Accelerating Proximal Policy Optimization Learning Using Task Prediction for Solving Environments with Delayed Rewards

Ahmad Ahmad AHMADGH@BU.EDU
Boston University, Boston, MA, USA

Mehdi Kermanshah MKER@BU.EDU
Boston University, Boston, MA, USA

Kevin Leahy KLEAHY@WPI.EDU
Worcester Polytechnic Institute, Worcester, MA, USA

Zachary Serlin ZACHARY.SERLIN@LL.MIT.EDU
MIT Lincoln Laboratory, Lexington, MA, USA

Ho Chit Siu HOCHIT.SIU@LL.MIT.EDU
MIT Lincoln Laboratory, Lexington, MA, USA

Makai Mann MAKAIM@MIT.EDU
MIT Lincoln Laboratory, Lexington, MA, USA

Cristian-Ioan Vasile CVR519@LEHIGH.EDU
Lehigh University, Bethlehem, PA, USA

Roberto Tron TRON@BU.EDU
Boston University, Boston, MA, USA.

Calin Belta CALIN@UMD.EDU
University of Maryland, College Park, MD, USA

Editors: N. Ozay, L. Balzano, D. Panagou, A. Abate

Abstract

In this paper, we tackle the challenging problem of delayed rewards in reinforcement learning (RL). While Proximal Policy Optimization (PPO) has emerged as a leading Policy Gradient method, its performance can degrade under delayed rewards. We introduce two key enhancements to PPO: a hybrid policy architecture that combines an offline policy (trained on expert demonstrations) with an online PPO policy, and a reward shaping mechanism using Time Window Temporal Logic (TWTL). The hybrid architecture leverages offline data throughout training while maintaining PPO's theoretical guarantees. Building on the monotonic improvement framework of Trust Region Policy Optimization (TRPO), we prove that our approach ensures improvement over both the offline policy and previous iterations, with a bounded performance gap of $(2\varsigma\gamma\alpha^2)/(1-\gamma)^2$, where α is the mixing parameter, γ is the discount factor, and ς bounds the expected advantage. Additionally, we prove that our TWTL-based reward shaping preserves the optimal policy of the original problem. TWTL enables formal translation of temporal objectives into immediate feedback signals that guide learning. We demonstrate the effectiveness of our approach through extensive experiments on an inverted pendulum and a lunar lander environment, showing improvements in both learning speed and final performance compared to standard PPO and offline-only approaches.

Keywords: Policy Gradient Methods, Behavior Cloning, Temporal Logic, Reward Shaping, and Trajectory Prediction.

1. Introduction

Reinforcement learning (RL) in environments with delayed rewards presents a fundamental challenge: actions that lead to successful outcomes may not receive immediate positive feedback, making it difficult for learning algorithms to identify and reinforce beneficial behaviors. This challenge is particularly acute in complex games like soccer, where the value of tactical decisions (e.g., passing plays) may only become apparent after multiple time steps later through their impact on strategic outcomes (e.g., scoring opportunities). Policy Gradient (PG) methods, particularly Proximal Policy Optimization (PPO) Schulman et al. (2017), have demonstrated remarkable success in RL tasks. PPO's effectiveness stems from its optimization of a surrogate objective that provides stable policy updates while maximizing expected rewards. However, in settings with delayed rewards, even PPO's carefully constructed optimization landscape becomes difficult to navigate, as the temporal gap between actions and their consequences creates a sparse and uninformative gradient signal.

Our work addresses these challenges through three main contributions. First, we introduce a hybrid policy architecture that combines an offline policy (trained on expert demonstrations) with an online PPO policy. Unlike previous approaches that use offline data merely for initialization Nair et al. (2020); Ross and Bagnell (2012); Schmitt et al. (2018); Chen et al. (2021); Kumar et al. (2020), our method maintains the offline policy as an active guide throughout training. We prove that this architecture guarantees monotonic improvement over both the offline policy and previous iterations. Second, we develop a reward-shaping mechanism using Time Window Temporal Logic (TWTL) Vasile et al. (2017) that provides immediate, semantically meaningful feedback while preserving the optimal policy of the original problem. This approach bridges the temporal gap in the reward signal by formally encoding desired temporal behaviors. Third, we extend existing convergence proofs for actor-critic methods to handle our hybrid architecture and reward shaping mechanism, providing theoretical foundations for our approach. Our analysis demonstrates that the proposed method maintains PPO's convergence properties, while accelerating learning in delayed-reward settings.

Related work. Our work builds upon three research directions: policy optimization, temporal logics in RL, and offline RL. In policy optimization, Trust Region Policy Optimization (TRPO) Schulman et al. (2015a) and PPO Schulman et al. (2017) provides stability guarantees. However, these methods, while effective for standard RL tasks, do not specifically address the challenges of delayed rewards. Our work extends PPO by incorporating temporal logic guidance and offline data while preserving its theoretical guarantees.

Most of the works that have investigated the use of temporal logics (TL) in RL primarily focus on task specification Asarkaya et al. (2021); Cai et al. (2023); Icarte et al. (2022); Xu et al. (2020); Neider et al. (2021); Alshiekh et al. (2018); Balakrishnan and Deshmukh (2019); Li et al. (2017); Aksaray et al. (2016). TLs formalize high-level tasks into propositional and temporal constraints Baier and Katoen (2008), with some, like Time Window Temporal Logic (TWTL) Vasile et al. (2017); Ahmad et al. (2023), handling delayed rewards through temporal constraints Aksaray et al. (2016); Balakrishnan and Deshmukh (2019). TWTL's clear syntax efficiently expresses sequential tasks. In Asarkaya et al. (2021), a Q-learning algorithm learns an optimal policy for TWTL-modeled tasks while maximizing external rewards. Our approach differs by actively using TWTL for reward shaping while maintaining optimality guarantees. Recent works like Cai et al. (2023) and Icarte et al. (2022) have shown promising results in combining TL with deep RL, but primarily focus on task specification rather than reward shaping.

