# Pausing Policy Learning in Non-stationary Reinforcement Learning

**Hyunin Lee** [1] **Ming Jin** [2] **Javad Lavaei** [1] **Somayeh Sojoudi** [1]

## Abstract

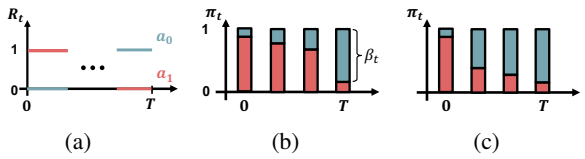
Real-time inference is a challenge of real-world reinforcement learning due to temporal differences in time-varying environments: the system collects data from the past, updates the decision model in the present, and deploys it in the future. We tackle a common belief that continually updating the decision is optimal to minimize the temporal gap. We propose forecasting an online reinforcement learning framework and show that strategically pausing decision updates yields better overall performance by effectively managing aleatoric uncertainty. Theoretically, we compute an optimal ratio between policy update and hold duration, and show that a non-zero policy hold duration provides a sharper upper bound on the dynamic regret. Our experimental evaluations on three different environments also reveal that a non-zero policy hold duration yields higher rewards compared to continuous decision updates.

## 1. Introduction

Real-world reinforcement learning (RL) bridges the gap between the current literature on RL and real-world problems. *Real-time inference*, a key challenge in real-world RL, requires that inference occur in real-time at the control frequency of the system (Dulac-Arnold et al., 2019). For RL deployment in a production system, policy inference must occur in real-time, matching the control frequency of the system. This could range from milliseconds for tasks such as recommendation systems (Covington et al., 2016; Steck et al., 2021) or autonomous vehicle control (Hester & Stone, 2013), to minutes for building control systems (Evans & Gao). This constraint prevents us from speeding up the task beyond real-time to rapidly generate extensive data (Silver et al., 2016; Espeholt et al., 2018) or slowing it down for more computationally intensive approaches (Levine et al., 2019; Schrittwieser et al., 2020). One strategy for real-time action is to employ a multi-threaded architecture, where model learning and planning occur in background threads while actions are returned in real-time (Hester & Stone, 2013; Imanberdiyev et al., 2016; Glavic et al., 2017).

In this paper, we show that intentionally pausing policy learning can lead to better overall performance than continuous policy updating. Our study is based on deriving an analytical solution for the optimal ratio between the pausing and updating phases. Perhaps most importantly, this paper offers the insight that the pausing phase is crucial to handling an aleatoric uncertainty that stems from the environment's intrinsic uncertainty.

This paper begins with a fundamental observation of the real-time inference mechanism based on prediction: the agent forecasts the *future* based on *past* data, and then continually updates decisions in the *present* based on future predictions. This highlights the significance of balancing conservatism or pessimism in decision-making, based on the three types of uncertainties: epistemic, aleatoric, and predictive uncertainties (Gal, 2016). We define conservatism as expecting past trends to continue in the future, and pessimism as anticipating future differences. Although accumulating extensive past data reduces aleatoric uncertainty, and a prediction model with high capacity lessens predictive uncertainty, the frequency of policy updates still remains a key factor due to unknown aleatoric uncertainty in the present.




(a) (b) (c)

*Figure 1.* (a) Non-stationary bandit setting, (b) conservative policy, (c) pessimistic policy

To elucidate the importance of the above problem, consider a recommendation system tasked with optimally suggesting item $x_0$ or $x_1$ to a user whose preference changes over time. This can be framed as a Bernoulli non-stationary bandit setting with a set of two actions $\mathcal{A} = \{a_0, a_1\}$, and a time-dependent policy $\pi_t : \mathcal{A} \to [0, 1]$, where $\pi_t(a_0) = \beta_t$ and $\pi_t(a_1) = 1 - \beta_t$, $0 \leq \beta_t \leq 1$. The rewards of each action,

[1]University of California, Berkeley [2]Virginia Tech. Correspondence to: Hyunin Lee <hyunin@berkeley.edu>.

*Proceedings of the 41st International Conference on Machine Learning*, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

denoted as $R_t$, switch (i.e., $R_t(a_0) \leftrightarrow R_t(a_1)$) once at an unpredictable time between 0 and $T$ (see Figure 1 (a)). The goal of the system is to maximize the average rewards over a period $T$, i.e., $\max_{\pi_1,\dots,\pi_T} \mathbb{E}\left[\sum_{t=0}^{T} R_t(a)\right]$. Initially, recommending $x_1$ yields a higher reward ($R_0(a_1) = 1$). However, the system anticipates a shift in the user preference towards $x_0$ by the end of period $T$. The system should optimize its policy $\pi_t$ during the interval from 0 to $T$, facing aleatoric uncertainty about when the user preferences will change. A conservative policy increases the preference weight $\beta_t$ associated with $x_0$ too quickly (Figure 1 (b)), while a pessimistic approach may adjust too slowly (Figure 1 (c)). The key challenge is to determine the optimal tempo of policy adjustment in anticipation of this unknown preference shift.

Based on the previous example, this paper challenges the belief that continually updating the decision always achieves an optimal bound of dynamic regret, a measurement of decision optimality in a time-varying environment. Our main contribution, Algorithm 1 and Theorem 5.8, demonstrates that strategically pausing decision updates provides a sharper upper bound on the dynamic regret by deriving an optimal ratio between the policy update duration and the pause duration.

To achieve this, we formulate the online interactive learning problem in Section 3 by determining three key aspects: 1) the frequency of policy updates, 2) the timing of policy updates, and 3) the extent of each update. First, we study the real-time inference mechanism by proposing a forecasting online reinforcement learning model-free framework in Section 4. In Section 5, we calculate an upper bound on the dynamic regret (Theorem 5.3) as a function of episodic and predictive uncertainties (Propositions 4.1 and 4.2), as well as aleatoric uncertainty (Proposition 5.6 and Lemma 5.7). This is achieved by separating it into the policy update phase (Lemma 5.1) and the policy hold phase (Lemma 5.2). In Subsection 5.3, we conduct numerical experiments to show how the optimal ratio minimizing the dynamic regret's upper bound (Theorem 5.8) varies with hyperparameters related to aleatoric uncertainty, highlighting the significance of the policy hold phase in this minimization. Finally, in Section 6, we empirically show two findings from three non-stationary environments: 1) a higher average reward of the forecasting method compared to the reactive method (Subsection 6.2), and 2) a non-positive correlation relationship between update ratios and average returns (Subsection 6.3).

### Notations

The sets of natural, real, and non-negative real numbers are denoted by $\mathbb{N}$, $\mathbb{R}$, and $\mathbb{R}_+$, respectively. For a finite set $Z$, the notation $|Z|$ represents its cardinality, and $\Delta(Z)$ denotes the probability simplex over $Z$. Given $X, Y \in \mathbb{N}$ with $X < Y$, we define $[X] := \{1, 2, \dots, X\}$, the closed interval $[X, Y] := \{X, X+1, \dots, Y\}$, and the half-open interval $[X, Y) := \{X, X+1, \dots, Y-1\}$. For $x \in \mathbb{R}_+$, the floor function $\lfloor x \rfloor$ is defined as $\max\{n \in \mathbb{N} \cup \{0\} \mid n \leq x\}$. For any functions $f, g : \mathbb{R}^m \to \mathbb{R}$ satisfying $f(x) \leq g(x)$ for all values of $x$, if $x_g^* = \arg\min_{x \in \mathbb{R}^m} g(x)$, then $x_g^*$ is referred to as a surrogate optimal solution of $f(x)$. We use the term surrogate optimal solution and suboptimal solution interchangeably.

## 2. Related works

### Real-time inference RL

One approach to real-time reinforcement learning is to adapt existing algorithms and validate their feasibility for real-time operation (Adam et al., 2012). Alternatively, some algorithms are specifically designed with the primary objective of functioning in real-time contexts (Cai et al., 2017; Wang & Yuan, 2015). A recent and distinct perspective on real-time inference was presented in (Ramstedt & Pal, 2019), which proposed a real-time markov reward process. In this process, the state evolves concurrently with the action selection. The anytime inference approach (Vlasselaer et al., 2015; Spirtes, 2001) encompasses a set of algorithms capable of returning a valid solution at any interruption point, with their performance improving over time.

### Non-stationary RL

The problem formulation of this paper draws inspiration from "desynchronized-time environment", initially proposed by (Lee et al., 2023). The desynchronized-time environment assigns the real-time duration of the learning process, where the agent is responsible for deciding both the timing and the duration of its interactions. (Finn et al., 2019) introduced the Follow-The-Meta-Leader algorithm to improve parameter initialization in a non-stationary environment, but it cannot efficiently handle delays in optimal policy tracking. To address this, (Chandak et al., 2020b;a) developed methods for forecasting policy evaluation, yet faced limitations in empirical analysis and theoretical bounds for policy performance. (Mao et al., 2021) proposed an adaptive $Q$-learning approach with a restart strategy, establishing a near-optimal dynamic regret bound.

We will further elaborate on related work on non-stationary RL in Appendix A.

## 3. Problem Statement

**Time-elapsing Markov Decision Process** (Lee et al., 2023). For a given time $t \in [0, T]$, we define the Markov Decision Process (MDP) at time $t$ as $\mathcal{M}_t := \langle \mathcal{S}, \mathcal{A}, P_t, R_t, \gamma, H \rangle$.

$\mathcal{S}$ is a state space, $\mathcal{A}$ is an action space, $P_t : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \Delta(\mathcal{S})$ is a transition probability at time $t$, and $R_t : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is a reward function at time $t$. For every time $t$, the agent interacts with the environment via a policy $\pi_t : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ where each episode takes $H$ steps to complete. We assume that a trajectory is finished within a second, implying that the agent will finish its trajectory within a temporally fixed MDP $\mathcal{M}_t$.

**Time elapsing variation budget.** In the real world, the time of the environment flows independently from $t = 0$ to $t = T$ regardless of the agent's behavior. For any time instances $t_1, t_2 \in [0, T)$ such that $t_1 < t_2$, we define *local variation budgets* $B_r(t_1, t_2)$ and $B_p(t_1, t_2)$ as

$$B_r(t_1, t_2) \coloneqq \sum_{t=t_1}^{t_2-1} \max_{s,a} |R_{t+1}(s, a) - R_t(s, a)| ,$$

$$B_p(t_1, t_2) \coloneqq \sum_{t=t_1}^{t_2-1} \max_{s,a} ||P_{t+1}(\cdot \mid s, a) - P_t(\cdot \mid s, a)||_1 .$$

Also, we define *cumulative variation budgets* $\bar{B}_p(t_1, t_2)$ and $\bar{B}_r(t_1, t_2)$ as the summation of local variation budgets between time $t_1$ and $t_2$, i.e.,

$$\bar{B}_r(t_1, t_2) \coloneqq \sum_{t=t_1}^{t_2-1} B_r(t_1, t), \bar{B}_p(t_1, t_2) \coloneqq \sum_{t=t_1}^{t_2-1} B_p(t_1, t).$$

To align with real-world scenarios where environmental changes do not normally occur too abruptly, we propose that these changes follow an exponential growth.

**Assumption 3.1** (Exponential order local variation budget)**.** For any time interval $[t_1, t_2] \subset [0, T)$, there exist constants $k_r, k_p > 1, B_p^{\max}, B_r^{\max} > 0$ such that $B_p(t_1, t) \leq B_p^{\max} k_p^{t-t_1}$ and $B_r(t_1, t) \leq B_r^{\max} k_r^{t-t_1}$ hold for $\forall t \in [t_1, t_2]$ .

Building on Assumption 3.1, we will derive cumulative variation budgets that also adhere to an exponential order.

**Corollary 3.2** (Exponential order cumulative variation budget)**.** *For arbitrary time instances $t_1, t_2 \in [0, T)$ satisfying $t_1 < t_2$, there exist constants $\alpha_r, \alpha_p > 1$ such that $\bar{B}_p(t_1, t_2) \leq B_p^{max} \alpha_p^{t_2-t_1}$ and $\bar{B}_r(t_1, t_2) \leq B_r^{max} \alpha_r^{t_2-t_1}$ hold.*

Next, we define stationary and non-stationary environments in the context of variation budget.

**Definition 3.3** (Stationary environment)**.** For arbitrary time instances $t_1, t_2 \in [0, T]$, if $B_r(t_1, t_2) = 0$ and $B_p(t_1, t_2) = 0$ are satisfied, then we call the corresponding environment a stationary environment.

**Definition 3.4** (Non-stationary environment)**.** If there exist $t_1, t_2 \in [0, T]$ such that $B_r(t_1, t_2) > 0$ or $B_p(t_1, t_2) > 0$, then we call the corresponding environment a non-stationary environment.

**State value function, State action value function.** For any policy $\pi$, we define the state value function $V_t^\pi : \mathcal{S} \to \mathbb{R}$ and the state action value function $Q_t^\pi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ at time $t$ as $V_t^\pi(s) := \mathbb{E}_{\mathcal{M}_t}\left[\sum_{h=0}^{H-1} \gamma^h r_{t,h} \mid s_t^0 = s\right]$ and $Q_t^\pi(s, a) := \mathbb{E}_{\mathcal{M}_t}\left[\sum_{h=0}^{H-1} \gamma^h r_{t,h} \mid s_t^0 = s, a_t^0 = a\right]$, where $r_{t,h} := R_t(s_t^h, a_t^h)$. We define the optimal policy at time $t$ as $\pi_t^* = \arg\max_\pi V_t^\pi$.

**Dynamic regret.** During the interval $[0, T]$, the agent operates according to a sequence of policies $\pi_1, \pi_2, \ldots, \pi_T$. Drawing from the learning procedure outlined previously, we define the time-varying dynamic regret $\mathfrak{R}(T) := \sum_{t=1}^{T} (V_t^* - V_t^{\pi_t})$, where $V_t^*$ represents the optimal policy value at time $t$ and $V_t^{\pi_t}$ is the value function obtained by executing policy $\pi_t$ in the MDP $\mathcal{M}_t$.

**Parallel process of policy learning and data collection.** In our formalization of policy learning in a non-stationary environment, the policy learning phase and the data collection phase (interaction) occur concurrently. In this context, the number of trajectories an agent can execute between the unit times $(t, t + 1), \forall t \in [T - 1]$, typically depends on the system's control frequency or its hardware capabilities. However, for the purpose of our analysis, we assume that the agent executes one trajectory per unit time. This means that at time $t$, the agent has rolled out a total of $t$ trajectories.

Before the first episode, the agent determines several key parameters:

1. **Frequency of Policy Updates:** The agent decides on the number of updates, denoted as $M \in \mathbb{N}$ times.

2. **Timing of Policy Updates:** The update times are set as a sequence $\{t_1, t_2, t_3, \ldots, t_M\}$ within $[0, T]$.

3. **Extent of Each Update:** The policy update iteration sequence is defined as $\{G_1, G_2, \ldots, G_M\}$.

Specifically, at each time $t_m \in [0, T]$ where $m \in [M]$, the agent updates its policy for $G_m \in \mathbb{N} \cup \{0\}$ iterations, using all previously collected trajectories. We assume that each policy iteration corresponds to one second in real-time. The policy then remains fixed for $N_m \in \mathbb{N} \cup \{0\}$ seconds after the updates, where it is determined as $N_m = t_{m+1} - (t_m + G_m)$. The next episode starts immediately at time $t_{m+1} = t_m + G_m + N_m$. Without loss of generality, we assume that $t_1 = 0$, and therefore $t_m = \sum_{i=1}^{m-1}(N_i + G_i)$ holds. Also, we define the $m^{\text{th}}$ policy update interval as $\mathcal{G}_m := [t_m, t_m + G_m)$ and the $m^{\text{th}}$ policy hold interval as $\mathcal{N}_m := [t_m + G_m, t_{m+1})$. For notational simplicity, we denote $\bar{B}_r(t_m, t_m + G_m)$, $\bar{B}_r(t_m + G_m, t_{m+1})$, $\bar{B}_p(t_m, t_m + G_m)$ and $\bar{B}_p(t_m + G_m, t_{m+1})$ as $\bar{B}_r(\mathcal{G}_m)$, $\bar{B}_r(\mathcal{N}_m)$, $\bar{B}_p(\mathcal{G}_m)$ and $\bar{B}_p(\mathcal{N}_m)$, respectively.

