# Overcoming Non-stationary Dynamics with Evidential Proximal Policy Optimization

**Abdullah Akgül** **Gulcin Baykal** **Manuel Haußmann** **Melih Kandemir**
Department of Mathematics and Computer Science
University of Southern Denmark
Odense, Denmark
{akgul,baykalg,haussmann,kandemir}@imada.sdu.dk

## Abstract

Continuous control of non-stationary environments is a major challenge for deep reinforcement learning algorithms. The time-dependency of the state transition dynamics aggravates the notorious stability problems of model-free deep actor-critic architectures. We posit that two properties will play a key role in overcoming non-stationarity in transition dynamics: (i) preserving the plasticity of the critic network, (ii) directed exploration for rapid adaptation to the changing dynamics. We show that performing on-policy reinforcement learning with an evidential critic provides both of these properties. The evidential design ensures a fast and sufficiently accurate approximation to the uncertainty around the state-value, which maintains the plasticity of the critic network by detecting the distributional shifts caused by the change in dynamics. The probabilistic critic also makes the actor training objective a random variable, enabling the use of directed exploration approaches as a by-product. We name the resulting algorithm as *Evidential Proximal Policy Optimization (EPPO)* due to the integral role of evidential uncertainty quantification in both policy evaluation and policy improvement stages. Through experiments on non-stationary continuous control tasks, where the environment dynamics change at regular intervals, we demonstrate that our algorithm outperforms state-of-the-art on-policy reinforcement learning variants in both task-specific and overall return.

## 1 Introduction

Most of the deep reinforcement learning algorithms are developed assuming stationary transition dynamics, even though in many real-world applications the transition distributions are time-dependent, i.e., *non-stationary* (Thrun, 1998). The non-stationarity of state transitions makes it essential for the agent to keep updating its policy. For example, a robotic arm may experience wear and tear, leading to changes in the ability of its joints to apply torque, or an autonomous robot navigating a terrain with varying ground conditions, such as friction, inclination, and roughness. In such environments, an agent can maintain high performance only by continually adapting its policy to changes. On-policy algorithms, such as Proximal Policy Optimization (PPO) (Schulman et al., 2017), are particularly well-suited for non-stationary environments (Sutton et al., 2007) because they rely solely on data from the most recent policy, ensuring policy improvement through sufficiently small updates (Kakade and Langford, 2002). This makes PPO an attractive choice for applications ranging from physical robotics (Melo and Máximo, 2019) to fine-tuning large language models (Touvron et al., 2023; Achiam et al., 2023; Zheng et al., 2023). Agents designed for open-world, non-stationary environments has

Full paper available at https://arxiv.org/pdf/2503.01468.

18th European Workshop on Reinforcement Learning (EWRL 2025).

to continually learn throughout their entire lifecycle, not just during a fixed training phase. Time-dependent changes in state transition dynamics result in non-stationary Markov decision processes (MDPs), where existing reinforcement learning algorithms often struggle to adapt effectively.

We posit that the simultaneous presence of two key features is essential for overcoming the challenges caused by non-stationarity in deep reinforcement learning:

*(i)* **Maintaining the plasticity of the critic network:** Plasticity refers to the ability of a neural network to change its wiring in response to new observations throughout the complete learning period. Deep reinforcement learning algorithms have been reported to suffer from the loss of plasticity in non-stationary settings by a vast body of earlier work (Dohare et al., 2021; Lyle et al., 2022; Nikishin et al., 2022; Abbas et al., 2023; Dohare et al., 2023; Lyle et al., 2023; Dohare et al., 2024; Lee et al., 2024; Moalla et al., 2024; Chung et al., 2024; Kumar et al., 2025; Lyle et al., 2025).

*(ii)* **Ensuring directed exploration for rapid adaptation to changing dynamics:** Directed exploration determines the degree of exploration based on an estimated uncertainty of an unobserved state. This way, the agent prioritizes underexplored, hence more informative, areas of the state-action space, thereby improves its sample-efficiency. Directed exploration is instrumental in fast-changing non-stationary environments where the agent has limited time to adapt to each new condition (Kaufmann et al., 2012; Besbes et al., 2014; Zhao et al., 2020).

We hypothesize that both sustained plasticity and directed exploration can be achieved by quantifying the uncertainty around the value function. An agent equipped with a probabilistic value function will systematically reduce the uncertainty of its value predictions as it collects more data. When confronted with a change in environment dynamics, the value function output will make predictions with reduced confidence. The increased uncertainty will increase the critic training loss, hence keeping the training process active. Furthermore, the probabilistic value predictor will make it possible to assign uncertainty estimates on the policy training objective, which can in turn be used as an exploration bonus to direct the policy search towards underexplored areas of the state-action space.

**Our Hypothesis:** *Equipping an agent with uncertainty quantification of the value function enables it to (i) preserve plasticity and (ii) explore effectively.*

Guided by the above hypothesis, we adopt *Evidential Deep Learning* (Sensoy et al., 2018) as a well-suited framework for learning probabilistic value functions. Evidential deep learning suggests modeling the uncertainty of each data point by a Bayesian data-generating process where the hyperparameters of the prior distribution are determined by input-dependent functions. The likelihood and priors are chosen as conjugate pairs to keep the calculation of the data point-specific posterior and the marginal likelihood analytically tractable. The prior hyperparameter functions are modeled as deep neural nets, the parameters of which are learned by empirical Bayes. Evidential approaches are observed to deliver high-quality uncertainty estimates in both regression (Amini et al., 2020) and classification (Kandemir et al., 2022) settings.

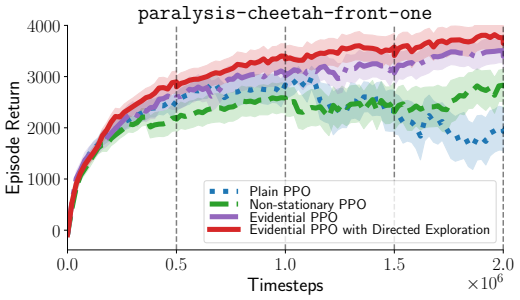


Figure 1: *Adaptation of on-policy agents to non-stationarity.* PPO and its non-stationary extension lose their adaptation capability after 1 million steps, while the evidential PPO variants continue to improve. Directed exploration further improves performance. See Appendix B.3 for details.

Figure 1 illustrates the learning profiles of on-policy deep actor-critics in a continuous control task with non-stationary dynamics. Both plain PPO and its recent extension to non-stationary environments (Moalla et al., 2024) struggle to preserve their adaptation capability at early stages of training. In contrast, our evidential version and its extension to directed exploration quickly adapt to new tasks. We posit that our new method, called *Evidential Proximal Policy Optimization (EPPO)*, brings such a performance boost as it fulfills both requirements of our hypothesis. Our contributions are as follows:

*(i)* We apply evidential deep learning for the first time to uncertainty-aware modeling of the value function in an on-policy deep actor-critic architecture. Our solution prescribes a hierarchical Bayesian generative process that maps the state observations to hyperpriors.

*(ii)* We use evidential value learning to develop two ways to construct a probabilistic extension of the *generalized advantage estimator* (Schulman et al., 2016). We show that performing directed exploration based on the probabilistic advantage estimators brings a consistent performance boost.

*(iii)* Due to the absence of a widely adopted benchmark, we introduce two new experiment designs tailored to evaluate the adaptation capabilities of continuous control agents to rapidly changing environment conditions. We benchmark our approach against two state-of-the-art PPO variants and observe that it outperforms them in the majority of cases.

## 2 Background

### 2.1 On-policy deep actor-critics

We define an infinite-horizon MDP as a tuple $\mathcal{M} \triangleq \langle \mathcal{S}, \mathcal{A}, P, r, \rho_0, \gamma \rangle$, where $\mathcal{S}$ represents the state space and $\mathcal{A}$ denotes the action space. Let $P$ be the state transition probability distribution such that $s' \sim P(\cdot|s,a)$ where $s \in \mathcal{S}$ and $a \in \mathcal{A}$. We assume a deterministic reward function $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ to facilitate presentation but without loss of generality. We denote the initial state distribution as $s_0 \sim \rho_0(\cdot)$ and the discount factor as $\gamma \in (0,1)$. We consider non-stationary environments with time-homogeneous reward functions and time-dependent state transition probabilities, i.e., $P_t(\cdot|s,a)$ for a time index $t$. We consider stationary stochastic policies defined as $a \sim \pi(\cdot|s)$. We use the following standard definitions of the action-value function $Q^\pi$, the value function $V^\pi$, and the advantage function $A^\pi$:

$$Q^\pi(s_t, a_t) \triangleq \mathbb{E}_{\substack{s_{t+1:\infty}, \\ a_{t+1:\infty}}} \left[ \sum_{l=0}^{\infty} \gamma^l r_{t+l} \right], \quad V^\pi(s_t) \triangleq \mathbb{E}_{\substack{s_{t+1:\infty}, \\ a_{t:\infty}}} \left[ \sum_{l=0}^{\infty} \gamma^l r_{t+l} \right],$$
$$A^\pi(s_t, a_t) \triangleq Q^\pi(s_t, a_t) - V^\pi(s_t),$$

where expectations are taken over trajectories induced by the policy $\pi$ and $r_{t+l} \triangleq r(s_{t+l}, a_{t+l})$. The colon notation $a : b$ refers to the inclusive range $(a, a+1, \ldots, b)$. We denote by $G_t \triangleq \sum_{l=0}^{\infty} \gamma^l r_{t+l}$ as the discounted sum of rewards.

