Fast Bandit-based Policy Adaptation in Diverse Environments

Ziyi Zhang and Guannan Qu and Yorie Nakahira

***Abstract*— Autonomous systems must have the ability to quickly adapt to various situations. However, adaptation methods often require strong assumptions about system structures, environmental homogeneity, and multiple rollouts. In this work, we integrate multi-armed bandit and model-based RL to design a fast adaptation algorithm on a single trajectory. Our approach achieves sublinear regret of $O(\sqrt{T})$, and the performance guarantee does not require homogeneity of the environment. This regret bound is achieved using a novel prediction error metric that is minimized in the ground-truth MDP. To the best of our knowledge, all existing results with provable guarantees depend on the Bregman divergence between the optimal policies among the MDP's. We show by simulation that our algorithm performs well in puzzle navigation and quadcopter path-tracking.**

I. INTRODUCTION

Rapid adaptation to environmental changes is essential in planning and control problems. Practical problems, such as autonomous vehicles [1], power systems [2], and robotics, may require a given system to rapidly adapt to an unknown environment, such as a sudden change in road conditions or power demand, respectively. Those changes in environments may require significant changes in policies than what is currently deployed. Furthermore, these problems often require the adaptation to be done on a single trajectory, as multiple rollouts might be costly and impractical.

Motivated by such needs, many adaptive control methods have been developed [3]. Those methods usually adjust control actions in real-time by augmenting the state space with conditional distribution for uncertain system parameters. Some identify system parameters [4] or analyze the relationship between the error in open-loop system identification and the error in closed-loop system control/prediction [5], [6]. Most adaptive control techniques require certain system structures to ensure convergence and do not work in a general MDP setting and focus mainly on stabilization. On the other hand, online switching control has gained much success [7], [8] recently. Online switching control often works in more general systems and can handle non-continuous uncertainty [9] and hybrid systems [10]. However, most works in online switching control also focus on stabilization and can not optimize on a user-defined cost.

Learning-to-learn and meta-learning have also been developed to perform fast adaption [11]. Some use gradient-based approaches, which improves learning performance by tuning parameters of a gradient-based update, either on an offline dataset [12] or during online training on a single task [13]. While some methods guarantee local convergence or stepwise improvement [14], [15], it is also generally difficult to achieve global convergence/regret. One notable work with a global convergence guarantee is [16], which offers a sublinear convergence guarantee on a convex objective function. The regret bound depends linearly on the Bregman divergence among the optimal policies, which would be large if the optimal policies are far apart from each other.

Other approaches directly use the difference between different Markov decision processes (MDPs), such as the Q functions or the transitional probability, to identify the ground-truth MDP and optimize the task during online training [17], [18]. For instance, [17] introduced an algorithm for meta-learning based on hierarchical RL [17]. Their work offers an algorithm that takes advantage of the latent hierarchical structure of MDPs. In a tabular setting, the latent hierarchical structure requires the MDPs to have identical dynamics except for a finite number of state-action pairs. Their work provides a bound on the number of samples needed to determine the set of exits in the ground-truth MDP. However, their work also requires multiple rollouts and needs the MDP's to have similar control tasks.

A. Contribution

In this work, we study the problem of quickly determining the optimal policy on a single trajectory in diverse environments.

- Our work proposes an algorithm for online policy adaptation in diverse environments on a single trajectory. By using a carefully designed prediction error, the proposed algorithm takes advantage of the differences among reward functions and transitional probabilities of the MDP's and quickly identifies the ground-truth MDP and is not confined to stabilization problems.
- Our algorithm achieves a provable sublinear regret with respect to the time horizon. Critically, our regret bound does not depend on the Bregman divergence of the optimal policies and the differences among the dynamics of the MDPs.
- We show by simulation that the proposed algorithm performs well in both navigation problems and quadcopter path tracking. In the puzzle navigation problem, the proposed algorithm generates very small regret. In the quadcopter path-tracking problem, compared to naively using the EXP3 algorithm on cost or reward as in previous online switching control papers [19], the proposed algorithm performs significantly better when minimizing the prediction error we designed.

Ziyi Zhang, Guannan Qu, Yorie Nakahira are with Electrical and Computer Engineering, Carnegie Mellon University, 5000 Forbes Ave, Pittsburgh, PA 15213, USA {ziyizhan, gqu, ynakahir}@andrew.cmu.edu

B. Related Work

Online switching control has enjoyed a long history of study [7], [8]. There are two major types of switching rules: model-based switching [7] and performance-based switching [20]. Our work draws inspiration from [19], which used EXP3 to choose the policy to minimize the cost of the actions and guarantee finite-gain stabilization and a sublinear regret. Most of the aforementioned works are primarily concerned with stabilization, and when trying to optimize cost or reward, simply run a bandit algorithm on the observed reward, which might not take full advantage of the difference of system dynamics in a typical MDP setting. Compared to their work, while offering a similar regret guarantee, our algorithm generalizes to classical MDP settings and, therefore, can be applied to more settings than stabilization by utilizing both reward and system dynamics.

Meta Learning is a relatively new field of study [11], [14] and have been studied from different perspectives. For instance, context-based offline meta-learning improves upon previous works in offline RL and offers a new meta-learning framework for offline RL [12]. Another work offers meta-learning from an optimization and meta-gradient perspective by minimizing the distance to a bootstrapped target under a chosen (pseudo)metric, which guarantees local improvement of meta-parameters [13]. Most work in meta-learning offers excellent simulation performance but lacks provable guarantees [21], [22]. Compared to their work, our work learns on a single trajectory and works better when the MDPs are significantly different.

II. Problem Formulation

We start by introducing the discrete-time finite time horizon dynamic system, characterized by the tuple $(\mathcal{M}, \mathcal{S}, \mathcal{A}, P, r, H, s_0)$. Here $\mathcal{M}$ is the space of MDP. $\mathcal{S}, \mathcal{A}$ denote the discrete state space and action space, respectively. $P = \{P^m(\cdot|s,a)\}_{(s,a)\in\mathcal{S}\times\mathcal{A}}$ is the collection of transition probability measures indexed by the state-action pair (s,a) and MDP m. The function $r = \{r_h^m(s,a)\}_{(s,a)\in\mathcal{S}\times\mathcal{A}}$ is the expected instantaneous reward, where $r_h^m(s,a)$ is the deterministic reward taking value in $[0,1]$ incurred by taking action a at state s in MDP m at time step h. Lastly, H is the time horizon, and $s_0 \in \mathcal{S}$ is the initial state. Our proof also easily generalizes to a stationary distribution μ_0 of initial states.