**How to determine $\{\pi_1, \pi_2, ..., \pi_T\}$.** At time $t_m$, the agent executes the policy $\pi_{t_m}$ and starts optimizing the policy for $G_m$ seconds. During this optimization, after $g$ iterations (seconds), where $g \in [G_m]$, the agent executes the most recently updated policy $\pi^g_{t_m}$. This updated policy represents the $g^{th}$ iteration of optimization from the initial policy $\pi_{t_m}$. Therefore, during the policy update interval $\mathcal{G}_m$, specifically at time $t_m + g$, the policy $\pi_{t_m+g}$ is equivalent to $\pi^g_{t_m}$. Subsequently, throughout the policy hold interval $\mathcal{N}_m$, the agent continues to execute the latest updated policy, denoted as $\pi_t = \pi^{G_m}_{t_m}$ for every $t$ within $\mathcal{N}_m$.

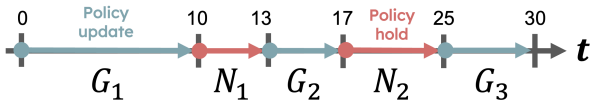



*Figure 2.* Parallel process of policy learning and data collection.

**Example.** Figure 2 illustrates our problem setting. For a given time duration between $t = 0$ and $t = 30$, suppose that the agent has chosen the frequency of policy updates as $M = 3$ and the update time sequence as $t_1 = 0, t_2 = 13, t_3 = 25$, along with the policy update durations $G_1 = 10, G_2 = 4, G_3 = 5$. The agent begins the first episode at $t = 0$ with a random policy $\pi_0$. Subsequently, during times $t = 1, 2, \ldots, 10$, the agent continuously executes updating policies $\pi^1_0, \pi^2_0, \ldots, \pi^{10}_0$, respectively, and then employs the latest updated policy $\pi^{10}_0$ at times $t = 11, 12, 13$. Following this, the agent operates with policies $\pi^1_{13}, \pi^2_{13}, \ldots, \pi^4_{13}$ during the period $t = 14, 15, 16, 17$, where $\pi_{13} = \pi^{10}_0$. Lastly, it executes with the most recently updated policy $\pi^4_{13}$ during the time $t = 18, \ldots, 25$.

## 4. Method

To implement a real-time inference mechanism, particularly emphasizing the prediction-based control approach of *"predicting the future in the past,"* we introduce a model-free proactive algorithm, detailed in Algorithm 1. This approach is based on the proactive evaluation of policies. At policy update time $t_m$, our proposed algorithm forecasts the future $Q$ value of time $t_{m+1}$ based on previous trajectories and then optimizes the future policy for duration $\mathcal{G}_m$ based on foreacasted future $Q$ value. For all $t \in [0, T]$, we denote the estimated value of $Q$ based on the past trajectories as $\widehat{Q}_t$ and the optimal value of $Q$ as $Q^*_t$. We also denote the future $Q$ value of time $t_{m+1}$ which was forcasted at time $t_m$ as $\widetilde{Q}_{t_{m+1}|t_m}$. During the time duration $\mathcal{G}_m$, we determine the policies $\{\pi^g_{t_m}\}_{g=1}^{G_m}$ by utilizing the Natural Policy Gradient (Kakade, 2001) with the entropy regularization method based on $\widetilde{Q}_{t_{m+1}|t_m}$ as follows:

$$\pi^{g+1}_{t_m}(\cdot|s) \propto \left(\pi^g_{t_m}(\cdot|s)\right)^{1-\frac{\eta\tau}{1-\gamma}} \exp\left(\frac{\eta \tilde{Q}_{t_{m+1}|t_m}}{1-\gamma}\right)$$
$$\text{s.t.} \quad ||\tilde{Q}_{t_{m+1}|t_m} - Q^*_{t_{m+1}}||_\infty = \delta^f_m$$

where $\eta$ is a learning rate, $\tau$ is an entropy regularization parameter and $\delta^f_m$ is the maximum forecasting error at time step $t_m$.

There are various methods to forecast $Q_{t_{m+1}|t_m}$ based on past $Q$ estimates $\{\widehat{Q}_t\}_{t=0}^{t_m}$. In this work, we provide analytical explanations on how the forecasting error can be bounded by the past $l$ uncertainties ($Q$ estimation errors) and the intrinsic uncertainty of the future environment (local variation budgets). For any $t$, we refer to $\epsilon_t$ as the maximum $Q$ estimation error if $||\widehat{Q}_t - Q^*_t||_\infty \leq \epsilon_t$ holds. To simplify the presentation, we drop the term "maximum" when it is clear from the context.

**Proposition 4.1** (Linear forecasting method with bounded $l_2$ norm)**.** *Consider a past reference length $l_p \in \mathbb{N}$ and define $\boldsymbol{w} := [w_{t_m-l_p+1}, \ldots, w_{t_m-1}, w_{t_m}]^\top$. We forecast $\tilde{Q}_{t_{m+1}|t_m}$ as a linear combination of the past $l_p$-estimated $Q$ values, namely $\tilde{Q}_{t_{m+1}|t_m} = \sum_{t=t_m-l_p+1}^{t_m} w_t \hat{Q}_t$, where the condition $\|\boldsymbol{w}\|_2 \leq L$ holds for some $L$. Then, $\delta_m$ can be bounded by*

$$\delta^f_m \leq L\sqrt{\sum_{t=t_m-l_p+1}^{t_m} 2\left(\max(u_t, \epsilon_t)\right)^2 + l_p(L+1)}$$
$$\left(\frac{1-\gamma^H}{1-\gamma} r_{max}\right)$$

*where* $u_t := \frac{1-\gamma^H}{1-\gamma}\left(B_r(t, t_{m+1}) + \frac{r_{max}}{1-\gamma} B_p(t, t_{m+1})\right)$ *and* $r_{max} := \max_{t,s,a} |R_t(s,a)|$.

Proposition 4.1 shows that utilizing a low-complexity forecasting model provides that the $m^{\text{th}}$ maximum forecasting error is bounded by intrinsic environment uncertainty of future $\{u_t\}_{t=t_m-l_p-1}^{t_m}$ and past uncertainties $\{\epsilon_t\}_{t=t_m-l_p-1}^{t_m}$ due to finite samples.

Compared to previous studies on finite-time $Q$ value convergence with asynchronous updates (Qu & Wierman, 2020; Even-Dar & Mansour, 2004), our work primarily focuses on how strategic policy update intervals affect an upper bound on the dynamic regret, leaving room for future exploration of $Q$ convergence rate improvement. This will be discussed in more detail in Section 5.

In the remainder of this section, we investigate in Proposition 4.2 and Corollary 4.3 how an $\epsilon_t$-accurate estimate of past $Q$ value establishes a lower bound condition on $\{N_i\}_{i=1}^{m-1}$ and $\{G_i\}_{i=1}^{m-1}$.

**Algorithm 1** Forecasting Online Reinforcement Learning

**Input:** Total time $T$, Policy update duration sets $\{H_1, .., H_M\}$, $\{G\}_{1:K}$, Dataset $\mathcal{D}$
**Init:** $m = 0$, $\pi_1 =$ random policy
**for** $t = \{1, 2, ..., T\}$ **do**
  Rollout $H$ steps trajectory with policy $\pi_t$ and save a trajectory to $\mathcal{D}$
  **if** $t \in \{t_1, t_2, ..., t_M\}$ **then**
    $m \leftarrow m + 1$
    $\widetilde{Q}_{t_{m+1}|t_m} = \texttt{ForQ}(\mathcal{D})$ /* Forecast future Q */
  **end if**
  **if** $t \in [t_m + G_m)$ **then**
    $\pi_{t+1} = \texttt{Update}(\pi_t, \eta, \tau, \gamma, \widetilde{Q}_{t_{m+1}|t_m})$ /* Update Policy */
  **else if** $t \in [t_m + G_m + N_m)$ **then**
    $\pi_{t+1} = \pi_t$ /* Pause policy update*/
  **end if**
**end for**

**Proposition 4.2** (Past uncertainty with sample complexity (Qu & Wierman, 2020))**.** *For any $\kappa > 0$ and under some conditions on stepsizes, if $t \geq \frac{(|S||A|)^{3.3}}{(1-\gamma)^{5.2}\epsilon_t^{2.6}}$, then $||\widehat{Q}_t - Q_t^*||_\infty \leq \epsilon_t$ holds.*

Proposition 4.2 highlights that the lower bound conditions of $\{N_i\}_{i=1}^{m-1}$ and $\{G_i\}_{i=1}^{m-1}$ are useful to reach $\epsilon_t$-accurate estimate of $Q$ value for asynchronous $Q$-learning method on a single trajectory. The upper bound of $\delta_m^f$ could be better minimized by taking $\max(u_t, \epsilon_t) = u_t$ for all $t \in [t_m - l + 1, t_m]$. This requires $t \geq \frac{(|S||A|)^{3.3}}{(1-\gamma)^{5.2}u_t^{2.6}}$ to hold for all $t \in [t_m - l + 1, t_m]$. Note that $t_m = \sum_{i=1}^{m-1}(N_i + G_i)$ holds. Therefore, for $j = 1, 2, ..., l_p$, we have $\sum_{i=1}^{m-1}(N_i + G_i) - j + 1 \geq \frac{(|S||A|)^{3.3}}{(1-\gamma)^{5.2}u_{t_m-j+1}^{2.6}}$. Then, the upper bound can be simplified without past uncertainty terms as follows.

**Corollary 4.3** (Maximum forecasting error bound)**.** *For $j = 1, 2, ..., l_p$, if $\{N_i\}_{i=1}^{m-1}$ and $\{G_i\}_{i=1}^{m-1}$ satisfy the condition $\sum_{i=1}^{m-1}(N_i + G_i) - j + 1 \geq \frac{(|S||A|)^{3.3}}{(1-\gamma)^{5.2}u_{t_m-j+1}^{2.6}}$, then $\delta_f$ is bounded by*

$$\delta_f \leq L u_{max}\sqrt{2l_p} + l_p(L+1)\left(\frac{1-\gamma^H}{1-\gamma} r_{max}\right)$$

*where $\delta_f := \max_{m\in[M]} \delta_m^f$ is a maximum forecasting error and $u_{max} := \max_{m\in[M]} u_{t_m - l_p + 1}$.*

Corollary 4.3 shows how the forecasting error $\delta_f$ is bounded with future environment's uncertainty $u_{max}$ with lower bound conditions on $\{N_i\}_{i=1}^{m-1}$ and $\{G_i\}_{i=1}^{m-1}$. By collecting more trajectories per the unit time $(t, t+1)$, we can significantly relax the lower bound condition, going beyond our initial assumption (see Section 3).

## 5. Theoretical Analysis

In this section, we provide a dynamic regret analysis to investigate how policy hold durations $\{N_1, N_2, \ldots, N_M\}$ influence the minimization of dynamic regret. We initially decompose the regret into two main components and calculate upper bounds on these components in Subsection 5.1. Subsequently, in Subsection 5.2, we further divide the overall upper bound of regret into three distinct terms and investigate how $N_m$ modulates each of these terms, except for the future forecasting regret term. Finally, in Subsection 5.3, we present numerical experiments that demonstrate variations in the regret upper bound in response to different $N_m$ values under different aleatoric uncertainties.

### 5.1. Regret analysis

We define the dynamic regret between times $t_m$ and $t_{m+1}$ as $\mathfrak{R}_m(T)$, which is given by $\mathfrak{R}_m(T) := \sum_{t=t_m}^{t_{m+1}}(V_t^* - V_t^{\pi_t})$. The $m^{\text{th}}$ dynamic regret, $\mathfrak{R}_m(T)$, can be decomposed into two components, named Policy update regret and Policy hold regret, as follows:

$$\mathfrak{R}(T) = \sum_{m=1}^{M}\Bigg(\underbrace{\sum_{t\in\mathcal{G}_m}(V_t^* - V_t^{\pi_t})}_{\text{Policy update regret}} + \underbrace{\sum_{t\in\mathcal{N}_m}(V_t^* - V_t^{\pi_t})}_{\text{Policy hold regret}}\Bigg).$$

The policy update regret and the policy hold regret will be studied next.

**Lemma 5.1** (Policy update regret)**.** *Let $\bar{B}(\mathcal{G}_m) := C_4\bar{B}_r(\mathcal{G}_m) + C_5\bar{B}_p(\mathcal{G}_m)$. For all $t \in \mathcal{G}_m$ where $m \in [M]$, it holds that*

$$\sum_{t\in\mathcal{G}_m}(V_t^* - V_t^{\pi_t}) \leq \frac{C_1}{\eta\tau}\cdot\Big(1 - (1-\eta\tau)^{G_m}\Big) + G_m\Big(C_2\delta_m^f + C_3\Big) + \bar{B}(\mathcal{G}_m)$$

*where* $C_1 = (\gamma+2)\big(||Q_{t_m}^* - Q_{t_m}||_\infty + 2\tau(1 - \frac{\eta\tau}{1-\gamma} || \log\pi_{t_m}^* - \log\pi_{t_m}||_\infty)\big), C_2 = \frac{2(\gamma+2)}{1-\gamma}\left(1 + \frac{\gamma}{\eta\tau}\right), C_3 = \frac{2\tau\log|\mathcal{A}|}{1-\gamma}, C_4 = \frac{2(1-\gamma^H)}{1-\gamma}, C_5 = \frac{\gamma}{1-\gamma}\cdot\Big(\frac{1-\gamma^H}{1-\gamma} - \gamma^{H-1}H\Big) + \frac{1-\gamma^H}{1-\gamma}\cdot\frac{r_{max}}{1-\gamma}$.

**Lemma 5.2** (Policy hold regret)**.** *Let $\bar{B}(\mathcal{N}_m) := C_4\bar{B}_r(\mathcal{N}_m) + C_5\bar{B}_p(\mathcal{N}_m)$. For all $t \in \mathcal{N}_m$ where $m \in [M]$, it holds that*

$$\sum_{t\in\mathcal{N}_m}(V_t^* - V_t^{\pi_t}) \leq N_m\cdot\Big(C_1(1-\eta\tau)^{G_m} + C_2\delta_m^f + C_3\Big) + \bar{B}(\mathcal{N}_m)$$

where $C_1, C_2, C_3, C_4, C_5$ are the constants defined in Lemma 5.1.

By leveraging Lemmas 5.1 and 5.2, the dynamic regret $\mathfrak{R}(T)$ will be bounded below.

**Theorem 5.3** (Dynamic regret)**.** *Let* $\bar{B}(t_m, t_{m+1}) := \bar{B}(\mathcal{N}_m) + \bar{B}(\mathcal{G}_m)$*. Then, it holds that*

$$\mathfrak{R}(T) \leq \sum_{m=1}^{M} \Bigg( \underbrace{\frac{C_1}{\eta\tau} + \left(N_m C_1 - \frac{C_1}{\eta\tau}\right)(1-\eta\tau)^{G_m}}_{\text{policy optimization regret}(\mathfrak{R}^{\pi}_m)} + \underbrace{(N_m + G_m)(C_2\delta^f_m + C_3)}_{Q \text{ function forecasting regret}(\mathfrak{R}^f_m)} + \underbrace{\bar{B}(t_m, t_{m+1})}_{\text{non-stationarity regret}(\mathfrak{R}^{env}_m)} \Bigg).$$

In Theorem 5.3, we articulate the decomposition of $\mathfrak{R}_m(T)$ into three terms: the policy optimization regret, denoted as $\mathfrak{R}^{\pi}_m$, $Q$ value forecasting regret, denoted as $\mathfrak{R}^f_m$, and non-stationarity regret, denoted by $\mathfrak{R}^{\text{env}}_m$. Now, by extending the upper bound of the forecasting error regret to $\sum_{m=1}^{M} \mathfrak{R}^f_m \leq \sum_{m=1}^{M}(N_m + G_m)(C_2\delta_f + C_3) = T(C_2\delta_f + C_3) \leq T\left(C_2\left(Lu_{\max}\sqrt{2l_p} + l_p(L+1)\left(\frac{1-\gamma^H}{1-\gamma} r_{\max}\right)\right) + C_3\right)$, we find that its upper bound is independent from $\{N_i, G_i\}_{i=1}^m$ and satisfies a sublinear convergence rate to the total time $T$ for any $l_p = (1/T)^{\alpha}, \alpha > 0$.