We focus our study on on-policy deep actor-critic algorithms. We adopt PPO (Schulman et al., 2017) as the state-of-the-art representative of the conservative policy iteration approaches (Kakade and Langford, 2002). This algorithm family has been adopted in real-world scenarios due to its relative robustness stemming from the conservative policy updates that promote slower but more stable training. Prime examples include the control of physical robotic platforms (Lopes et al., 2018; Melo and Máximo, 2019) and fine-tuning large language models (Christiano et al., 2017; Bai et al., 2022; Touvron et al., 2023; Achiam et al., 2023; Zheng et al., 2023). PPO is a policy gradient method that updates the policy using a surrogate objective, ensuring that policy updates remain constrained to ensure an average policy improvement (Schulman et al., 2015). We follow the established practice and adopt the clipped objective as the surrogate function. PPO updates its policy $\pi_\theta$, parametrized by $\theta \in \Theta$:

$$\mathcal{L}_{\text{clip}}(\theta) = \mathbb{E}_{(s,a)\sim\pi_{\text{old}}} \left[ \min \left( \frac{\pi_\theta(a|s)}{\pi_{\text{old}}(a|s)} \hat{A}^{\pi_{\text{old}}}(s,a), \text{clip}\left( \frac{\pi_\theta(a|s)}{\pi_{\text{old}}(a|s)}, 1-\epsilon, 1+\epsilon \right) \hat{A}^{\pi_{\text{old}}}(s,a) \right) \right],$$

where $\hat{A}^{\pi_{\text{old}}}(s,a)$ is an estimate of the advantage function, and $\text{clip}(\frac{\pi_\theta(a|s)}{\pi_{\text{old}}(a|s)}, 1-\epsilon, 1+\epsilon)$ bounds the probability ratio within the range $[1-\epsilon, 1+\epsilon]$ for $\epsilon > 0$. PPO approximates the value function with $V_\phi$ parametrized by $\phi \in \Phi$. It uses the squared-error loss $\mathcal{L}_{\text{VF}}(\phi) = \mathbb{E}_{s_t}\left[(V_\phi(s_t) - G_t)^2\right]$ to learn $V_\phi$. The learned $V_\phi$ is then used to compute advantage estimates, guiding policy updates for more stable and efficient learning.

Modern PPO implementations use the *Generalized Advantage Estimation (GAE) (Schulman et al., 2016)* which is a technique for computing advantage estimates. This method helps reduce the variance on the return estimate while enabling step-wise updates via bootstrapping. GAE constructs the

advantage function using a weighted sum of multi-step temporal-difference errors. Let temporal-difference residual at time step $t$ be $\delta_t \triangleq r_t + \gamma V_\phi(s_{t+1}) - V_\phi(s_t)$. The GAE estimate is defined as the exponentially weighted sum of temporal difference residuals:

$$\hat{A}_t^{\mathrm{GAE}(\lambda),\pi} = \sum_{l=0}^{\infty} (\gamma\lambda)^l \delta_{t+l}, \tag{1}$$

where $\lambda \in [0, 1]$ is a hyperparameter that controls the bias-variance trade-off. GAE provides a flexible mechanism for estimating advantages, allowing reinforcement learning algorithms to achieve improved stability and faster convergence (Schulman et al., 2015, 2017).

**Directed exploration and non-stationarity.** Prior work has explored enhancing the exploration scheme of PPO to improve its sample-efficiency (Wang et al., 2019; Zhang et al., 2022) mainly in stationary settings. Directed exploration with PPO under non-stationary environments is an underexplored topic. Among very few prior works, Steinparz et al. (2022) proposes a model-based solution where the exploration is direct with intrinsic rewards derived from prediction errors in the learned transition dynamics model. We keep such approaches outside our scope due to their distinct computational requirements. Non-stationary RL is similar to but still different from meta-learning and continual learning. Meta-learning concerns with solving multiple tasks using a single model with the main motivation of increasing the data pool and reducing model development time. The setup is heavily studied in control scenarios (Al-Shedivat et al., 2018; Berseth et al., 2021; Bing et al., 2023). Continual learning has the same motivation as meta-learning but assumes a sequential generation of tasks. As each task is meant as a separate goal, continual learning algorithms aim to minimize catastrophic forgetting. Applications to RL also exist (Rusu et al., 2016; Kirkpatrick et al., 2017; Traoré et al., 2019; Kaplanis et al., 2019). Non-stationary RL aims to develop an agent that quickly adapts to a perpetually changing environment in short time intervals. Rapid adaptation to each new situation is desired instead of remembering all the previous situations. Non-stationary RL is much less studied than meta-learning and continual learning (Khetarpal et al., 2020).

### 2.2 Evidential deep learning

Bayesian inference (Bishop, 2006; Gelman et al., 2013) computes a posterior distribution over model parameters from a given likelihood function evaluated on data and a prior distribution chosen without access to data. Evidential deep learning (Sensoy et al., 2018) applies the classical Bayesian framework in a particular way where posteriors are fit to data-specific random variables from again data-specific prior distributions, the parameters of which are amortized by input observations. The amortized prior and the likelihood are chosen from conjugate families to ensure an analytically tractable computation of the posterior and the marginal likelihood, the latter of which is used as a training objective. Marginal likelihood optimization is also known as *Type II Maximum Likelihood* or *Empirical Bayes* (Efron, 2012).

We build our solution on Amini et al. (2020)'s adaptation of the evidential framework to regression problems as a typical continuous control task has real-valued reward functions. Amini et al. (2020)'s approach assumes that the output label $y$ corresponding to an input observation $\boldsymbol{x}$ follows a normally distributed likelihood with mean $\mu$ and variance $\sigma^2$. This distribution is assigned a Normal Inverse-Gamma ($\mathcal{NIG}$) distributed evidential prior:

$$\begin{aligned}(\mu, \sigma^2)|\boldsymbol{m}(\boldsymbol{x}) &\sim \mathcal{NIG}\left(\mu, \sigma^2|\omega(\boldsymbol{x}), \nu(\boldsymbol{x}), \alpha(\boldsymbol{x}), \beta(\boldsymbol{x})\right) \\ &= \mathcal{N}\left(\mu|\omega(\boldsymbol{x}), \sigma^2\nu(\boldsymbol{x})^{-1}\right) \mathcal{Inv}\mathcal{G}am\big(\sigma^2|\alpha(\boldsymbol{x}), \beta(\boldsymbol{x})\big),\end{aligned}$$

where the hyperparameters $\omega, \nu, \alpha, \beta$ are modeled as input-dependent functions, specifically neural networks with weights $\phi$. Throughout the paper, we suppress the dependency of the variables on $\phi$ and $\boldsymbol{x}$ for notational clarity, e.g., $\omega = \omega_\phi(\boldsymbol{x})$, and refer to them jointly as $\boldsymbol{m} \triangleq \boldsymbol{m}_\phi = (\omega, \nu, \alpha, \beta)$. Due to its conjugacy with the normal likelihood $p(y|\mu, \sigma^2) = \mathcal{N}(y|\mu, \sigma^2)$, the posterior $p(\mu, \sigma^2|y, \boldsymbol{m})$ and the marginal likelihood $p(y|\boldsymbol{m})$ are analytically tractable. This marginal is the well-known Student-t distribution:

$$y|\boldsymbol{m} \sim \mathrm{St}\left(y\middle|\omega, \frac{\beta(1-\nu)}{\nu\alpha}, 2\alpha\right).$$

The parameters of this distribution can be fit by maximizing the logarithm of the marginal likelihood function as

$$\mathcal{L}_{\text{NLL}}(\boldsymbol{m}) = \frac{1}{2}\log\left(\frac{\pi}{\nu}\right) - \alpha\log(\Omega) + \left(\alpha + \frac{1}{2}\right)\log\left((y-\omega)^2\nu + \Omega\right) + \log\left(\frac{\Gamma(\alpha)}{\Gamma\left(\alpha+\frac{1}{2}\right)}\right), \quad (2)$$

where $\Omega = 2\beta(1+\nu)$ and $\Gamma(\cdot)$ is the Gamma function. See Appendix A for the derivation of the posterior distribution.

Evidential deep learning has been extensively used in numerous machine learning frameworks and practical tasks (Gao et al., 2024). It has also been integrated into deep reinforcement learning for recommendation systems to provide uncertainty-aware recommendations (Wang et al., 2024), modeling policy network uncertainty to guide evidence-based exploration in behavioral analysis (Wang et al., 2023), incorporating uncertainty measures as rewards for decision-making in opinion inference tasks (Zhao et al., 2019), and calibrating prediction risk in safety-critical vision tasks through fine-grained reward optimization (Yang et al., 2024). However, we instead use it to model uncertainty in value function estimates, which enables confidence-based exploration and helps to preserve the plasticity of the neural network.

## 3 Method

We present a method that adapts the evidential approach to learn a distribution over the value function $V(s_t)$. The inferred distribution induces a corresponding distribution over the GAE, which both helps the model to detect the distributional shifts caused by the non-stationarity of the dynamics and guide directed exploration, thereby promotes rapid adaptation.

### 3.1 Evidential value learning

We assume our value function estimates $V(s_t)$ to be normally distributed with unknown mean $\mu$ and variance $\sigma^2$ which are jointly $\mathcal{NIG}$-distributed. We shorten the notation to $V_t = V(s_t)$ when the relation is clear from context. Naïvely following Amini et al. (2020)'s method would result in training instabilities similar to those found by Meinert et al. (2023) for regular supervised regression. We extend their non-Bayesian heuristic to a fully Bayesian hierarchical design. We summarize the model as a plate diagram in Figure 2. Introducing hyperpriors on each of the four evidential parameters, we get the forward model below:

$$\begin{aligned}
\omega(s) &\sim \mathcal{N}\left(\omega(s)|\mu_\omega^0, (\sigma_\omega^0)^2\right), \\
\nu(s) &\sim \mathcal{G}am\left(\nu(s)|\alpha_\nu^0, \beta_\nu^0\right), \\
\alpha(s) &\sim \mathcal{G}am\left(\alpha(s)|\alpha_\alpha^0, \beta_\alpha^0\right), \\
\beta(s) &\sim \mathcal{G}am\left(\beta(s)|\alpha_\beta^0, \beta_\beta^0\right), \\
\sigma^2 &\sim \mathcal{I}nv\mathcal{G}am\left(\sigma^2|\alpha(s), \beta(s)\right), \\
\mu|\sigma^2 &\sim \mathcal{N}\left(\mu|\omega(s), \sigma^2\nu(s)^{-1}\right), \\
V|\mu,\sigma^2 &\sim \mathcal{N}(V|\mu,\sigma^2),
\end{aligned}$$

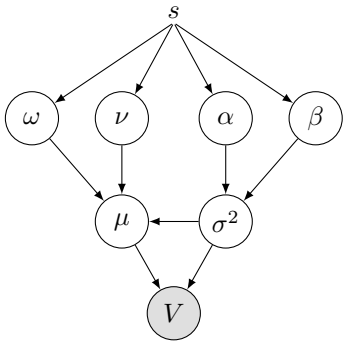


Figure 2: Plate diagram of our evidential value learning model.

where $\mathcal{G}am(\cdot)$ is the Gamma distribution, $\mu_\omega^0, \ldots, \beta_\beta^0$ are fixed hyperparameters.[1] We adopt a fixed set of hyperpriors to provide relatively flat and uninformative priors for all experiments. See Table 3 in the Appendix for further details. Following our notational convention, we suppress the dependency on $s$, e.g., $\omega = \omega(s)$, and combine the evidential parameters into $\boldsymbol{m} = (\omega, \nu, \alpha, \beta)$. Marginalizing over $(\mu, \sigma^2)$ yields

$$p(V, \boldsymbol{m}) = \int p(V|\mu,\sigma^2)p(\mu,\sigma^2|\boldsymbol{m})d(\mu,\sigma^2)\, p(\boldsymbol{m}) = p(V|\boldsymbol{m})p(\boldsymbol{m}),$$

[1] As $\omega$ is a deterministic transformation of the state $s$, the notation $\omega(s) \sim \mathcal{N}(\cdot)$ implies that its parameters $\phi$ are random variable such that $\omega(s)$ is normally distributed.

where $p(V|\boldsymbol{m})$ is a Student-t distribution parameterized as in Section 2.2. The hyperprior $p(\boldsymbol{m})$ acts as a regularizer in the log-joint objective. The training objective of evidential value learning is $\mathcal{L}(\boldsymbol{m}) = \mathcal{L}_{\text{NLL}}(\boldsymbol{m}) - \xi \log p(\boldsymbol{m})$, where $\xi \geq 0$ is a regularization coefficient.