In offline RL, the authors of Ball et al. (2023) developed an algorithm that has shown the benefits of learning from demonstrations, but typically treats offline data as a fixed dataset rather than an active component of the policy. In Hu et al. (2023), the policy chooses between an imitation learning (IL) policy, trained offline, and an online RL policy based on the action with a higher Q-value. Our work differs in that we consider the choosing mechanism as a learnable parameter in the context of deep RL.

Paper structure. The rest of the paper is organized as follows. Sec. 2 contains definitions and preliminaries. In Sec. 4, we formally state the problem of learning with delayed rewards using TWTL specifications. Sec. 4 introduces our main theoretical contributions: the hybrid policy architecture and TWTL-based reward shaping, along with convergence guarantees. In Sec. 5, we present our case study using some benchmark gymnasium environments Towers et al. (2023), namely the inverted pendulum and the lunar lander, including the task predictor architecture and experimental results. The paper concludes with a discussion of limitations and future work in Sec. 6.

2. Preliminaries

Finite-horizon Markov Decision Process (MDP)

Definition 1 *A finite horizon MDP is a tuple* $(\mathcal{X}, \mathcal{U}, p(\cdot|\cdot,\cdot), r(\cdot,\cdot), l(.), \mathcal{O})$*, where* $\mathcal{X}$*,* $\mathcal{U}$ *and* $\mathcal{O}$ *are the state, control, and output spaces, respectively;* $p(\cdot|\cdot,\cdot)$ *is the state-action pair transition probability;* $r : \mathcal{X} \times \mathcal{U} \mapsto [0,1]$ *is the reward function; and* $l : \mathcal{X} \mapsto \mathcal{O}$ *is a labeling function that maps the state to an output observation.*

We denote a state trajectory of the MDP as $\mathbf{x}_{i,i+N} := x_i x_{i+1} \dots x_{i+N}$, where $x_i \in \mathcal{X}$ and $i \in \mathbb{N}$. $\mathbf{x}_{i,i+N}$ generates a word, $\mathbf{o}_{i,i+N} = o_i o_{i+1} \dots o_{i+N}$, where $o_i = l(x_i)$, where $o_i \in \mathcal{O}$. Let $\pi : \mathcal{X} \mapsto \mathcal{U}$ be a stochastic policy. In an episodic RL setting Sutton and Barto (2018), for K learning episodes, the state value function and the state-action value, at iteration t of episode k, are defined, respectively, as follows Schulman et al. (2017):

$$V_t^{\pi,k}(x) := \mathop{\mathrm{E}}_{\pi}\left[\sum_{i=t}^{N} r_i^k(x_i,u_i)\middle| x_t = x\right], \quad Q_t^{\pi,k}(x,u) := \mathop{\mathrm{E}}_{\pi}\left[\sum_{i=t}^{N} r_i^k(x_i,u_i)\middle| x_t = x,\ u_t = u\right]. \tag{1}$$

where E_π is the expectation over the stochastic policy π. Correspondingly, the advantage function at iteration t of episode k:

$$A_t^{\pi,k}(x,u) := Q_t^{\pi,k}(x,u) - V_t^{\pi,k}(x) \tag{2}$$

quantifies how much better (or worse) an action u is compared to the average action that would be taken by policy π in state x. A positive advantage indicates that action u is better than π's average action, while a negative advantage suggests it is worse. This function plays a crucial role in policy gradient methods by identifying which actions to encourage or discourage during policy updates.

Time Window Temporal Logic We use TWTL to define tasks and to create reward functions that guide the agent towards specific goals.

An atomic proposition, $\mathrm{AP} \in \Pi$ (Π is the set of atomic propositions) is true ($\top$) if $o := l(x)$ satisfies all components of $h_{\mathrm{AP}}(o) > \sigma_{\mathrm{AP}}$, where $h_{\mathrm{AP}} : \mathcal{O} \mapsto \mathbb{R}^d$ is a predicate function corresponding to the atomic proposition AP and $\sigma_{\mathrm{AP}} \in \mathbb{R}^d$. Otherwise, it is false ($\bot$).

The syntax of TWTL is defined inductively as $\phi ::= H^d s | \phi_1 \cdot \phi_2 | \phi_1 \vee \phi_2 | [\phi]^{[a,b]}$ Vasile et al. (2017); where s is either true, $\top$, or an atomic proposition in Π; H^d is the *hold* operator where $d \in \mathbb{N}$; $\cdot$ is the *concatenation* operator; $\vee$ is the Boolean conjunction operator; $[]^{[a,b]}$ is the *within* operator, where $d, a, b \in \mathbb{Z}_{\geq 0}$ and $a \geq b$; and ϕ_1 and ϕ_2 are TWTL formulae.

The Boolean semantics over an observation word $\mathbf{o}_{i_1,i_2}$, $i_1, i_2 \in \mathbb{N}$ and $i_1 < i_2$, are defined recursively as follows. for the *hold* operator, $\mathbf{o}_{i_1,i_2} \models H^d s \Leftrightarrow s \in o_t,\ \forall t \in [i_1, i_2] \wedge (i_2 - i_1 \geq d)$; the concatenation, $\mathbf{o}_{i_1,i_2} \models \phi_1 \cdot \phi_2 \Leftrightarrow \exists t =_{i \in [i_1,i_2]} \{\mathbf{o}_{i_1,i} \models \phi_1\} \wedge (\mathbf{o}_{i+1,i_2} \models \phi_2)$; and the within operator, $\mathbf{o}_{i_1,i_2} \models [\phi]^{[a,b]} \Leftrightarrow \exists t \geq i_1 + a$ s.t. $\mathbf{o}_{i,i_1+b} \models \phi \wedge (i_2 - i_1 \geq b)$. The rest of the semantics are detailed in Ahmad et al. (2023).