Expanding on the independence of $\{N_i, G_i\}_{i=1}^m$ from the upper bound of $\sum_{m=1}^{M} \mathfrak{R}^f_m$, we will show how $N_m$ balances between $\mathfrak{R}^{\pi}_m$ and $\mathfrak{R}^{env}_m$, followed by minimizing the upper bound of $\mathfrak{R}_m(T)$ in the next subsection.

### 5.2. Theoretical insight

One crucial theoretical insight to be deduced from Theorem 5.3 is what nonzero value of $N_m$ strikes a balance between $\mathfrak{R}^{\pi}_m$ and $\mathfrak{R}^{env}_m$. Our insights begin with the analysis of $\mathfrak{R}_m(T)$. We start by considering a fixed time interval $[t_m, t_{m+1}]$, which brings up the constraint $N_m + G_m = t_{m+1} - t_m$. The initial aspect of our investigation addresses whether a nonzero value of $N_m$ offers any advantage in a *stationary* environment.

**Lemma 5.4** (Optimal $N^*_m, G^*_m$ for $\mathfrak{R}^{\pi}_m$)**.** *Given a fixed time interval* $[t_m, t_{m+1}]$*, the optimal values* $N^*_m$ *and* $G^*_m$ *that minimize* $\mathfrak{R}^{\pi}_m$ *are determined as* $N^*_m = 0$ *and* $G^*_m = t_{m+1} - t_m$*, respectively.*

Since $\mathfrak{R}^{env}_m = 0$ is satisfied in stationary environments (see Definition 3.3), Corollary 5.5 ensues from Lemma 5.4.

**Corollary 5.5** (Optimal $N^*_m, G^*_m$ in Stationary Environments)**.** *Consider a stationary environment. The upper bound of* $\mathfrak{R}_m$ *achieves its minimum when* $N_m = 0$ *and* $G_m = t_{m+1} - t_m$*.*

What Corollary 5.5 states is intuitively straightforward. This is because in scenarios where the time sequence of the policy update $(t_1, t_2, \ldots, t_m)$ is fixed, maximizing the policy update duration is advantageous without considering forecasting errors. However, we claim that $N_m$ plays an important role in a non-stationary environment, i.e., positive $N^*_m$ minimizes the upper bound of $\mathfrak{R}_m(T)$. We first develop the following proposition.

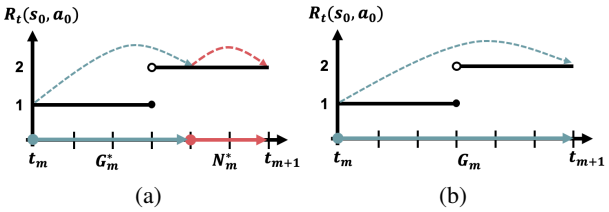



*Figure 3.* Optimal solutions for $\min_{G_m, N_m} \bar{B}(t_m, t_{m+1})$ are $(G^*_m, N^*_m) = (4, 2), (2, 4)$. (a) $G_m = 4, N_m = 2$. (b) $G_m = 6, N_m = 0$

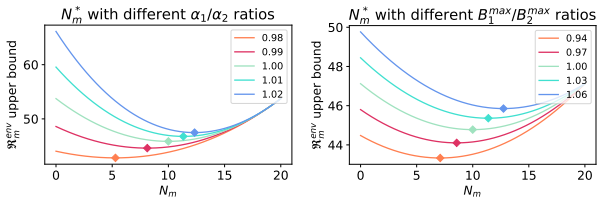
**Proposition 5.6** (Existence of Positive $N^*_m$ for $\mathfrak{R}^{\text{env}}_m$)**.** *In a non-stationary environment, consider any given time interval* $[t_m, t_{m+1}]$ *satisfying* $t_{m+1} - t_m \geq 2$*. Under these conditions, there exists a number* $N_m$ *within the open interval* $(0, t_{m+1} - t_m)$ *that minimizes* $\bar{B}(t_m, t_{m+1})$*.*

One way to intuitively understand Proposition 5.6 is exemplified in Figure 3. Consider a non-stationary environment where the reward abruptly changes only at state $s_0$ and action $a_0$. Suppose that $C_4 = C_5 = 1$. Then $\min \bar{B}(t_m, t_{m+1}) = 1$ and its solution is attainable at $(G^*_m, N^*_m) = (4, 2)$ and $(2, 4)$ (Figure 3 (a)), while in the case where $G_m = 6, N_m = 0$ yields $\bar{B}(t_m, t_{m+1}) = 3$ (Figure 3 (b)). Both subfigures optimize the policy toward the forecasted future $Q$ value of time $t_{m+1}$, but the time that the agent stops to update the policy ($t = t_m + G_m$) determines how much the agent would be conservative with respect to the future reward prediction.

Based on Proposition 5.6, we introduce the surrogate optimal solution $(G^*_m, N^*_m)$ for the non-stationarity regret $\mathfrak{R}^{env}_m$. According to Corollary 3.2, it holds that $\bar{B}_r(\mathcal{N}_m)$ is bounded by $\sum_{t=t_m+G_m}^{t=t_m+G_m+N_m-1} \alpha_r^{t-(t_m+G_m)} B^{\max}_r(\mathcal{N}_m)$, and similarly, $\bar{B}_r(\mathcal{G}_m)$ is bounded by $\sum_{t=t_m}^{t=t_m+G_m-1} \alpha_r^{t-t_m} B^{\max}_r(\mathcal{G}_m)$. For brevity, we use the notation $\alpha_{\diamond,1} = \alpha_{\diamond}(\mathcal{G}_m)$ and $\alpha_{\diamond,2} = \alpha_{\diamond}(\mathcal{N}_m)$, and similarly for $B^{\max}_{\diamond,1} = B^{\max}_{\diamond}(\mathcal{G}_m)$ and $B^{\max}_{\diamond,2} = B^{\max}_{\diamond}(\mathcal{N}_m)$, where $\diamond$ is either $r$ or $p$. Furthermore, we define $\alpha_{\square}$ as the $\max\left(\alpha_{r,\square}, \alpha_{p,\square}\right)$, and $B^{\max}_{\square}$ as the $\max\left(B^{\max}_{r,\square}, B^{\max}_{p,\square}\right)$, where $\square$ is either 1 or 2.

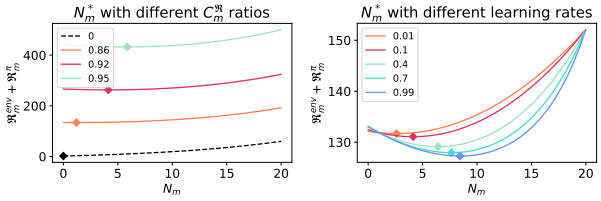
**Lemma 5.7** (Surrogate optimal $(G^*_m, N^*_m)$ for $\mathfrak{R}^{env}_m$)**.** *For given* $m, t_m, t_{m+1}$*, the surrogate optimal policy update and policy hold variables that minimize the upper bound of* $\mathfrak{R}^{env}_m$ *are*

$$N^*_m = \underset{N_m \in \{\lfloor \tilde{N}^*_m \rfloor, \lfloor \tilde{N}^*_m \rfloor + 1\}}{\arg\min} \mathfrak{R}^{env}_m(N_m, G_m)$$




(a) (b)

*Figure 4.* $\mathfrak{R}_m^{\text{env}}$ upper bound with different environmental hyperparameters. ⧫ denotes the minimum of each function graph. (a) $\alpha_1/\alpha_2 \in \{0.98, 0.99, 1.0, 1.01, 1.02\}$. (b) $B_1^{\max}/B_2^{\max} \in \{0.94, 0.97, 1.0, 1.03, 1.06\}$.




(a) (b)

*Figure 5.* $\mathfrak{R}_m^{\text{env}} + \mathfrak{R}_m^{\pi}$ upper bound with different $C_{\mathfrak{R}}$ ratios and learning rates. ⧫ denotes the minimum of each function graph. (a) $C_m^{\mathfrak{R}} \in \{0, 0.86, 0.92, 0.95\}$. (b) Learning rates $\in \{0.01, 0.1, 0.3, 0.7, 0.99\}$.

*and*

$$G_m^* = t_{m+1} - t_m - N_m^*,$$

*where*

$$\tilde{N}_m^* = \frac{1}{\ln(\alpha_2/\alpha_1)} \cdot \ln\left(\frac{\ln \alpha_1/(\alpha_1-1)}{\ln \alpha_2/(\alpha_2-1)} \cdot \alpha_1^{t_{m+1}-t_m} \cdot \frac{B_1^{max}}{B_2^{max}}\right).$$

Note that Lemma 5.7 provides a nonzero suboptimal $N_m^*$ that minimizes the non-stationary regret $\mathfrak{R}_m^{env}$. Now, we combine Lemmas 5.4 and 5.7 to find the suboptimal $N_m^*$ and $G_m^*$ that minimize the upper bound of $\mathfrak{R}_m(T)$.

**Theorem 5.8** (Surrogate optimal $(G_m^*, N_m^*)$ for $\mathfrak{R}_m$)**.** *For given $m, t_m, t_{m+1}$, the surrogate optimal policy update variable $N_m$ and surrogate policy hold variable $G_m$ that minimize the upper bound of $\mathfrak{R}_m$ satisfy the following equation:*

$$C_1\left((N_m-1)\ln(1-\eta\tau)-1\right)(1-\eta\tau)^{G_m} + (C_4+C_5)\left(\frac{\ln\alpha_1}{\alpha_1-1}B_1^{max}\alpha_1^{G_m} - \frac{\ln\alpha_2}{\alpha_2-1}B_2^{max}\alpha_2^{N_m}\right) = 0,$$

*where $C_1, C_4, C_5, \eta, \tau, \alpha_1, \alpha_2, B_1^{max}$, and $B_2^{max}$ are constants or parameters specific to the system under consideration.*

Apart from Lemma 5.7, Theorem 5.8 does not provide a closed-form solution. Consequently, we will conduct some numerical experiments to understand how $N_m^*$ and $G_m^*$ change to the hyperparameters of Theorem 5.8 in the next subsection.

### 5.3. Numerical analysis of theoretical insights

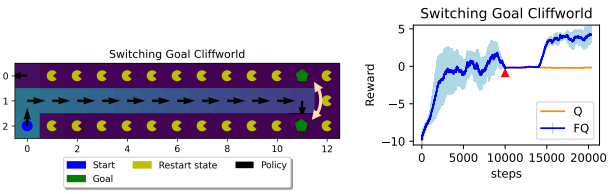
Figures 4 and 5 show how the surrogate optimal $N_m^*$ changes with different parameter choices. Figure 4 shows how $N_m^*$ changes with different parameters of the environment intrinsic uncertainty. Note that $(\alpha_1, B_1^{\max})$ and $(\alpha_2, B_2^{\max})$ represent the magnitude (severity) of the intrinsic uncertainty of the environment during the policy update phase ($\mathcal{G}_m$) and the policy hold phase ($\mathcal{N}_m$), respectively. The two subfigures of Figure 4 not only support the importance of holding $N_m^*$, but also show the necessity of keeping the policy hold phase longer if the uncertainty of the environment during the policy update phase $(\alpha_1, B_1^{\max})$ is greater than that of the policy hold phase $(\alpha_2, B_2^{\max})$. Moreover, Figure 5 (a) shows that increasing $N_m^*$ provides a better performance if the environment regret term dominates the regret $\mathfrak{R}_m^{\text{env}} + \mathfrak{R}_m^{\pi}$. We define the dominant ratio $C_m^{\mathfrak{R}}$ as $C_m^{\mathfrak{R}} := \int_{t_m}^{t_{m+1}} \mathfrak{R}_m^{env}/(\mathfrak{R}_m^{env} + \mathfrak{R}_m^{\pi})dt$. Finally, Figure 5 (b) validates that the surrogate optimal solution is still an acceptable solution and illustrates that the suboptimal gap resulting from relaxing the non-convex upper bound into a convex one is tolerable, as a higher learning rate leads to a fast convergence of $\mathfrak{R}_m^{\pi}$ and, in turn, intuitively results in a longer $N_m$ within fixed $t_m, t_{m+1}$.

## 6. Experiments

In this section, we demonstrate the effectiveness of two key components of the proposed algorithm, forecasting $Q$ value (line 7 of Algorithm 1) and the strategic policy update (line $9 \sim 12$ of Algorithm 1). In Subsection 6.2, we illustrate how utilizing forecasted $Q$ value yields higher rewards compared to a reactive method in a finite-dimensional environment. Subsequently, in Subsection 6.3, we will show how strategically assigning different policy update frequencies provides a higher performance than the continually updating policy method in an infinite-dimensional Mujoco environment, swimmer and halfcheetah. Details of environments and experiments are specified in Appendix C.

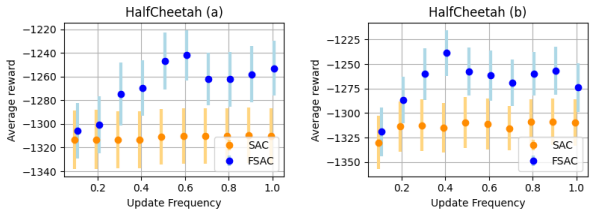
### 6.1. Future $Q$ value estimator

For the following experiments in Subsections 6.2 and 6.3, we design the `ForQ` function as the least-squares estimator (Chandak et al., 2020b), namely $\tilde{Q}_{t_{m+1}|t_m}(s,a) = \phi(t_{m+1})^\top w^*(s,a)$ where $\phi : [0,T) \to \mathbb{R}^d$ is a basis function for encoding the time index. For example, an identity basis is $\phi(x) := \{x, 1\}$. Then $w^*(s,a)$ denotes an optimal solution of the least-squares problem for any $s \in \mathcal{S}, a \in \mathcal{A}$, namely $w^*(s,a) = \arg\min_{w \in \mathbb{R}^{d\times 1}} ||\boldsymbol{Q}(s,a) - \Phi(X)^\top w||_2$




(a) (b)

*Figure 6.* (a) Swithcing goal cliffworld. (b) Reward per step. A red triangle means the goal point switches at step $= 10000$. A shaded area denotes one standard deviation among five different hyperparameter results.



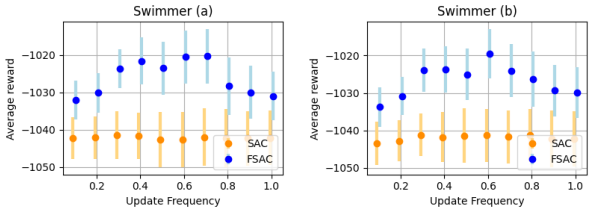

(a) (b)

*Figure 7.* Halfcheetah environment: blue dots are FSAC and orange dots are SAC. An error bar is 0.5 standard deviation over 36 different hyperparmeter results. (a) Average reward $l_f = 5$. (b) Average reward $l_f = 20$.

where $\boldsymbol{Q}(s,a) := [Q_{t_m-l_p+1}(s,a), ..., Q_{t_m}(s,a)]^\top \in \mathbb{R}^{l_p \times 1}$, $X := [t_m - l_p + 1, ..., t_m]^\top \in \mathbb{R}^{l_p \times 1}$, and $\Phi(X) := [\phi(t_m - l_p + 1), ..., \phi(t_m)] \in \mathbb{R}^{d \times l_p}$. The solution to the above least-squares problem is $w^*(s,a) = (\Phi(X)^\top \Phi(X))^{-1} \Phi(X)^\top \boldsymbol{Q}(s,a)$.

