathcal{I}'^c_2} = 0$$

and

$$\bar{A} \cdot \begin{pmatrix} \boldsymbol{x}^*_{\mathcal{I}_1} \\ \boldsymbol{s}^*_{\mathcal{I}_2} \end{pmatrix} = \boldsymbol{c}. \tag{33}$$

Since the linear system described in (31) and (32) is uniquely determined, we know that the square matrix $\bar{A}$ is non-singular.

The non-singularity of the matrix $\bar{A}$ would implies the non-singularity of the matrix $A_{\mathcal{I}'^c_2,\, \mathcal{I}_1}$. To see this, we first note that since $\boldsymbol{s} \in \mathbb{R}^K$, we have $|\mathcal{I}'_2| + |\mathcal{I}'^c_2| = K$ and we also have $|\mathcal{I}'_1| + |\mathcal{I}'_2| = K$, which implies that $|\mathcal{I}'_1| = |\mathcal{I}'^c_2|$ and thus the matrix $A_{\mathcal{I}'^c_2,\, \mathcal{I}_1}$ is a square matrix. Moreover, the non-singularity of $\bar{A}$ must imply that the rows of the matrix $A_{\mathcal{I}'^c_2,\, \mathcal{I}_1}$ must be linearly independent from each other. Thus, we know that the square matrix $A_{\mathcal{I}'^c_2,\, \mathcal{I}_1}$ is non-singular. Therefore, we know that $\boldsymbol{x}^*_{\mathcal{I}_1}$ can be uniquely determined as the unique solution to the linear system

$$A_{\mathcal{I}'^c_2,\, \mathcal{I}_1} \cdot \boldsymbol{x}^*_{\mathcal{I}_1} = \boldsymbol{c}. \tag{34}$$

We now set the index set

$$\mathcal{I} = \mathcal{I}'_1, \text{ and } \mathcal{J} = \mathcal{I}'^c_2.$$

Then, we know that

$$|\mathcal{I}| = |\mathcal{J}|$$

and a solution $\boldsymbol{x}^*$ uniquely defined as

$$\boldsymbol{x}^*_{\mathcal{I}^c} = 0$$

and

$$A_{\mathcal{I}'^c_2,\, \mathcal{I}_1} \cdot \boldsymbol{x}^*_{\mathcal{I}_1} = \boldsymbol{c}$$

is an optimal solution to V^{LP}, where $\boldsymbol{s}$ is the slackness variables determined corresponding to $\boldsymbol{x}^*$. Our proof is thus completed.

H Proof of Theorem D.1

We now condition on the event that

$$\mathcal{E} = \left\{ \left| \frac{\gamma}{N} \cdot \sum_{n=1}^{N} \phi_i(s_n) - \phi_i(s) - A_{(s,a),i} \right| \leq \mathrm{Rad}(N, \varepsilon), \ \forall (s,a) \in \mathcal{K}, \ \forall i \in [d_1] \right\}. \tag{35}$$

We know that this event $\mathcal{E}$ happens with probability at least $1 - d_1 \cdot |\mathcal{K}| \cdot \varepsilon$.

We first bound the gap between V^{LP} and $\hat{V}^{\mathrm{LP}}(\mathcal{H})$, for any set $\mathcal{I}$. The result is formalized in the following claim, where the proof is presented at the end of this proof.

Claim H.1. *Conditional on the event $\mathcal{E}$ (35) happens, for any set $\mathcal{I}$, it holds that*

$$\left| V^{\mathrm{LP}} - \hat{V}^{\mathrm{LP}}(\mathcal{H}) \right| \leq C_1 \cdot \mathrm{Rad}(N, \varepsilon), \tag{36}$$

where C_1 is a constant that is independent of N and ε.

Note that Claim H.1 shows that an optimal solution to $\hat{V}^{\mathrm{LP}}(\mathcal{H})$ would be at most $O(\mathrm{Rad}(N, \varepsilon))$ distance away from the optimal solution to V^{LP}. We now restrict to an optimal basic solution of $\hat{V}^{\mathrm{LP}}(\mathcal{H})$ and from Claim H.1, it is at most $O(\mathrm{Rad}(N, \varepsilon))$ distance away from the optimal solution to V^{LP}. We denote by $\hat{I}$ and $\hat{J}$ an optimal basis to $\hat{V}^{\mathrm{LP}}(\mathcal{H})$, and denote by $\hat{\boldsymbol{x}}$ the corresponding optimal basic solution to $\hat{V}^{\mathrm{LP}}(\mathcal{H})$. We know that $\hat{\boldsymbol{x}}$ satisfies the condition

$$\hat{\boldsymbol{x}}_{\hat{\mathcal{I}}^c} = 0 \tag{37}$$

and

$$\hat{A}_{\hat{\mathcal{J}}, \hat{\mathcal{I}}}(\mathcal{H}) \cdot \hat{x}_{\hat{\mathcal{I}}} = \boldsymbol{c}_{\hat{\mathcal{J}}}. \tag{38}$$

We now consider the gap of the basis $\hat{I}$ and $\hat{J}$ with respect to the optimal basis of V^{LP}. Denote by $\boldsymbol{x}'$ the basic solution to V^{LP}, corresponding to the basis $\hat{\mathcal{I}}$ and $\hat{\mathcal{J}}$. (Note that the non-singularity of the square matrix $\hat{A}_{\hat{\mathcal{J}},\hat{\mathcal{I}}}(\mathcal{H})$ implies the non-singularity of the square matrix $A_{\mathcal{J},\mathcal{I}}$, as long as $\mathrm{Rad}(N,\varepsilon) \leq C_1$ for some constant C_1). Then, $\boldsymbol{x}'$ can be described as

$$\boldsymbol{x}'_{\hat{\mathcal{I}}^c} = 0 \text{ and } A_{\hat{\mathcal{J}},\hat{\mathcal{I}}}(\mathcal{H}) \cdot \boldsymbol{x}_{\mathcal{I}} = \boldsymbol{c}_{\mathcal{J}}. \tag{39}$$