The time horizon of a TWTL formula ϕ represents the minimum time required to evaluate whether the formula is satisfied. It is defined as follows. $||\phi|| := \max\{\max(||\phi_1||, ||\phi_2||) \cdot \mathbf{1}_{\phi=\phi_1 \vee \phi_2}, (||\phi_1|| + ||\phi_2|| + 1) \cdot \mathbf{1}_{\phi=\phi_1 \cdot \phi_2}, d \cdot \mathbf{1}_{\phi=H^d s}, b \cdot \mathbf{1}_{\phi=[\phi_1]^{[a,b]}}\}$, where $\mathbf{1}$ is the indicator function.

Example 1 (Lunar Lander) *The lunar lander environment from Gymnasium (Towers et al. (2023)) has state $X = (p_x, p_y, \dot{p}_x, \dot{p}_y, \psi, \dot{\psi}) \in \mathbb{R}^6$, with observation $o = X$, where (p_x, p_y) is the position, $(\dot{p}_x, \dot{p}_y)$ is the velocity, ψ is the angle, and $\dot{\psi}$ is the angular velocity. The control input space $\mathcal{U}$ consists of discrete commands for the main engine and side thrusters. Based on the environment's success criteria, we define the landing task using TWTL: $\phi_{\text{landing}} := [H^{100}\text{AP}_{\text{hover}}]^{[0,\ 150]} \cdot [H^{150}\text{AP}_{\text{align}}]^{[100,\ 300]} \cdot [H^{150}\text{AP}_{\text{descend}}]^{[250,\ 450]} \cdot [H^{50}\text{AP}_{\text{land}}]^{[400,\ 500]}$; with predicate functions with fixed parameters motivated by the environment's success criteria: $h_{\text{AP}_{\text{hover}}}(o) = \min\{h_0 - 0.8h_0 - |p_y - 0.8h_0|, 0.1 - |\dot{p}_y|\} \geq 0$; $h_{\text{AP}_{\text{align}}}(o) = \min\{0.2 - |p_x|, 0.1 - |\psi|\} \geq 0$; $h_{\text{AP}_{\text{descend}}}(o) = \min\{-0.2 - \dot{p}_y, \dot{p}_y + 0.5, 0.15 - |\psi|\} \geq 0$; and $h_{\text{AP}_{\text{land}}}(o) = \min\{0.1 - \sqrt{\dot{p}_y^2 + \dot{p}_x^2}, 0.1 - |\psi|, \mathbf{1}_{p_y \leq 0}\} \geq 0$. The time horizon $||\phi_{\text{landing}}|| = 1403$. The formula ϕ_{landing} reads: "Within time 0 and time 150, the lander must be hovering for a 100 time steps (subformula $H^{100}\ \text{AP}_{\text{hover}}$). Then, within time 100 and time 300, the lander must be aligned with the landing position for a 150 time steps (subformula $H^{150}\ \text{AP}_{\text{align}}$). After ensuring that the lander is aligned, descend and then dwelling in the landing position." We use the concatenation operator to ensure the correct landing sequence, which is, in high level, hovering, aligning, descending, then landing.*

A formula is feasible if the time window of each *within* operator in the formula is longer than the enclosed task (expressed via the *Hold* operators) (Definition IV.1 in Vasile et al. (2017)). Let Φ be the set of feasible TWTL formulas.

In the following, we define a TWTL robustness which measures how close or far a sequence of observations, $\mathbf{o}_{i_1,i_2}$, is from satisfying the TWTL task. A positive value indicates task satisfaction, with higher values signifying greater robustness. Conversely, a negative value indicates task violation Ahmad et al. (2023).

Definition 2 *(TWTL Robustness) Given a TWTL formula ϕ and an output word $\mathbf{o}_{i_1,i_2}$ of MDP (Def. 1), we define the robustness degree $\varrho(\mathbf{o}_{i_1,i_2}, \phi)$ at time 0 recursively as follows:*

$$\begin{aligned}
&\varrho(\mathbf{o}_{i_1,i_2}, H^d \text{AP}_\text{A}) := \min_{i \in [i_1, d+i_1]} h(o_i) \text{ if } i_2 - i_1 \geq d, \text{ else } -\infty : \\
&\varrho(\mathbf{o}_{i_1,i_2}, \phi_1 \cdot \phi_2) := \max_{i \in [i_1,i_2)} \{\min\{\varrho(\mathbf{o}_{i_1,i}, \phi_1), \varrho(\mathbf{o}_{i+1,i_2}, \phi_2)\}\}; \\
&\varrho(\mathbf{o}_{i_1,i_2}, [\phi]^{[a,b]}) := \max_{i \geq i_1 + a} \{\varrho(\mathbf{o}_{i,i_1+b}, \phi)\} \text{ if } i_2 - i_1 \geq b, \text{ else } -\infty.
\end{aligned} \tag{3}$$

3. Problem Formulation

We consider a model-free RL setting where an agent interacts with a complex environment with delayed rewards. To address the temporal credit assignment challenge, we introduce a reward function based on Time Window Temporal Logic (TWTL) specifications that provides more immediate feedback about task progress.