The mean and variance of the state-value function output $V$ can be computed analytically as

$$\mathbb{E}_{V|\boldsymbol{m}}[V] = \mathbb{E}_{(\mu,\sigma^2)|\boldsymbol{m}}\left[\mathbb{E}_{V|\mu,\sigma^2}[V]\right] = \mathbb{E}_{(\mu,\sigma^2)|\boldsymbol{m}}[\mu] = \omega,$$

and

$$\begin{aligned}\text{var}_{V|\boldsymbol{m}}[V] &= \mathbb{E}_{(\mu,\sigma^2)|\boldsymbol{m}}\left[\text{var}_{V|\mu,\sigma^2}[V]\right] + \text{var}_{(\mu,\sigma^2)|\boldsymbol{m}}\left[\mathbb{E}_{V|\mu,\sigma^2}[V]\right] \\ &= \mathbb{E}_{(\mu,\sigma^2)|\boldsymbol{m}}\left[\sigma^2\right] + \text{var}_{(\mu,\sigma^2)|\boldsymbol{m}}[\mu] \\ &= \frac{\beta}{\alpha-1} + \frac{\beta}{\nu(\alpha-1)} = \frac{\beta}{\alpha-1}\left(1+\frac{1}{\nu}\right),\end{aligned}$$

where we assume $\alpha > 1$,[2] and the first equality follows from the law of total variance, which splits the marginal variance into aleatoric and epistemic uncertainty components. Reliance on $\text{var}_{y|\boldsymbol{m}}[y]$ therefore provides us with a principled way of incorporating irreducible uncertainty inherent in the environmental structure and reducible uncertainty due to improvable approximation errors in EPPO.

Evidential value learning belongs to a broader research field that incorporates distributional information into the reinforcement learning model, which can be roughly divided into two sub-fields. The first aims to account for aleatoric uncertainty caused by the inherent stochasticity of the environment. It focuses on accurately modeling the resulting distribution over the returns $G_t$, e.g., to infer risk-averse policies (Keramati et al., 2020). See Bellemare et al. (2023) for a recent textbook introduction. The other focuses on accounting for epistemic uncertainty inherent in the inference of a value function, usually relying on methods from Bayesian inference (Ghavamzadeh et al., 2015; Luis et al., 2024), e.g., to use it as a guide for exploration (e.g., Deisenroth and Rasmussen, 2011; Osband et al., 2019). Evidential value learning is relative to this second area of research. It uses an evidential model over the value function to induce a distribution over an advantage function that incorporates aleatoric and epistemic uncertainty for regularization and optimistic exploration.

### 3.2 Directed exploration via probabilistic advantages

Evidential value learning provides an uncertainty quantifier of the value function that detects shifts in the data distribution caused by the non-stationary state transition dynamics. It achieves this by increasing the uncertainty assigned to value predictions in response to distributional shifts, which subsequently contributes to the critic training loss, maintaining gradient flows and preserving plasticity. The uncertainty around the value function modeled by the distribution $p(V|\boldsymbol{m})$ also propagates through the advantage calculation, making GAE a random variable. Such a probabilistic modeling of the GAE enables uncertainty-aware policy updates by quantifying the uncertainty in the expected consequences of policy changes. Moreover, it facilitates the assignment of an exploration bonus that directs the policy toward underexplored regions of the state-action space. We implement this directed exploration scheme using an Upper Confidence Bound (UCB) on the advantage estimator:

$$\hat{A}_t^{\text{UCB}} = \mathbb{E}\left[\hat{A}_t^{\text{GAE}}\right] + \kappa\sqrt{\text{var}\left[\hat{A}_t^{\text{GAE}}\right]}, \tag{3}$$

where $\kappa \geq 0$ controls the confidence radius, and $\text{var}\left[\hat{A}_t^{\text{GAE}}\right]$ represents the variance estimate for GAE. The mean estimate for GAE is

$$\mathbb{E}\left[\hat{A}_t^{\text{GAE}}\right] = \sum_{l=0}^{\infty}(\gamma\lambda)^l\mathbb{E}[\delta_{t+l}],$$

and it is tractable due to the linearity of expectations and because the mean of the temporal difference $\mathbb{E}[\delta_t] = r_t + \gamma\mathbb{E}[V_{t+1}] - \mathbb{E}[V_t]$ is tractable. We propose two variants of EPPO that differ in the way the variance term in Equation (3) is computed.

[2]We enforce this condition by adding one to the neural network's output.

Table 1: *Performance evaluation on the slippery environments.* Area Under the Learning Curve (AULC) and Final Return (mean$\pm$se) scores are averaged over 15 repetitions. The highest mean values are highlighted in bold and underlined if they fall within one standard error of the best score. The average score represents the mean across all environments, while the average ranking is determined based on the ranking of the mean scores.

| Metric | Model | decreasing | | increasing | | Average | |
|---|---|---|---|---|---|---|---|
| | | Ant | HalfCheetah | Ant | HalfCheetah | Score | Ranking |
| AULC (↑) | PPO | 2355$\pm$203 | 2495$\pm$201 | 2237$\pm$254 | 2536$\pm$297 | 2406 | 4.3 |
| | PFO | 2522$\pm$109 | 2300$\pm$189 | 2485$\pm$90 | 1809$\pm$430 | 2279 | 4.3 |
| | EPPO$_{\text{mean}}$ | 2504$\pm$127 | 2432$\pm$299 | 2875$\pm$77 | 2822$\pm$219 | 2658 | 3.3 |
| | EPPO$_{\text{cor}}$ | 2561$\pm$128 | 2699$\pm$256 | **2944$\pm$80** | **3645$\pm$240** | **2962** | **1.5** |
| | EPPO$_{\text{ind}}$ | **2614$\pm$138** | **2866$\pm$218** | 2779$\pm$86 | 3374$\pm$220 | 2908 | 1.8 |
| FINAL RETURN (↑) | PPO | 2357$\pm$230 | 2483$\pm$212 | 2341$\pm$270 | 2720$\pm$310 | 2475 | 4.5 |
| | PFO | 2613$\pm$110 | 2346$\pm$214 | 2620$\pm$99 | 1906$\pm$462 | 2371 | 4.5 |
| | EPPO$_{\text{mean}}$ | 2660$\pm$131 | 2522$\pm$331 | 3002$\pm$92 | 2978$\pm$227 | 2790 | 2.8 |
| | EPPO$_{\text{cor}}$ | 2714$\pm$128 | 2821$\pm$274 | **3071$\pm$88** | **3872$\pm$248** | **3120** | **1.5** |
| | EPPO$_{\text{ind}}$ | **2741$\pm$145** | **2970$\pm$231** | 2941$\pm$89 | 3559$\pm$227 | 3053 | 1.8 |

**(EPPO$_{\text{cor}}$) Exploration via correlated uncertainties.** We derive the variance of $\hat{A}_t$ by focusing on its definition as the exponentially-weighted average of the $k$-step estimators $\hat{A}_t^{(k)} = -V_t + \gamma^k V_{t+k} + \sum_{l=0}^{k-1} \gamma^l r_{t+l}$. Because the rewards are deterministic in our setup and therefore have zero variance, we combine them into a generic constant term and obtain

$$\hat{A}_t^{\text{GAE}} \triangleq (1-\lambda)\sum_{l=1}^{\infty} \lambda^{l-1}\hat{A}_t^{(l)} = (1-\lambda)\Big(-V_t\sum_{l=0}^{\infty}\lambda^l + \sum_{l=1}^{\infty}\gamma^l\lambda^{l-1}V_{t+l}\Big) + \text{const}$$
$$= -V_t + \frac{1-\lambda}{\lambda}\sum_{l=1}^{\infty}(\gamma\lambda)^l V_{t+l} + \text{const}.$$

Given the conditional independence of the states, the resulting variance is

$$\text{var}\left[\hat{A}_t^{\text{GAE}}\right] = \text{var}\left[V_t\right] + \left(\frac{1-\lambda}{\lambda}\right)^2 \sum_{l=1}^{\infty}(\gamma\lambda)^2 \text{var}\left[V_{t+l}\right]. \tag{4}$$

We use this variance to construct the UCB in Equation (3) with $\kappa > 0$. We refer to this estimator as EPPO$_{\text{cor}}$ in the experiments.