Comparing the linear system (38) and (39), we can bound the distance between $\hat{\boldsymbol{x}}$ and $\boldsymbol{x}'$, and thus bound the feasibility gap and the sub-optimality gap of the solution $\boldsymbol{x}'$ to V^{LP}. For notation brevity, we denote by

$$\Delta A = A_{\hat{\mathcal{J}},\hat{\mathcal{I}}} - \hat{A}_{\hat{\mathcal{J}},\hat{\mathcal{I}}}.$$

Then, following standard perturbation analysis of linear equations [22], we have that

$$\begin{aligned}\frac{\|\hat{\boldsymbol{x}}^n_{\hat{\mathcal{I}}} - \boldsymbol{x}'_{\hat{\mathcal{I}}}\|_1}{\|\boldsymbol{x}'_{\hat{\mathcal{I}}}\|_1} &\leq \frac{\kappa(A_{\hat{\mathcal{J}},\hat{\mathcal{I}}})}{1-\kappa(A_{\hat{\mathcal{J}},\hat{\mathcal{I}}}) \cdot \frac{\|\Delta A\|_1}{\|A_{\hat{\mathcal{J}},\hat{\mathcal{I}}}\|_1}} \cdot \frac{\|\Delta A\|_1}{\|A_{\hat{\mathcal{J}},\hat{\mathcal{I}}}\|_1} \leq 2 \cdot \kappa(A_{\hat{\mathcal{J}},\hat{\mathcal{I}}}) \cdot \frac{\|\Delta A\|_1}{\|A_{\hat{\mathcal{J}},\hat{\mathcal{I}}}\|_1} \\ &\leq 2\|\Delta A\|_1/\sigma,\end{aligned} \tag{40}$$

where we denote by $\sigma > 0$ the smallest absolute value of the singular values of the square matrix $A_{\hat{\mathcal{J}},\hat{\mathcal{I}}}$. Therefore, further noting that $\hat{\boldsymbol{x}}_{\hat{\mathcal{I}}^c} = \boldsymbol{x}'_{\hat{\mathcal{I}}^c} = 0$ and $\|\Delta A\|_1 \leq C_1 \cdot \mathrm{Rad}(N,\varepsilon)$ for a constant C_1, we know that

$$\|\hat{\boldsymbol{x}} - \boldsymbol{x}'\|_1 \leq C_2 \cdot \mathrm{Rad}(N,\varepsilon) \tag{41}$$

for a constant C_2. We now show that when N is sufficiently large, $\boldsymbol{x}'$ becomes an optimal solution to V^{LP}. If not, we classify into the two possible situations:

Situation (i): the basic solution $\boldsymbol{x}'$ is an infeasible solution to V^{LP}. We now bound the infeasibility gap of $\boldsymbol{x}'$ to V^{LP}. From the feasibility of $\hat{\boldsymbol{x}}$ to the LP $\hat{V}^{\mathrm{LP}}(\mathcal{H})$, we know that

$$\hat{A}(\mathcal{H}) \cdot \hat{\boldsymbol{x}} \leq \boldsymbol{c}.$$

Then, combining with (41), we know that

$$\hat{A}(\mathcal{H}) \cdot \boldsymbol{x}' \leq \hat{A}(\mathcal{H}) \cdot \hat{\boldsymbol{x}} + \|\hat{A}(\mathcal{H})\|_1 \cdot \|\hat{\boldsymbol{x}} - \boldsymbol{x}'\|_1 \leq \boldsymbol{c} + C_3 \cdot \mathrm{Rad}(N,\varepsilon)$$

for a constant C_3. Therefore, we have shown that the infeasibility gap of the basic solution $\boldsymbol{x}'$ to V^{LP} is $C_3 \cdot \mathrm{Rad}(N,\varepsilon)$. However, recalling the definition of the parameter δ_1 in (16), we know that in order for the basic solution $\boldsymbol{x}'$ to become infeasible to V^{LP}, it must hold that

$$\delta_1 \leq C_3 \cdot \mathrm{Rad}(N,\varepsilon).$$

In other words, when the sample size N is large enough such that the condition

$$\mathrm{Rad}(N,\varepsilon) \leq \frac{1}{C_3} \cdot \delta_1 \tag{42}$$

holds, we know that the basic solution $\boldsymbol{x}'$ must be a feasible solution to V^{LP}.

Situation (ii): We now consider the situation where the basic solution $\boldsymbol{x}'$ is a feasible solution to V^{LP} but not optimal. We now bound the suboptimality gap of $\boldsymbol{x}'$ to V^{LP}. Noting the bound in (41), we have that

$$\boldsymbol{r}^\top \boldsymbol{x}' \geq \boldsymbol{r}^\top \hat{\boldsymbol{x}} - \|\boldsymbol{r}\|_1 \cdot \|\boldsymbol{x}' - \hat{\boldsymbol{x}}\| \geq \boldsymbol{r}^\top \hat{\boldsymbol{x}} - C_4 \cdot \mathrm{Rad}(N,\varepsilon)$$

for a constant C_4. From the optimality of $\hat{\boldsymbol{x}}$ to the LP $\hat{V}^{\mathrm{LP}}(\mathcal{H})$ and the gap between $\hat{V}^{\mathrm{LP}}(\mathcal{H})$ and V^{LP}, as shown in Claim H.1, we know that

$$\boldsymbol{r}^\top \boldsymbol{x}' \geq \hat{V}^{\mathrm{LP}}(\mathcal{H}) - C_4 \cdot \mathrm{Rad}(N,\varepsilon) \geq V^{\mathrm{LP}} - (C_1 + C_4) \cdot \mathrm{Rad}(N,\varepsilon).$$

However, recalling the definition of the parameter δ_2 in (17), we know that in order for the feasible basic solution $\boldsymbol{x}'$ to become infeasible to V^{LP}, it must hold that

$$\delta_2 \leq (C_1 + C_4) \cdot \mathrm{Rad}(N,\varepsilon).$$

In other words, when the sample size N is large enough such that the condition

$$\mathrm{Rad}(N,\varepsilon) \leq \frac{1}{C_1 + C_4} \cdot \delta_2$$

holds, we know that $\boldsymbol{x}'$ must be an optimal solution to V^{LP}.