Definition 3 (Concrete Time Reward) *Let $\phi \in \Phi$ be a TWTL formula. For a finite-horizon MDP with a trajectory $\mathbf{x}_{i,i+||\phi||}$ produced by applying $\mathbf{u}_{i-1,i+||\phi||-1} := u_{i-1}u_t \dots u_{N-i+||\phi||-1}$, we define an episodic concrete-time reward, over the generated observation word $\mathbf{o}_{i,i+||\phi||}$, at time i, as follows:*

$$r_{\phi,i}(\mathbf{o}_{i,i+||\phi||}) := \mathbf{1}_{\mathbf{o}_{i,i+||\phi||} \models \phi}. \tag{4}$$

This reward function provides a binary signal indicating whether the observed sequence satisfies the TWTL ϕ within its time horizon $||\phi||$.

Given a finite horizon MDP with an episodic concrete-time reward $r_{\phi,i}$, compute the optimal parameter θ^* of the stochastic policy, π_θ, that maximizes the total expected episodic reward. (i.e. $\pi_{\theta^*} = _{\theta \in \Theta}\ \mathrm{E}_{\pi_\theta}\left[\sum_{i=1}^{N} r_{\phi,i}\right]$).

4. Accelerated Proximal Policy Optimization

We enhance PPO with an offline policy and a reward-shaping function based on TWTL. The offline policy improves performance, while the reward shaping guides learning towards desired temporal goals.

4.1. Proximal Policy Optimization

PPO is a policy optimization algorithm that uses the policy gradient to optimize a parameterized policy. PPO provides stability to the learning process. At each iteration, the algorithm aims to find a better policy that is close to the previous iteration Schulman et al. (2017). This, in turn, helps keep the learning process away from degenerate policies.

Consider the state and state-action value functions, (1)) –and subsequently the advantage function (2)– to be defined over the *concrete time reward* (4). For a parameterized policy π_θ, where $\theta \in \Theta$ is the parameter and Θ is the parameter space, PPO optimizes the policy at each iteration according to $\theta_{i+1} \leftarrow_{\theta \in \Theta}\ \mathrm{E}[J^k(x, u, \theta, \theta_i)]$, where $J^k(x, u, \theta, \theta_i)$ is a clipped objective and defined as follows Schulman et al. (2017):

$$J^k(x, u, \theta, \theta_i) := \min\left[\frac{\pi_\theta(u|x)}{\pi_{\theta_i}(u|x)} \hat{A}_t^{\pi_{\theta_i},k}(u, x),\ g(\epsilon, \hat{A}_i^{\pi_{\theta_i},k}(u, x))\right], \tag{5}$$

where $g(\epsilon, \hat{A}) := \begin{cases} (1+\epsilon)\hat{A};\ \hat{A} \geq 0 \\ (1-\epsilon)\hat{A};\ \hat{A} < 0 \end{cases}$, and $\epsilon \in (0, 1)$ is a tunable hyperparameter.

The value $\hat{A}_t^{\pi_{\theta_i},k}$ estimates the advantage $A_t^{\pi_{\theta_i},k}$ at time t and can be computed as follows Schulman et al. (2015b):

$$\hat{A}_t^{\pi_{\theta_i},k} = \delta_t^k + (\gamma\lambda)\delta_{t+1}^k + \dots + (\gamma\lambda)^{N-t+1}\delta_{N-1}^k, \tag{6}$$

where δ_t^k is the temporal difference (TD) residual of $V^{\pi_{\theta_i},k}$ discounted by γ, and is defined as $\delta_t^k := r_t^k + \gamma V_t^{\pi_{\theta_i},k} - V_{t+1}^{\pi_{\theta_i},k}$. In our setting the value function (1) is parameterized by a neural network.

The choice of maximizing the objective (5) implies maximizing the advantage function Grudzien et al. (2022). This choice yields low variance in the gradient; where the advantage function measures how much better or worse the policy is from the default control Schulman et al. (2015b).

4.2. Temporal Logic Reward-Shapping

Consider an episodic RL framework where tasks are formulated using TWTL formulae. To simplify the analysis, we consider a single task ϕ, with time Horizon $||\phi|| < N$, where N is the number of steps in the training episode.

Recall the concrete time reward (4), computing this reward requires complete MDP trajectories that span the entire task horizon, $||\phi||$. However, obtaining these full trajectories during real-world implementation might be impractical. To overcome this challenge, we need a method to complete or extend partially observed trajectories. Existing approaches for trajectory completion include trajectory prediction Salzmann et al. (2020), or runtime monitoring techniques Ahmad et al. (2023).

We propose using an observation predictor. This predictor takes an MDP state and generates a sequence of predicted observations spanning the entire task horizon ($||\phi||$). We present Long Short-Term Memory (LSTM)-based task predictor in Example 2. For formulation simplicity, assume we have a pre-defined predictor function, $\texttt{Pred} : \mathcal{X} \mapsto \mathcal{O}$, which maps a state x_t to a sequence of observations $(\hat{o}_t, \hat{o}_{t+1}, \ldots, \hat{o}_{t+||\phi||})$.