Before time step 0, a ground-truth MDP m^* unknown to the agent is fixed throughout the time horizon. The agent has previously interacted with all potential MDP's in $\mathcal{M}$ and obtained controller $\pi^m : \mathcal{S} \times h \to \mathcal{A}$ with control function in the form of $\pi_h^m(s) = a, h \in [H], s \in \mathcal{S}, a \in \mathcal{A}$ for all $m \in \mathcal{M}$. We also assume that, during learning, the agent has estimated the reward function r^m and transitional probability P^m for all $m \in \mathcal{M}$, which is common to model-based RL algorithms [23]. At each time step h, the agent starts at state s_h, takes action a_h and observes reward $r^{m^*}_{\text{observed},h}$, which can be decomposed as follows:

$$r^{m^*}_{\text{observed},h}(s_h,a_h) := r_h^{m^*}(s_h,a_h) + w_h,$$

where w_h is the noise in the reward when taking action a at state s in MDP m at time step h. Here, we assume that $\mathbb{E}[w_h] = 0$, so $\mathbb{E}[r^{m^*}_{\text{observed},h}] = \mathbb{E}[r_h^{m^*}(s_h,a_h)]$.

The goal of the agent is to determine the MDP it is operating on by interacting with it and minimize the regret defined as

$$\mathcal{R}_H = C^{m^*}(\pi^{m^*}) - \mathbb{E}\left[\sum_{h=0}^{H-1} r_h^{m^*}(s_h,a_h)\right], \qquad (1)$$

where the expectation is taken over the trajectory taken by the agent $\{s_0,\dots,s_{H-1},a_0,\dots,a_{H-1}\}$, and $C^{m^*}(\pi) = \mathbb{E}^{\pi,m^*}\left[\sum_{h=0}^{H-1} r_h^{m^*}(s_h,a_h)\right]$ denotes the expected reward when operating under policy π.

III. Notation

In this section, we introduce the notations that are used in the rest of this paper. We denote that $[H] := \{0,1,\dots,H\}$. We represent the positive real numbers as $\mathbb{R}_+ := \{x \in \mathbb{R} : x > 0\}$ denotes the positive real space. Let $\mathbb{E}_{\pi,m}[\cdot]$ denote the expectation taken conditioned on policy π and MDP m. We use $\mathbb{P}(\cdot) \to [0,1]$ to denote the probability of an event.

IV. Algorithm Design

Instead of optimizing the agent's performance in a greedy algorithm, we design a prediction error that is minimized when the algorithm makes the right prediction. Therefore, our method is able to identify the ground-truth MDP with a bandit algorithm and achieve optimal performance.

In order for a bandit-based algorithm to work in a classical MDP setting, we need an estimation that depends on both probability distributions and rewards of the MDP's and is minimized when the expected behavior of the chosen MDP is in accordance with the system dynamics observed by the learner. More formally, we define

Definition 1: At time step h, given s_h,a_h, and the reward function r_h^m of MDP m and the observed reward $r_{\text{observed},h}(s_h,a_h)$, the observation error at time step h is defined as

$$\tilde{\epsilon}_h^m := (r^{m^*}_{\text{observed},h}(s_h,a_h) - r_h^m(s_h,a_h))^2 + (1 - \rho^m(s_{h+1}|s_h,a_h)), \qquad (2)$$

where $\rho^m(s_{h+1}|s_h,a_h) := \frac{P^m(s_{h+1}|s_h,a_h)}{\left\|P^m(\cdot|s_h,a_h)\right\|_2} \in [0,1]$ is the normalized transition probability of MDP m.

By definition, $\tilde{\epsilon}_h^m$ is the summation of a quadratic loss function on the error in reward and a one-step estimation of the transitional probability. This quantity $\tilde{\epsilon}_h^m$ plays an important role in the algorithm (line 7 in Algorithm 1) and we later show in Lemma 1 that $\mathbb{E}[\tilde{\epsilon}_h^m]$ is minimized when the algorithm makes the correct prediction.

Given the above definition, the algorithm pseudo is provided in Algorithm 1. We explain the steps of the algorithm in detail below.

After initialization, at each time step h, we randomly select MDP $m_h \in \mathcal{M}$ from the distribution p_h and obtain action a_h from the policy π^{m_h}. The agent execute a_h and obtain $r^{m^*}_{\text{observed},h}$ and s_{h+1} (Line 4-6 in Algorithm 1). The algorithm then computes the observation error $\tilde{\epsilon}_h^m$ for all $m \in \mathcal{M}$ (Line

Algorithm 1 EXP3 for Meta Learning

1: Fix ground-truth MDP $m^* \in \mathcal{M}$
2: Initialize estimated gap $S_0^m = 0$, $p_0(m) = \frac{1}{|\mathcal{M}|}$ for all $m \in \mathcal{M}$.
3: **for** $h = 1, \ldots, H$ **do**
4: Sample m_h from p_h.
5: Update s_{h+1}, a_h with π^{m_h} under the ground-truth MDP m^*.
6: Update $r^{m^*}_{\text{observed},h}(s_h, a_h)$.
7: $\tilde{\epsilon}_h^m \leftarrow \left(r^{m^*}_{\text{observed},h}(s_h, a_h) - r_h^m(s_h, a_h)\right)^2 + \left(1 - \frac{P^m(s_{h+1}|s_h,a_h)}{\left\|P^m(\cdot|s_h,a_h)\right\|_2}\right)$
8: **for** $m \in \mathcal{M}$ **do**
9: $S_h^m = S_{h-1}^m + \tilde{\epsilon}_h^m$
10: **end for**
11: **for** $m \in \mathcal{M}$ **do**
12: $p_{h+1}(m) \propto \exp\left(-\eta S_h^m\right)$
13: **end for**
14: **end for**

7 in Algorithm 1). We then update the cumulative error S_h^m for all $m \in \mathcal{M}$ and update p_h (Line 8-13 in Algorithm 1).

The proposed algorithm achieves the convergence guarantee by gradually becoming more likely to select the policy that minimizes $\tilde{\epsilon}_h^m$ in expectation, which is also the optimal controller. The key technical approach of this algorithm is that the best controller π^m for the ground-truth MDP is the controller that minimizes the expected observation error. We will prove this statement in Lemma 1 in Section VI. Therefore, over time, $\mathbb{E}[S_h^m]$ will be minimized when $m = m^*$.