From the arguments above, we summarize that when the conditions

$$\mathrm{Rad}(N,\varepsilon) \le \frac{1}{C_3}\cdot\delta_1 \ \text{ and } \ \mathrm{Rad}(N,\varepsilon) \le \frac{1}{C_1+C_4}\cdot\delta_2$$

are satisfied, the basic solution $\boldsymbol{x}'$, corresponding to $\hat{I}$ and $\hat{\mathcal{J}}$, must be an optimal solution to V^{LP}. Therefore, we conclude that the basis $\hat{I}$ and $\hat{J}$ must be an optimal basis to V^{LP}, when the condition

$$\mathrm{Rad}(N,\varepsilon) \le O(\Delta) = O(\min\{\delta_1,\delta_2\})$$

is satisfied. Our proof is thus completed.

H.1 Proof of Claim H.1

Denote by $\boldsymbol{x}^*$ an optimal solution to V^{LP}. We now construct a feasible solution to $\hat{V}^{\mathrm{LP}}(\mathcal{H})$ based on $\boldsymbol{x}^*$. Note that conditional on the event $\mathcal{E}$ happens, we have that

$$\hat{A}(\mathcal{H})\boldsymbol{x}^* \le A\boldsymbol{x}^* + \|\boldsymbol{x}^*\|_1\cdot\mathrm{Rad}(N,\varepsilon)\cdot\boldsymbol{e} \le \boldsymbol{c} + \|\boldsymbol{x}^*\|_1\cdot\mathrm{Rad}(N,\varepsilon)\cdot\boldsymbol{e}, \tag{43}$$

where $\boldsymbol{e} = (1,\dots,1)^\top \in \mathbb{R}^K$ is an all-one vector. The feasible solution, denoted by $\hat{\boldsymbol{x}}$, can be given as $\hat{\boldsymbol{x}} = \boldsymbol{x}^* + \Delta\boldsymbol{x}$. As long as $\Delta\boldsymbol{x}$ satisfies the condition

$$\hat{A}(\mathcal{H})\Delta\boldsymbol{x} \le -\|\boldsymbol{x}^*\|_1\cdot\mathrm{Rad}(N,\varepsilon)\cdot\boldsymbol{e}, \tag{44}$$

we know that $\hat{\boldsymbol{x}}$ will be a feasible solution to $\hat{V}^{\mathrm{LP}}(\mathcal{H})$. Further note that for any possible $\hat{A}(\mathcal{H})$, the LP

$$c(\mathcal{H}) = \max\ \boldsymbol{r}^\top\boldsymbol{x} \quad \text{s.t. } \hat{A}(\mathcal{H})\boldsymbol{x} \le -\boldsymbol{e}, \quad \boldsymbol{x}\in\mathbb{R}^{d_1}, \tag{45}$$

describe the LP applying to a reinforcement learning instance with the transition kernel given by the empirical estimate constructed using the dataset $\mathcal{H}$ and the cost vector given as $-\boldsymbol{e}$. Since the cost vector $-\boldsymbol{e}$ is uniform over all state-action pair, the V-value is always $-\frac{1}{1-\gamma}$ for any possible state, and the objective value of LP (45) well approximates the aggregated V-value with initial distribution μ_1, which is exactly $-\frac{1}{1-\gamma}$. In fact, from Theorem 2 of [14], we know that

$$\left|-\frac{1}{1-\gamma} - c(\mathcal{H})\right| \le \frac{2}{1-\gamma}\cdot\min_{\boldsymbol{x}}\left\|-\frac{1}{1-\gamma}\cdot\boldsymbol{e} - \Phi\boldsymbol{x}\right\|_\infty \le \frac{2}{(1-\gamma)^2}.$$

Therefore, we know that

$$-c(\mathcal{H}) \le \frac{1}{1-\gamma} + \frac{2}{(1-\gamma)^2}. \tag{46}$$

Denote by $\boldsymbol{x}'$ one optimal solution to LP (45), we can in fact set

$$\Delta\boldsymbol{x} = \|\boldsymbol{x}^*\|_1\cdot\mathrm{Rad}(N,\varepsilon)\cdot\boldsymbol{x}'. \tag{47}$$

We know that $\hat{\boldsymbol{x}} = \boldsymbol{x}^* + \Delta\boldsymbol{x}$ forms a feasible solution to $\hat{V}^{\mathrm{LP}}(\mathcal{H})$, with the formulation of $\Delta\boldsymbol{x}$ given in (47). As a result, it holds that

$$\begin{aligned} V^{\mathrm{LP}} &= \boldsymbol{r}^\top\boldsymbol{x}^* \le \boldsymbol{r}^\top\hat{\boldsymbol{x}} - \boldsymbol{r}^\top\Delta\boldsymbol{x}^* = \boldsymbol{r}^\top\hat{\boldsymbol{x}} - c(\mathcal{H}) \\ &\le \hat{V}^{\mathrm{LP}}(\mathcal{H}) + \left(\frac{1}{1-\gamma} + \frac{2}{(1-\gamma)^2}\right)\cdot\|\boldsymbol{x}^*\|_1\cdot\mathrm{Rad}(N,\varepsilon). \end{aligned} \tag{48}$$

In a same way, we can show that

$$\hat{V}^{\mathrm{LP}}(\mathcal{H}) \le V^{\mathrm{LP}} + \left(\frac{1}{1-\gamma} + \frac{2}{(1-\gamma)^2}\right)\cdot\|\boldsymbol{x}^*\|_1\cdot\mathrm{Rad}(N,\varepsilon), \tag{49}$$

which completes our proof.