**(EPPO$_{\text{ind}}$) Exploration via uncorrelated uncertainties.** We also consider the case where the $k$-step estimators $\hat{A}_t^{(k)}$ are assumed to be independent of each other. We then build the overall variance as the exponentially weighted sum of the individual $k$-step estimators. It can be shown easily (see Appendix A.2) that the resulting variance approximation is

$$\text{var}\left[\hat{A}_t^{\text{GAE}}\right] \approx \frac{1-\lambda}{1+\lambda}\text{var}\left[V_t\right] + \left(\frac{1-\lambda}{\lambda}\right)^2 \sum_{l=1}^{\infty}(\gamma\lambda)^2 \text{var}\left[V_{t+l}\right], \tag{5}$$

i.e., the influence of the current value variance is down-scaled by a factor $(1-\lambda)/(1+\lambda) < 1$ relative to the future time steps in EPPO$_{\text{ind}}$ compared with EPPO$_{\text{cor}}$. This adjustment makes EPPO$_{\text{ind}}$ more far-sighted for the same $\kappa$. We use the variance estimate in Equation (5) to construct the UCB in Equation (3) with $\kappa > 0$.

## 4 Experiments

We design experiments to benchmark EPPO variants against state-of-the-art on-policy deep actor-critic algorithms in non-stationary continuous control environments. To amplify the effect of non-stationarity on model performance, we define tasks over short time intervals and introduce changes in the environment dynamics. In each interval, agents are required to detect the change, explore effectively, and adapt rapidly to maximize the overall return during learning. We run our simulations

Table 2: *Performance evaluation on the paralysis environments.* Area Under the Learning Curve (AULC) and Final Return (mean$\pm$se) scores are averaged over 15 repetitions. The highest mean values are highlighted in bold and underlined if they fall within one standard error of the best score. The average score represents the mean across all environments, while the average ranking is determined based on the ranking of the mean scores.

| Metric | Environment | Strategy | Model | | | | |
|---|---|---|---|---|---|---|---|
| | | | PPO | PFO | EPPO$_{\text{mean}}$ | EPPO$_{\text{cor}}$ | EPPO$_{\text{ind}}$ |
| AULC (↑) | Ant | back-one | 2009$\pm$312 | 2259$\pm$113 | 2455$\pm$78 | 2608$\pm$129 | **2724$\pm$174** |
| | | front-one | 2054$\pm$260 | 2098$\pm$87 | 2407$\pm$92 | **2749$\pm$112** | 2743$\pm$121 |
| | | back-two | 1928$\pm$174 | 2136$\pm$57 | **2203$\pm$79** | 2099$\pm$80 | 2088$\pm$94 |
| | | front-two | 1975$\pm$174 | 2000$\pm$56 | 2259$\pm$85 | **2294$\pm$93** | 2275$\pm$76 |
| | | parallel | 2162$\pm$175 | 2298$\pm$86 | 2350$\pm$103 | 2348$\pm$127 | **2558$\pm$159** |
| | | cross | 1898$\pm$185 | 2161$\pm$71 | 2167$\pm$91 | 2197$\pm$72 | **2281$\pm$72** |
| | Average AULC on Ant | | 2004 | 2159 | 2307 | 2383 | **2445** |
| | HalfCheetah | back-one | 2444$\pm$223 | 2181$\pm$282 | 3160$\pm$270 | 3502$\pm$173 | **3515$\pm$131** |
| | | front-one | 2076$\pm$299 | 2485$\pm$271 | 3384$\pm$227 | 3558$\pm$224 | **3695$\pm$241** |
| | | cross-v1 | 2311$\pm$235 | 2314$\pm$287 | 3002$\pm$238 | **3205$\pm$224** | 3120$\pm$207 |
| | | cross-v2 | 2477$\pm$220 | 1903$\pm$245 | 3039$\pm$195 | 3250$\pm$195 | **3283$\pm$212** |
| | Average AULC on HalfCheetah | | 2327 | 2221 | 3146 | 3379 | **3403** |
| Overall Average AULC Score | | | 2133 | 2184 | 2643 | 2781 | **2828** |
| Overall Average Ranking on AULC | | | 4.8 | 4.0 | 2.7 | 1.9 | **1.6** |
| FINAL RETURN (↑) | Ant | back-one | 2261$\pm$325 | 2503$\pm$114 | 2709$\pm$83 | 2891$\pm$129 | **2977$\pm$169** |
| | | front-one | 2253$\pm$284 | 2337$\pm$87 | 2649$\pm$96 | **3020$\pm$107** | 2956$\pm$135 |
| | | back-two | 2188$\pm$205 | 2454$\pm$62 | **2533$\pm$96** | 2400$\pm$86 | 2327$\pm$112 |
| | | front-two | 2282$\pm$189 | 2249$\pm$61 | 2536$\pm$98 | **2633$\pm$97** | 2605$\pm$94 |
| | | parallel | 2397$\pm$195 | 2601$\pm$95 | 2649$\pm$114 | 2653$\pm$144 | **2883$\pm$163** |
| | | cross | 2144$\pm$220 | 2467$\pm$79 | 2495$\pm$103 | 2500$\pm$75 | **2570$\pm$72** |
| | Average Final Return on Ant | | 2254 | 2435 | 2595 | 2683 | **2720** |
| | HalfCheetah | back-one | 2504$\pm$260 | 2275$\pm$303 | 3320$\pm$287 | 3696$\pm$178 | **3718$\pm$133** |
| | | front-one | 2115$\pm$325 | 2577$\pm$295 | 3540$\pm$235 | 3724$\pm$232 | **3892$\pm$248** |
| | | cross-v1 | 2405$\pm$271 | 2349$\pm$310 | 3159$\pm$254 | **3420$\pm$231** | 3341$\pm$220 |
| | | cross-v2 | 2550$\pm$235 | 1953$\pm$260 | 3217$\pm$204 | 3450$\pm$208 | **3468$\pm$228** |
| | Average Final Return on HalfCheetah | | 2394 | 2288 | 3309 | 3573 | **3605** |
| Overall Average Final Return Score | | | 2310 | 2376 | 2881 | 3039 | **3074** |
| Overall Average Ranking on Final Return | | | 4.6 | 4.2 | 2.8 | 1.8 | **1.6** |

on the Ant and HalfCheetah environments using the 'v5' versions of MuJoCo environments (Todorov et al., 2012). For further details on the experimental pipeline and hyperparameters, see Appendix B.

We benchmark our evidential approach against two baselines: *(i)* PPO (Schulman et al., 2017): A widely used on-policy deep actor-critic reinforcement learning algorithm that serves as the foundation for EPPO. We follow the most recent implementation practices. We use the GAE method (Schulman et al., 2016) to estimate value function targets. *(ii)* PFO (Moalla et al., 2024): A recent PPO variant that addresses the plasticity problem under non-stationarity by extending the trust region constraint to the feature space. We also evaluate the EPPO variant with $\kappa = 0$ which performs evidential value learning without directed exploration. We denote this model as $\texttt{EPPO}_{\texttt{mean}}$. Its relative performance indicates the contribution of directed exploration.

We propose two experimental setups to assess the models' ability to adapt to non-stationarity. The setups are as follows:

*(i)* **Slippery environments.** Inspired by Dohare et al. (2021, 2024), we construct a non-stationary environment by varying the friction coefficient of the floor in locomotion tasks using the Ant and HalfCheetah environments. We induce non-stationarity to the environments by changing friction every 500 000 steps. To create more challenging task changes, we implement two strategies: decreasing, where friction starts at its maximum value and gradually decreases, and increasing, where friction starts at its minimum value and gradually increases. This setup ensures that agents encounter non-stationarity in both increasing and decreasing friction scenarios. The minimum friction is set to $0.5$ and the maximum to $4.0$, based on the feasibility of solving the tasks—extreme friction values may make movement too difficult because of slipping or an inability to move forward. We define 15 tasks by changing the friction with a positive or negative offset of $0.25$.

*(ii)* **Paralysis environments.** We design a new set of non-stationarity experiments by dynamically altering the torque capabilities of the leg joints in the `Ant` and `HalfCheetah` environments, inspired by Al-Shedivat et al. (2018). Each experiment involves paralyzing different joints to diversify the control tasks across experiments. We generate six torque modification schemes for `Ant` and four for `HalfCheetah`. In each scheme, we select specific joints and progressively reduce their torque capability until they become fully paralyzed. Then, we gradually restore their functionality, returning to the fully operational state. This yields a sequence of nine tasks, where each joint either loses or regains $25\%$ of its torque capacity in each step, following the pattern: $[100, 75, 50, 25, 0, 25, 50, 75, 100]$.

**Results.** We evaluate model performance based on two scores: *(i) Area Under Learning Curve (AULC):* The average return computed throughout training. *(ii) Final Return:* The average return computed during the final steps of training for each task. AULC indicates how much reward is collected throughout the entire training process and how quickly a model converges to its final performance. A higher AULC suggests greater adaptation ability during training, assuming other factors remain equal. The final return measures an agent's ability to adapt to each individual task. A higher final return score reflects better performance on a specific task. These metrics evaluate an agent's adaptation performance and plasticity. We provide experiment result visualizations in Appendix B.3, illustrating episode returns throughout the changing tasks and demonstrating both quantitative and qualitative performance differences between EPPO variants and our baselines. Our experimental findings are as follows:

*(i) Evidential value learning accelerates convergence and improves training stability while preserving plasticity.* EPPO variants achieve better task adaptation than the baselines, as reflected in the final return scores and the learning curves. Furthermore, evidential value learning reaches its final performance more quickly and improves the training stability, as evidenced by the AULC scores. By maintaining flexibility in the value function throughout training, evidential value learning maintains the plasticity through probabilistic modeling.

*(ii) Directed exploration boosts performance.* EPPO variants with directed exploration outperform the baselines in both metrics. They also outperform the EPPO variant that uses the value function mean for policy improvement. This outcome highlights the unique contribution that directed exploration makes to performance.

We highlight that our best-performing algorithms, equipped with a value function incorporating uncertainty quantification, enable the agent to maintain plasticity and perform directed exploration. This facilitates quick and continual adaptation to non-stationary environments, supporting our key hypothesis.

**Compute time.** We perform our experiments using two computers equipped with GeForce RTX 4090 GPUs, an Intel(R) Core(TM) i7-14700K CPU running at 5.6 GHz, and 96 GB of memory. Our experiments are conducted on these two machines with four parallel seeds. We measure approximately the total wall-clock time for the computation of $15$ seeds across all environments in $74.8$ hours for PPO, $75$ hours for PFO, $75.3$ hours for $\text{EPPO}_{\text{mean}}$, $75.4$ hours for $\text{EPPO}_{\text{cor}}$, and $75.6$ hours for $\text{EPPO}_{\text{ind}}$. The total execution time for all experiments reported in this work is approximately $376.1$ hours, equivalent to $15.6$ days on two GPU-supported workstations.