V. Performance Guarantee

In order to achieve a bound in theoretical guarantee, we need the following assumption on the relative optimality of controllers:

Assumption 1: For any two MDP's $m, m^* \in \mathcal{M}$,

$$V_0^{\pi^m, m^*}(s_0) \leq V_0^{\pi^{m^*}, m^*}(s_0)$$

where

$$V_h^{\pi,m}(s) = \mathbb{E}_{\pi,m}\left[\sum_{h'=h}^{H} r_{h'}^m(s_{h'}, a_{h'}) \middle| s_h = s\right], \quad (3)$$

is the value function.

Intuitively, the above assumption states that the control policy we learned for each MDP m is relatively optimal when operating on m compared to any controller learned from other MDPs.

We also need an assumption on a bound on the reward function, which can be easily generalized to settings with sub-Gaussian noise.

Assumption 2: There exists constant C such that noise $|w_h| < C$ for all $h \in [H]$ almost surely. Moreover, and w_h is i.i.d for all $h \in [H]$, independent from the other randomness in the algorithm, and $\mathbb{E}[w_h] = 0$.

In order to bound the regret generated by each incorrectly predicted m_h, we need the following bound on the advantage function:

Assumption 3: There exists $d \in \mathbb{R}_+$, such that the advantage function $\mathcal{A}$ satisfy that

$$\mathcal{A}_h^{\pi^{m^*}, m^*}(s_h, a_h) := Q_h^{\pi^{m^*}, m^*}(s_h, a_h) - V_h^{\pi^{m^*}, m^*}(s_h) \geq -d \quad (4)$$

for all $(s_h, a_h, h) \in \mathcal{S} \times \mathcal{A} \times [H]$, where

$$Q_h^{\pi,m}(s, a) = \mathbb{E}_{\pi,m}\left[\sum_{h'=h}^{H} r_h^m(s_{h'}, a_{h'}) \middle| s_h = s, a_h = a\right], \quad (5)$$

is the Q function. We may omit π if we assume the agent is operating on the optimal policy.

The above assumption is reasonable in a number of scenarios. For example, in a maze system with a stationary reward, the object of control can move up and down, left and right, which, when paired, has a 0 net effect. Since the reward is upper bounded by 1 at each time step, we have $V_h^{\pi^{m^*}, m^*}(s_h) \leq V_{h+2}^{\pi^{m^*}, m^*}(s_h) + 2$, because if the agent makes a wrong move, it can immediately move back and lose at most 2 points in cumulative reward. Therefore, the MDP would satisfy

$$\begin{aligned}\mathcal{A}_h^{\pi^{m^*}, m^*}(s_h, a_h) &= Q_h^{\pi^{m^*}, m^*}(s_h, a_h) - V_h^{\pi^{m^*}, m^*}(s_h) \\ &\geq Q_h^{\pi^{m^*}, m^*}(s_h, a_h) - V_{h+2}^{\pi^{m^*}, m^*}(s_h) - 2 \geq -2,\end{aligned}$$

where we used $Q_h^{\pi^{m^*}, m^*}(s_h, a_h) - V_{h+2}^{\pi^{m^*}, m^*}(s_h) \geq 0$, as in the case of maze and puzzle, the agent can always step back to the original square in two steps.

Another example that satisfies the above assumption is when the MDP is irreducible and aperiodic under the optimal policy, which is a common assumption in many papers [24], [25]. Please see Proposition 1 in Appendix D for a detailed explanation.

Lastly, since we are using reward deviation and difference of transitional probability to estimate the accuracy of MDP prediction, we need the following assumption to get a fixed regret bound:

Assumption 4: We assume that at each step the prediction makes a mistake such that $a_h = \pi_h^{m_h}(s_h) \neq \pi_h^{m^*}(s_h)$, the reward or probability transition has a minimal deviation from the ground truth MDP. For all $(s_h, a_h, h) \in \mathcal{S} \times \mathcal{A} \times [H]$, MDP $m \neq m^*$ and policy π,

$$\left| r_h^m(s_h, a_h) - r_h^{m^*}(s_h, a_h) \right|^2 + \left\langle P^{m^*}(\cdot|s_h, a_h), \rho^{m^*}(\cdot|s_h, a_h) - \rho^m(\cdot|s_h, a_h) \right\rangle > c^2, \tag{6}$$

for some constant c.

In particular, if the MDP's are significantly apart, c^2 defined in Assumption 4 would increase, improving the bound in Theorem 1. Assumption 4 is satisfied in many classical control problems, such as quadcopter stabilization, as we will show in Section VII-B. Even for MDP's that do not satisfy Assumption 4, the proposed algorithm works reasonably well, as shown in simulation in Section VII-A.

Under the above assumptions, we have the following sublinear bound for the regret:

Theorem 1: Given Assumption 1, Assumption 2, Assumption 3, and Assumption 4,, the total regret bound of reward is bounded as

$$\mathcal{R}_H \leq \frac{2d}{c^2}(5 + 4C + C^2)\sqrt{H \log |\mathcal{M}|}$$

This bound is sublinear in H and $|\mathcal{M}|$. In contrast to [16], our bound does not depend on the Bregman divergence between policies. Therefore, our approach achieves smaller regret regardless of the MDP's similarity. We also demonstrate this in experiments in Section VII.

VI. Proof Outline

In this section, we briefly introduce the three steps to prove Theorem 1. In step 1, we show that the observation error in (2) is an unbiased estimator of the true regret, and that the expectation of $\tilde{\epsilon}_h^m$ is minimized when $m = m^*$. In step 2, we upper bound the regret in observation error $\tilde{\epsilon}$. In step 3, we transfer the regret in observation error back to regret in $\mathcal{R}_H$. The full proof can be found in the Appendix.

Step 1. First, we show that $\tilde{\epsilon}_h^m$ is an unbiased observation error, and $\tilde{\epsilon}_h^m = 0$ when $m = m^*$.

Lemma 1: The expectation of observation error is minimized with $m = m^*$, i.e.,

$$m^* = \arg\min_{m \in \mathcal{M}} \mathbb{E}[\tilde{\epsilon}_h^m].$$

The full proof of the Lemma 1 can be found in Appendix A. With Lemma 1, we establish that by minimizing the observation error $\tilde{\epsilon}_h^m$, we ensure that the policy we pick is the optimal policy in the given set of controllers. In the next step, we offer a bound on the regret regarding the cumulative observation error.