I Proof of Lemma E.1

From the update of Algorithm 1, we know that

$$\sum_{n=1}^{N} (\boldsymbol{r})^\top \mathbb{E}[\boldsymbol{x}^n] = \sum_{n=1}^{N} (\boldsymbol{r}_{\mathcal{I}^*})^\top \mathbb{E}\left[\boldsymbol{x}^n_{\mathcal{I}^*}\right]$$

Denote by $\boldsymbol{x}^*$ and $\boldsymbol{y}^*$ the optimal primal-dual variable corresponding to the optimal basis $\mathcal{I}^*$ and $\mathcal{J}^*$. From the complementary slackness condition and noting that $\boldsymbol{x}^*_{\mathcal{I}^*} > 0$, we know that

$$A^\top_{\mathcal{J}^*,\mathcal{I}^*} \boldsymbol{y}_{\mathcal{J}^*} = \boldsymbol{r}_{\mathcal{I}^*}. \tag{50}$$

Further note that

$$\mathbb{E}[A^n] = A_{\mathcal{J}^*,\mathcal{I}^*}$$

and the distribution of A^n is independent of the distribution of $\boldsymbol{x}^n$. Then, it holds that

$$\begin{aligned} \sum_{n=1}^{N} (\boldsymbol{r}_{\mathcal{I}^*})^\top \mathbb{E}\left[\boldsymbol{x}^n_{\mathcal{I}^*}\right] &= \sum_{n=1}^{N} \left((A^\top_{\mathcal{J}^*,\mathcal{I}^*} \boldsymbol{y}^*_{\mathcal{J}^*}\right)^\top \mathbb{E}[\boldsymbol{x}^n_{\mathcal{I}^*}] = \mathbb{E}\left[\sum_{n=1}^{N} \left((A^n)^\top \boldsymbol{y}^*_{\mathcal{J}^*}\right)^\top \boldsymbol{x}^n_{\mathcal{I}^*}\right] \\ &= \mathbb{E}\left[\sum_{n=1}^{N} (\boldsymbol{y}^*)^\top A^n \boldsymbol{x}^n_{\mathcal{I}^*}\right] \end{aligned} \tag{51}$$

From the update rule (8), we have

$$\sum_{n=1}^{N} A^n \boldsymbol{x}^n_{\mathcal{I}^*} = \boldsymbol{c}^1_{\mathcal{J}^*} - \boldsymbol{c}^N_{\mathcal{J}^*}. \tag{52}$$

Plugging (52) back into (51), we get that

$$\sum_{n=1}^{N} (\boldsymbol{r}_{\mathcal{I}^*})^\top \mathbb{E}\left[\boldsymbol{x}^n_{\mathcal{I}^*}\right] = (\boldsymbol{y}^*_{\mathcal{J}^*})^\top \boldsymbol{c}^1_{\mathcal{J}^*} - (\boldsymbol{y}^*_{\mathcal{J}^*})^\top \mathbb{E}\left[\boldsymbol{c}^N_{\mathcal{J}^*}\right].$$

Note that from the strong duality of V^{LP}, we have

$$N \cdot V^{\mathrm{LP}} = (\boldsymbol{y}^*_{\mathcal{J}^*})^\top \boldsymbol{c}^1_{\mathcal{J}^*}.$$

Then, we have that

$$N \cdot V^{\mathrm{LP}} - \sum_{n=1}^{N} (\boldsymbol{r})^\top \mathbb{E}[\boldsymbol{x}^n] \le (\boldsymbol{y}^*_{\mathcal{J}^*})^\top \mathbb{E}\left[\boldsymbol{c}^N_{\mathcal{J}^*}\right]. \tag{53}$$

Our proof is thus completed.

J Proof of Theorem E.2

We now condition on the event that Algorithm 2 has successfully identified the optimal basis $\mathcal{I}^*$ and $\mathcal{J}^*$, which happens with probability at least $1 - \varepsilon$ from Theorem D.1. We consider the stochastic process $\tilde{\boldsymbol{c}}_{(s,a)}(n)$ defined in (22). For a fixed $\nu > 0$ which we specify later, we define a set

$$\mathcal{C} = \{\boldsymbol{c}' \in \mathbb{R}^{|\mathcal{J}^*|} : c'_{(s,a)} \in [c_{(s,a)} - \nu, c_{(s,a)} + \nu], \forall (s,a) \in \mathcal{J}^*\}. \tag{54}$$

It is easy to see that initially, $\tilde{\boldsymbol{c}}_{\mathcal{J}^*}(1) \in \mathcal{C}$. We show that $\tilde{\boldsymbol{c}}_{\mathcal{J}^*}(n)$ behaves well as long as they stay in the region $\mathcal{C}$ for a sufficiently long time. To this end, we define a stopping time

$$\tau = \min_{n \in [N]} \{\tilde{\boldsymbol{c}}_{\mathcal{J}^*}(n) \notin \mathcal{C}\}. \tag{55}$$

Note that in Algorithm 1, to prevent $\boldsymbol{x}^n$ from behaving ill when n is small, we project it to a set that guarantees $\|\boldsymbol{x}^n\|_1 \le C$. We now show in the following lemma that when n is large enough but smaller than the stopping time τ, it is automatically satisfied that $\|\boldsymbol{x}^n\|_1 \le C$.