## 5 Limitations and broader impact

We observe EPPO to be sensitive to the choice of some hyperparameters such as the regularization coefficient ($\xi$) and the confidence radius ($\kappa$). While this is a common weakness of most deep reinforcement learning algorithms, the effect of the resulting brittleness may be larger in non-stationary environments. Choosing the confidence radius based on a generalization bound as practiced commonly in bandit research (Li et al., 2010; Srinivas et al., 2010; Kaufmann et al., 2012; Lattimore and Szepesvári, 2020) and increasing the Bayesian modeling hierarchy may make EPPO more robust to hyperparameters. As an on-policy policy-gradient algorithm, EPPO shares similar theoretical properties to other PPO variants. The effect of the evidential learning extension on non-asymptotic convergence is a challenging problem, hence requires a special investigation. Although our study demonstrates that evidential value learning improves the control of non-stationary systems, we did not

investigate whether the quantified uncertainties are calibrated and how big the correlation is between their calibration and performance. We leave this interesting problem to a separate study. Our results are limited to rigid-body locomotors of a single physics engine, despite covering comprehensive variations of challenging scenarios at non-stationarity levels exceeding prior studies. We do not expect extending our results to even more tasks to bring any additional insights. We view testing our approach on physical robotic systems to be the natural next step.

Continuous control of a non-stationary environment is the core problem of building an agentic system on a physical platform. Non-stationarity is the essential element of developing co-adaptive environments where robots and humans learn via bilateral feedback. Such a co-adaptation is crucial to ensure a human-centric growth of the capabilities of agentic systems of the future. Our work contributes to the responsible AI initiative by facilitating the application of the powerful PPO algorithm to co-adaptive system development.

## References


Abbas, Z., Zhao, R., Modayil, J., White, A., and Machado, M. C. (2023). Loss of plasticity in continual deep reinforcement learning. In *Proceedings of The 2nd Conference on Lifelong Learning Agents*.

Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al. (2023). GPT-4 technical report. *arXiv preprint arXiv:2303.08774*.

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

Amini, A., Schwarting, W., Soleimany, A., and Rus, D. (2020). Deep evidential regression. In *Advances in Neural Information Processing Systems*.

Ba, J. L., Kiros, J. R., and Hinton, G. E. (2016). Layer normalization. *arXiv preprint arXiv:1607.06450*.

Bai, Y., Jones, A., Ndousse, K., Askell, A., Chen, A., DasSarma, N., Drain, D., Fort, S., Ganguli, D., Henighan, T., et al. (2022). Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.

Bellemare, M. G., Dabney, W., and Rowland, M. (2023). *Distributional Reinforcement Learning*. MIT Press.

Berseth, G., Zhang, Z., Zhang, G., Finn, C., and Levine, S. (2021). CoMPS: Continual meta policy search. *arXiv preprint arXiv:2112.04467*.

Besbes, O., Gur, Y., and Zeevi, A. (2014). Stochastic multi-armed-bandit problem with non-stationary rewards. In *Advances in Neural Information Processing Systems*.

Bing, Z., Lerch, D., Huang, K., and Knoll, A. (2023). Meta-reinforcement learning in non-stationary and dynamic environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Bishop, C. (2006). *Pattern Recognition and Machine Learning*. Springer.

Christiano, P. F., Leike, J., Brown, T., Martic, M., Legg, S., and Amodei, D. (2017). Deep reinforcement learning from human preferences. *Advances in Neural Information Processing Systems*.

Chung, W., Cherif, L., Precup, D., and Meger, D. (2024). Parseval regularization for continual reinforcement learning. In *Advances in Neural Information Processing Systems*.

Deisenroth, M. and Rasmussen, C. E. (2011). PILCO: A model-based and data-efficient approach to policy search. In *International Conference on Machine Learning*.

Dohare, S., Hernandez-Garcia, J. F., Lan, Q., Rahman, P., Mahmood, A. R., and Sutton, R. S. (2024). Loss of plasticity in deep continual learning. *Nature*.

Dohare, S., Lan, Q., and Mahmood, A. R. (2023). Overcoming policy collapse in deep reinforcement learning. In *Sixteenth European Workshop on Reinforcement Learning*.

Dohare, S., Sutton, R. S., and Mahmood, A. R. (2021). Continual backprop: Stochastic gradient descent with persistent randomness. *arXiv preprint arXiv:2108.06325*.

Efron, B. (2012). *Large-scale Inference: Empirical Bayes Methods for Estimation, Testing, and Prediction*. Cambridge University Press.

Gao, J., Chen, M., Xiang, L., and Xu, C. (2024). A comprehensive survey on evidential deep learning and its applications. *arXiv preprint arXiv:2409.04720*.

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., and Rubin, D. B. (2013). *Bayesian Data Analysis*. Chapman and Hall/CRC.

Ghavamzadeh, M., Mannor, S., Pineau, J., Tamar, A., et al. (2015). Bayesian reinforcement learning: A survey. *Foundations and Trends® in Machine Learning*.

Kakade, S. and Langford, J. (2002). Approximately optimal approximate reinforcement learning. In *International Conference on Machine Learning*.

Kandemir, M., Akgül, A., Haussmann, M., and Unal, G. (2022). Evidential Turing processes. In *International Conference on Learning Representations*.

Kaplanis, C., Shanahan, M., and Clopath, C. (2019). Policy consolidation for continual reinforcement learning. In *International Conference on Machine Learning*.

Kaufmann, E., Cappe, O., and Garivier, A. (2012). On Bayesian upper confidence bounds for bandit problems. In *International Conference on Artificial Intelligence and Statistics*.

Keramati, R., Dann, C., Tamkin, A., and Brunskill, E. (2020). Being optimistic to be conservative: Quickly learning a CVaR policy. *Proceedings of the AAAI Conference on Artificial Intelligence*.

Khetarpal, K., Riemer, M., Rish, I., and Precup, D. (2020). Towards continual reinforcement learning: A review and perspectives. *Journal of Artificial Intelligence Research*.

Kingma, D. P. and Ba, J. L. (2015). Adam: A method for stochastic optimization. In *International Conference on Learning Representations*.

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., et al. (2017). Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*.

Kumar, S., Marklund, H., and Roy, B. V. (2025). Maintaining plasticity in continual learning via regenerative regularization. In *Proceedings of The 3rd Conference on Lifelong Learning Agents*.

Lattimore, T. and Szepesvári, C. (2020). *Bandit Algorithms*. Cambridge University Press.

Lee, H., Cho, H., Kim, H., Kim, D., Min, D., Choo, J., and Lyle, C. (2024). Slow and steady wins the race: Maintaining plasticity with Hare and Tortoise networks. In *International Conference on Machine Learning*.

Li, L., Chu, W., Langford, J., and Schapire, R. E. (2010). A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th International Conference on World Wide Web*.

Lopes, G. C., Ferreira, M., da Silva Simões, A., and Colombini, E. L. (2018). Intelligent control of a quadrotor with proximal policy optimization reinforcement learning. In *2018 Latin American Robotic Symposium, 2018 Brazilian Symposium on Robotics (SBR) and 2018 Workshop on Robotics in Education (WRE)*.

Luis, C. E., Bottero, A. G., Vinogradska, J., Berkenkamp, F., and Peters, J. (2024). Value-distributional model-based reinforcement learning. *Journal of Machine Learning Research*.

Lyle, C., Rowland, M., and Dabney, W. (2022). Understanding and preventing capacity loss in reinforcement learning. In *International Conference on Learning Representations*.

Lyle, C., Zheng, Z., Khetarpal, K., Hasselt, H. v., Pascanu, R., Martens, J., and Dabney, W. (2025). Disentangling the causes of plasticity loss in neural networks. In *Proceedings of The 3rd Conference on Lifelong Learning Agents*.

Lyle, C., Zheng, Z., Nikishin, E., Avila Pires, B., Pascanu, R., and Dabney, W. (2023). Understanding plasticity in neural networks. In *International Conference on Machine Learning*.

Meinert, N., Gawlikowski, J., and Lavin, A. (2023). The unreasonable effectiveness of deep evidential regression. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Melo, L. C. and Máximo, M. R. O. A. (2019). Learning humanoid robot running skills through proximal policy optimization. In *2019 Latin American robotics symposium (LARS), 2019 Brazilian symposium on robotics (SBR) and 2019 workshop on robotics in education (WRE)*.

Moalla, S., Miele, A., Pyatko, D., Pascanu, R., and Gulcehre, C. (2024). No representation, no trust: Connecting representation, collapse, and trust issues in PPO. In *Advances in Neural Information Processing Systems*.

Nair, V. and Hinton, G. E. (2010). Rectified linear units improve restricted Boltzmann machines. In *International Conference on Machine Learning*.

Nikishin, E., Schwarzer, M., D'Oro, P., Bacon, P.-L., and Courville, A. (2022). The primacy bias in deep reinforcement learning. In *International Conference on Machine Learning*.

Osband, I., Van Roy, B., Russo, D. J., and Wen, Z. (2019). Deep exploration via randomized value functions. *Journal of Machine Learning Research*.

Rusu, A. A., Rabinowitz, N. C., Desjardins, G., Soyer, H., Kirkpatrick, J., Kavukcuoglu, K., Pascanu, R., and Hadsell, R. (2016). Progressive neural networks. *arXiv preprint arXiv:1606.04671*.

Schulman, J., Levine, S., Abbeel, P., Jordan, M., and Moritz, P. (2015). Trust region policy optimization. In *International Conference on Machine Learning*.

Schulman, J., Moritz, P., Levine, S., Jordan, M., and Abbeel, P. (2016). High-dimensional continuous control using generalized advantage estimation. *International Conference on Learning Representations*.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. (2017). Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Sensoy, M., Kaplan, L., and Kandemir, M. (2018). Evidential deep learning to quantify classification uncertainty. In *Advances in Neural Information Processing Systems*.

Srinivas, N., Krause, A., Kakade, S., and Seeger, M. (2010). Gaussian process optimization in the bandit setting: No regret and experimental design. In *International Conference on Machine Learning*.

Steinparz, C. A., Schmied, T., Paischer, F., Dinu, M.-c., Patil, V. P., Bitto-nemling, A., Eghbal-zadeh, H., and Hochreiter, S. (2022). Reactive exploration to cope with non-stationarity in lifelong reinforcement learning. In *Proceedings of The 1st Conference on Lifelong Learning Agents*.