Step 2. In step 2, we bound the cumulative observation error in $\tilde{\epsilon}$.

$$S_h^m = \sum_{h'=1}^{h} \tilde{\epsilon}_{h'}^m. \tag{7}$$

which we calculated in line 9 of Algorithm 1. Recall m^* is the ground-truth MDP, the expected $\tilde{\epsilon}$-error bound R_H is defined as

$$R_H = \left(\sum_{h=0}^{H} \tilde{\epsilon}_h^{m_h} \right) - S_H^{m^*}. \tag{8}$$

In the following theorem, we upper bound the expected cumulative observation error.

Theorem 2: Given Assumption 1, Assumption 2, Assumption 3, and Assumption 4, for $\eta = \sqrt{\frac{\log |\mathcal{M}|}{(2+C+C^2)H}}$, our adaptation algorithm has an expected $\tilde{\epsilon}$-error regret bound of

$$\mathbb{E}[R_H] \leq 2(5 + 4C + C^2)\sqrt{H \log |\mathcal{M}|}.$$

We obtain the above theorem by following a similar proof structure to the EXP3 method. The advantage of the above bound is that, given Assumption 4, we can simultaneously evaluate all policies regardless of which policy we pick at any time step.

Step 3. In the last step, we convert the bound on R_H to a bound on $\mathcal{R}_H$. We follow two steps. First, we bound the expected number of non-optimal actions from R_H. Then, we use the bound on the non-optimal action to bound the overall regret as shown in Theorem 1.

VII. Experiments

In the following section, we present simulated results of our algorithm in for quadcopter path-tracking. The source code for the simulation can be found in the supplementary material. The quadcopter simulation is developed on top of the source code of [26]. We show that the proposed algorithm does converge to the optimal policy in very short order and generates the corresponding regret.

A. Mud Walk Navigation

We examine how our proposed algorithm performs in a navigation problem. In this problem, the agent is navigating on a 21×21 grid with a pre-defined path $\mathcal{P}^m$. The reward function r is defined as follows:

$$r_h^m(s, a) = \begin{cases} 1 & s = \text{destination} \\ -0.1 & s \in \mathcal{P}^m \\ -10 & \text{otherwise} \end{cases}$$

In other words, the reward function encourages the agent to get to the destination as fast as possible while staying on the path, as there is a large penalty for being off-path. There are two MDP's, the maps of which are shown in Figure 1. The agent has 5 actions (up/down/left/right/stay). In MDP 1, there is a 0.05 chance that the agent will move to one of its four neighboring states with uniform probability; 0.95 chance of moving to a state indicated by the action. Similarly, in MDP 2, there is a 0.1 chance that the agent move to one of its four neighboring states with uniform probability. After 10,000 trials, the reward of taking the optimal policy averages to 432.55, and our algorithm has an average reward of 430.22. That means the meta-learning result arrives 2 steps later than the optimal policy on average.

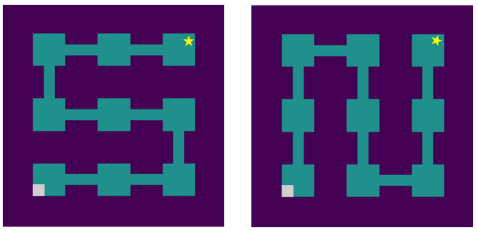

Fig. 1: The two maps of Mud Walk problem are shown above. The agent navigates from the gray square to the star(destination) while trying to stay on the path shown in green.

B. *Quadcopter path tracking*

In this section, we show that our algorithm successfully identifies the weight of a quadcopter and applies a controller to follow a given track with bounded error. In this simulation, we only vary the weight of the quadcopter, but the algorithm easily generalizes to different dimensions and types of drones.

There are ten possible quadcopter, each with mass $m \in \{1\text{kg}, \ldots, 10\text{kg}\}$. The ground truth quadcopter has mass 1 kg. Each quadcopter has a PID controller that tracks a path with 5 waypoints, as shown in Figure 2. Each PID controller takes in the weight of the quadcopter, so the height and velocity of the quadcopter will not be correctly maintained if the wrong controller is chosen. Our algorithm needs to choose between two policies, determine the ground truth MDP, and use the correct controller.

We can see in Table I, that the regret is indeed independent of the distance between the sub-optimal controller and optimal controller, in this case, differentiated by weight. As shown in Figure 3, the tracking errors of the adaptation algorithms are relatively large in the first 2.5 seconds, but

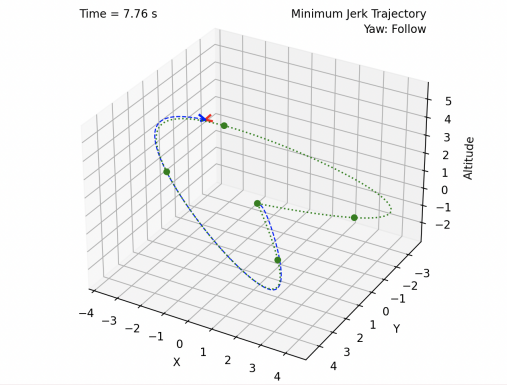

Fig. 2: The above figure shows one trajectory of our algorithm choosing between the ground-truth controller and another PID controller, which assumes the weight of the quadcopter is 5 kg. The green trajectory is the desired trajectory of the ground-truth controller, defined by the 5 waypoints. The blue trajectory is the trajectory of the quadcopter under our algorithm. We see that there is a small gap at the start of the trajectories, but they mostly coincide for the latter parts.

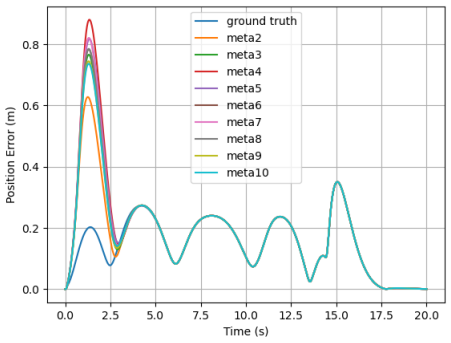

Fig. 3: The above figure shows the deviation from the optimal trajectory when the algorithm takes in different controller pairs.

they quickly diminish and converge to the tracking error of the ground-truth controller. Therefore, although the regrets shown in Table I are relatively high, they would quickly diminish to 0 as the time horizon increases, as demonstrated in Theorem 1 earlier.