Consider the TWTL robustness function, $\varrho : \mathcal{O} \times \Phi \mapsto \mathbb{R}$, as defined in (3). For a TWTL task ϕ, we define a reward shaping function $F : \mathcal{X} \times \mathcal{U} \times \mathcal{X} \times \Phi \mapsto \mathbb{R}$, with $0 < \kappa < 1$, as the following:

$$F(x_t, u_t, x_{t+1}, \phi) := \kappa \cdot \varrho(\texttt{Pred}(x_t), \phi) - \varrho(\texttt{Pred}(x_{t+1}), \phi) \tag{7}$$

Lemma 4 *The optimal policy of the original MDP is the same as the optimal policy of the MDP with the shaped reward function ($r'^k_{\phi,t} := r^k_{\phi,t} + F$).*

Proof By Theorem 1 in Ng et al. (1999), F being a potential function, as defined in (7), guarantees optimal policy consistency. ■

Example 2 (Continue) *For the Lunar Lander environment, we implement a sequence prediction network to forecast future state trajectories, which is similar to Trajectron ++ Salzmann et al. (2020) but for a single agent case, which are then used to evaluate TWTL robustness. The network architecture consists of: (i) an input embedding layer, $\texttt{embed}$, that processes a window, $w \in \mathbb{N}_{>0}$, of state vectors $\mathbf{x}_{t-w:t} \in \mathbb{R}^{8\times w}$; (ii) an LSTM encoder $\texttt{LSTM}_\text{e}$ and decoder $\texttt{LSTM}_\text{d}$ for temporal sequence modeling; and (iii) prediction heads for future state estimation. Given a window of states $\mathbf{x}_{t-w:t}$, the network predicts future state trajectories: $\hat{\mathbf{x}}_{t:t+||\phi||} = \texttt{LSTM}_\text{d}(\texttt{LSTM}_\text{e}(\texttt{embed}(\mathbf{x}_{t-w:t})))$, where $\hat{\mathbf{x}}_{t:t+||\phi||}$ represents the predicted state sequence over the TWTL formula horizon.*

The predictor is trained on expert demonstrations to minimize the mean squared error of state predictions, $\mathcal{L} = \sum_{i=t}^{t+||\phi||} \|\hat{\mathbf{x}}_i - \mathbf{x}_i\|^2$ Using these predicted state trajectories, we can compute the TWTL robustness degree ϱ for the landing task formula $\phi_{landing}$ by evaluating the atomic predicates (hover, align, descend, land) on the predicted states. This robustness value is then used in the reward shaping function (7) to provide immediate feedback about the predicted satisfaction or violation of the landing requirements.

4.3. Online-Offline Policy Architecture

In this section, we present two key theoretical results for policy improvement in episodic learning. First, we introduce a novel policy architecture that combines an offline policy (π_ρ) with an online policy (π_β) and prove its convergence properties. Second, we establish guarantees for iterative improvement of this mixed policy over time. Both results build on the same theoretical foundation while addressing different aspects of the learning process.

The core idea of our architecture is to combine an offline policy, pre-trained on expert demonstrations, with an online policy continuously optimized by PPO. Initially, the offline policy guides the online policy's learning, while a mixing parameter gradually reduces the offline policy's influence, allowing the online policy to take over action selection.

Let $\mathrm{D}_{\mathcal{U}}$ be the set of distributions over $\mathcal{U}$. We introduce a deep policy architecture, $\pi_\theta \in \mathrm{D}_{\mathcal{U}}$, that we constitute using two parallel deep policies $\pi_\rho, \pi_\beta \in \mathrm{D}_{\mathcal{U}}$ and combine them using a fully connect layer (FCL), see Figure 1. We compute π_θ as follows.

$$\pi_\theta(u|x) := (1-\alpha)\cdot\pi_\rho(u|x) + \alpha\cdot\pi_\beta(u|x) \tag{8}$$

where α is mixing parameters which are the weights of the FCL of the architecture.

In the following, we demonstrate that policy π_θ is guaranteed to improve upon the offline policy π_ρ at every PPO iteration. After that, and using similar arguments, we prove that policy π_θ improves at every iteration.

We interpret π_ρ as a fixed prior of policy π_θ, where we aim to learn the optimal π_β and the optimal mixing parameter α through optimizing the weights of the FCL.

We base our analysis on the works of Kakade and Langford (2002) and Schulman et al. (2015a). The total return of the policy π, at episode k, is given by: $\eta^k(\pi) := \mathrm{E}_{u\sim\pi}\left[\sum_{i=1}^{N} r_i^k(x_i,u_i)\right]$, we drop the superscript k from η for notation simplicity.

Lemma 5 *In our framework, where we mix an offline policy π_ρ with an online policy π_β, we can express the return of the mixed policy π_θ in terms of the offline policy return. This relationship follows a similar structure to policy improvement bounds established in Kakade and Langford (2002); Schulman et al. (2015a). Specifically, as follows.*

$$\eta(\pi_\theta) = \eta(\pi_\rho) + \mathop{\mathrm{E}}_{u\sim\pi_\theta}\left[\sum_{i=1}^{N} A_i^{\pi_\rho}\right] \tag{9}$$

Proof The proof is omitted due to space limitations and it can be found in the extended version of this manuscript Ahmad et al. (2024). ■

Lemma 5 provides the basic formula to compute an improvement bound on π_θ upon π_ρ, however, it the expectation depends on π_θ which makes it hard to derive an update rule during optimization. Hence, we use a local approximation of the total return, $L : \mathrm{D}_{\mathcal{U}} \mapsto \mathbb{R}$, introduced by Schulman et al. Schulman et al. (2015a), which we require for introduce a bound guaranteeing that π_θ improves upon π_ρ. Let $u \sim \pi_\rho$ and the visitation frequency $P_{\pi_\rho}(x) = p(x_0 = x|x,u) + p(x_1 = x|x,u) + \cdots + p(x_N = x|x,u)$, the return estimation of the offline policy, π_ρ, as a function of a general policy π is given as follows.

$$L_{\pi_\rho}(\pi) = \eta(\pi_\rho) + \sum_{x\in\mathcal{X}} P_{\pi_\rho}(x) \sum_{u\in\mathcal{U}} \pi(u|x) A_0^{\pi_\rho}(x,u) \tag{10}$$

Choose $\pi^*_\beta =_{\pi_\beta} L_{\pi_\rho}(\pi_\beta)$. In the following theorem, we represent a fundamental bound to demonstrate the effectiveness of policy architecture (8).