Lemma J.1. *There exist two constants N_0' and ν_0. When $N_0' \le n \le \tau$, and $\nu \le \nu_0$, it holds that $\|\tilde{\boldsymbol{x}}^n_{\mathcal{I}^*}\|_1 \le C$, where $\tilde{\boldsymbol{x}}^n_{\mathcal{I}^*}$ denotes the solution to the linear equations* (7)*. Specifically, N_0' is given as follows*

$$N_0' = \frac{8d_2^2}{\sigma^2 \cdot C^2} \cdot \log(1/\varepsilon) \tag{56}$$

Also, ν_0 is given as follows

$$\nu_0 := \frac{\sigma \cdot \|\boldsymbol{c}_{\mathcal{J}^*}\|_1 \cdot C}{8d_2 \cdot \|A_{\mathcal{J}^*,\mathcal{I}^*}\|_1}. \tag{57}$$

We set ν to satisfy the condition $\nu \le \nu_0$ with ν_0 satisfies the condition in Lemma J.1. We bound $\mathbb{E}[N-\tau]$ in the following lemma.

Lemma J.2. *Let the stopping time τ be defined in* (55)*. It holds that*

$$\mathbb{E}[N-\tau] \le N_0' + 2d_2 \cdot \exp(-\nu^2/8)$$

where N_0' is given in (56)*, as long as*

$$N \ge N_0' \text{ and } N \ge \frac{8}{\nu^2} \ge \frac{8}{\nu_0^2} = \frac{512 \cdot d_2^2 \cdot \|A_{\mathcal{J}^*,\mathcal{I}^*}\|_1^2}{\sigma^2 \cdot \|\boldsymbol{c}_{\mathcal{J}^*}\|_1^3 \cdot C^2}. \tag{58}$$

Also, for any N' such that $N_0' \le N' \le N$, it holds that

$$P(\tau \le N') \le \frac{d_2 \cdot \nu^2}{4} \cdot \exp\left(-\frac{\nu^2 \cdot (N - N' + 1)}{8}\right). \tag{59}$$

From the definition of the stopping time τ in (55), we know that for each $(s,a) \in \mathcal{J}^*$, it holds

$$c^{\tau-1}_{(s,a)} \in [(N-\tau+1) \cdot (c_{(s,a)} - \nu), (N-\tau+1) \cdot (c_{(s,a)} + \nu)]$$

Thus, we have that

$$|c^N_{(s,a)}| \le |c^{\tau-1}_{(s,a)}| + \left| \sum_{n=\tau}^{N} A^n_{(s,a),:} \cdot \boldsymbol{x}^n_{\mathcal{I}^*} \right| \tag{60}$$

and thus

$$\begin{aligned} \left|\mathbb{E}[c^N_{(s,a)}]\right| &\le (\|\boldsymbol{c}_{\mathcal{J}^*}\|_\infty + \|A_{\mathcal{J}^*,\mathcal{I}^*}\|_\infty \cdot C) \cdot \mathbb{E}[N-\tau] \\ &\le (\|\boldsymbol{c}_{\mathcal{J}^*}\|_\infty + \|A_{\mathcal{J}^*,\mathcal{I}^*}\|_\infty \cdot C) \cdot N_0' + 2(\|\boldsymbol{c}_{\mathcal{J}^*}\|_\infty + \|A_{\mathcal{J}^*,\mathcal{I}^*}\|_\infty \cdot C) \cdot d_2 \cdot \exp(-\nu^2/8). \end{aligned} \tag{61}$$

Our proof is thus completed by plugging in the formulation of N_0' in (56).

J.1 Proof of Lemma J.1

Denote by $\boldsymbol{x}^*$ the optimal solution corresponding to the optimal basis $\mathcal{I}^*$ and $\mathcal{J}^*$. Then, it holds that

$$A_{\mathcal{J}^*,\mathcal{I}^*} \cdot \boldsymbol{x}^*_{\mathcal{I}^*} = \boldsymbol{c}_{\mathcal{J}^*}. \tag{62}$$

We compare $\tilde{\boldsymbol{x}}^n_{\mathcal{I}^*}$ with $\boldsymbol{x}^*_{\mathcal{I}^*}$ when n large enough. Note that when $n \ge N_0$, $\tilde{\boldsymbol{x}}^n$ is the solution to the following linear equations

$$\hat{A}_{\mathcal{J}^*,\mathcal{I}^*}(\mathcal{H}^n) \cdot \tilde{\boldsymbol{x}}^n_{\mathcal{I}^*} = \frac{\boldsymbol{c}^n_{\mathcal{J}^*}}{N-n+1}. \tag{63}$$

When $n \le \tau$, we know that

$$\left| \boldsymbol{c}_{\mathcal{J}^*} - \frac{\boldsymbol{c}^n_{\mathcal{J}^*}}{N-n+1} \right| \le \nu. \tag{64}$$

Moreover, we know that the absolute value of each element of $\hat{A}_{\mathcal{J}^*,\mathcal{I}^*}(\mathcal{H}^n) - A_{\mathcal{J}^*,\mathcal{I}^*}$ is upper bounded by $\mathrm{Rad}(n,\varepsilon)$, given that the following event

$$\mathcal{E} = \left\{ \left| \frac{\gamma}{N} \cdot \sum_{n=1}^{N} \phi_i(s_n) - \phi_i(s) - A_{(s,a),i} \right| \le \mathrm{Rad}(N,\varepsilon), \ \forall (s,a) \in \mathcal{J}^*, \ \forall i \in [d_1] \right\}. \tag{65}$$

is assumed to be satisfied (it holds with probability at least $1 - O(\varepsilon)$ following standard Chernoff bound). We now bound the distance between the solutions to the linear equations (62) and (63). The perturbation of the matrix is denoted as

$$\Delta A = A_{\mathcal{J}^*,\mathcal{I}^*} - \hat{A}_{\mathcal{J}^*,\mathcal{I}^*}(\mathcal{H}^n).$$