VIII. Conclusion

In this work, we proposed an algorithm for fast adaptation to potentially dissimilar environments and offered a provable regret guarantee. We also showed that our regret bound does

Controller regret		
mass assumed by the non-ground-truth PID controller (kg)	proposed algorithm regret as %	EXP3 regret as %
2	20.49 ± 7.17	79.93 ± 9.64
3	27.54 ± 13.53	81.33 ± 9.28
4	33.51 ± 16.63	81.41 ± 9.30
5	30.94 ± 23.50	79.20 ± 9.47
6	28.41 ± 18.87	79.66 ± 8.91
7	30.50 ± 20.21	80.81 ± 10.45
8	28.61 ± 20.01	78.48 ± 8.20
9	26.46 ± 18.05	79.60 ± 9.12
10	26.50 ± 21.72	80.44 ± 8.88

TABLE I: The average regret and standard deviation of the tracking error of the quadcopter.

not depend on the diversity of the MDP's we adapt to. In the future, we intend to extend this work to constraint meta RL and develop an algorithm that can adapt to the optimal MDP with provable guarantees on both regret and constraint violation. Another direction that we aim to study is to integrate our study with existing learning algorithms. In this work, we assumed the reward and transition probability for each MDP is given and accurate, but, in practice, there is usually an error bound on those estimations. It is also important to analyze how those estimation errors would affect our algorithm.

REFERENCES

[1] A. P. Aguiar and J. Hespanha, "Logic-based switching control for trajectory-tracking and path-following of underactuated autonomous vehicles with parametric modeling uncertainty," vol. 4, pp. 3004 – 3010 vol.4, 01 2004.

[2] L. Meng, E. R. Sanseverino, A. Luna, T. Dragicevic, J. C. Vasquez, and J. M. Guerrero, "Microgrid supervisory controllers and energy management systems: A literature review," *Renewable and Sustainable Energy Reviews*, vol. 60, no. C, pp. 1263–1273, 2016.

[3] R. Bellman, *Adaptive Control Processes: A Guided Tour*. Princeton University Press, 1961.

[4] M. Gevers, "Identification for control: From the early achievements to the revival of experiment design*," *European Journal of Control*, vol. 11, no. 4, pp. 335–352, 2005.

[5] U. Forssell and L. Ljung, "Some results on optimal experiment design," *Automatica*, vol. 36, no. 5, pp. 749–756, 2000.

[6] H. Hjalmarsson, M. Gevers, and F. de Bruyne, "For model-based control design, closed-loop identification gives better performance," *Automatica*, vol. 32, no. 12, pp. 1659–1673, 1996.

[7] J. Hespanha, D. Liberzon, and A. Morse, "Overcoming the limitations of adaptive control by means of logic-based switching," *Systems and Control Letters*, vol. 49, pp. 49–65, May 2003.

[8] P. Rosa, J. Shamma, C. Silvestre, and M. Athans, "Stability overlay for adaptive control laws applied to linear time-invariant systems," pp. 1934 – 1939, 07 2009.

[9] L. Liu and X. Yang, "Robust adaptive state constraint control for uncertain switched high-order nonlinear systems," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 10, pp. 8108–8117, 2017.

[10] P. García, J. P. Torreglosa, L. M. Fernández, and F. Jurado, "Optimal energy management system for stand-alone wind turbine/photovoltaic/hydrogen/battery hybrid system with supervisory control based on fuzzy logic," *International Journal of Hydrogen Energy*, vol. 38, no. 33, pp. 14146–14158, 2013.

[11] Y. Duan, J. Schulman, X. Chen, P. L. Bartlett, I. Sutskever, and P. Abbeel, "RI^2: Fast reinforcement learning via slow reinforcement learning," 2016.

[12] L. Li, Y. Huang, and D. Luo, "Improved context-based offline meta-rl with attention and contrastive learning," *CoRR*, vol. abs/2102.10774, 2021.

[13] S. Flennerhag, Y. Schroecker, T. Zahavy, H. van Hasselt, D. Silver, and S. Singh, "Bootstrapped meta-learning," *CoRR*, vol. abs/2109.04504, 2021.

[14] A. Fallah, A. Mokhtari, and A. Ozdaglar, "On the convergence theory of gradient-based model-agnostic meta-learning algorithms," in *AISTATS* (S. Chiappa and R. Calandra, eds.), vol. 108, pp. 1082–1092, PMLR, 2020.

[15] Y. Song, A. Mavalankar, W. Sun, and S. Gao, "Provably efficient model-based policy adaptation," *CoRR*, vol. abs/2006.08051, 2020.

[16] M. Khodak, M. Balcan, and A. Talwalkar, "Adaptive gradient-based meta-learning methods," *CoRR*, vol. abs/1906.02717, 2019.

[17] K. Chua, Q. Lei, and J. D. Lee, "Provable hierarchy-based meta-reinforcement learning," *AISTATS*, 2021.

[18] Y. Song, A. Mavalankar, W. Sun, and S. Gao, "Provably efficient model-based policy adaptation," in *ICML*, vol. 119, pp. 9088–9098, 2020.

[19] Y. Li, J. A. Preiss, N. Li, Y. Lin, A. Wierman, and J. Shamma, "Online switching control with stability and regret guarantees," in *L4DC* (N. Matni, M. Morari, and G. Pappas, eds.), vol. 211, pp. 1138–1151, Mar. 2023.

[20] I. Al-Shyoukh and J. S. Shamma, "Switching supervisory control using calibrated forecasts," *IEEE Transactions on Automatic Control*, vol. 54, no. 4, pp. 705–716, 2009.

[21] P. Mirowski, R. Pascanu, F. Viola, H. Soyer, A. J. Ballard, A. Banino, M. Denil, R. Goroshin, L. Sifre, K. Kavukcuoglu, D. Kumaran, and R. Hadsell, "Learning to navigate in complex environments," *CoRR*, vol. abs/1611.03673, 2016.

[22] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *ICML*, vol. 70, p. 1126–1135, 2017.

[23] G. Li, Y. Wei, Y. Chi, Y. Gu, and Y. Chen, "Breaking the sample size barrier in model-based reinforcement learning with a generative model," in *NeurIPS*, (Red Hook, NY, USA), 2020.

[24] Y. Wu, W. Zhang, P. Xu, and Q. Gu, "A finite-time analysis of two time-scale actor-critic methods," in *NeurIPS*, (Red Hook, NY, USA), Curran Associates Inc., 2020.

[25] V. Konda and J. Tsitsiklis, "Actor-critic algorithms," *Society for Industrial and Applied Mathematics*, vol. 42, 04 2001.

[26] John, "Quadcopter simcon." `https://github.com/bobzwik/Quadcopter_SimCon`, 2023.