Proposition 6 implies that the mixing policy, π_θ improves upon π_ρ at every iteration.

Proposition 6 *Consider the total return of π_θ, $\eta(\pi_\theta)$, and the estimated performance of π_ρ, L_{π_ρ}, the following bound holds.*

$$\eta(\pi_\theta) \geq L_{\pi_\rho}(\pi_\theta) - \frac{2\varsigma\gamma\alpha^2}{(1-\gamma)^2} \tag{11}$$

where $\varsigma = \max_{x\in\mathcal{X}} |\, \mathrm{E}_{u\sim\pi^*_\beta}[A_0^{\pi_\rho}(x,u)]|$, and recall that γ is a discount factor the GAE (6).

Proof The proof is omitted due to space limitations and it can be found in the extended version of this manuscript Ahmad et al. (2024). ■

In the following, we derive the main result of the paper and the update rule of policy π_θ (defined in Equation 8). First, we consider the policy at iteration $i+1$ to be given as the following: $\pi_{\theta_{i+1}}(u|x) := (1-\mathrm{TV}^{\theta_{i+1}}_{\theta_i})\cdot\pi_{\theta_i}(u|x) + \mathrm{TV}^{\theta_{i+1}}_{\theta_i}\cdot\tilde{\pi}'_{\theta_{i+1}}(u|x)$, where $\mathrm{TV}^{\theta_{i+1}}_{\theta_i} = \max_{x\in\mathcal{X}} \mathrm{D}_{\mathrm{TV}}(\theta_i||\theta_{i+1})$; $\mathrm{D}_{\mathrm{TV}}(\theta_i||\theta_{i+1}) := \frac{1}{2}\sum_j |(u_j \sim \pi_{\theta_i}(x)) - (u'_j \sim \pi_{\theta_{i+1}}(x))|$ is the total variation between π_{θ_i} and $\pi_{\theta_{i+1}}$, and $\tilde{\pi}'_{\theta_{i+1}}(u|x) :=_{\pi'_{\theta_{i+1}}} L_{\pi_{\theta_i}}(\pi'_{\theta_{i+1}})$.

In the next Theorem, we show that the policy π_θ (defined in Equation 8) consistently improves with each iteration. This conclusion is a direct consequence of Proposition 6.

Theorem 7 *Consider π_{θ_i}, as in (8), with $\theta_i = (\beta_i, \alpha_i)$, where the i stands for the optimization iteration. Then for policy $\pi_{\theta_{i+1}}$, π_{θ_i} and $\varsigma = \max_{x\in\mathcal{X}} |\, \mathrm{E}_{u\sim\pi^*_\beta}[A_0^{\pi_{\theta_i}}(x,u)]|$, the following bound holds:*

$$\eta(\pi_{\theta_{i+1}}) \geq L_{\pi_{\theta_i}}(\eta(\pi_{\theta_{i+1}})) - \frac{2\varsigma\gamma\cdot(\mathrm{TV}^{\theta_{i+1}}_{\theta_i})^2}{(1-\gamma)^2} \tag{12}$$

Proof The proof is omitted due to space limitations and it can be found in the extended version of this manuscript Ahmad et al. (2024). ■

We use the sampling-based estimation, similar to the original PPO Schulman et al. (2017) and TRPO Schulman et al. (2015a) algorithms, which yield the surrogate clipped objective (5) and is defined based on policy architecture (8).

Considering the general class of majorization maximization (MM) algorithms Hunter and Lange (2004), maximizing the r.h.s. of (12) implies maximizing the total return of $\pi_{\theta_{i+1}}$ Schulman et al. (2015a). Similar to (10), with a slight abuse of notation we define the return $L_{\pi_{\theta_{i+1}}}$ as a function of parameter θ:

$$L_{\theta_i}(\theta_{i+1}) = \eta(\theta_i) + \sum_{x\in\mathcal{X}} P_{\theta_i}(x) \sum_{u\in\mathcal{U}} \pi_{\theta_{i+1}}(u|x) A_0^{\theta_i}(x,u) \tag{13}$$

4.4. Algorithm Details

We need to find θ_{i+1} that maximizes $L_{\theta_i}(\theta_{i+1})$. We use importance sampling (IS) to estimate θ_{i+1} that maximizes the term $\sum_{x\in\mathcal{X}} P_{\theta_i}(x) \sum_{u\in\mathcal{U}} \pi_{\theta_{i+1}}(u|x) A_0^{\theta_i}(x,u)$ in (13) which implies maximizing $L_{\theta_i}(\theta_{i+1})$. By using IS and including penalized $\mathrm{TV}^{\theta_{i+1}}_{\theta_i}$ in the objective that we're maximizing,

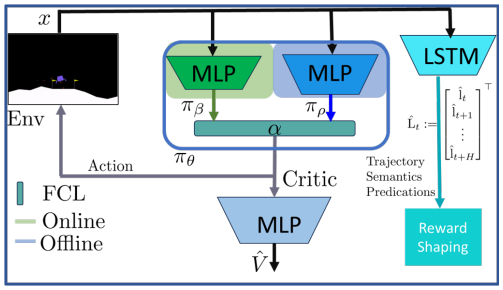

Figure 1: The Actor-Critic RL framework.The Actor's architecture, policy π_θ, consists of an offline policy, π_ρ, and an adaptive policy, π_β, where the two policies are mixed using the parameters of the FCL, α. The critic consists of an MLP that approximates the value function. The task predictor LSTM network and the reward shaping are depicted in cyan.