### 6.2. Goal switching cliffworld

We first experiment with a low-dimensional tabular MDP to verify that evaluating the policy by the forecasting method yields a better performance than the reactive method. The environment is the switching goal cliffworld where the agent always starts in the blue circle and a goal switches between two green pentagons (Figure 6 (a)). We use the $Q$-learning algorithm (Watkins & Dayan, 1992), denoted as `Q` in Figure 6 (b), to evaluate the current policy and compute future policy with future $Q$ estimator, denoted as `FQ` in Figure 6 (b), proposed in Subsection 6.1. Figure 6 (b) illustrates that after the goal point switches at step $= 10000$, the reactive method fails to obtain an optimal policy for the remaining steps. In contrast, the forecasting $Q$ method successfully identifies an optimal policy shortly after step $= 15000$.




(a) (b)

*Figure 8.* Swimmer environment: blue dots are FSAC and orange dots are SAC. An error bar is 0.5 standard deviation over 36 different hyperparmeter results. (a) Average reward $l_f = 5$. (b) Average reward $l_f = 15$.

### 6.3. Mujoco environment

To verify our findings in a large-scale environment, we propose a practical deep learning algorithm, Forecasting Soft-Actor Critic (FSAC), that specifies Algorithm 1. The FSAC algorithm is detailed in Algorithm 3 (see Appendix B). Then, we conduct experiments in high-dimensional non-stationary Mujoco environments (Todorov et al., 2012), swimmer, and halfcheetah where the reward changes as the episode goes by (Feng et al., 2022). We utilize the Soft-Actor Critic (SAC) algorithm (Haarnoja et al., 2018) as a baseline.

In particular, the distinctions between the FSAC and the SAC are the lines 2, $9 \sim 11$, and $16 \sim 18$ of Algorithm 3. In FSAC, the prediction length $l_f \in \mathbb{N}$ and the update frequency $\gamma_f \in (0, 1]$ are set as hyperparameters, with $t_m = l_f m$ for all $m \in [M]$ (line 2). The algorithm forecasts future $Q$ values at every $l_f$ iteration (lines $9 \sim 11$), updating the policy during the interval $(t_m, t_m + \lfloor l_f \gamma_f \rfloor]$ and keeping it between $(t_m + \lfloor l_f \gamma_f \rfloor, t_{m+1}]$ (lines $16 \sim 18$).

Figures 7 and 8 depict the results. In most cases, the FSAC algorithm (indicated by blue dots) yields a higher average return compared to the SAC algorithm (indicated by orange dots). These practical experiments aim to emphasize that $\gamma_f = 1.0$ does not necessarily lead to the best average reward. This observation aligns with our theoretical analysis presented in Section 5.2, where we demonstrate that a non-negative $N_m^*$ minimizes the upper bound on dynamic regret. We will elaborate on training and result details in Appendix C.2.

## 7. Conclusion

This paper introduces a forecasting online reinforcement learning framework, demonstrating that non-zero policy hold durations improve dynamic regret's upper bound. Empirical results show the forecasting method's advantage over reactive approaches and indicate that continuous policy updates do not always maximize average rewards. For future

work, it is crucial to explore methods to minimize the forecasting error to achieve a sharper upper bound. This paper presents work whose goal is implementing real-time control with prediction in environments with unknown uncertainties. A significant societal impact of our research is the narrowing of the gap between simulation-based RL and its real-world applications, along with demonstrating the advantages of pausing policy learning in continual learning settings.

## Acknowledgements

This work was supported by grants from ARO, ONR, AFOSR, NSF, and Noyce Initiative.

## Impact Statement

This paper presents work whose goal is to advance the field of reinforcement learning for real-world application. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

## References

Adam, S., Busoniu, L., and Babuska, R. Experience replay for real-time reinforcement learning control. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(2):201–212, 2012.

Al-Shedivat, M., Bansal, T., Burda, Y., Sutskever, I., Mordatch, I., and Abbeel, P. Continuous adaptation via meta-learning in nonstationary and competitive environments. In *International Conference on Learning Representations*, 2018.

Cai, H., Ren, K., Zhang, W., Malialis, K., Wang, J., Yu, Y., and Guo, D. Real-time bidding by reinforcement learning in display advertising. In *Proceedings of the tenth ACM international conference on web search and data mining*, pp. 661–670, 2017.

Cen, S., Cheng, C., Chen, Y., Wei, Y., and Chi, Y. Fast global convergence of natural policy gradient methods with entropy regularization. *Operations Research*, 70(4): 2563–2578, 2022.

Chandak, Y., Jordan, S., Theocharous, G., White, M., and Thomas, P. S. Towards safe policy improvement for non-stationary mdps. *Advances in Neural Information Processing Systems*, 33:9156–9168, 2020a.

Chandak, Y., Theocharous, G., Shankar, S., White, M., Mahadevan, S., and Thomas, P. Optimizing for the future in non-stationary mdps. In *International Conference on Machine Learning*, pp. 1414–1425. PMLR, 2020b.

Chen, X., Zhu, X., Zheng, Y., Zhang, P., Zhao, L., Cheng, W., CHENG, P., Xiong, Y., Qin, T., Chen, J., and Liu, T.-Y. An adaptive deep rl method for non-stationary environments with piecewise stable context. In *Advances in Neural Information Processing Systems*, volume 35, pp. 35449–35461, 2022.

Cheung, W. C., Simchi-Levi, D., and Zhu, R. Reinforcement learning for non-stationary markov decision processes: The blessing of (more) optimism. In *International Conference on Machine Learning*, pp. 1843–1854. PMLR, 2020.

Covington, P., Adams, J., and Sargin, E. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems*, 2016.

Ding, Y. and Lavaei, J. Provably efficient primal-dual reinforcement learning for cmdps with non-stationary objectives and constraints. AAAI'23/IAAI'23/EAAI'23. AAAI Press, 2023. ISBN 978-1-57735-880-0.

Ding, Y., Jin, M., and Lavaei, J. Non-stationary risk-sensitive reinforcement learning: Near-optimal dynamic regret, adaptive detection, and separation design. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37 (6):7405–7413, 2022.

Dulac-Arnold, G., Mankowitz, D., and Hester, T. Challenges of real-world reinforcement learning. *arXiv preprint arXiv:1904.12901*, 2019.

Espeholt, L., Soyer, H., Munos, R., Simonyan, K., Mnih, V., Ward, T., Doron, Y., Firoiu, V., Harley, T., Dunning, I., et al. Impala: Scalable distributed deep-rl with importance weighted actor-learner architectures. In *International conference on machine learning*, pp. 1407–1416. PMLR, 2018.

Evans, R. and Gao, J. Deepmind ai reduces google data centre cooling bill by 40 URL https://deepmind.google/discover/blog/.

Even-Dar, E. and Mansour, Y. Learning rates for q-learning. *J. Mach. Learn. Res.*, 5:1–25, dec 2004. ISSN 1532-4435.

Fei, Y., Yang, Z., Wang, Z., and Xie, Q. Dynamic regret of policy optimization in non-stationary environments. *Advances in Neural Information Processing Systems*, 33: 6743–6754, 2020.

Feng, F., Huang, B., Zhang, K., and Magliacane, S. Factored adaptation for non-stationary reinforcement learning. *Advances in Neural Information Processing Systems*, 35: 31957–31971, 2022.

Finn, C., Rajeswaran, A., Kakade, S., and Levine, S. Online meta-learning. In *International Conference on Machine Learning*, pp. 1920–1930. PMLR, 2019.

Gal, Y. Uncertainty in deep learning. phd thesis, University of Cambridge, 2016.

Glavic, M., Fonteneau, R., and Ernst, D. Reinforcement learning for electric power system decision and control: Past considerations and perspectives. *International Federation of Automatic Control*, 50:6918–6927, 2017.

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pp. 1861–1870. PMLR, 2018.

Hafner, D., Pasukonis, J., Ba, J., and Lillicrap, T. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023.

Hester, T. and Stone, P. Texplore: real-time sample-efficient reinforcement learning for robots. *Machine learning*, 90: 385–429, 2013.

Huang, B., Feng, F., Lu, C., Magliacane, S., and Zhang, K. Adarl: What, where, and how to adapt in transfer reinforcement learning. In *International Conference on Learning Representations*, 2022.

Imanberdiyev, N., Fu, C., Kayacan, E., and Chen, I.-M. Autonomous navigation of uav by using real-time model-based reinforcement learning. *2016 14th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, 2016.

Janner, M., Fu, J., Zhang, M., and Levine, S. When to trust your model: Model-based policy optimization. *Advances in neural information processing systems*, 32, 2019.

Kakade, S. M. A natural policy gradient. *Advances in neural information processing systems*, 14, 2001.

Kwon, J., Efroni, Y., Caramanis, C., and Mannor, S. Rl for latent mdps: Regret guarantees and a lower bound. *Advances in Neural Information Processing Systems*, 34: 24523–24534, 2021.

Lee, H., Ding, Y., Lee, J., Jin, M., Lavaei, J., and Sojoudi, S. Tempo adaptation in non-stationary reinforcement learning. *arXiv preprint arXiv:2309.14989*, 2023.

Levine, N., Chow, Y., Shu, R., Li, A., Ghavamzadeh, M., and Bui, H. Prediction, consistency, curvature: Representation learning for locally-linear control. *arXiv preprint arXiv:1909.01506*, 2019.

Mao, W., Zhang, K., Zhu, R., Simchi-Levi, D., and Başar, T. Model-free non-stationary rl: Near-optimal regret and applications in multi-agent rl and inventory control. *arXiv preprint arXiv:2010.03161*, 2020.

Mao, W., Zhang, K., Zhu, R., Simchi-Levi, D., and Basar, T. Near-optimal model-free reinforcement learning in non-stationary episodic mdps. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 7447–7458. PMLR, 2021.

Qu, G. and Wierman, A. Finite-time analysis of asynchronous stochastic approximation and $q$-learning. In Abernethy, J. and Agarwal, S. (eds.), *Proceedings of Thirty Third Conference on Learning Theory*, volume 125 of *Proceedings of Machine Learning Research*, pp. 3185–3205. PMLR, 09–12 Jul 2020.

Ramstedt, S. and Pal, C. Real-time reinforcement learning. *Advances in neural information processing systems*, 32, 2019.

Schrittwieser, J., Antonoglou, I., Hubert, T., Simonyan, K., Sifre, L., Schmitt, S., Guez, A., Lockhart, E., Hassabis, D., Graepel, T., et al. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588(7839): 604–609, 2020.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., van den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leach, M., Kavukcuoglu, K., Graepel, T., and Hassabis, D. Mastering the game of go with deep neural networks and tree search. *Nature*, 529:484–503, 2016.

Spirtes, P. An anytime algorithm for causal inference. In *Proceedings of the Eighth International Workshop on Artificial Intelligence and Statistics*, volume R3 of *Proceedings of Machine Learning Research*, pp. 278–285. PMLR, 04–07 Jan 2001.

Steck, H., Baltrunas, L., Elahi, E., Liang, D., Raimond, Y., and Basilico, J. Deep learning for recommender systems: A netflix case study. *AI Magazine*, 42(3):7–18, 2021.

Todorov, E., Erez, T., and Tassa, Y. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033. IEEE, 2012.

Vlasselaer, J., Van den Broeck, G., Kimmig, A., Meert, W., and De Raedt, L. Anytime inference in probabilistic logic programs with tp-compilation. In *Proceedings of 24th International Joint Conference on Artificial Intelligence (IJCAI)*, volume 2015, pp. 1852–1858. IJCAI-INT JOINT CONF ARTIF INTELL, 2015.

Wang, J. and Yuan, S. Real-time bidding: A new frontier of computational advertising research. *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, 2015.

Watkins, C. J. and Dayan, P. Q-learning. *Machine learning*, 8:279–292, 1992.

Zintgraf, L., Schulze, S., Lu, C., Feng, L., Igl, M., Shiarlis, K., Gal, Y., Hofmann, K., and Whiteson, S. Varibad: Variational bayes-adaptive deep rl via meta-learning. *Journal of Machine Learning Research*, 22(289):1–39, 2021.

## A. Related works

In this work, we have introduced a forecasting method for non-stationary environments. Before proceeding with our contributions, we first review the existing methods for addressing non-stationary environments in reinforcement learning (RL). Those can be categorized into three main approaches.

One naive approach is to utilize previous RL algorithms that were designed for stationary environments to solve non-stationary environments. Namely, this involves directly applying established RL frameworks for stationary MDPs without additional mechanisms. Usually, this approach involves restarting strategies to handle longer horizon problems in a decision making.

The second approach is model-based methods, which update models to adapt to changing environments. Techniques include using rollout data from the model (Janner et al., 2019; Hafner et al., 2023). A few well-known methods include online model updates and identifying latent factors (Zintgraf et al., 2021; Chen et al., 2022; Huang et al., 2022; Feng et al., 2022; Kwon et al., 2021). Model-based methods face challenges in non-stationary settings due to difficulties in estimating accurate non-stationary models (Cheung et al., 2020; Ding et al., 2022). To be more specific, (Huang et al., 2022) explored learning factors of non-stationarity and their representations in heterogeneous domains with varying reward functions and dynamics. (Zintgraf et al., 2021) proposed a Bayesian policy learning algorithm by conditioning actions on both states and latent tensors that capture the agent's uncertainty in the environment. In a similar manner, (Feng et al., 2022) incorporated insights from the causality literature to model non-stationarity as latent change factors across different environments, learning policies conditioned on these latent factors of causal graphs. Despite these advancements, learning optimal policies conditioned on latent states (Zintgraf et al., 2021; Chen et al., 2022; Huang et al., 2022; Feng et al., 2022; Kwon et al., 2021) presents significant challenges for theoretical analysis. Recent works (Cheung et al., 2020; Ding et al., 2022; Ding & Lavaei, 2023) have proposed model-based algorithms with provable guarantees. However, these algorithms are not scalable for complex environments and lack empirical evaluation.

The third approach is model-free methods. (Al-Shedivat et al., 2018) utilized meta-learning among training tasks to find initial hyperparameters of policy networks that can be quickly fine-tuned for new, unseen tasks. However, this method assumes access to a prior distribution of training tasks, which is often unavailable in real-world scenarios. To address this limitation, (Finn et al., 2019) proposed the Follow-The-Meta-Leader (FTML) algorithm, which continuously improves parameter initialization for non-stationary input data. Despite its innovation, FTML suffers from a lag in tracking the optimal policy, as it maximizes current performance uniformly over all past samples.

To mitigate this lag, (Mao et al., 2020) introduced adaptive Q-learning with a restart strategy, establishing a near-optimal dynamic regret bound. (Chandak et al., 2020b;a) focused on forecasting the non-stationary performance gradient to adapt to time-varying optimal policies. Nevertheless, these approaches are limited by empirical analyses on bandit settings or low-dimensional environments and lack a theoretical performance bound for the adapted policies. Also, (Fei et al., 2020) proposed two model-free policy optimization algorithms based on the restart strategy, demonstrating that their dynamic regret exhibits polynomial space and time complexities. However, these methods (Mao et al., 2020; Fei et al., 2020) still lack empirical validation and adaptability in complex environments.