APPENDIX

In this section, we prove that the meta-learning regret of our algorithm is indeed sublinear. Our proof has two steps. First, we prove that the regret of the EXP3 algorithm on the observation error $\tilde{\epsilon}$ is sublinear in expectation. In the second step, we prove that a sublinear regret of the EXP3 algorithm leads to a sublinear meta-learning regret $\mathcal{R}_H$.

A. Proof of Lemma 1

For simplicity of proof, we first introduce two new notations:

$$\bar{\epsilon}_h^m := \mathbb{E}\left[\tilde{\epsilon}_h^m | s_h, p_h\right] \tag{9}$$

$$\epsilon_h^m := \mathbb{E}\Big[\Big(r_h^m(s_h,a_h) - r_h^{m^*}(s_h,a_h)\Big)^2 + \langle P^{m^*}(\cdot|s_h,a_h), \rho^{m^*}(\cdot|s_h,a_h) - \rho^m(\cdot|s_h,a_h)\rangle | s_h, p_h\Big] \tag{10}$$

We will show that ϵ_h^m is the unbiased observation error and equals to 0 when $m = m^*$ and prove the optimality of π^m that minimizes $\mathbb{E}[\tilde{\epsilon}_h^m]$.

Proof: (Proof of Lemma 1) First, we express the expectation of the observation error by taking expectation of (2) and get

$$\begin{aligned}
\bar{\epsilon}_h^m = \mathbb{E}\left[\tilde{\epsilon}_h^m | s_h, p_h\right] =& \mathbb{E}\Big[\Big(r_h^m(s_h,a_h) - r_h^{m^*}(s_h,a_h) - w_h\Big)^2 + \sum_{s'\in\mathcal{S}} P^{m^*}(s'|s_h,a_h)(1-\rho^m(s'|s_h,a_h))|s_h,p_h\Big] \\
=& \mathbb{E}\Big[\Big(r_h^m(s_h,a_h) - r_h^{m^*}(s_h,a_h)\Big)^2 + w_h^2 + 1 - \sum_{s'\in\mathcal{S}} P^{m^*}(s'|s_h,a_h)\rho^m(s'|s_h,a_h)|s_h,p_h\Big] \\
=& \epsilon_h^m + \mathbb{E}\Big[w_h^2 + \Big(1 - \Big\|P^{m^*}(\cdot|s_h,a_h)\Big\|_2\Big)|s_h,p_h\Big],
\end{aligned} \tag{11}$$

where $\sum_{s'\in\mathcal{S}} P^{m^*}(s'|s_h,a_h)\rho^{m^*}(s'|s_h,a_h)$ is written in inner product form, i.e.

$$\sum_{s'\in\mathcal{S}} P^{m^*}(s'|s_h,a_h)\rho^{m^*}(s'|s_h,a_h) = \langle P^{m^*}(\cdot|s_h,a_h), \rho^{m^*}(\cdot|s_h,a_h)\rangle = \Big\|P^{m^*}(\cdot|s_h,a_h)\Big\|_2.$$

The second term of (11) does not depend on the choice of m, and ϵ_h^m can be further expanded as follows:

$$\begin{aligned}
\epsilon_h^m =& \mathbb{E}\Big[\Big(r_h^m(s_h,a_h) - r_h^{m^*}(s_h,a_h)\Big)^2 + \langle P^{m^*}(\cdot|s_h,a_h), \rho^{m^*}(\cdot|s_h,a_h) - \rho^m(\cdot|s_h,a_h)\rangle|s_h,p_h\Big] \\
=& \mathbb{E}\Big[\sum_{m'\in\mathcal{M}} p_h(m')\Big(\Big(r_h^m(s_h,\pi_h^{m'}(s_h)) - r_h^{m^*}(s_h,\pi_h^{m'}(s_h))\Big)^2 + \langle P^{m^*}(\cdot|s_h,\pi_h^{m'}(s_h)), \rho^{m^*}(\cdot|s_h,\pi_h^{m'}(s_h)) \\
& - \rho^m(\cdot|s_h,\pi_h^{m'}(s_h))\rangle\Big)|s_h,p_h\Big].
\end{aligned} \tag{12}$$

Intuitively, ϵ_h^m is the true observation error, and $\tilde{\epsilon}_h^m$ is the summation of ϵ_h^m and noise from both reward and transition. We see that for each m' term, (12) is minimized with $m = m^*$, as both $\Big(r_h^m(s_h,\pi_h^{m'}(s_h)) - r_h^{m^*}(s_h,\pi_h^{m'}(s_h))\Big)^2$ and $\langle P^{m^*}(\cdot|s_h,\pi_h^{m'}(s_h)), \rho^{m^*}(\cdot|s_h,\pi_h^{m'}(s_h)) - \rho^m(\cdot|s_h,\pi_h^{m'}(s_h))\rangle$ are minimized at $m = m^*$. The former by quadratic function, and latter by Cauchy-Shwarz inequality, as ρ^m is a normalized unit vector, the inner product between P^{m^*} and ρ^m will be maximized when the two vectors are aligned, which is achieved when $m = m^*$. Therefore, by tower property of expectation, we have proved $m^* = \arg\min_{m\in\mathcal{M}} \mathbb{E}[\tilde{\epsilon}_h^m]$. □

B. Proof of Theorem 2

Proof: (Proof of Theorem 2) Define Φ_h as follows:

$$\Phi_h = \frac{1}{\eta}\log\left(\sum_{m\in\mathcal{M}} \exp(-\eta S_h^m)\right). \tag{13}$$

In particular, we have the following conditions:

$$\Phi_0 = \frac{1}{\eta}\log|\mathcal{M}|, \tag{14}$$

$$\Phi_H = \frac{1}{\eta}\log\left(\sum_{m\in\mathcal{M}}\exp(-\eta S_H^m)\right) \geq \frac{1}{\eta}\log\left(\exp(-\eta S_H^{m^*})\right) = -S_H^{m^*}. \tag{15}$$

In (15), we used that $\log$ is monotonically increasing and $\exp(-\eta S_H^m) > 0$ for all $m \in \mathcal{M}$. Furthermore, recall that

$$p_{h+1}(m) = \frac{\exp\left(-\eta S_h^m\right)}{\sum_{m'\in\mathcal{M}}\exp\left(-\eta S_h^{m'}\right)}. \tag{16}$$