Sutton, R. S., Koop, A., and Silver, D. (2007). On the role of tracking in stationary environments. In *International Conference on Machine Learning*.

Thrun, S. (1998). Lifelong learning algorithms. In *Learning to learn*. Springer.

Todorov, E., Erez, T., and Tassa, Y. (2012). MuJoCo: A physics engine for model-based control. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al. (2023). Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Traoré, R., Caselles-Dupré, H., Lesort, T., Sun, T., Cai, G., Díaz-Rodríguez, N., and Filliat, D. (2019). DisCoRL: Continual reinforcement learning via policy distillation. *arXiv preprint arXiv:1907.05855*.

Wang, D., Neupane, K. P., Zheng, E., and Yu, Q. (2024). Evidential conservative Q-learning for dynamic recommendations.

Wang, D., Pandey, D. S., Neupane, K. P., Yu, Z., Zheng, E., Zheng, Z., and Yu, Q. (2023). Deep temporal sets with evidential reinforced attentions for unique behavioral pattern discovery. In *International Conference on Machine Learning*.

Wang, Y., He, H., Tan, X., and Gan, Y. (2019). Trust region-guided proximal policy optimization. In *Advances in Neural Information Processing Systems*.

Yang, H., Chen, C., Chen, Y., Scheppach, M., Yip, H. C., and Dou, Q. (2024). Uncertainty estimation for safety-critical scene segmentation via fine-grained reward maximization. In *Advances in Neural Information Processing Systems*.

Zhang, J., Zhang, Z., Han, S., and Lü, S. (2022). Proximal policy optimization via enhanced exploration efficiency. *Information Sciences*.

Zhao, P., Zhang, L., Jiang, Y., and Zhou, Z.-H. (2020). A simple approach for non-stationary linear bandits. In *International Conference on Machine Learning*.

Zhao, X., Hu, S., Cho, J.-H., and Chen, F. (2019). Uncertainty-based decision making using deep reinforcement learning. In *2019 22th International Conference on Information Fusion (FUSION)*.

Zheng, R., Dou, S., Gao, S., Hua, Y., Shen, W., Wang, B., Liu, Y., Jin, S., Liu, Q., Zhou, Y., et al. (2023). Secrets of RLHF in large language models part i: PPO. *arXiv preprint arXiv:2307.04964*.

# APPENDIX

## A Derivations

### A.1 Derivations for evidential deep learning

We follow the derivations from Amini et al. (2020), adapting them to our notation whenever necessary.

**Normal inverse-gamma ($\mathcal{NIG}$) distribution** We use the notation

$$\begin{aligned}(\mu, \sigma^2)|\boldsymbol{m} &\sim \mathcal{NIG}\left(\mu, \sigma^2|\omega, \nu, \alpha, \beta\right) \\ &= \mathcal{N}(\mu|\omega, \sigma^2\nu^{-1})\mathcal{I}nv\mathcal{G}am(\sigma^2|\alpha, \beta) \\ &= \frac{\beta^\alpha\sqrt{\nu}}{\Gamma(\alpha)\sqrt{2\pi\sigma^2}}\left(\frac{1}{\sigma^2}\right)^{\alpha+1}\exp\left(-\frac{2\beta+\nu(\omega-\mu)^2}{2\sigma^2}\right),\end{aligned}$$

where $\omega \in \mathbb{R}$ and $\lambda, \alpha, \beta > 0$. The mean, mode, and variance are given by

$$\mathbb{E}\left[\mu\right] = \omega, \quad \mathbb{E}\left[\sigma^2\right] = \frac{\beta}{\alpha-1}, \quad \text{var}\left[\mu\right] = \frac{\beta}{\nu(\alpha-1)}, \qquad \text{for } \alpha > 1.$$

The second and third terms respectively correspond to aleatoric and epistemic uncertainty.

**Model evidence and type II maximum likelihood loss** We derive the model evidence of an $\mathcal{NIG}$ distribution. We marginalize out $\mu$ and $\sigma$:

$$\begin{aligned}p(y|\boldsymbol{m}) &= \int_{(\mu,\sigma^2)} p(y|\mu,\sigma^2)p(\mu,\sigma^2|\boldsymbol{m})d(\mu,\sigma^2) \\ &= \int_{\sigma^2=0}^{\infty}\int_{\mu=-\infty}^{\infty} p\left(y|\mu,\sigma^2\right)p\left(\mu,\sigma^2|\boldsymbol{m}\right)d\mu\, d\sigma^2 \\ &= \int_{\sigma^2=0}^{\infty}\int_{\mu=-\infty}^{\infty} p\left(y|\mu,\sigma^2\right)p\left(\mu,\sigma^2|\omega,\nu,\alpha,\beta\right)d\mu\, d\sigma^2 \\ &= \int_{\sigma^2=0}^{\infty}\int_{\mu=-\infty}^{\infty}\left[\sqrt{\frac{1}{2\pi\sigma^2}}\exp\left(-\frac{(y-\mu)^2}{2\sigma^2}\right)\right] \\ &\qquad \left[\frac{\beta^\alpha\sqrt{\nu}}{\Gamma(\alpha)\sqrt{2\pi\sigma^2}}\left(\frac{1}{\sigma^2}\right)^{\alpha+1}\exp\left(-\frac{2\beta+\nu(\omega-\mu)^2}{2\sigma^2}\right)\right]d\mu\, d\sigma^2 \\ &= \int_{\sigma^2=0}^{\infty}\frac{\beta^\alpha\sigma^{-3-2\alpha}}{\sqrt{2\pi}\sqrt{1+1/\nu}\Gamma(\alpha)}\exp\left(-\frac{2\beta+\frac{\nu(y-\omega)^2}{1+\nu}}{2\sigma^2}\right)d\sigma^2 \\ &= \int_{\sigma=0}^{\infty}\frac{\beta^\alpha\sigma^{-3-2\alpha}}{\sqrt{2\pi}\sqrt{1+1/\nu}\Gamma(\alpha)}\exp\left(-\frac{2\beta+\frac{\nu(y-\omega)^2}{1+\nu}}{2\sigma^2}\right)2\sigma\, d\sigma \\ &= \frac{\Gamma(1/2+\alpha)}{\Gamma(\alpha)}\sqrt{\frac{\nu}{\pi}}(2\beta(1+\nu))^\alpha\left(\nu\left(y-\gamma\right)^2+2\beta(1+\nu)\right)^{-\left(\frac{1}{2}+\alpha\right)},\end{aligned}$$

where $\Gamma(\cdot)$ is the Gamma function. Therefore, we have that the evidence distribution $p(y|\boldsymbol{m})$ is a Student-t distribution, i.e.,

$$p(y|\boldsymbol{m}) = \text{St}\left(y\middle|\omega, \frac{\beta(1-\nu)}{\nu\alpha}, 2\alpha\right),$$

which is evaluated at $y$ with location parameter $\omega$, scale parameter $\beta(1-\nu)/\nu\alpha$, and degrees of freedom $2\alpha$. We can compute the negative log-likelihood (NLL) loss as:

$$\begin{aligned}\mathcal{L}_{\text{NLL}}(\boldsymbol{m}) &= -\log p(y|\boldsymbol{m}) \\ &= -\log\left(\text{St}\left(y\middle|\omega, \frac{\beta(1-\lambda)}{\nu\alpha}, 2\alpha\right)\right) \\ &= \frac{1}{2}\log\left(\frac{\pi}{\nu}\right) - \alpha\log(\Omega) + \left(\alpha+\frac{1}{2}\right)\log\left((y-\omega)^2\nu+\Omega\right) + \log\left(\frac{\Gamma(\alpha)}{\Gamma\left(\alpha+\frac{1}{2}\right)}\right)\end{aligned}$$

where $\Omega = 2\beta(1+\nu)$.

### A.2 Derivations for the generalized advantage estimator

Given the definition of the $k$-step estimator as $\hat{A}_t^{(k)} = -V_t + \gamma^k V_{t+k} + \sum_{l=0}^{k-1}\gamma^l r_{t+l}$, we have that

$$\text{var}\left[\hat{A}_t^{(k)}\right] = \text{var}\left[V_t\right] + \gamma^{2k}\text{var}\left[V_{t+k}\right].$$

We adapt our estimator's variance approximation for EPPO$_{\text{ind}}$ to

$$\begin{aligned}\text{var}\left[\hat{A}_t^{\text{GAE}}\right] &\approx (1-\lambda)^2\sum_{l=1}^{\infty}\lambda^{2(l-1)}\text{var}\left[\hat{A}_t^{(l)}\right] \\ &= (1-\lambda)^2\left(\text{var}\left[V_t\right]\sum_{l=0}^{\infty}\lambda^{2l} + \sum_{l=1}^{\infty}\gamma^{2l}\lambda^{2(l-1)}\text{var}\left[V_{t+l}\right]\right) \\ &= \frac{(1-\lambda)^2}{1-\lambda^2}\text{var}\left[V_t\right] + \left(\frac{1-\lambda}{\lambda}\right)^2\sum_{l=1}(\gamma\lambda)^{2l}\text{var}\left[V_{t+l}\right],\end{aligned}$$

i.e., the form we have in (5).

## B Further details on experiments

### B.1 Experiment Details

In this section, we outline the details and design choices for our experiments and non-stationary environments. We use the `Ant` and `HalfCheetah` environments with the 'v5' versions of MuJoCo (Todorov et al., 2012), as these tasks do not reward the agent for maintaining stability.

#### B.1.1 Slippery environments

Our experimental design is inspired by Dohare et al. (2021, 2024). We construct a non-stationary environment by varying the floor's friction coefficient. Searching for feasible friction values we set the minimum at $0.5$ and the maximum at $4.0$. Outside of this range, solving the tasks either become infeasible or yield low rewards due to excessive action costs, limited movement, or the agent simply falling.

To introduce variation across tasks while ensuring differences between tasks, we incrementally change the friction by $0.25$, resulting in 15 distinct tasks. We implement two strategies for these changes:

- `decreasing`: Friction starts at its maximum value and gradually decreases.
- `increasing`: Friction starts at its minimum value and gradually increases.

These setups ensures that the agents experience non-stationarity in both increasing and decreasing friction scenarios. We implement these changes by modifying the publicly available environment XML files[3] [4] to adjust the floor friction coefficients.