## B. Algorithms

**Algorithm 2** `Update`: Update policy $\pi$

**Input:** policy $\pi$, learning rate $\eta$, entropy regularization constant $\tau$, discount factor $\gamma$, policy evaluation $\widehat{Q}$

$Z(s) = \sum_{a\in\mathcal{A}} \left(\pi(a|s)\right)^{1-\frac{\eta\tau}{1-\gamma}} \exp\left(\eta\widehat{Q}(s,a)/(1-\gamma)\right)$

$\pi'(\cdot|s) = \frac{1}{Z(s)} \cdot \left(\pi(\cdot|s)\right)^{1-\frac{\eta\tau}{1-\gamma}} \exp\left(\frac{\eta\widehat{Q}(s,\cdot)}{1-\gamma}\right)$

**Return** $\pi'$

## C. Experiments

### C.1. Environments and experiments details

**Goal switching cliffword**

**Algorithm 3** Forecasting Soft Actor-Critic

Initialize parameter vectors $\psi, \bar{\psi}, \theta, \phi$.
Set prediction length $l_f$, update frequency $\gamma_f$
**for** each iteration **do**
  **for** each environment step **do**
    Sample action $a_t \sim \pi_\theta(a_t|s_t)$.
    Sample next state $s_{t+1} \sim p(s_{t+1}|s_t, a_t)$.
    $D \leftarrow D \cup \{(s_t, a_t, r(s_t, a_t), s_{t+1})\}$.
  **end for**
  **if** iteration % $l_f = 0$ **then**
    $\tilde{Q} = \texttt{ForQ}(D)$.
  **end if**
  **for** each gradient step **do**
    $\psi \leftarrow \psi - \lambda_\psi \nabla_\psi J_V(\psi)$.
    $\theta_i \leftarrow \theta_i - \lambda_Q \nabla_{\theta_i} J_Q(\theta_i)$ for $i \in \{1, 2\}$.
    $\bar{\psi} \leftarrow \tau_s \psi + (1 - \tau_s)\bar{\psi}$.
    **if** iteration % $l_f \leq l_f \gamma_f$ **then**
      $\phi \leftarrow \phi - \lambda_\pi \nabla_\phi \tilde{J}_\pi(\phi)$.
    **end if**
  **end for**
**end for**

The environment is $12 \times 3$ tabular MDP where $(0, 2)$ is a fixed initial state (blue point), and the possible goal points are $(11, 0)$ and $(11, 2)$ (for the $x, y$ axis, see Figure 6 (a)). The agent executes $4$ actions (up, left, right, down). If the agent reaches the restart states ($(1, 2), (2, 2), ..., (10, 2)$ and $(1, 0), (2, 0), ..., (10, 0)$), denoted by yellow points, then the agent goes back to the initial state with a failure reward $-100$. If the agent reaches the goal point, then it receives the success reward $+100$. For taking every step (for every time the agent executes an action), the agent receives a step reward of $-100$.

For experiments, we use the $Q$-learning algorithm (Watkins & Dayan, 1992). In Figure 6 (b), we denote "reactive" label as $Q$-learning algorithm proposed by (Watkins & Dayan, 1992) and "future $Q$" label as a method that combines $Q$-learning algorithm to evaluate the current policy and use future $Q$ estimator to compute future policy that was proposed in section 6.1. We set the maximum number of steps as $100$. The experiments have been carried out by changing hyperparameters of $Q$-learning: step size $\alpha$ and $\epsilon$ from the $\epsilon$-greedy method. We have done experiments with different $(\alpha, \epsilon) = (0.05, 0.05), (0.1, 0.1), (0.1, 0.05), (0.2, 0.1), (0.2, 0.05), (0.3, 0.1)$.

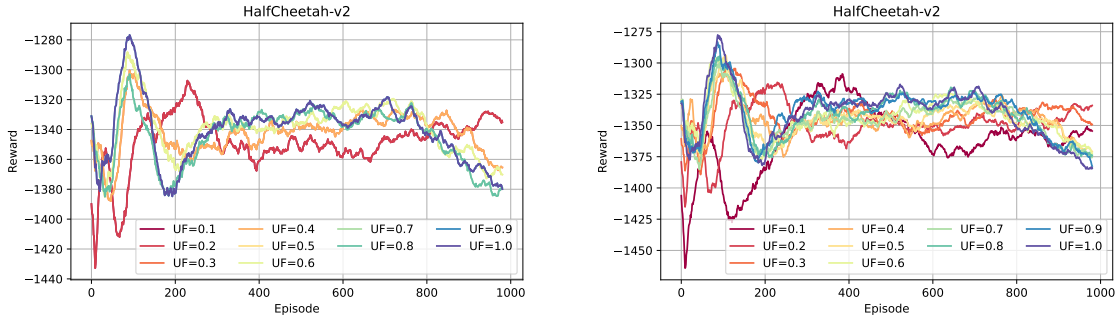
**Swimmer, Halfcheetah**

The Swimmer and Halfcheetah environments share the same reward function at step $h$ as $r_h = r_h^{(1)} + r_h^{(2)} + r_h^{(3)}$. It comprises a healthy reward ($r_h^{(1)}$), a forward reward ($r_h^{(2)} = k_f \frac{x_{h+1} - x_h}{\Delta t_{frame}}, k_f > 0$), and a control cost ($r_h^{(3)}$). We modify the environment to be non-stationary by the agent's desired velocity changes as time goes by. Specifically, we modify the forward reward $r_h^{(2)}$ varies as $r_h^{(2)} = -\left| k_f \frac{x_{h+1} - x_h}{\Delta t_{frame}} - v_d(t) \right|$, with $v_d(t) = a \sin(wt)$ and $t$ representing the episode. Here, $a, w > 0$ are constants.

For our experiments, we varied hyperparameters such as learning rates $\lambda_\pi \in \{0.0001, 0.0003, 0.0005, 0.0007\}$, soft update parameters $\tau_s \in \{0.001, 0.005, 0.003\}$ and the entropy regularization parameters $\{0.01, 0.03, 0.1\}$ and also experimented with different prediction lengths $l_f \in \{5, 15, 20\}$. We selected the average reward per episode as the performance metric, in line with the definition of dynamic regret. For given hyperparameters, we compare the average reward between FSAC and SAC for different update frequencies $\gamma_f \in \{0.1, 0.2, \ldots, 1.0\}$. The experiments were conducted in two different Mujoco environments: HalfCheetah and Swimmer (see Figures 7 and 8). In Figures 7 and 8, error bars denote 0.5 standard deviations.

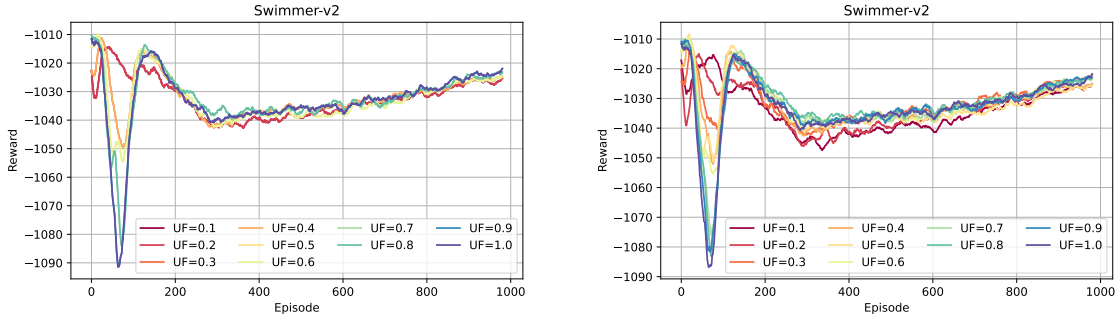
### C.2. Results

In this subsection, we have elaborated on the results of the experiment on Halfcheetah and Swimmer. Note that Figures 9,11 and 12 are detailed results for Figure 7 of the main paper, and Figures 10,13 and 14 are detailed result for Figure 8 of the main paper. Figures 9 and 10 show the reward return per episode for different update frequencies $\gamma_f \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$. Figures 11,12,13 and 14 compare the FSAC and SAC reward return per episode. Note that the plotted lines are mean rewards calculated over 36 different hyperparameters (learning rates $\lambda_\pi \in \{0.0001, 0.0003, 0.0005, 0.0007\}$, soft update parameters $\tau_s \in \{0.001, 0.005, 0.003\}$ and the entropy regularization parameters $\{0.01, 0.03, 0.1\}$).



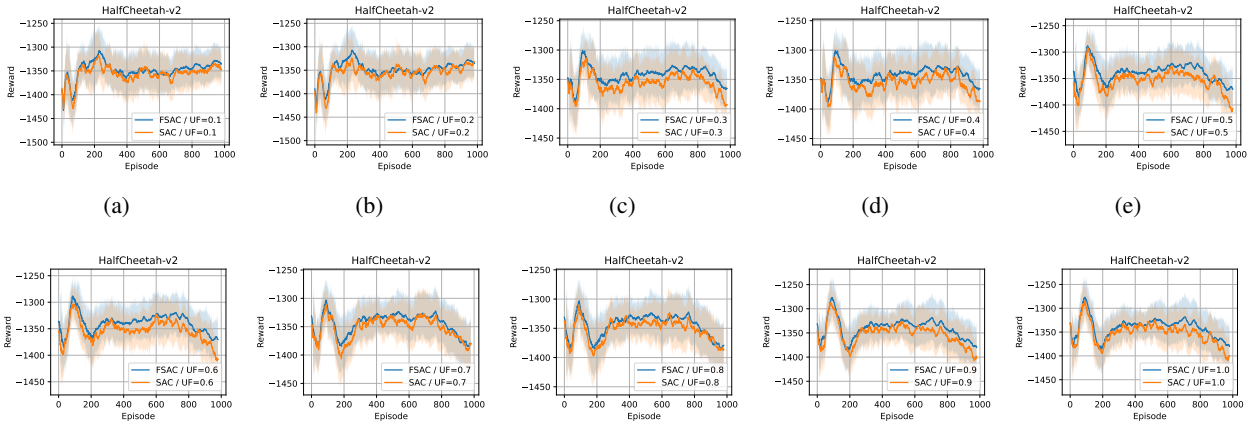

(a) (b)

*Figure 9.* Reward per episode in the Halfcheetah environment for various update frequencies $\gamma_f \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$. The plotted lines represent the mean reward across 36 different hyperparameters. (a) For $l_f = 5$. (b) For $l_f = 20$.




(a) (b)

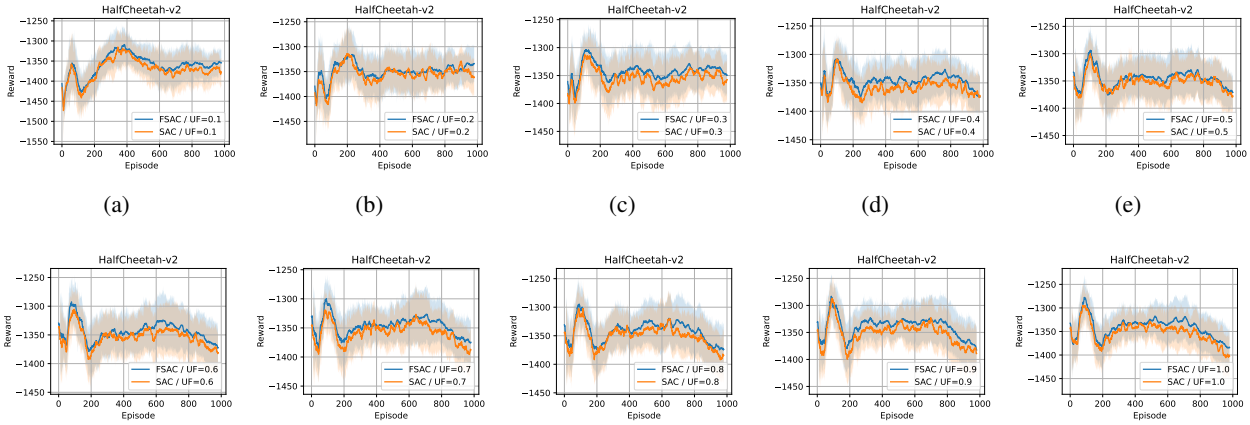
*Figure 10.* Reward per episode in the Swimmer environment for various update frequencies $\gamma_f \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$. The plotted lines represent the mean reward across 36 different hyperparameters. (a) For $l_f = 5$. (b) For $l_f = 15$.




(a) (b) (c) (d) (e)

(f) (g) (h) (i) (j)

*Figure 11.* Reward per episode for Halfcheetah environment when $l_f = 5$. The blue lines are FSAC, and the orange lines are SAC. The shaded areas are 0.5 standard deviations over 36 different hyperparameter results. (a) $\gamma_f = 0.1$. (b) $\gamma_f = 0.2$. (c) $\gamma_f = 0.3$. (d) $\gamma_f = 0.4$. (e) $\gamma_f = 0.5$. (f) $\gamma_f = 0.6$. (g) $\gamma_f = 0.7$. (h) $\gamma_f = 0.8$. (i) $\gamma_f = 0.9$. (j) $\gamma_f = 1.0$.




(a) (b) (c) (d) (e)

(f) (g) (h) (i) (j)

*Figure 12.* Reward per episode for Halfcheetah environment when $l_f = 20$. The blue lines are FSAC, and the orange lines are SAC. The shaded areas are 0.5 standard deviations over 36 different hyperparameter results. (a) $\gamma_f = 0.1$. (b) $\gamma_f = 0.2$. (c) $\gamma_f = 0.3$. (d) $\gamma_f = 0.4$. (e) $\gamma_f = 0.5$. (f) $\gamma_f = 0.6$. (g) $\gamma_f = 0.7$. (h) $\gamma_f = 0.8$. (i) $\gamma_f = 0.9$. (j) $\gamma_f = 1.0$.

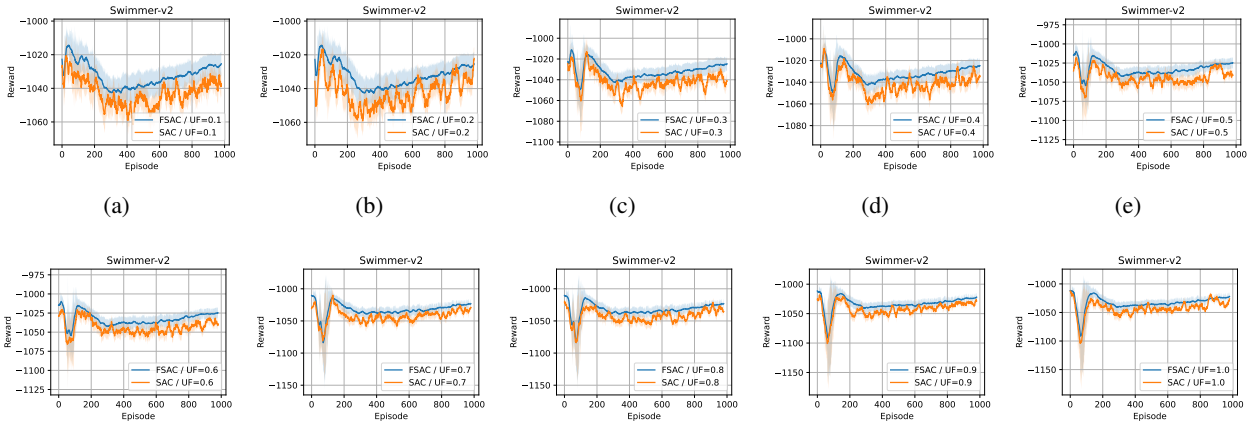



*Figure 13.* Reward per episode for Swimmer environment when $l_f = 5$. The blue lines are FSAC, and the orange lines are SAC. The shaded areas are 0.5 standard deviations over 36 different hyperparameter results. (a) $\gamma_f = 0.1$. (b) $\gamma_f = 0.2$. (c) $\gamma_f = 0.3$. (d) $\gamma_f = 0.4$. (e) $\gamma_f = 0.5$. (f) $\gamma_f = 0.6$. (g) $\gamma_f = 0.7$. (h) $\gamma_f = 0.8$ (i) $\gamma_f = 0.9$. (j) $\gamma_f = 1.0$.