We then bound the gap between Φ_h and Φ_{h-1} as follows:

$$\begin{aligned}
\Phi_h - \Phi_{h-1} =& \frac{1}{\eta}\log\left(\sum_{m\in\mathcal{M}} e^{-\eta S_h^m}\right) - \frac{1}{\eta}\log\left(\sum_{m\in\mathcal{M}} e^{-\eta S_{h-1}^m}\right) \\
=& \frac{1}{\eta}\log\frac{\sum_{m\in\mathcal{M}} e^{-\eta S_{h-1}^m} e^{-\eta\tilde{\epsilon}_h^m}}{\sum_{m\in\mathcal{M}} e^{-\eta S_{h-1}^m}} \\
=& \frac{1}{\eta}\log\sum_{m\in\mathcal{M}} p_h(m) e^{-\eta\tilde{\epsilon}_h^m} && (17) \\
\leq& \frac{1}{\eta}\log\sum_{m\in\mathcal{M}} p_h(m)\left(1 - \eta\tilde{\epsilon}_h^m + \frac{\eta^2}{2}\left(\tilde{\epsilon}_h^m\right)^2\right) && (18) \\
\leq& \frac{1}{\eta}\log\left(1 - \eta\sum_{m\in\mathcal{M}} p_h(m)\tilde{\epsilon}_h^m + \frac{\eta^2}{2}\sum_{m\in\mathcal{M}} p_h(m)\left(\tilde{\epsilon}_h^m\right)^2\right) && (19) \\
\leq& -\sum_{m\in\mathcal{M}} p_h(m)\tilde{\epsilon}_h^m + \frac{\eta}{2}\sum_{m\in\mathcal{M}} p_h(m)\left(\tilde{\epsilon}_h^m\right)^2. && (20)
\end{aligned}$$

In (17), we used (16). In (18), we used a variant of Taylor's expansion, i.e. $\forall\alpha \geq 0$, $e^{-\alpha} \leq 1 - \alpha + \alpha^2/2$. In (20), we used $\forall\alpha \geq 0, \log(\alpha) \leq \alpha - 1$.

We take a telescoping sum from $h = 1$ to H,

$$\Phi_H - \Phi_0 = \sum_{h=1}^{H}(\Phi_h - \Phi_{h-1}) \leq -\sum_{h=1}^{H}\sum_{m\in\mathcal{M}} p_h(m)\tilde{\epsilon}_h^m + \frac{\eta}{2}\sum_{h=1}^{H}\sum_{m\in\mathcal{M}} p_h(m)\left(\tilde{\epsilon}_h^m\right)^2.$$

Take the expectation of the above summation leads to

$$\begin{aligned}
\mathbb{E}\left[\Phi_H - \Phi_0\right] \leq& \mathbb{E}\Bigg[-\sum_{h=1}^{H}\sum_{m\in\mathcal{M}} p_h(m)\tilde{\epsilon}_h^m + \frac{\eta}{2}\sum_{h=1}^{H}\sum_{m\in\mathcal{M}} p_h(m)\left(\tilde{\epsilon}_h^m\right)^2\Bigg] \\
\leq& \mathbb{E}\Bigg[-\sum_{h=1}^{H}\mathbb{E}\Bigg[\sum_{m\in\mathcal{M}} p_h(m)\tilde{\epsilon}_h^m | s_h, p_h\Bigg] + (5 + 4C + C^2)^2\eta\sum_{h=1}^{H}\sum_{m\in\mathcal{M}} p_h(m)\Bigg] && (21) \\
\leq& -\sum_{h=1}^{H}\mathbb{E}\Bigg[\mathbb{E}\Bigg[\sum_{m\in\mathcal{M}} p_h(m)\bar{\epsilon}_h^m | s_h, p_h\Bigg]\Bigg] + (5 + 4C + C^2)^2\eta H && (22) \\
=& -\sum_{h=1}^{H}\mathbb{E}\left[\bar{\epsilon}_h^{m_h}\right] + (5 + 4C + C^2)^2\eta H. && (23)
\end{aligned}$$

In (21), we bounded $(\tilde{\epsilon}_h^m)^2$ with Assumption 2 and (2) as follows:

$$(\tilde{\epsilon}_h^m)^2 \leq ((2+C)^2+1)^2 \leq (5+4C+C^2)^2.$$

In (22), we used tower property and (11). We are now ready to bound the regret of the cumulative error of prediction.

$$\begin{aligned}
\mathbb{E}\left[R_H\right] =& \mathbb{E}\left[\sum_h \tilde{\epsilon}_h^{m_h} - S_H^{m^*}\right] && (24)\\
\leq& \mathbb{E}\left[\sum_h \tilde{\epsilon}_h^{m_h} + \Phi_H\right] && (25)\\
\leq& \Phi_0 + (5+4C+C^2)^2\eta H && (26)\\
\leq& \frac{\log|\mathcal{M}|}{\eta} + (5+4C+C^2)^2\eta H. && (27)
\end{aligned}$$

where in (24) we took expectation of (8), in (25) we substituted (15), and in (26) we subtracted (23). By picking $\eta = \frac{1}{5+4C+C^2}\sqrt{\frac{\log|\mathcal{M}|}{H}}$, we get the bound in Theorem 2. □

C. Proof of Theorem 1

Given the regret bound in Thereom 2, we can use this to upper bound the number of mistakes the algorithm makes in the following theorem.

Theorem 3: Let k denote the number of mistakes the algorithm makes, i.e.

$$k := \sum_{h=0}^{H-1} \mathbf{1}(m_h \neq m^*), \tag{28}$$

then

$$\mathbb{E}[k] \leq \frac{\mathbb{E}\left[R_H\right]}{c^2}.$$

Proof: We connect the expected $\tilde{\epsilon}$-error regret R_H to the number of wrong predictions we made in the trajectory.

$$\begin{aligned}
\mathbb{E}\left[R_H\right] =& \mathbb{E}\left[\sum_{h=1}^{H} \tilde{\epsilon}_h^{m_h} - S_H^{m^*}\right]\\
=& \mathbb{E}\left[\sum_{h=1}^{H} \tilde{\epsilon}_h^{m_h} - \sum_{h=1}^{H} \tilde{\epsilon}_h^{m^*}\right]\\
=& \mathbb{E}\left[\sum_{h=0}^{H-1}\left(\epsilon_h^{m_h} + \mathbb{E}\left[w_h^2 + (1 - \left\|P^{m^*}(\cdot|s_h,a_h)\right\|_2)|s_h,p_h\right]\right) - \sum_{h=0}^{H-1}\left(\mathbb{E}\left[w_h^2 + (1 - \left\|P^{m^*}(\cdot|s_h,a_h)\right\|_2)|s_h,p_h\right]\right)\right]\\
=& \mathbb{E}\left[\sum_{h=0}^{H-1} \epsilon_h^{m_h}\right]\\
\geq& \mathbb{E}\left[c^2 \sum_{h=0}^{H-1} \mathbf{1}(m_h \neq m^*)\right]. && (29)
\end{aligned}$$

where we simply substituted (7) and (11) in (8) and cancelled out the noise term in $\tilde{\epsilon}_h^{m_h}$. In (29), we used Assumption 4, which guarantees that $\epsilon_h^m > c^2$ for all m.