Clearly, it holds that

$$\|\Delta A\|_1 \leq \mathrm{Rad}(n, \varepsilon) \cdot d_2. \tag{66}$$

Therefore, as long as

$$\|\Delta A\|_1 \leq \mathrm{Rad}(n, \varepsilon) \cdot d_2 \leq \frac{1}{2\|(A_{\mathcal{J}^*,\mathcal{I}^*})^{-1}\|_1} \leq \frac{1}{2\sigma}, \tag{67}$$

following standard perturbation analysis of linear equations [22], we have that

$$\begin{aligned}
\frac{\|\tilde{\boldsymbol{x}}^n_{\mathcal{I}^*} - \boldsymbol{x}^*_{\mathcal{I}^*}\|_1}{\|\boldsymbol{x}^*_{\mathcal{I}^*}\|_1} &\leq \frac{\kappa(A_{\mathcal{J}^*,\mathcal{I}^*})}{1 - \kappa(A_{\mathcal{J}^*,\mathcal{I}^*}) \cdot \frac{\|\Delta A\|_1}{\|A_{\mathcal{J}^*,\mathcal{I}^*}\|_1}} \cdot \left(\frac{\|\Delta A\|_1}{\|A_{\mathcal{J}^*,\mathcal{I}^*}\|_1} + \frac{d_2 \cdot \nu}{\|\boldsymbol{c}_{\mathcal{J}^*}\|_1} \right) \\
&\leq 2 \cdot \kappa(A_{\mathcal{J}^*,\mathcal{I}^*}) \cdot \left(\frac{\|\Delta A\|_1}{\|A_{\mathcal{J}^*,\mathcal{I}^*}\|_1} + \frac{d_2 \cdot \nu}{\|\boldsymbol{c}_{\mathcal{J}^*}\|_1} \right) \\
&\leq 2 \cdot \kappa(A_{\mathcal{J}^*,\mathcal{I}^*}) \cdot \left(\frac{\|\Delta A\|_1}{\|A_{\mathcal{J}^*,\mathcal{I}^*}\|_1} + \frac{d_2 \cdot \nu}{c_3} \right),
\end{aligned} \tag{68}$$

where we set $c_3 = \|\boldsymbol{c}_{\mathcal{J}^*}\|_1$ and $\kappa(A_{\mathcal{J}^*,\mathcal{I}^*}) = \|A_{\mathcal{J}^*,\mathcal{I}^*}\|_1 \cdot \|(A_{\mathcal{J}^*,\mathcal{I}^*})^{-1}\|_1$ denotes the conditional number of $A_{\mathcal{J}^*,\mathcal{I}^*}$. The last inequality follows from defining the constant c_3 to be a lower bound of $\|\boldsymbol{c}_{\mathcal{J}^*}\|_1$. Further, note that $\|\boldsymbol{x}^*_{\mathcal{I}^*}\|_1 \leq \frac{C}{2}$. Therefore, in order to satisfy the condition $\|\tilde{\boldsymbol{x}}^n_{\mathcal{I}^*}\|_1 \leq C$, we only need the right hand side of (68) to be upper bounded by $\frac{C}{2}$. Clearly, as long as n satisfies the condition (67) and the following condition

$$2 \cdot \kappa(A_{\mathcal{J}^*,\mathcal{I}^*}) \cdot \frac{\|\Delta A\|_1}{\|A_{\mathcal{J}^*,\mathcal{I}^*}\|_1} \leq 2 \cdot \frac{\mathrm{Rad}(n, \varepsilon) \cdot d_2}{\sigma} \leq \frac{C}{4}, \tag{69}$$

we only need to select a ν such that

$$2 \cdot \kappa(A_{\mathcal{J}^*,\mathcal{I}^*}) \cdot \frac{d_2 \cdot \nu}{c_3} \leq \frac{C}{4}. \tag{70}$$

Combining (67) and (69), we know that n needs to satisfy the following condition:

$$n \geq N'_0 := 8 \cdot \frac{d_2^2}{\sigma^2 \cdot C^2} \cdot \log(1/\varepsilon). \tag{71}$$

Also, ν is selected to satisfy the following condiont

$$\nu \leq \nu_0 := \frac{\sigma \cdot c_3 \cdot C}{8d_2 \cdot \|A_{\mathcal{J}^*,\mathcal{I}^*}\|_1}. \tag{72}$$

Our proof is thus completed.

J.2 Proof of Lemma J.2

Now we fix a $(s, a) \in \mathcal{J}^*$. We specify a $\bar{N}_0 = N'_0 = 8 \cdot \frac{d_2^2}{\sigma^2 \cdot C^2} \cdot \log(1/\varepsilon)$. For any $\bar{N}_0 \leq N' \leq N$, it holds that

$$\tilde{c}_{(s,a)}(N') - \tilde{c}_{(s,a)}(\bar{N}_0) = \sum_{n=\bar{N}_0}^{N'-1} (\tilde{c}_{(s,a)}(n+1) - \tilde{c}_{(s,a)}(n)).$$

We define $\xi_{(s,a)}(n) = \tilde{c}_{(s,a)}(n+1) - \tilde{c}_{(s,a)}(n)$. Then, we have

$$\tilde{c}_{(s,a)}(N') - \tilde{c}_{(s,a)}(\bar{N}_0) = \sum_{n=\bar{N}_0}^{N'-1} (\xi_{(s,a)}(n) - \mathbb{E}[\xi_{(s,a)}(n)|\mathcal{H}^n]) + \sum_{n=\bar{N}_0}^{N'-1} \mathbb{E}[\xi_{(s,a)}(n)|\mathcal{H}^n].$$

where $\mathcal{H}^n$ denotes the filtration of information up to step n. Note that due to the update in (23), we have

$$\xi_{(s,a)}(n) = \frac{\tilde{c}_{(s,a)}(n) - A^n \cdot \boldsymbol{x}^n_{\mathcal{I}^*}}{N-n-1}.$$