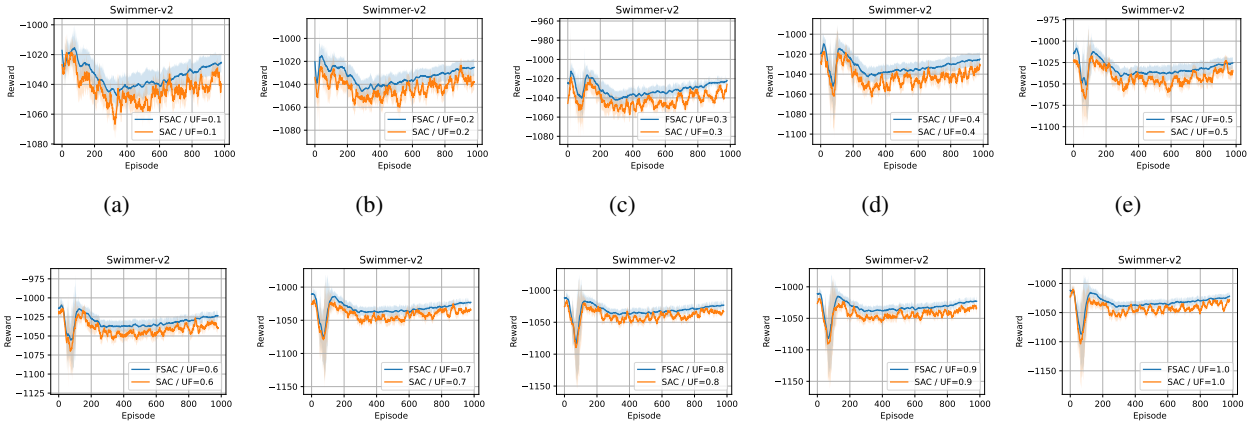



*Figure 14.* Reward per episode for Swimmer environment when $l_f = 15$. The blue lines are FSAC, and the orange lines are SAC. The shaded areas are 0.5 standard deviations over 36 different hyperparameter results. (a) $\gamma_f = 0.1$. (b) $\gamma_f = 0.2$. (c) $\gamma_f = 0.3$. (d) $\gamma_f = 0.4$. (e) $\gamma_f = 0.5$. (f) $\gamma_f = 0.6$. (g) $\gamma_f = 0.7$. (h) $\gamma_f = 0.8$. (i) $\gamma_f = 0.9$. (j) $\gamma_f = 1.0$.

## D. Proofs

*Proof of Proposition 4.1.*

$$
\begin{aligned}
||\widetilde{Q}_{t_{m+1}} - Q^*_{t_{m+1}}||_\infty &= \left|\left|\sum_{t=t_m-l_p+1}^{t_m} w_t \left(\widehat{Q}_t - Q^*_{t_{m+1}}\right)\right|\right|_\infty + \left|\left|\sum_{t=t_m-l_p+1}^{t_m} (w_t - 1)\, Q^*_{t_{m+1}}\right|\right|_\infty \\
&\leq \sum_{t=t_m-l_p+1}^{t_m} |w_t| \left(||Q^*_t - Q^*_{t_{m+1}}||_\infty + ||Q^*_t - \widehat{Q}_t||_\infty\right) + \sum_{t=t_m-l_p+1}^{t_m} |w_t - 1| \left|\left|Q^*_{t_{m+1}}\right|\right|_\infty \\
&\leq \sqrt{\sum_{t=t_m-l_p+1}^{t_m} |w_t|^2} \sqrt{\sum_{t=t_m-l_p+1}^{t_m} \left(||Q^*_t - Q^*_{t_{m+1}}||_\infty + ||Q^*_t - \widehat{Q}_t||_\infty\right)^2} \\
&\quad + \left(\sum_{t=t_m-l_p+1}^{t_m} |w_t| + l_p\right) \left|\left|Q^*_{t_{m+1}}\right|\right|_\infty \\
&\leq L \cdot \sqrt{\sum_{t=t_m-l_p+1}^{t_m} \left(||Q^*_t - Q^*_{t_{m+1}}||^2_\infty + 2||Q^*_t - Q^*_{t_{m+1}}||_\infty ||Q^*_t - \widehat{Q}_t||_\infty ||Q^*_t - \widehat{Q}_t||^2_\infty\right)} \\
&\quad + \left(l_p \sqrt{\sum_{t=t_m-l+1}^{t_m} |w_t|^2} + l_p\right) \left(\frac{1-\gamma^H}{1-\gamma} r_{max}\right). \qquad (1)
\end{aligned}
$$

We use Lemma E.2 to conclude that

$$
||Q^*_t - Q^*_{t_{m+1}}||_\infty \leq \frac{1-\gamma^H}{1-\gamma} \left(B_r(t, t_{m+1}) + \frac{r_{max}}{1-\gamma} B_p(t, t_{m+1})\right).
$$

Moreover, the assumption

$$
||Q^*_t - \widehat{Q}_t||_\infty \leq \epsilon_t
$$

holds. As a result,

$$
\begin{aligned}
&\left|\left|\widetilde{Q}_{t_{m+1}} - Q^*_{t_{m+1}}\right|\right|_\infty \\
&\leq L \sqrt{\sum_{t=t_m-l+1}^{t_m} \begin{bmatrix} \left(\frac{1-\gamma^H}{1-\gamma}\left(B_r(t, t_{m+1}) + \frac{r_{max}}{1-\gamma} B_p(t, t_{m+1})\right)\right)^2 \\ +2\frac{1-\gamma^H}{1-\gamma}\left(B_r(t, t_{m+1}) + \frac{r_{max}}{1-\gamma} B_p(t, t_{m+1})\right)\epsilon_t + \epsilon_t^2 \end{bmatrix}} \\
&\quad + l_p(L+1)\left(\frac{1-\gamma^H}{1-\gamma} r_{max}\right).
\end{aligned}
$$

To simplify the expression, define $u_t := \frac{1-\gamma^H}{1-\gamma}\left(B_r(t, t_{m+1}) + \frac{r_{max}}{1-\gamma} B_p(t, t_{m+1})\right)$. Then, the inequality (1) can be rewritten in a simpler form as follows:

$$
\begin{aligned}
\left|\left|\widetilde{Q}_{t_{m+1}} - Q^*_{t_{m+1}}\right|\right|_\infty &\leq L \sqrt{\sum_{t=t_m-l+1}^{t_m} 2\left(\max(u_t, \epsilon_t)\right)^2} + l_p(L+1)\left(\frac{1-\gamma^H}{1-\gamma} r_{max}\right) \\
&\leq \sqrt{2} L l_p \max_{t \in [t_m-l+1, t_m]} \left(\max(u_t, \epsilon_t)\right) + l_p(L+1)\left(\frac{1-\gamma^H}{1-\gamma} r_{max}\right).
\end{aligned}
$$

*Proof of Proposition 4.2.* Refer to Theorem 7 of (Qu & Wierman, 2020). □

□

*Proof of Lemma 5.1.* The policy update term is divided into three terms:

$$\sum_{t\in\mathcal{G}_m}(V_t^*-V_t^{\pi_t}) = \sum_{g=0}^{G_m-1}\left(V_{t_m+g}^*-V_{t_m+g}^{\pi_{t_m+g}}\right)$$
$$= \sum_{g=0}^{G_m-1}\left(\underbrace{\left(V_{t_m+G_m-1}^*-V_{t_m+G_m-1}^{\pi_{t_m+g}}\right)}_{\text{(1-I)}}+\underbrace{\left(V_{t_m+G_m-1}^{\pi_{t_m+g}}-V_{t_m+g}^{\pi_{t_m+g}}\right)}_{\text{(1-II)}}+\underbrace{\left(V_{t_m+g}^*-V_{t_m+G_m-1}^*\right)}_{\text{(1-III)}}\right).$$

Note that the term (1-I), the term (1-II), and the term (1-III) are upper bounded by Lemma E.1, Corollary E.3, and Lemma E.4.

For any $g\in[0,G_m-1]$ and for any $s\in\mathcal{S}$, one can write:

- $V_{t_m+G_m-1}^*-V_{t_m+G_m-1}^{\pi_{t_m+g}}\le(\gamma+2)((1-\eta\tau)^g C')+\frac{2(\gamma+2)}{1-\gamma}\left(1+\frac{\gamma}{\eta\tau}\right)\cdot\epsilon_f+\frac{2\tau\log|\mathcal{A}|}{1-\gamma}$
- $V_{t_m+G_m-1}^{\pi_{t_m+g}}(s)-V_{t_m+g}^{\pi_{t_m+g}}(s)\le\frac{1-\gamma^H}{1-\gamma}\cdot B_r(t_m+g,t_m+G_m-1)+\frac{\gamma}{1-\gamma}\cdot\left(\frac{1-\gamma^H}{1-\gamma}-\gamma^{H-1}H\right)\cdot B_p(t_m+g,t_m+G_m-1)$
- $V_{t_m+g}^*(s)-V_{t_m+G_m-1}^*(s)\le\frac{1-\gamma^H}{1-\gamma}\left(B_r(t_m+g,t_m+G_m-1)+\frac{r_{max}}{1-\gamma}B_p(t_m+g,t_m+G_m-1)\right)$

where $C'=||Q_\tau^*-Q_\tau^{t_m}||_\infty+2\tau(1-\frac{\eta\tau}{1-\gamma}||\log\pi_\tau^*-\log\pi_\tau^{t_m}||_\infty)$. Now, taking the summation over $g=0,...,G_m-1$ gives rise to

$$\begin{aligned}
&\sum_{t\in\mathcal{G}_m}(V_t^*-V_t^{\pi_t})\\
=&(\gamma+2)C'\frac{1-(1-\eta\tau)^{G_m}}{\eta\tau}+G_m\cdot\left(\frac{2(\gamma+2)}{1-\gamma}\left(1+\frac{\gamma}{\eta\tau}\right)\cdot\epsilon_f+\frac{2\tau\log|\mathcal{A}|}{1-\gamma}\right)\\
&+\frac{1-\gamma^H}{1-\gamma}\cdot\left(\sum_{g=0}^{G_m-1}B_r(t_m+g,t_m+G_m-1)\right)\\
&+\frac{\gamma}{1-\gamma}\cdot\left(\frac{1-\gamma^H}{1-\gamma}-\gamma^{H-1}H\right)\cdot\left(\sum_{g=0}^{G_m-1}B_p(t_m+g,t_m+G_m-1)\right)\\
&+\frac{1-\gamma^H}{1-\gamma}\left(\sum_{g=0}^{G_m-1}B_r(t_m+g,t_m+G_m-1)+\frac{r_{max}}{1-\gamma}\sum_{g=0}^{G_m-1}B_p(t_m+g,t_m+G_m-1)\right)\\
\le&(\gamma+2)C'\frac{1-(1-\eta\tau)^{G_m}}{\eta\tau}+G_m\cdot\left(\frac{2(\gamma+2)}{1-\gamma}\left(1+\frac{\gamma}{\eta\tau}\right)\cdot\epsilon_f+\frac{2\tau\log|\mathcal{A}|}{1-\gamma}\right)\\
&+\frac{2(1-\gamma^H)}{1-\gamma}\cdot\left(\bar{B}_r(t_m,t_m+G_m-1)\right)\\
&+\left(\frac{\gamma}{1-\gamma}\cdot\left(\frac{1-\gamma^H}{1-\gamma}-\gamma^{H-1}H\right)+\frac{1-\gamma^H}{1-\gamma}\cdot\frac{r_{max}}{1-\gamma}\right)\cdot\left(\bar{B}_r(t_m,t_m+G_m-1)\right)\\
=&\frac{C_1}{\eta\tau}\cdot\left(1-(1-\eta\tau)^{G_m}\right)+G_m\cdot(C_2\epsilon_f+C_3)+C_4\bar{B}_r(\mathcal{G}_m)+C_5\bar{B}_p(\mathcal{G}_m)
\end{aligned}$$

where $C_1=(\gamma+2)\left(||Q_{t_m}^*-Q_{t_m}||_\infty+2\tau(1-\frac{\eta\tau}{1-\gamma}||\log\pi_{t_m}^*-\log\pi_{t_m}||_\infty)\right), C_2=\frac{2(\gamma+2)}{1-\gamma}\left(1+\frac{\gamma}{\eta\tau}\right), C_3=\frac{2\tau\log|\mathcal{A}|}{1-\gamma}, C_4=\frac{2(1-\gamma^H)}{1-\gamma}, C_5=\frac{\gamma}{1-\gamma}\cdot\left(\frac{1-\gamma^H}{1-\gamma}-\gamma^{H-1}H\right)+\frac{1-\gamma^H}{1-\gamma}\cdot\frac{r_{max}}{1-\gamma}$. □

*Proof of Lemma 5.2.* The policy hold error can be divided into three terms:

$$\begin{aligned}\sum_{t\in\mathcal{N}_m}(V_t^*-V_t^{\pi_t}) &= \sum_{n=0}^{N_m-1}(V_{t_m+G_m+n}^*-V_{t_m+G_m+n}^{\pi_{t_m+G_m}})\\ &= \sum_{n=0}^{N_m-1}\Bigg(\underbrace{(V_{t_m+G_m+n}^*-V_{t_m+G_m}^*)}_{\text{(2-I)}}+\underbrace{(V_{t_m+G_m}^*-V_{t_m+G_m}^{\pi_{t_m+G_m}})}_{\text{(2-II)}}+\underbrace{(V_{t_m+G_m}^{\pi_{t_m+G_m}}-V_{t_m+G_m+n}^{\pi_{t_m+G_m}})}_{\text{(2-III)}}\Bigg).\end{aligned}$$

The terms (2-I), (2-II) and (2-III) can be bounded using Corollary E.3, Lemma E.1 and Lemma E.4. Recall that we have defined the time interval $\mathcal{N}_m=[t_m+G_m,t_{m+1})$, where $t_{m+1}=t_m+G_m+N_m$. One can write:

- $V_{t_m+G_m+n}^*-V_{t_m+G_m}^*\le\frac{1-\gamma^H}{1-\gamma}\Big(B_r(t_m+G_m,t_m+G_m+n)+\frac{r_{max}}{1-\gamma}B_p(t_m+G_m,t_m+G_m+n)\Big)$
- $V_{t_m+G_m}^*-V_{t_m+G_m}^{\pi_{t_m+G_m}}\le(\gamma+2)((1-\eta\tau)^{G_m}C')+\frac{2(\gamma+2)}{1-\gamma}\Big(1+\frac{\gamma}{\eta\tau}\Big)\cdot\epsilon_f+\frac{2\tau\log|\mathcal{A}|}{1-\gamma}$
- $V_{t_m+G_m}^{\pi_{t_m+G_m}}-V_{t_m+G_m+n}^{\pi_{t_m+G_m}}\le\frac{1-\gamma^H}{1-\gamma}\cdot B_r(t_m+G_m,t_m+G_m+n)+\frac{\gamma}{1-\gamma}\cdot\Big(\frac{1-\gamma^H}{1-\gamma}-\gamma^{H-1}H\Big)\cdot B_p(t_m+G_m,t_m+G_m+n).$

Now, taking the summation over $n=0,1,...,N_m-1$ leads to

$$\begin{aligned}\sum_{t\in\mathcal{N}_m}(V_t^*-V_t^{\pi_t}) \le& N_m\cdot\Big((\gamma+2)((1-\eta\tau)^{G_m}C')+\frac{2(\gamma+2)}{1-\gamma}\Big(1+\frac{\gamma}{\eta\tau}\Big)\cdot\epsilon_f+\frac{2\tau\log|\mathcal{A}|}{1-\gamma}\Big)\\ &+\frac{1-\gamma^H}{1-\gamma}\cdot\Big(\sum_{n=0}^{N_m-1}B_r(t_m+G_m,t_m+G_m+n)\Big)\\ &+\frac{\gamma}{1-\gamma}\cdot\Big(\frac{1-\gamma^H}{1-\gamma}-\gamma^{H-1}H\Big)\cdot\Big(\sum_{n=0}^{N_m-1}B_p(t_m+G_m,t_m+G_m+n)\Big)\\ &+\frac{1-\gamma^H}{1-\gamma}\Big(\sum_{n=0}^{N_m-1}B_r(t_m+G_m,t_m+G_m+n)+\frac{r_{max}}{1-\gamma}\sum_{n=0}^{N_m-1}B_p(t_m+G_m,t_m+G_m+n)\Big)\\ \le& N_m\cdot\Big((\gamma+2)((1-\eta\tau)^{G_m}C')+\frac{2(\gamma+2)}{1-\gamma}\Big(1+\frac{\gamma}{\eta\tau}\Big)\cdot\epsilon_f+\frac{2\tau\log|\mathcal{A}|}{1-\gamma}\Big)\\ &+\frac{2(1-\gamma^H)}{1-\gamma}\cdot\big(\bar{B}_r(t_m+G_m,t_m+G_m+N_m-1)\big)\\ &+\Big(\frac{\gamma}{1-\gamma}\cdot\Big(\frac{1-\gamma^H}{1-\gamma}-\gamma^{H-1}H\Big)+\frac{1-\gamma^H}{1-\gamma}\cdot\frac{r_{max}}{1-\gamma}\Big)\cdot\big(\bar{B}_r(t_m+G_m,t_m+G_m+N_m-1)\big)\\ =& N_m\cdot\big(C_1(1-\eta\tau)^{G_m}+C_2\epsilon_f+C_3\big)+C_4\bar{B}_r(\mathcal{N}_m)+C_5\bar{B}_p(\mathcal{N}_m)\end{aligned}$$

where $C_1,C_2,C_3,C_4,C_5$ are the constants defined in the Lemma 5.1. □

*Proof of Theorem 5.3.* Note that the following relationship holds for the dynamic regret $\mathfrak{R}(T)$:

$$\mathfrak{R}(T)=\sum_{m=1}^{M}\Bigg(\underbrace{\sum_{t\in\mathcal{G}_m}(V_t^*-V_t^{\pi_t})}_{\text{Policy update error}}+\underbrace{\sum_{t\in\mathcal{N}_m}(V_t^*-V_t^{\pi_t})}_{\text{Policy hold error}}\Bigg).$$

We use Lemma 5.1 to upper bound the use policy update error and use Lemma 5.2 to upper bound the policy hold error.