From the above, the theorem statement follows naturally. □

We are now ready to prove Theorem 1.

Proof: (Proof of Theorem 1) Therefore, let $\{h_1, \cdots, h_k\}$ represent the time steps Algorithm 1 made the wrong prediction ($m_h \neq m^*$). In addition to the distribution of the trajectory (s_h, a_h) at time h of our proposed approach, we also define a series of other trajectories for each $i \in \{0, \ldots, k\}$: $\{(s_h^i, a_h^i)\}_{h=0,\ldots,H}$ defined as for $h \leq h_i$,$(s_h^i, a_h^i) = (s_h, a_h)$; after h_i, the trajectores are generated following π^{m^*}. Intuitively, $\{(s_h^i, a_h^i)\}_{h=0,\ldots,H}$ makes the same first i mistakes with our trajectory and then follows by the optimal policy. It is clear that $\{(s_h^0, a_h^0)\}_{h=0,\ldots,H}$ is a trajectory generated by the optimal policy, so we can write the expected regret $\mathcal{R}_H$ as follows:

$$\mathcal{R}_H = \mathbb{E}\Big[\sum_{h=0}^{H-1} r_h^{m^*}(s_h^0, a_h^0) - \sum_{h=0}^{H-1} r_h^{m^*}(s_h, a_h)\Big] \tag{30}$$

$$= \mathbb{E}\Big[\sum_{h=h_1}^{H-1} r_h^{m^*}(s_h^0, a_h^0) - \sum_{h=h_1}^{H-1} r_h^{m^*}(s_h, a_h)\Big] \tag{31}$$

$$= \mathbb{E}\Big[\sum_{h=h_1}^{H-1} r_h^{m^*}(s_h^0, a_h^0) - \sum_{h=h_1}^{H-1} r_h^{m^*}(s_h^1, a_h^1) + \sum_{h=h_1}^{H-1} r_h^{m^*}(s_h^1, a_h^1) - \sum_{h=h_1}^{H-1} r_h^{m^*}(s_h, a_h)\Big] \tag{32}$$

$$= \mathbb{E}\left[V_{h_1}^{m^*}(s_{h_1}) - Q_{h_1}^{m^*}(s_{h_1}, a_{h_1})\right] + \mathbb{E}\Big[\sum_{h=h_2}^{H-1} r_h^{m^*}(s_h^1, a_h^1) - \sum_{h=h_2}^{H-1} r_h^{m^*}(s_h, a_h)\Big] \tag{33}$$

$$= \sum_{i=1}^{k} \mathbb{E}\left[V_{h_i}^{m^*}(s_{h_i}) - Q_{h_i}^{m^*}(s_{h_i}, a_{h_i})\right] \tag{34}$$

$$\leq d\frac{\mathbb{E}[R_H]}{c^2}. \tag{35}$$

In (30), we plug in (1) and use the fact that $\{(s_h^0, a_h^0)\}_{h=0,\ldots,H}$ is a trajectory generated by the optimal policy. In (31), we used $(s_h, a_h) = (s_h^0, a_h^0)$ are the same for $h < h_1$. In (32), we added and subtracted $\sum_{h=h_1}^{H-1} r_h^{m^*}(s_h^1, a_h^1)$. In (33), we substituted in (5) and (3). We repeat (30)-(33) to get (34). Using Assumption 3, we get (35). Applying Theorem 2 to (35) gives the result in the theorem statement. □

D. Proof of irreducible, aperiodic MDP has bounded advantage function

In this section, we prove that the advantage function of a MDP is bounded if it is irreducible and aperiodic under the optimal policy.

Proposition 1: Assumption 3 is satisfied if the MDP is irreducible and aperiodic under the optimal policy.

Proof: Let A denote the transitional probability under the optimal policy. Given any $(s_h, a_h) \in \mathcal{S} \times \mathcal{A}$ and the optimal action a_h^*, where $a_h \neq a_h^*$, let ν denote the state distribution of s_{h+1} if the agent takes action a_h^*, and let ν' denote the state distribution of s_{h+1} if the agent take action a_h. Then, there exists $\rho \in (0, 1)$, such that the advantage function can be bounded as follows:

$$\begin{aligned}
&\mathcal{A}_h^{\pi^{m^*}, m^*}(s_h, a_h) \\
=&Q_h^{\pi^{m^*}, m^*}(s_h, a_h) - V_h^{\pi^{m^*}, m^*}(s_h) \\
=&(r_h(s_h, a_h) - r_h(s_h, a_h^*)) + \mathbb{E}_{s\sim\nu}[V_{h+1}(s)] - \mathbb{E}_{s\sim\nu'}[V(s)] && (36) \\
=&(r_h(s_h, a_h) - r_h(s_h, a_h^*)) + \sum_{i=1}^{H-h} \left(\mathbb{E}_{s\sim A^i\nu, s'\sim A^i\nu'}[r_{h+i}(s, \pi^*(s)) - r_{h+i}(s, \pi^*(s))]\right) && (37) \\
\geq& -1 - \sum_{i=1}^{\infty} \left\| A^i\nu - A^i\nu' \right\|_{\text{TV}} && (38)
\end{aligned}$$

$$\geq -1 - \sum_{i=1}^{\infty} D\rho^i \tag{39}$$

$$= -1 - \frac{D\rho}{1-\rho}, \tag{40}$$

where $\|\cdot\|_{\text{TV}}$ denote the total variation distance. In (36) and (37), we use the Bellman's operator. In (38), we use the assumption that reward takes value in $[0, 1]$. Furthermore, we use the irreducible and aperiodic assumption, which leads to the existence of a stationary distribution, so there exists $\rho \in (0, 1)$ such that (39) holds. Therefore, we have proven that the advantage function of an irreducible, aperiodic MDP under optimal policy is indeed bounded. □