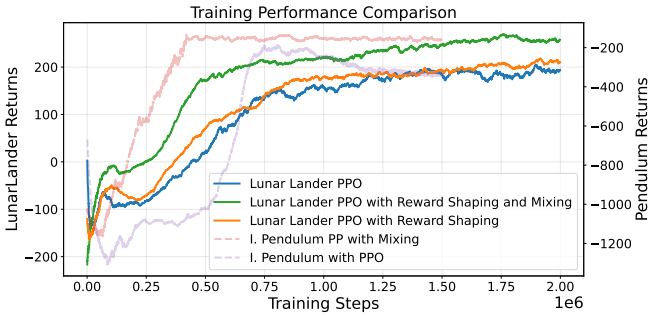

Figure 2: Training performance comparison across different variants of PPO in LunarLander-v2 and Pendulum environments. For LunarLander-v2 (left y-axis), we compare vanilla PPO against variants with reward shaping and mixing. The results show that combining reward shaping with policy mixing achieves faster learning and better performance compared to baseline PPO and reward shaping alone. For Pendulum (right y-axis, dashed lines), we include PPO with and without mixing for reference. Training steps are shown in millions on the x-axis.

we conclude to use a clipped objective function that has the same structure as the objective of PPO Schulman et al. (2017), $J^k(x, u, \theta, \theta_t)$ (see Eq (5)), but is defined with respect to (8).

Following the formulation of Cai et al. (2020) we introduce APPO in Algorithm 1. Given the offline policy, π_ρ, we initialize for episode 1 a series of the policy, π_θ, with an initial parameter θ_0. Then at every training episode k we observe the state after applying the 0-th step policy, $\pi_{\theta,0}$, Line 4.4. Then at every iteration i of the episode k, the parameter θ_i of the policy is optimized.

Input $\pi_\rho(\cdot)$ a fixed task predictor trained offline
Initialize $\{\pi_{\theta,t}\}_{t=0}^{N-1}$ with $\theta = \theta_0$
foreach *episode* $k \in \{1, \ldots, K\}$ **do**
  Observe x_1
  foreach *iteration* $i \in \{1, \ldots, N\}$ **do**
    $\theta_{i+1}^k \leftarrow_{\theta\in\Theta} \mathrm{E}[J_\phi^k(x, u, \theta, \theta_i)]$
    Apply the control $u_i \sim \pi_{\theta_i}$
    Compute $\hat{A}_{\phi,t}^{\pi_{\theta_i},t,k}$ according to (6)
  end
end

Algorithm 1: Accelerated PPO

Given a concrete-time reward (Def 3), we introduce $J_\phi^k(\cdot,\cdot,\cdot,\cdot)$, a variant of the clipped objective (5) defined based on $A_{\phi,t}^k$ and is given as $J_\phi^k(x, u, \theta, \theta_i) := \min\left[\frac{\pi_\theta(u|x)}{\pi_{\theta_i}(u|x)}\hat{A}_{\phi,t}^{\pi_{\theta_i\cdot\rho},k}, g(\epsilon, \hat{A}_{\phi,t}^{\pi_{\theta_i\cdot\rho},k})\right]$. Consequently, the parameterized part of the policy update is computed according to $\theta_{i+1} \leftarrow_{\theta\in\Theta} \mathrm{E}[J_\phi^k(x, u, \theta, \theta_i)]$.

5. Case Study

As a continuation of Examples 1 and 2, we demonstrate the empirical performance of our approach in Figure 2. For the Lunar Lander environment, where we previously defined the landing task TWTL formula ϕ_{landing} and the LSTM-based task predictor, combining policy mixing with reward shaping (green) achieves faster learning and better asymptotic performance compared to both vanilla PPO (blue) and reward shaping alone (orange). The improvement is particularly noticeable in the early training phase (0-0.5M steps), where our method accelerates learning.

To test the robustness of our mixing approach, we intentionally degraded pre-trained PPO policies by adding controlled noise to their parameters, using these as our offline policies π_ρ. Despite starting with these suboptimal policies, our method successfully leverages their partial knowledge while learning to improve upon them. For comparison, we also tested our approach on the Inverted Pendulum environment with similarly degraded offline policies (right y-axis, dashed lines), where policy mixing again demonstrates improved learning dynamics over the baseline PPO implementation.

6. Conclusion and Future Work

We presented an approach to accelerate reinforcement learning in environments with delayed rewards through two key innovations: a mixed policy architecture that combines potentially suboptimal offline policies with online learning, and a TWTL-based reward shaping mechanism enabled by a task predictor. For the mixed policy architecture, we established theoretical guarantees showing consistent improvement over both the offline policy and previous iterations. Our empirical results on the Lunar Lander and Inverted Pendulum environments demonstrate that even with degraded offline policies, our method achieves faster learning and better asymptotic performance compared to vanilla PPO.

Several directions for future work emerge from this study. First, exploring more complex TWTL specifications that capture intricate temporal dependencies could extend our framework to more challenging tasks, such as soccer games. Second, investigating adaptive mixing strategies that automatically adjust based on the relative performance of offline and online policies could further improve learning efficiency. The integration of other temporal logic formalisms and their corresponding quantitative semantics could also provide interesting avenues for reward shaping in reinforcement learning.

Acknowledgments

The authors wish to thank Bassel El Mabsout and Abdelrahman AbdelGawad for their insightful comments and suggestions regarding the practical issues of delayed rewards settings and environments, and the challenges of reward design. This work was supported in part by MIT Lincoln Laboratory MAESTRO program.