This leads to

$$\begin{aligned}\mathfrak{R}(T) &= \sum_{m=1}^{M}\left(\sum_{t\in\mathcal{G}_m}(V_t^* - V_t^{\pi_t}) + \sum_{t\in\mathcal{N}_m}(V_t^* - V_t^{\pi_t})\right)\\ &\le \sum_{m=1}^{M}\left(\frac{C_1}{\eta\tau}\cdot\Big(1-(1-\eta\tau)^{G_m}\Big) + G_m\Big(C_2\delta_m^f + C_3\Big) + C_4\bar{B}_r(\mathcal{G}_m) + C_5\bar{B}_p(\mathcal{G}_m)\right.\\ &\quad\left. + N_m\cdot\Big(C_1(1-\eta\tau)^{G_m} + C_2\delta_m^f + C_3\Big) + C_4\bar{B}_r(\mathcal{N}_m) + C_5\bar{B}_p(\mathcal{N}_m)\right)\\ &= \sum_{m=1}^{M}\left(\frac{C_1}{\eta\tau} + \left(N_mC_1 - \frac{C_1}{\eta\tau}\right)(1-\eta\tau)^{G_m} + (N_m+G_m)(C_2\delta_m^f + C_3) + \bar{B}(t_m, t_{m+1})\right).\end{aligned}$$

□

*Proof of Lemma 5.4.* The reader is referred to the proof of Theorem 5.8. □

*Proof of Proposition 5.6.* For fixed $t_m, t_{m+1}$, note that $\bar{B}(t_m, t_{m+1})$ is a function of $G_m, N_m$ with the constraint $G_m + N_m = t_{m+1} - t_m$. In this proof, we let $\bar{B}(t_m, t_{m+1})$ to be denoted as a function $g(G_m, N_m)$. Recall that we have defined $\bar{B}(t_{m+1}, t_m) := \bar{B}(\mathcal{N}_m) + \bar{B}(\mathcal{G}_m)$. Now, since $g(0, t_{m+1} - t_m) = g(t_{m+1} - t_m, 0) = \sum_{t=t_m}^{t_{m+1}-1}(C_4B_r(t_m, t) + C_5B_p(t_m, t))$, it is sufficient to show the existence of $G_m^\dagger \in (0, t_{m+1}, t_m)$ and $N_m^\dagger \in (0, t_{m+1}, t_m)$ that satisfy $g(G_m^\dagger, N_m^\dagger) < g(0, t_{m+1} - t_m) = g(t_{m+1} - t_m, 0)$. By the definition of non-stationary environments (see Definition 3.4), let $t_1^\dagger, t_2^\dagger$ satisfy $B_r(t_1^\dagger, t_2^\dagger) > 0$ or $B_p(t_1^\dagger, t_2^\dagger) > 0$. Now, letting $G_m^\dagger = t_2^\dagger$, we have $B_r(t_m, G_m^\dagger) > 0$ or $B_p(t_m, G_m^\dagger) > 0$. As a result, either $\sum_{t_m}^{t_m+G_m^\dagger-1} B_r(t_m, t) + \sum_{t_m+G_m^\dagger}^{t_{m+1}-1} B_r(t_m + G_m^\dagger, t) < \sum_{t_m}^{t_{m+1}-1} B_r(t_m, t)$ or $\sum_{t_m}^{t_m+G_m^\dagger-1} B_p(t_m, t) + \sum_{t_m+G_m^\dagger}^{t_{m+1}-1} B_p(t_m + G_m^\dagger, t) < \sum_{t_m}^{t_{m+1}-1} B_p(t_m, t)$ holds. Now, by combining the two inequalities with the constants $C_4, C_5 > 0$ defined in Lemma 5.1, we obtain that

$$\begin{aligned}&C_4\bar{B}_r(t_m, t_m + G_m^\dagger) + C_4\bar{B}_r(t_m + G_m^\dagger, T_{m+1}) + C_5\bar{B}_p(t_m, t_m + G_m^\dagger) + C_5\bar{B}_p(t_m + G_m^\dagger, t_{m+1})\\ &< C_4\bar{B}_r(t_m, t_{m+1}) + C_5\bar{B}_p(t_m, t_{m+1})\end{aligned}$$

if and only if

$$\bar{B}(t_m, t_m + G_m^\dagger) + \bar{B}(t_m + G_m^\dagger, t_{m+1}) < \bar{B}(t_m, t_{m+1}).$$

Therefore, $G_m^\dagger = t_2^\dagger, N_m^\dagger = t_{m+1} - t_m - t_m^\dagger$ satisfies the condition $g(G_m^\dagger, N_m^\dagger) < g(0, t_{m+1} - t_m) = g(t_{m+1} - t_m, 0)$. This completes the proof. □

*Proof of Theorem 5.8.* We first show that the policy optimization error is a convex function of $G_m$ (or $N_m$. Let $f_1(N_m, G_m) = C_1(1 - (1-\eta\tau)^{G_m}) + N_mC_1(1-\eta\tau)^{G_m}$, where $N_m + G_m = t_{m+1} - t_m$ is a constant. Note that $\partial N_m/\partial G_m = -1$. It holds that

$$\frac{1}{C_1}\cdot\frac{\partial f_1}{\partial G_m} = \big\{\ln(1-\eta\tau)\,(N_m - 1) - 1\big\}(1-\eta\tau)^{G_m}$$

and

$$\frac{1}{C_1}\cdot\frac{\partial^2 f_1}{\partial G_m^2} = \big\{(\ln(1-\eta\tau))^2(N_m - 1) - 2\ln(1-\eta\tau)\big\}(1-\eta\tau)^{G_m}.$$

Therefore, $\partial^2 f_1/\partial G_m^2 > 0$ and $\partial^2 f_1/\partial N_m^2 > 0$ holds for $\forall N_m, G_m \ge 0$, where $N_m + G_m = t_{m+1} - t_m$ holds. The non-stationary terms are bounded as follows:

$$\bar{B}(\mathcal{N}_m) + \bar{B}(\mathcal{G}_m) = (C_4 + C_5)\left(\bar{B}_r(\mathcal{N}_m) + \bar{B}_r(\mathcal{G}_m)\right).$$

Note that by Assumption 3.1, $\bar{B}_r(\mathcal{N}_m) \leq \sum_{t=t_m+G_m}^{t=t_m+G_m+N_m-1} \alpha_r^{t-(t_m+G_m)} B^{\max}(\mathcal{N}_m)$ and $\bar{B}_r(\mathcal{G}_m) \leq \sum_{t=t_m}^{t=t_m+G_m-1} \alpha_r^{t-t_m} B^{\max}(\mathcal{G}_m)$. For the short notation, we use $\alpha_\diamond(\mathcal{G}_m) = \alpha_{\diamond,1}, \alpha_\diamond(\mathcal{N}_m) = \alpha_{\diamond,2}$ and $B_\diamond^{\max}(\mathcal{G}_m) = B_{\diamond,1}^{\max}, B_\diamond^{\max}(\mathcal{N}_m) = B_{\diamond,2}^{\max}$ where $\diamond = r$ or $p$. Also, we let $\alpha_\square = \max(\alpha_{r,\square}, \alpha_{p,\square})$ and $B_\square^{\max} = \max(B_{r,\square}^{\max}, B_{p,\square}^{\max})$, where $\square = 1$ or 2. One can write:

$$\begin{aligned}\bar{B}(\mathcal{N}_m) + \bar{B}(\mathcal{G}_m) &= C_4 \left(\bar{B}_r(\mathcal{N}_m) + \bar{B}_r(\mathcal{G}_m)\right) + C_5 \left(\bar{B}_p(\mathcal{N}_m) + \bar{B}_p(\mathcal{G}_m)\right) \\ &\leq (C_4 + C_5) \cdot \left( \frac{\alpha_1^{G_m} - 1}{\alpha_1 - 1} \cdot B_1^{\max} + \frac{\alpha_2^{N_m} - 1}{\alpha_2 - 1} \cdot B_2^{\max} \right).\end{aligned}$$

We denote the upper bound as a function $f_2(N_m, G_m)$. Note that $B_1^{\max}$ and $B_2^{\max} > 0$ hold for a non-stationary environment. If $0 < \alpha_1, \alpha_2 < 1$, then $f_2(N_m, G_m)$ is a concave function with respect to $(N_m, G_m)$. If $\alpha_1, \alpha_2 > 1$, then $f_2(N_m, G_m)$ is a convex function with respect to $(N_m, G_m)$. □

## E. Supplementary lemmas

**Lemma E.1** (NPG Convergence)**.** *Assume that we have an inexact Q value estimation at time $t_m + G_m - 1$, $\hat{Q}_{t_m+G_m-1}$, where we denote $Q_{t_m+G_m-1}$ as the exact Q value. Now, define the error of estimation as $\epsilon$, that is, $||Q_{t_m+G_m-1} - \hat{Q}_{t_m+G_m-1}||_\infty \leq \epsilon_f$. For any $g \in [G_m]$, it holds that*

$$V_{t_m+G_m-1}^* - V_{t_m+G_m-1}^{\pi^g} \leq (\gamma+2)((1-\eta\tau)^{g-1} C_1) + \frac{2(\gamma+2)}{1-\gamma}\left(1 + \frac{\gamma}{\eta\tau}\right) \cdot \epsilon_f + \frac{2\tau \log|\mathcal{A}|}{1-\gamma}$$

*where*

$$C_1 = \left|\left| Q_\tau^* - Q_\tau^{(0)} \right|\right|_\infty + 2\tau \left(1 - \frac{n\tau}{1-\gamma}\right) \left|\left| \log \pi_\tau^* - \log \pi^{(0)} \right|\right|_\infty.$$

*Proof of Lemma E.1.* We omit the underscript $t$ for simplicity of notation, i.e., $V_t, V_{\tau,t}, V_t^*$ denote $V, V_\tau, V^*$, respectively. For any $m \in [M]$ and any $t \in \mathcal{G}_m$, the inequality

$$\begin{aligned}V^*(s) - V(s) &\leq ||V^*(\cdot) - V_\tau^*(\cdot)||_\infty + ||V_\tau^*(\cdot) - V_\tau(\cdot)||_\infty + ||V_\tau(\cdot) - V(\cdot)||_\infty \\ &\leq ||V_\tau^*(\cdot) - V_\tau(\cdot)||_\infty + \frac{2\tau \log|\mathcal{A}|}{1-\gamma}\end{aligned}$$

holds since for any policy $\pi$, $||V_\tau^\pi - V^\pi||_\infty = \tau \max_s |\mathcal{H}(s,\pi)| \leq \frac{\tau \log|\mathcal{A}|}{1-\gamma}$ holds. Now, note that $V_\tau$ is a value function of a policy $\pi_\tau$ that we obtain after updating $g$ iterations. As a result,

$$\begin{aligned}||V_\tau^*(\cdot) - V_\tau(\cdot)||_\infty &\leq \tau \, ||\log \pi_\tau^* - \log \pi_\tau^g||_\infty + ||Q_\tau^*(\cdot) - Q_\tau(\cdot)||_\infty \\ &\leq \tau \cdot \frac{2}{\tau}\left((1-\eta\tau)^{g-1} C_1 + C_2\right) + \gamma\left((1-\eta\tau)^{g-1} C_1 + C_2\right) \\ &= (\gamma+2)\left((1-\eta\tau)^{g-1} C_1 + C_2\right)\end{aligned} \tag{2}$$

where

$$C_1 = \left|\left| Q_\tau^* - Q_\tau^{(0)} \right|\right|_\infty + 2\tau \left(1 - \frac{n\tau}{1-\gamma}\right) \left|\left| \log \pi_\tau^* - \log \pi^{(0)} \right|\right|_\infty, \; C_2 = \frac{2\epsilon_f}{1-\gamma}\left(1 + \frac{\gamma}{\eta\tau}\right).$$

The equation (2) holds cue to Theorem 2 in (Cen et al., 2022). □

**Lemma E.2** (Difference between optimal state action value functions of two MDPs)**.** *For any two time steps $t_1, t_2 \in T$, we denote the optimal Q functions at step $h \in [H]$ as $Q_{t_1,h}^*(s,a), Q_{t_2,h}^*(s,a)$. Then, for any state and action pair $s, a \in \mathcal{S} \times \mathcal{A}$,*

$$Q_{t_1,h}^*(s,a) - Q_{t_2,h}^*(s,a) \leq \sum_{h'=h}^{H-1} \gamma^{h'-h} B_r(t_1,t_2) + \frac{r_{max}}{1-\gamma} \sum_{h'=h}^{H-1} \gamma^{h'-h} B_p(t_1,t_2)$$

*holds, where $B_r(t_1,t_2)$ and $B_p(t_1,t_2)$ denote the local time-elapsing variation budgets between the time steps $\{t_1, t_1+1, t_1+2, ..., t_2\}$.*

*Proof of Lemma E.2.* Only for the purpose of the proof of Lemma E.2, we define the state value function $V^{\pi}_{t,h} : \mathcal{S} \to \mathbb{R}$ and the state action value function $Q^{\pi}_{t,h} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ at step $h$ of time $t$ as

$$V^{\pi}_{t,h}(s) := \mathbb{E}_{\mathcal{M}_t}\left[\sum_{h'=h}^{H-1} \gamma^{h'-h} r_{t,h'} \,\middle|\, s^0_t = s\right]$$

and

$$Q^{\pi}_{t,h}(s,a) := \mathbb{E}_{\mathcal{M}_t}\left[\sum_{h'=h}^{H-1} \gamma^{h'-h} r_{t,h'} \,\middle|\, s^0_t = s, a^0_t = a\right].$$

Note that the optimal state value function and the state action value function satisfy the following Bellman equation.

$$Q^*_{t,h}(s,a) = \left(R_{t,h} + \gamma P_t V^*_{t,h}\right)(s,a), \pi^*_t = \arg\max_{a\in\mathcal{A}} Q^*_{t,h}(s,a).$$

The proof depends on a backward induction. First, the statement holds when $h = H-1$ since

$$\left|\left|Q^*_{t_1,H-1}(s,a) - Q^*_{t_2,H-1}(s,a)\right|\right|_\infty = ||r_{t_1,H-1} - r_{t_2,H-1}||_\infty = ||R_{t_1} - R_{t_2}||_\infty .$$

Now, we assume that the statement of Lemma E.2 holds for $h+1$. Then, for $h$ it holds that

$$\begin{aligned}
Q^*_{t_1,h}(s,a) - Q^*_{t_2,h}(s,a) &= \left(R_{t_1,h} - R_{t_2,h}\right)(s,a) + \gamma \sum_{s'\in\mathcal{S}} \left(P_{t_1}(s'|s,a) V^*_{t_1,h+1}(s') - P_{t_2}(s'|s,a) V^*_{t_2,h+1}(s')\right) \\
&\le B_r(t_1,t_2) + \gamma \sum_{s'\in\mathcal{S}} \left(P_{t_1}(s'|s,a) Q^*_{t_1,h+1}(s',\pi^*_{t_1}(s')) - P_{t_2}(s'|s,a) Q^*_{t_2,h+1}(s',\pi^*_{t_2}(s')(s')\right).
\end{aligned}$$