[3] https://github.com/Farama-Foundation/Gymnasium/blob/main/gymnasium/envs/mujoco/assets/ant.xml

[4] https://github.com/Farama-Foundation/Gymnasium/blob/main/gymnasium/envs/mujoco/assets/half_cheetah.xml

#### B.1.2 Paralysis environments

We introduce a novel set of non-stationarity experiments by dynamically modifying the torque capabilities of leg joints in the `Ant` and `HalfCheetah` environments, inspired by Al-Shedivat et al. (2018). Specifically, we define six torque modification schemes for `Ant` and four for HalfCheetah. Each scheme targets selected joints, progressively reducing their torque capacity until they become completely paralyzed, after which their functionality is gradually restored to the fully operational state. This process results in a sequence of nine tasks, where each joint's torque capacity changes in increments of $25\%$, following the pattern: $[100, 75, 50, 25, 0, 25, 50, 75, 100]$. Note that while the policy can still output full torques, the applied torque is scaled according to the specified coefficients.

**Paralysis on ant.** The `Ant` environment consists of four legs and eight joints. We design distinct experiments by paralyzing different joints, ensuring that control tasks remain unique across experiments. For instance, if we paralyze the right back leg, we do not conduct a separate experiment on the left back leg, as the locomotion is symmetric and would result in an equivalent control task. We create the following experiments:

- `back-one`: Paralyzing a single back leg. The affected joints are $6$ and $7$.
- `front-one`: Paralyzing a single front leg. The affected joints are $2$ and $3$.
- `back-two`: Paralyzing both back legs. The affected joints are $0, 1, 6$, and $7$.
- `front-two`: Paralyzing both front legs. The affected joints are $2, 3, 4$, and $5$.
- `cross`: Paralyzing diagonally opposite legs (right back and left front). The affected joints are $0, 1, 2$, and $3$.
- `parallel`: Paralyzing the left-side legs (one back and one front). The affected joints are $2, 3, 6$, and $7$.

**Paralysis on halfCheetah.** The `HalfCheetah` environment consists of two legs and four joints. To prevent the agent from resorting to crawling, we modify only one joint per leg. We create the following experiments:

- `back-one`: Paralyzing a single joint in the back leg. The affected joint is $2$.
- `front-one`: Paralyzing a single joint in the front leg. The affected joint is $5$.
- `cross-v1`: Paralyzing diagonally opposite joints in the back and front legs. The affected joints are $2$ and $4$.
- `cross-v2`: Paralyzing a different pair of diagonally opposite joints in the back and front legs. The affected joints are $1$ and $5$.

### B.2 Hyperparameters

In this section, we provide all the necessary details to reproduce EPPO. We evaluate EPPO with $15$ repetitions using the following seeds: $[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]$. Our implementation will be made public upon acceptance. We list the hyperparameters for the experimental pipeline in Table 3.

#### B.2.1 Training

**Architecture and optimization details.** We train EPPO for $500\,000$ steps per task, performing updates to the policy and critic $10$ times every $2048$ step with a batch size of $256$. The learning rate is set to $0.0003$ for both the actor and critic, optimized using Adam (Kingma and Ba, 2015). The actor and critic networks each consist of a 2-layer feedforward neural network with $256$ hidden units. Unlike other baselines, our critic network outputs four values instead of one to predict the evidential priors. We apply Layer Normalization (Ba et al., 2016) and ReLU activations (Nair and Hinton, 2010) for both networks. The policy follows a diagonal normal distribution. Following common practice in the literature, we set the discount factor to $\gamma = 0.99$, the GAE parameter to $\lambda = 0.95$, and the clipping rate to $\epsilon = 0.2$. Gradient norms are clipped at $0.5$, and GAE advantage estimates are normalized within each batch.

Table 3: Common hyperparameters used in the experimental pipeline.

| Policy learning | |
|---|---|
| Seeds | $[1, 2, \ldots, 15]$ |
| Number of steps per task | $500\,000$ |
| Learning rate for actor | $0.0003$ |
| Learning rate for critic | $0.0003$ |
| Horizon | $2048$ |
| Number of epochs | $10$ |
| Minibatch size | $256$ |
| Clip rate $\epsilon$ | $0.2$ |
| GAE parameter $\lambda$ | $0.95$ |
| Hidden dimensions of actor | $[256, 256]$ |
| Hidden dimensions of critic | $[256, 256]$ |
| Activation functions of actor | ReLU |
| Activation functions of critic | ReLU |
| Normalization layers of actor | Layer Norm |
| Normalization layers of critic | Layer Norm |
| Discount factor $\gamma$ | $0.99$ |
| Maximum gradient norm | $0.5$ |
| **Evaluation-related** | |
| Evaluation frequency (steps) | $20\,000$ and end of the tasks |
| Evaluation episodes | $10$ |
| **EPPO-related** | |
| Regularization coefficient ($\xi$) | $0.01$ |
| Hyperprior distribution of $w$ | $\mathcal{N}\left(\omega \mid 0, 100^2\right)$ |
| Hyperprior distribution of $\nu$ | $\mathcal{G}am\left(\nu \mid 5, 1\right)$ |
| Hyperprior distribution of $\alpha$ | $\mathcal{G}am\left(\alpha \mid 5, 1\right) + 1^{\dagger}$ |
| Hyperprior distribution of $\beta$ | $\mathcal{G}am\left(\beta \mid 5, 1\right)$ |
| **Grid Search-related** | |
| Seeds | $[1001, 1002, 1003]$ |
| Radius parameter $\kappa$ for $\texttt{EPPO}_{\text{cor}}$ | $[0.01, 0.1, 0.25]$ |
| Radius parameter $\kappa$ for $\texttt{EPPO}_{\text{ind}}$ | $[0.01, 0.05, 0.1]$ |

$^{\dagger}$The $+1$ ensures a finite mean for $\alpha$.

**Evaluation details.** We evaluate the models at the beginning and final steps of each task, as well as every $20\,000$ steps, using $10$ evaluation episodes. The evaluation environment seeds are set to the training seed plus $100$. For metric calculation, we use the mean return across the evaluation episodes.

**EPPO details.** We set the regularization coefficient ($\xi$) to $0.01$ to scale it down, selecting this value heuristically based on its contribution to the total loss. To prevent overfitting and allow flexibility in learning, we use uninformative, flat priors for the hyperprior distributions. Specifically, we choose a normal distribution $\mathcal{N}(\omega|0, 100^2)$ for $\omega$, though a positively skewed distribution may further improve performance. For $\nu$, $\alpha$, and $\beta$, we use a gamma distribution $\mathcal{G}am(5, 1)$ to ensure positivity. Additionally, we shift the hyperprior distribution of $\alpha$ by $+1$ to ensure a finite mean.

**Grid search details for $\kappa$ of EPPO.** We introduce a confidence radius parameter ($\kappa$) that controls the level of optimism incorporated into exploration. To determine an appropriate value, we perform a grid search over $\kappa \in [0.01, 0.05, 0.1]$ for $\texttt{EPPO}_{\text{ind}}$ and $\kappa \in [0.01, 0.1, 0.25]$ for $\texttt{EPPO}_{\text{cor}}$, selecting these ranges based on their influence on the advantage estimate. We train models using three seeds $(1001, 1002, 1003)$ and exclude them from the main results. After evaluating the AULC metric, we select the optimal $\kappa$ values and use them for EPPO's final evaluation. Table 4 presents $\kappa$ values selected for the training.

Table 4: Radius parameters ($\kappa$) of EPPO.

| Experiment | Environment | Strategy | Confidence radius parameter ($\kappa$) | |
|---|---|---|---|---|
| | | | $\texttt{EPPO}_{\text{cor}}$ | $\texttt{EPPO}_{\text{ind}}$ |
| Slippery | Ant | decreasing | 0.05 | 0.1 |
| | | increasing | 0.1 | 0.25 |
| | HalfCheetah | decreasing | 0.05 | 0.1 |
| | | increasing | 0.1 | 0.1 |
| Paralysis | Ant | back-one | 0.05 | 0.01 |
| | | front-one | 0.1 | 0.1 |
| | | back-two | 0.01 | 0.25 |
| | | front-two | 0.1 | 0.01 |
| | | cross | 0.01 | 0.1 |
| | | parallel | 0.05 | 0.01 |
| | HalfCheetah | back-one | 0.05 | 0.01 |
| | | front-one | 0.1 | 0.1 |
| | | cross-v1 | 0.05 | 0.25 |
| | | cross-v2 | 0.05 | 0.1 |

### B.3 Result visualizations

The learning curves across environment steps are illustrated in Figures 3 to 5. In these figures, the thick (dashed/dotted/dash-dotted/solid) curve represents the mean returns across ten evaluation episodes and 15 random seeds, with the shaded area indicating one standard error from the mean. The legend provides the mean and standard error for the AULC and final return scores, listed in this order. The vertical black dotted lines mark the task changes.