Then, it holds that

$$|\xi_{(s,a)}(n) - \mathbb{E}[\xi_{(s,a)}(n)|\mathcal{H}^n]| \le \frac{c_4}{N-n+1} \tag{73}$$

for some constant $c_4 > 0$, where the inequality follows from the fact that the value of $\tilde{c}_k(n)$ is deterministic given the filtration $\mathcal{H}^n$ which falls into the region $\mathcal{C}$ and $\|\boldsymbol{x}^n\|_1 \le C$ for any n. Note that

$$\{\xi_{(s,a)}(n) - \mathbb{E}[\xi_{(s,a)}(n)|\mathcal{H}^n]\}_{\forall n=\bar{N}_0,\dots,N'}$$

forms a martingale difference sequence. Following Hoeffding's inequality, for any $N'' \le N'$ and any $b > 0$, it holds that

$$\begin{aligned}
P\left(\left| \sum_{n=\bar{N}_0}^{N''} (\xi_{(s,a)}(n) - \mathbb{E}[\xi_{(s,a)}(n)|\mathcal{H}^n]) \right| \ge b \right) &\le 2\exp\left(-\frac{b^2}{2 \cdot \sum_{n=\bar{N}_0}^{N''} 1/(N-n+1)^2} \right) \\
&\le 2\exp\left(-\frac{b^2 \cdot (N - N'' + 1)}{2} \right).
\end{aligned}$$

Therefore, we have that

$$\begin{aligned}
&P\left(\left| \sum_{n=\bar{N}_0}^{N''} (\xi_{(s,a)}(n) - \mathbb{E}[\xi_{(s,a)}(n)|\mathcal{H}^n]) \right| \ge b \text{ for some } \bar{N}_0 \le N'' \le N' \right) \\
&\le \sum_{N''=\bar{N}_0}^{N'} 2\exp\left(-\frac{b^2 \cdot (N - N'' + 1)}{2} \right) \le b^2 \cdot \exp\left(-\frac{b^2 \cdot (N - N' + 1)}{2} \right)
\end{aligned} \tag{74}$$

holds for any $b > 0$.

We now bound the probability that $\tau > N'$ for one particular N' such that $\bar{N}_0 \le N' \le N$. Suppose that $N' \le \tau$, then, from Lemma J.1, for each $n \le N'$, we know that $\|\tilde{\boldsymbol{x}}^n\|_1 \le C$ and therefore $\boldsymbol{x}^n = \tilde{\boldsymbol{x}}^n$ as the solution to (7). We have

$$\tilde{c}_{(s,a)}(n) = \hat{A}_{(s,a),\mathcal{I}^*}(\mathcal{H}^n) \cdot \boldsymbol{x}^n_{\mathcal{I}^*}.$$

It holds that

$$\left|\mathbb{E}[\xi_{(s,a)}(n)|\mathcal{H}^n]\right| \le \frac{1}{N-n+1} \cdot \|\hat{A}_{(s,a),\mathcal{I}^*}(\mathcal{H}^n) - A^n_{(s,a),:}\|_1 \cdot \|\boldsymbol{x}^n_{\mathcal{I}^*}\| \le \frac{d_2 \cdot C \cdot \mathrm{Rad}(n, \varepsilon)}{N-n+1}. \tag{75}$$

Then, we know that

$$\begin{aligned}
\frac{\sum_{n=\bar{N}_0}^{N'-1} \left|\mathbb{E}[\xi_{(s,a)}(n)|\mathcal{H}^n]\right|}{d_2 \cdot C} &\le \sqrt{\frac{\log(2/\varepsilon)}{2}} \cdot \sum_{n=\bar{N}_0}^{N'-1} \frac{1}{\sqrt{n} \cdot (N-n)} \\
&\le \sqrt{\frac{\log(2/\varepsilon)}{2}} \cdot \sqrt{N'-1} \cdot \sum_{n=\bar{N}_0}^{N'-1} \frac{1}{n \cdot (N-n)} \\
&= \sqrt{\frac{\log(2/\varepsilon)}{2}} \cdot \frac{\sqrt{N'-1}}{N} \cdot \sum_{n=\bar{N}_0}^{N'-1} \left(\frac{1}{n} + \frac{1}{N-n} \right) \\
&\le \sqrt{2\log(2/\varepsilon)} \cdot \frac{\sqrt{N'-1}}{N} \cdot \log(N) \le \frac{\sqrt{2\log(2/\varepsilon)}}{\sqrt{N}} \cdot \log(N) \\
&\le \frac{\nu}{2}
\end{aligned} \tag{76}$$

for a N large enough such that

$$N \ge \frac{8}{\nu^2} \ge \frac{8}{\nu_0^2} = \frac{512 \cdot d_2^2 \cdot \|A_{\mathcal{J}^*,\mathcal{I}^*}\|_1^2}{\sigma^2 \cdot \|\boldsymbol{c}_{\mathcal{J}^*}\|_1^3 \cdot C^2} \tag{77}$$

Combining (76) and (74) with $b = \nu/2$, and apply a union bound over all $(s,a) \in \mathcal{J}^*$, we know that

$$P(\tau \le N') \le \frac{d_2 \cdot \nu^2}{4} \cdot \exp\left(-\frac{\nu^2 \cdot (N - N' + 1)}{8}\right). \tag{78}$$

Therefore, we know that

$$\mathbb{E}[N - \tau] = \sum_{N'=1}^{N} P(\tau \le N') \le \bar{N}_0 + \sum_{N'=\bar{N}_0}^{N} P(\tau \le N') \le \bar{N}_0 + 2d_2 \cdot \exp(-\nu^2/8)$$

which completes our proof.