Then by the induction hypothesis on $h+1$, the following holds for any $s' \in \mathcal{S}$:

$$\begin{aligned}
Q^*_{t_1,h+1}(s',\pi^*_{t_1}(s')) &\le Q^*_{t_2,h+1}(s',\pi^*_{t_1}(s')) + \sum_{h'=h+1}^{H-1} \gamma^{h'-(h+1)} B_r(t_1,t_2) + \frac{r_{\max}}{1-\gamma} \sum_{h'=h+1}^{H-1} \gamma^{h'-(h+1)} B_p(t_1,t_2) \\
&\le Q^*_{t_2,h+1}(s',\pi^*_{t_2}(s')) + \sum_{h'=h+1}^{H-1} \gamma^{h'-(h+1)} B_r(t_1,t_2) + \frac{r_{\max}}{1-\gamma} \sum_{h'=h+1}^{H-1} \gamma^{h'-(h+1)} B_p(t_1,t_2).
\end{aligned}$$

Therefore,

$$\begin{aligned}
Q^*_{t_1,h}(s,a) - Q^*_{t_2,h''}(s,a) &\le B_r(t_1,t_2) + \gamma \sum_{s'\in\mathcal{S}} \left(\left(P_{t_1}(s'|s,a) - P_{t_2}(s'|s,a)\right) Q^*_{t_2,h+1}(s',\pi^*_{t_2}(s'))\right) \\
&\quad + \sum_{h'=h+1}^{H-1} \gamma^{h'-h} B_r(t_1,t_2) + \frac{r_{\max}}{1-\gamma} \sum_{h'=h+1}^{H-1} \gamma^{h'-h} B_p(t_1,t_2) \\
&\le \gamma \left|\left|\left(P_{t_1}(s'|s,a) - P_{t_2}(s'|s,a)\right)\right|\right|_1 \left|\left|Q^*_{t_2,h+1}(s',\pi^*_{t_2}(s'))\right|\right|_\infty + \sum_{h'=h}^{H-1} \gamma^{h'-h} B_r(t_1,t_2) \\
&\quad + \frac{r_{\max}}{1-\gamma} \sum_{h'=h+1}^{H-1} \gamma^{h'-h} B_p(t_1,t_2) \\
&\le \gamma B_p(t_1,t_2) \cdot \frac{r_{max}}{1-\gamma} + \sum_{h'=h}^{H-1} \gamma^{h'-h} B_r(t_1,t_2) + \frac{r_{\max}}{1-\gamma} \sum_{h'=h+1}^{H-1} \gamma^{h'-h} B_p(t_1,t_2) \\
&= \sum_{h'=h}^{H-1} \gamma^{h'-h} B_r(t_1,t_2) + \frac{r_{\max}}{1-\gamma} \sum_{h'=h}^{H-1} \gamma^{h'-h} B_p(t_1,t_2).
\end{aligned}$$

This completes the proof. □

**Corollary E.3** (Difference between optimal state value functions of two MDPs)**.** *For any two times $t_1 < t_2 \in T$, the gap between the two value functions at times $t_1$ and $t_2$ is bounded as*

$$||V^*_{t_1}(s) - V^*_{t_2}(s)||_\infty \leq \frac{1-\gamma^H}{1-\gamma}\left(B_r(t_1,t_2) + \frac{r_{max}}{1-\gamma}B_p(t_1,t_2)\right).$$

*Proof of Corollary E.3.* Corollary E.3 comes from Lemma E.2. □

**Lemma E.4** (Difference between value functions of two MDPs with same policy)**.** *For any two times $t_1, t_2 \in T$, any policy $\pi$, and any state $s \in \mathcal{S}$, the gap between the two value functions $V^\pi_{t_1}$ and $V^\pi_{t_2}$ is bounded as follows:*

$$V^\pi_{t_1}(s) - V^\pi_{t_2}(s) \leq \frac{1-\gamma^H}{1-\gamma}\cdot B_r(t_1,t_2) + \frac{\gamma}{1-\gamma}\cdot\left(\frac{1-\gamma^H}{1-\gamma} - \gamma^{H-1}H\right)\cdot B_p(t_1,t_2).$$

*Proof of Lemma E.4.* For a given initial state $s_0$, we first define the occupancy measure of state and action $(s,a)$ as

$$\rho^\pi_t(s,a) := \sum_{h=0}^{H-1}\gamma^h \mathbb{P}\left(s_h = s, a_h = a | P_t, \pi\right).$$

It is worth noting that $\mathbb{P}\left(s_h = s, a_h = a|P_t,\pi\right) = \mathbb{P}\left(s_h = s|P_t,\pi\right)\cdot\pi(a_h = a|s_s = s)$. Now, note that the value function can be rewritten using the occupancy measure as

$$V^\pi_t(s) := \mathbb{E}_{\mathcal{M}_t}\left[\sum_{h=0}^{H-1}\gamma^h r_{t,h} \mid s^0_t = s\right] = \mathbb{E}_{(s,a)\sim\rho^\pi_t}\left[R_t(s,a)\right].$$

Then for any $t_1, t_2 \in T$, the gap between the two value functions can be expressed as

$$\begin{aligned}
V^\pi_{t_1}(s) - V^\pi_{t_2}(s) &= \mathbb{E}_{(s,a)\sim d^\pi_{t_1}}\left[R_{t_1}(s,a)\right] - \mathbb{E}_{(s,a)\sim d^\pi_{t_2}}\left[R_{t_2}(s,a)\right] \\
&= \mathbb{E}_{(s,a)\sim d^\pi_{t_1}}\left[R_{t_1}(s,a) - R_{t_2}(s,a)\right] - \mathbb{E}_{(s,a)\sim d^\pi_{t_1}}\left[R_{t_2}(s,a)\right] + \mathbb{E}_{(s,a)\sim d^\pi_{t_2}}\left[R_{t_2}(s,a)\right] \\
&\leq \frac{1-\gamma^H}{1-\gamma}\cdot\max_{(s,a)}\left(|R_{t_2}(s,a) - R_{t_1}(s,a)|\right) + \left(\mathbb{E}_{(s,a)\sim d^\pi_{t_2}}\left[R_{t_2}(s,a)\right] - \mathbb{E}_{(s,a)\sim d^\pi_{t_1}}\left[R_{t_2}(s,a)\right]\right) \\
&= \frac{1-\gamma^H}{1-\gamma}\cdot B_r(t_1,t_2) + \left(\mathbb{E}_{(s,a)\sim d^\pi_{t_2}}\left[R_{t_2}(s,a)\right] - \mathbb{E}_{(s,a)\sim d^\pi_{t_1}}\left[R_{t_2}(s,a)\right]\right). \qquad (3)
\end{aligned}$$

Now, the gap $\mathbb{E}_{(s,a)\sim d^\pi_{t_2}}\left[R_{t_2}(s,a)\right] - \mathbb{E}_{(s,a)\sim d^\pi_{t_1}}\left[R_{t_2}(s,a)\right]$ is upper bounded as follows:

$$\begin{aligned}
\mathbb{E}_{(s,a)\sim d^\pi_{t_2}}\left[R_{t_2}(s,a)\right] - \mathbb{E}_{(s,a)\sim d^\pi_{t_1}}\left[R_{t_2}(s,a)\right] &\leq ||\rho^\pi_{t_2}(\cdot,\cdot) - \rho^\pi_{t_1}(\cdot,\cdot)||_1 \cdot ||R_{t_2}(\cdot,\cdot)||_\infty \\
&= ||\rho^\pi_{t_2}(\cdot,\cdot) - \rho^\pi_{t_1}(\cdot,\cdot)||_1 \cdot r_{\max}. \qquad (4)
\end{aligned}$$

Now, the term $\sum_{(s,a)} \left|\rho^{\pi}_{t_2}(s,a) - \rho^{\pi}_{t_1}(s,a)\right|$ is bounded as follows:

$$\begin{aligned}
\sum_{(s,a)} \left|\rho^{\pi}_{t_2}(s,a) - \rho^{\pi}_{t_1}(s,a)\right| &= \sum_{(s,a)} \left| \sum_{h=0}^{H-1} \left( \gamma^h \cdot \left( \mathbb{P}\left(s_h = s | P_{t_2}, \pi\right) - \mathbb{P}\left(s_h = s | P_{t_1}, \pi\right) \right) \cdot \pi(a_h = a | s_h = s) \right) \right| \\
&= \sum_{(s,a)} \left( \sum_{h=0}^{H-1} \left| \gamma^h \cdot \left( \mathbb{P}\left(s_h = s | P_{t_2}, \pi\right) - \mathbb{P}\left(s_h = s | P_{t_1}, \pi\right) \right) \right| \cdot \left| \pi(a_h = a | s_h = s) \right| \right) \\
&= \sum_{s} \left( \sum_{h=0}^{H-1} \left| \gamma^h \cdot \left( \mathbb{P}\left(s_h = s | P_{t_2}, \pi\right) - \mathbb{P}\left(s_h = s | P_{t_1}, \pi\right) \right) \right| \cdot \sum_{a \in \mathcal{A}} \left| \pi(a_h = a | s_h = s) \right| \right) \\
&= \sum_{s \in \mathcal{S}} \left( \sum_{h=0}^{H-1} \left| \gamma^h \cdot \left( \mathbb{P}\left(s_h = s | P_{t_2}, \pi\right) - \mathbb{P}\left(s_h = s | P_{t_1}, \pi\right) \right) \right| \cdot 1 \right) \\
&= \sum_{h=0}^{H-1} \gamma^h \cdot \left( \sum_{s \in \mathcal{S}} \left| \left( \mathbb{P}\left(s_h = s | P_{t_2}, \pi\right) - \mathbb{P}\left(s_h = s | P_{t_1}, \pi\right) \right) \right| \right). \qquad (5)
\end{aligned}$$

Now, for simplicity of notation, we denote $\mathbb{P}(s_h = s | P_t, \pi)$ as $\mathbb{P}^h_t(s)$, $P_t(s_h = s | s_{h-1} = s', a_{h-1} = a')$ as $\mathbb{P}^h_t(s|s',a')$, and $\pi(a_h = a | s_h = s)$ as $\pi^h(a|s)$. Then, we have

$$\begin{aligned}
&\sum_{s \in \mathcal{S}} \left| \mathbb{P}^h_{t_2}(s) - \mathbb{P}^h_{t_1}(s) \right| \\
&= \sum_{s \in \mathcal{S}} \left| \sum_{s',a'} \left( P^h_{t_2}(s|s',a') \cdot \pi^{h-1}(a'|s') \cdot \mathbb{P}^{h-1}_{t_2}(s') - P^h_{t_1}(s|s',a') \cdot \pi^{h-1}(a'|s') \cdot \mathbb{P}^{h-1}_{t_1}(s') \right) \right| \\
&\leq \sum_{s \in \mathcal{S}} \sum_{s',a'} \left| \left( P^h_{t_2}(s|s',a') \cdot \mathbb{P}^{h-1}_{t_2}(s') - P^h_{t_1}(s|s',a') \cdot \mathbb{P}^{h-1}_{t_1}(s') \right) \cdot \pi^{h-1}(a'|s') \right| \\
&= \sum_{s',a'} \sum_{s \in \mathcal{S}} \left| \left( P^h_{t_2}(s|s',a') \cdot \mathbb{P}^{h-1}_{t_2}(s') - P^h_{t_1}(s|s',a') \cdot \mathbb{P}^{h-1}_{t_1}(s') \right) \cdot \pi^{h-1}(a'|s') \right| \\
&\leq \sum_{s',a'} \sum_{s \in \mathcal{S}} \Bigg( \left| \left( P^h_{t_2}(s|s',a') - P^h_{t_1}(s|s',a') \right) \cdot \mathbb{P}^{h-1}_{t_2}(s') \cdot \pi^{h-1}(a'|s') \right| \\
&\quad + \left| \left( \mathbb{P}^{h-1}_{t_2}(s') - \mathbb{P}^{h-1}_{t_1}(s') \right) \cdot P^h_{t_1}(s|s',a') \cdot \pi^{h-1}(a'|s') \right| \Bigg) \\
&\leq \max_{s',a'} \left( ||P^h_{t_2}(\cdot|s',a') - P^h_{t_1}(\cdot|s',a')||_1 \right) \cdot \left( \sum_{s',a'} \left( \mathbb{P}^{h-1}_{t_2}(s') \cdot \pi^{h-1}(a'|s') \right) \right) \\
&\qquad + \left( \sum_{s',a'} \left( \left| \mathbb{P}^{h-1}_{t_2}(s') - \mathbb{P}^{h-1}_{t_1}(s') \right| \right) \cdot \pi^{h-1}(a'|s') \right) \cdot \left( \sum_{s \in \mathcal{S}} P^h_{t_1}(s|s',a') \right) \\
&= B_p(t_1,t_2) \cdot \left( \sum_{s' \in \mathcal{S}} \mathbb{P}^{h-1}_{t_2}(s') \cdot \sum_{a' \in \mathcal{A}} \pi^{h-1}(a'|s') \right) + \left( \sum_{s' \in \mathcal{S}} \left| \mathbb{P}^{h-1}_{t_2}(s') - \mathbb{P}^{h-1}_{t_1}(s') \right| \cdot \sum_{a' \in \mathcal{A}} \pi^{h-1}(a'|s') \right) \cdot 1 \\
&= B_p(t_1,t_2) + \sum_{s' \in \mathcal{S}} \left| \mathbb{P}^{h-1}_{t_2}(s') - \mathbb{P}^{h-1}_{t_1}(s') \right|.
\end{aligned}$$

Now, note that $\sum_{s \in \mathcal{S}} \left| \mathbb{P}^0_{t_2}(s) - \mathbb{P}^0_{t_1}(s) \right| = 0$ and $\sum_{s \in \mathcal{S}} \left| \mathbb{P}^1_{t_2}(s) - \mathbb{P}^1_{t_1}(s) \right| = B_p(t_1,t_2)$ hold. Therefore,

$$\sum_{s \in \mathcal{S}} \left| \mathbb{P}^h_{t_2}(s) - \mathbb{P}^h_{t_1}(s) \right| \leq h B_p(t_1,t_2)$$

holds. Then, substituting the above inequality into the inequality (5) gives that

$$\begin{aligned}\sum_{(s,a)} \left|\rho^{\pi}_{t_2}(s,a) - \rho^{\pi}_{t_1}(s,a)\right| &\leq \sum_{h=0}^{H-1} \gamma^h h B_p(t_1,t_2) \\ &\leq \frac{\gamma}{1-\gamma} \cdot \left(\frac{1-\gamma^H}{1-\gamma} - \gamma^{H-1}H\right) \cdot B_p(t_1,t_2).\end{aligned}$$

Now, it follows from the inequalities (3) and (4) that

$$V^{\pi}_{t_1}(s) - V^{\pi}_{t_2}(s) \leq \frac{1-\gamma^H}{1-\gamma} \cdot B_r(t_1,t_2) + \frac{\gamma}{1-\gamma} \cdot \left(\frac{1-\gamma^H}{1-\gamma} - \gamma^{H-1}H\right) \cdot B_p(t_1,t_2).$$

□

## F. Experiment Platforms and Licenses

### F.1. Platforms

All experiments are conducted on 12 Intel Xeon CPU E5-2690 v4 and 2 Tesla V100 GPUs.

### F.2. Licenses

We have used the following libraries/ repos for our Python codes:

- Pytorch (BSD 3-Clause "New" or "Revised" License).
- OpenAI Gym (MIT License).
- Numpy (BSD 3-Clause "New" or "Revised" License).
- Official codes distributed from *https://github.com/pranz24/pytorch-soft-actor-critic*: to compare the performance of SAC and FSAC in the Mujoco environment.
- Official codes distributed from the *https://github.com/linesd/tabular-methods*: to compare SAC and FSAC in the goal-switching cliff world.