References

Ahmad Ahmad, Cristian-Ioan Vasile, Roberto Tron, and Calin Belta. Robustness measures and monitors for time window temporal logic. In *2023 62nd IEEE Conference on Decision and Control (CDC)*, pages 6841–6846. IEEE, 2023.

Ahmad Ahmad, Mehdi Kermanshah, Kevin Leahy, Zachary Serlin, Ho Chit Siu, Makai Mann, Cristian-Ioan Vasile, Roberto Tron, and Calin Belta. Accelerating proximal policy optimization learning using task prediction for solving environments with delayed rewards, 2024. URL https://arxiv.org/abs/2411.17861.

Derya Aksaray, Austin Jones, Zhaodan Kong, Mac Schwager, and Calin Belta. Q-learning for robust satisfaction of signal temporal logic specifications. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pages 6565–6570. IEEE, 2016.

Mohammed Alshiekh, Roderick Bloem, Rüdiger Ehlers, Bettina Könighofer, Scott Niekum, and Ufuk Topcu. Safe reinforcement learning via shielding. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Ahmet Semi Asarkaya, Derya Aksaray, and Yasin Yazıcıoğlu. Temporal-logic-constrained hybrid reinforcement learning to perform optimal aerial monitoring with delivery drones. In *2021 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 285–294. IEEE, 2021.

Christel Baier and Joost-Pieter Katoen. *Principles of model checking*. MIT press, 2008.

Anand Balakrishnan and Jyotirmoy V Deshmukh. Structured reward shaping using signal temporal logic specifications. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3481–3486. IEEE, 2019.

Philip J Ball, Laura Smith, Ilya Kostrikov, and Sergey Levine. Efficient online reinforcement learning with offline data. In *International Conference on Machine Learning*, pages 1577–1594. PMLR, 2023.

Mingyu Cai, Erfan Aasi, Calin Belta, and Cristian-Ioan Vasile. Overcoming exploration: Deep reinforcement learning for continuous control in cluttered environments from temporal logic specifications. *IEEE Robotics and Automation Letters*, 8(4):2158–2165, 2023.

Qi Cai, Zhuoran Yang, Chi Jin, and Zhaoran Wang. Provably efficient exploration in policy optimization. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1283–1294. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/cai20d.html.

Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Misha Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision transformer: Reinforcement learning via sequence modeling. *Advances in neural information processing systems*, 34:15084–15097, 2021.

Jakub Grudzien, Christian A Schroeder De Witt, and Jakob Foerster. Mirror learning: A unifying framework of policy optimisation. In *International Conference on Machine Learning*, pages 7825–7844. PMLR, 2022.

Hengyuan Hu, Suvir Mirchandani, and Dorsa Sadigh. Imitation bootstrapped reinforcement learning. *arXiv preprint arXiv:2311.02198*, 2023.

David R Hunter and Kenneth Lange. A tutorial on mm algorithms. *The American Statistician*, 58 (1):30–37, 2004.

Rodrigo Toro Icarte, Toryn Q Klassen, Richard Valenzano, and Sheila A McIlraith. Reward machines: Exploiting reward function structure in reinforcement learning. *Journal of Artificial Intelligence Research*, 73:173–208, 2022.

Sham Kakade and John Langford. Approximately optimal approximate reinforcement learning. In *Proceedings of the Nineteenth International Conference on Machine Learning*, pages 267–274, 2002.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 33:1179–1191, 2020.

Xiao Li, Cristian-Ioan Vasile, and Calin Belta. Reinforcement learning with temporal logic rewards. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3834–3839. IEEE, 2017.

Ashvin Nair, Abhishek Gupta, Murtaza Dalal, and Sergey Levine. Awac: Accelerating online reinforcement learning with offline datasets. *arXiv preprint arXiv:2006.09359*, 2020.

Daniel Neider, Jean-Raphael Gaglione, Ivan Gavran, Ufuk Topcu, Bo Wu, and Zhe Xu. Advice-guided reinforcement learning in a non-markovian environment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 9073–9080, 2021.

Andrew Y Ng, Daishi Harada, and Stuart Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *Icml*, volume 99, pages 278–287, 1999.

Stéphane Ross and J Andrew Bagnell. Agnostic system identification for model-based reinforcement learning. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, pages 1905–1912, 2012.

Tim Salzmann, Boris Ivanovic, Punarjay Chakravarty, and Marco Pavone. Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVIII 16*, pages 683–700. Springer, 2020.

Simon Schmitt, Jonathan J Hudson, Augustin Zidek, Simon Osindero, Carl Doersch, Wojciech M Czarnecki, Joel Z Leibo, Heinrich Kuttler, Andrew Zisserman, Karen Simonyan, et al. Kickstarting deep reinforcement learning. *arXiv preprint arXiv:1803.03835*, 2018.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pages 1889–1897. PMLR, 2015a.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015b.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

Mark Towers, Jordan K. Terry, Ariel Kwiatkowski, Andrew Copplestone-Bruce, Gianluca Lutati, Rousslan Fernand Julien Dossa, Antonin Famularo, Filip Pytlak, Marc Ballester, Hao Guo, Jiayi Hu, and Mikayel Awad. Gymnasium. https://github.com/Farama-Foundation/Gymnasium, 2023. URL https://gymnasium.farama.org/.

Cristian-Ioan Vasile, Derya Aksaray, and Calin Belta. Time window temporal logic. *Theoretical Computer Science*, 691:27–54, 2017.

Zhe Xu, Ivan Gavran, Yousef Ahmad, Rupak Majumdar, Daniel Neider, Ufuk Topcu, and Bo Wu. Joint inference of reward machines and policies for reinforcement learning. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 30, pages 590–598, 2020.