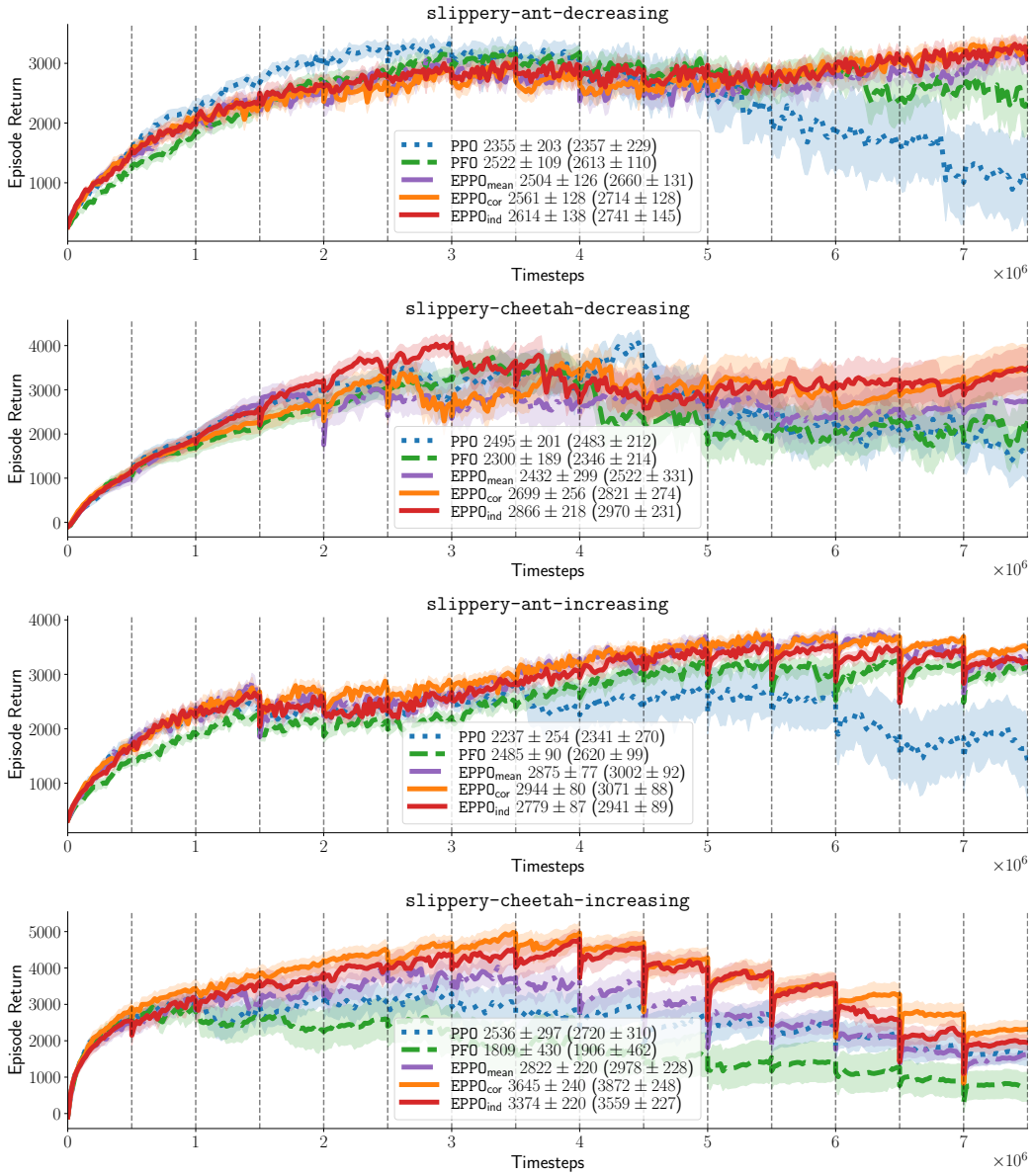


Figure 3: Learning curves for the slippery experiment.

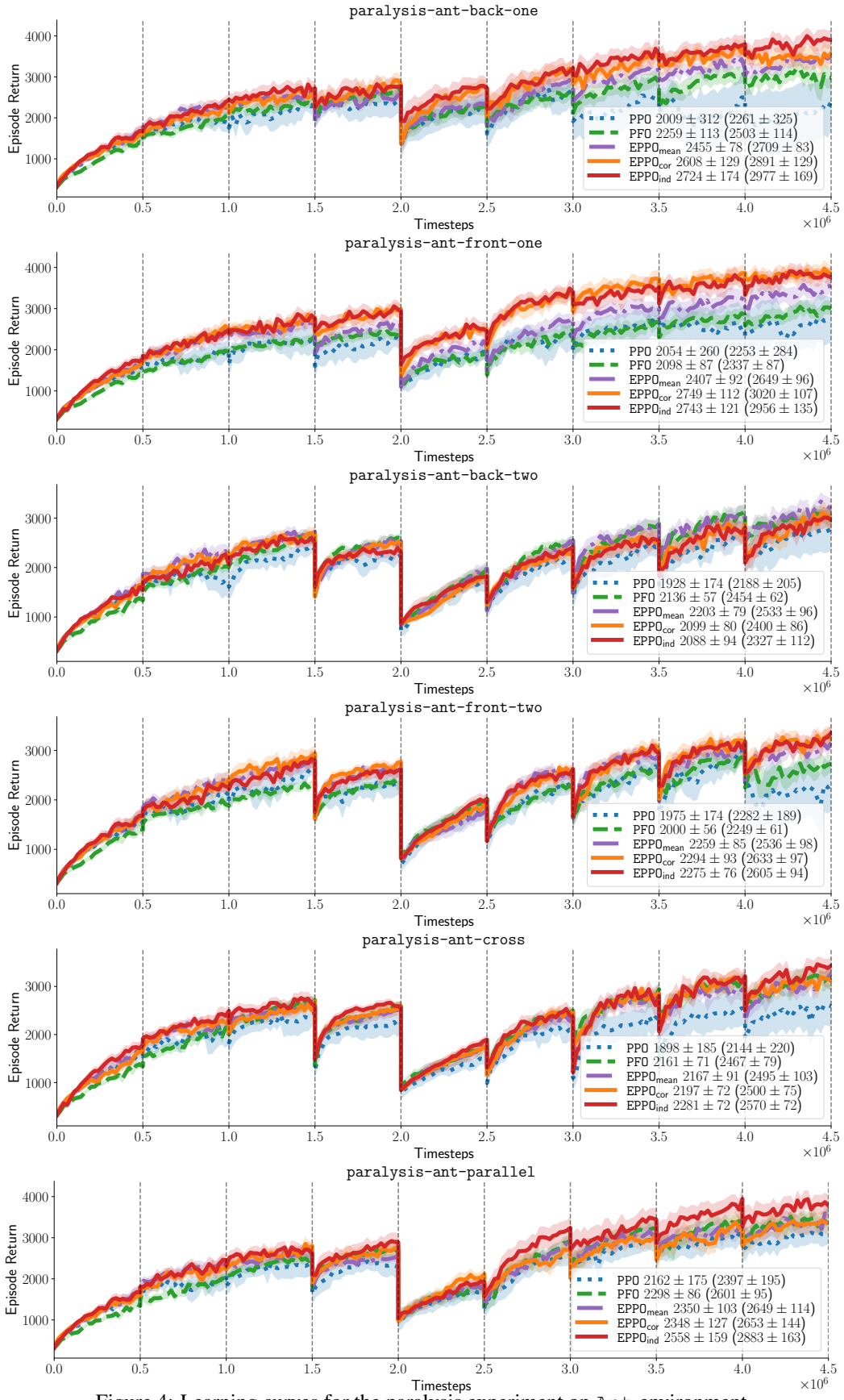


Figure 4: Learning curves for the paralysis experiment on `Ant` environment.

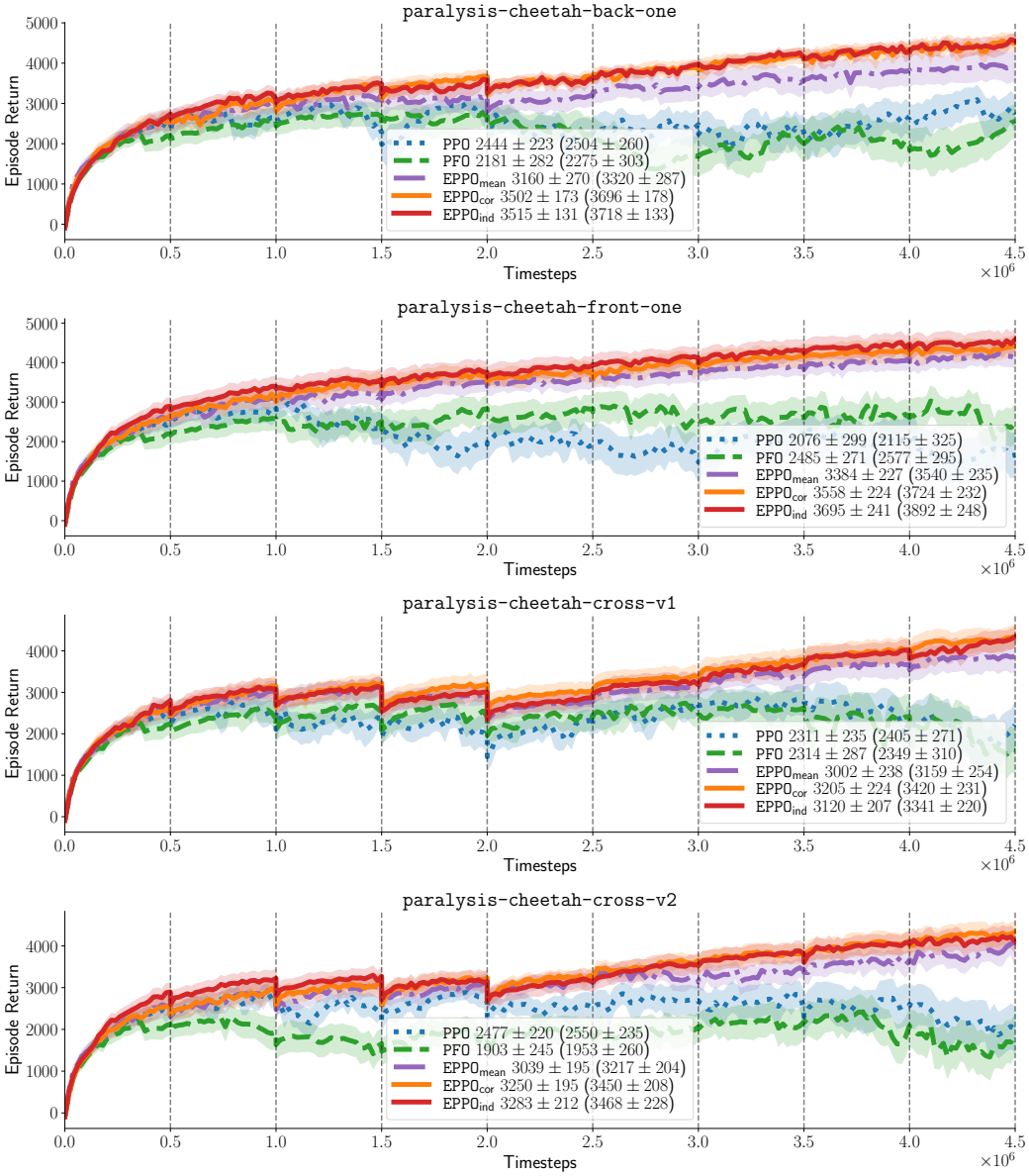


Figure 5: Learning curves for the paralysis experiment on `HalfCheetah` environment.