K Proof of Theorem E.3

We first consider the other constraints $(s,a) \in \mathcal{J}^{*c}$, where $\mathcal{J}^{*c}$ denotes the complementary set of $\mathcal{J}$ in $\mathcal{K}$. Note that following the definition of $\boldsymbol{c}^n$, we have the following relationship

$$A_{\mathcal{J}^*,\mathcal{I}^*} \cdot \left(\sum_{n=1}^{N} \mathbb{E}[\boldsymbol{x}^n_{\mathcal{I}^*}]\right) = \boldsymbol{c}^1_{\mathcal{J}^*} - \mathbb{E}\left[\boldsymbol{c}^N_{\mathcal{J}^*}\right]. \tag{79}$$

Also, from the bindingness of $\boldsymbol{x}^*$ regarding the optimal basis $\mathcal{I}^*$ and $\mathcal{J}^*$, we have

$$N \cdot A_{\mathcal{J}^*,\mathcal{I}^*} \cdot \boldsymbol{x}^*_{\mathcal{I}^*} = \boldsymbol{c}^1_{\mathcal{J}^*}. \tag{80}$$

Therefore, it holds that

$$\sum_{n=1}^{N} \mathbb{E}\left[\boldsymbol{x}^n_{\mathcal{I}^*}\right] = N \cdot \boldsymbol{x}^*_{\mathcal{I}^*} - \left(A_{\mathcal{J}^*,\mathcal{I}^*}\right)^{-1} \cdot \mathbb{E}\left[\boldsymbol{c}^N_{\mathcal{J}^*}\right], \tag{81}$$

and

$$\left\|\sum_{n=1}^{N} \mathbb{E}\left[\boldsymbol{x}^n_{\mathcal{I}^{*c}}\right]\right\|_1 = 0 \tag{82}$$

following the definition of $\boldsymbol{x}^n$. Finally, for any $(s,a) \in \mathcal{J}^{*c}$, we have

$$\begin{aligned} A_{(s,a),:} \cdot \left(\sum_{n=1}^{N} \mathbb{E}\left[\boldsymbol{x}^n\right]\right) &= A_{(s,a),:} \cdot N \cdot \boldsymbol{x}^* + A_{(s,a),:} \cdot \left(\sum_{n=1}^{N} \left(\mathbb{E}\left[\boldsymbol{x}^n\right] - \boldsymbol{x}^*\right)\right) \\ &= A_{(s,a),:} \cdot N \cdot \boldsymbol{x}^* - A_{(s,a),:} \cdot \left[\left(A_{\mathcal{J}^*,\mathcal{I}^*}\right)^{-1} \cdot \mathbb{E}\left[\boldsymbol{c}^N_{\mathcal{J}^*}\right]; \mathbb{E}\left[\boldsymbol{x}^n_{\mathcal{I}^{*c}} - \boldsymbol{x}^*_{\mathcal{I}^{*c}}\right]\right] \\ &= N \cdot A_{(s,a),\mathcal{I}^*} \cdot \boldsymbol{x}^*_{\mathcal{I}^*} - A_{(s,a),\mathcal{I}^*} \cdot \left(A_{\mathcal{J}^*,\mathcal{I}^*}\right)^{-1} \cdot \mathbb{E}\left[\boldsymbol{c}^N_{\mathcal{J}^*}\right]. \end{aligned} \tag{83}$$

Further from the feasibility of $\boldsymbol{x}^*$, we know that

$$N \cdot c_{(s,a)} \ge N \cdot A_{(s,a),:} \cdot \boldsymbol{x}^* = N \cdot A_{(s,a),\mathcal{I}^*} \cdot \boldsymbol{x}^*_{\mathcal{I}^*}.$$

Therefore, for any $(s,a) \in \mathcal{J}^{*c}$, it holds that

$$\begin{aligned} &\left|N \cdot c_{(s,a)} - A_{(s,a),:} \cdot \left(\sum_{n=1}^{N} \mathbb{E}\left[\boldsymbol{x}^n\right]\right)\right| \\ \le &\left|A_{(s,a),\mathcal{I}^*} \cdot \left(A_{\mathcal{J}^*,\mathcal{I}^*}\right)^{-1} \cdot \mathbb{E}\left[\boldsymbol{c}^N_{\mathcal{J}^*}\right]\right| \\ \le &\frac{c_4}{\sigma} \cdot (\|\boldsymbol{c}_{\mathcal{J}^*}\|_\infty + \|A_{\mathcal{J}^*,\mathcal{I}^*}\|_\infty \cdot C) \cdot N'_0 + 2(\|\boldsymbol{c}_{\mathcal{J}^*}\|_\infty + \|A_{\mathcal{J}^*,\mathcal{I}^*}\|_\infty \cdot C) \cdot d_2 \cdot \exp(-\nu^2/8), \end{aligned} \tag{84}$$

where the final bound on $\left|\mathbb{E}[\boldsymbol{c}^N_{\mathcal{J}^*}]\right|$ follows from (61).

We finally convert the regret bound established in (25), (26), and (84) into the sample complexity bound. Let ε satisfy

$$\varepsilon = O\left(\frac{d_2^2 \cdot (1 + \|A_{\mathcal{J}^*,\mathcal{I}^*}\|_\infty)}{\sigma^2} \cdot \frac{\log(N)}{N}\right).$$

We know that

$$N = O\left(\frac{d_2^2 \cdot (1 + \|A_{\mathcal{J}^*,\mathcal{I}^*}\|_\infty)}{\sigma^2} \cdot \frac{\log(1/\varepsilon)}{\varepsilon}\right).$$

We further combine with the number of samples that are required for Algorithm 2. To be specific, in Theorem D.1, we have shown that the number of samples needed for Algorithm 2 can be bounded as

$$O\left(K \cdot \frac{\log(K/\epsilon)}{\Delta^2}\right),$$

where K refers to the number of constraints in LP (3). Therefore, we know that the total number of required samples can be bounded as

$$O\left(K \cdot \frac{\log(K/\varepsilon)}{\Delta^2} + \frac{d_2^2(1 + \|A_{\mathcal{J}^*,\mathcal{I}^*}\|_\infty)}{\sigma^2} \cdot \frac{\log(1/\varepsilon)}{\varepsilon}\right).$$

Our proof is thus completed.