Overcoming Non-stationary Dynamics with Evidential Proximal Policy Optimization

Anonymous authors
Paper under double-blind review

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 and (ii) directed exploration for rapid adaptation to the changing dynamics. We show that performing on-policy reinforcement learning with an evidential critic provides both. 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 *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 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 & Langford, 2002). This makes PPO an attractive choice for applications ranging from physical robotics (Melo & 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 have 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. In this way, the agent prioritizes underexplored, hence more informative, areas of the state-action space, thereby improving 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, thereby keeping the training process active. Furthermore, the probabilistic value predictor will make it possible to assign uncertainty estimates to the policy training objective, which can in turn be used as an exploration bonus to direct the policy search toward underexplored areas of the state-action space.

Our Hypothesis: *Equipping an agent with a mechanism to quantify the uncertainty of the value function enables it to (i) preserve plasticity and (ii) explore effectively under non-stationary dynamics.*

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 networks, 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.

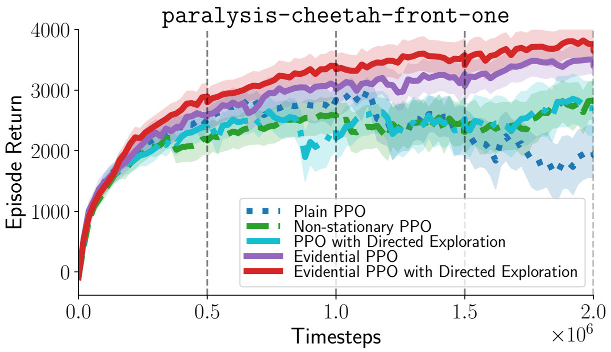

Figure 1: PPO, its non-stationary extension, and PPO equipped with directed exploration all lose their adaptation capability after 1 million steps. In contrast, evidential PPO variants continue to improve, and directed exploration further enhances evidential PPO's performance. See Appendix B.3 for details.

Figure 1 illustrates the learning profiles of on-policy deep actor-critics in a continuous control task with non-stationary dynamics. Plain PPO, its recent extension to non-stationary environments (Moalla et al., 2024), and a state-of-the-art variant equipped with directed exploration (Yang et al., 2024b) all lose their adaptation capability at early stages of training. Conversely, 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 above. 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 state observations to hyperpriors.

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

(iii) Due to the absence of a widely adopted benchmark, we introduce two new experimental 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^π, the value function V^π, and the advantage function A^π:

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

where expectations are taken over trajectories induced by the policy π 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 $G_t \triangleq \sum_{l=0}^{\infty} \gamma^l r_{t+l}$ 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 & 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 & 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 π_θ, 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_ϕ 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_ϕ. The learned V_ϕ is then used to compute advantage estimates, guiding policy updates for more stable and efficient learning.

Modern PPO implementations use *Generalized Advantage Estimation (GAE)* (Schulman et al., 2016), which is a technique for computing advantage estimates. This method helps reduce the variance in 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 the 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:

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

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. Directed exploration encourages agents to seek out novel or informative states. Prior work has explored enhancing the exploration scheme of PPO in both stationary and non-stationary settings (Burda et al., 2019; Wang et al., 2019; Zhang et al., 2022; Steinparz et al., 2022; Yang et al., 2024b). Non-stationary settings, where the environment dynamics shift over time, introduce unique challenges for reinforcement learning that differ from those in stationary environments. Non-stationary RL is similar to but distinct from meta-learning and continual learning. Meta-learning is concerned 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 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) infers 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 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 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 μ and variance σ^2. This distribution is assigned a Normal Inverse-Gamma ($\mathcal{NIG}$) distributed evidential prior:

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

where the hyperparameters ω,ν,α,β are modeled as input-dependent functions, specifically neural networks with weights ϕ. Throughout the paper, we suppress the dependency of the variables on ϕ 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

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

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 in deep reinforcement learning. 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., 2024a). However, we instead use it to model uncertainty in value function estimates, which enables confidence-based exploration and helps 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 enables the model to detect distributional shifts resulting from the non-stationarity of the dynamics and to guide directed exploration, thereby promoting rapid adaptation.

3.1 Evidential value learning

We assume our value function estimates $V(s_t)$ to be normally distributed with unknown mean μ and variance σ^2, which are jointly $\mathcal{NIG}$-distributed. We shorten the notation to $V_t = V(s_t)$ when the relation is clear from context. Although evidential deep learning has demonstrated promising results in epistemic uncertainty estimation, including for unseen out-of-distribution data, naïvely following Amini et al. (2020)'s method often results in training instabilities similar to those reported by Meinert et al. (2023) for standard supervised regression. We extend their non-Bayesian heuristic to a principled, fully Bayesian hierarchical design. We provide a plate diagram of the model in Figure 2. Introducing hyperpriors on each of the four evidential parameters, our model is:

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

where $\mathcal{G}am(\cdot)$ is the Gamma distribution, and $\mu_\omega^0, \ldots, \beta_\beta^0$ are fixed hyperparameters.[1] We adopt a fixed set of hyperpriors to provide relatively flat and uninformative priors for all experiments. See Table 10 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 (μ, σ^2) yields

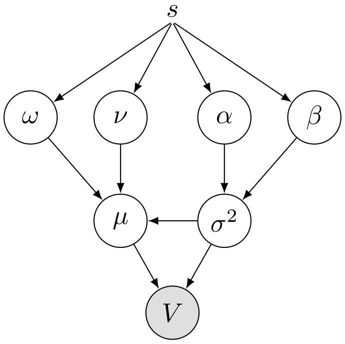

Figure 2: Plate diagram of our evidential value learning model.

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

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.

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

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

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

where we assume $\alpha > 1$.[2] The first equality follows from the law of total variance, which splits the marginal variance into aleatoric and epistemic uncertainty components. Reliance on $\operatorname{var}_{y|\boldsymbol{m}}[y]$ 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.

Distributional reinforcement learning. Evidential value learning belongs to a broader research field that incorporates distributional information into reinforcement learning models, 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 second focuses on accounting for epistemic uncertainty inherent in value function inference, 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 & Rasmussen, 2011; Osband et al., 2019). Evidential value learning differs from standard distributional RL approaches in several key ways. While methods in the first category focus solely on modeling aleatoric uncertainty for risk-sensitive control, evidential value learning is related to the second area of research. It uses an evidential model over the value function to induce a distribution over an advantage function that simultaneously incorporates both aleatoric and epistemic uncertainty. This dual uncertainty quantification enables both regularization benefits and optimistic exploration strategies, distinguishing it from approaches that target only adaptive or risk-sensitive settings through aleatoric uncertainty modeling alone.

3.2 Directed exploration via probabilistic advantages

Evidential value learning provides an uncertainty quantifier for the value function that detects shifts in the data distribution caused by non-stationary state transition dynamics. By parameterizing the value function as a distribution $p(V|\boldsymbol{m})$ rather than as a point estimate, our model naturally increases uncertainty in regions of distributional shift, thereby maintaining gradient flows and preserving plasticity. This uncertainty propagates through the advantage calculation, turning the generalized advantage estimator $\hat{A}_t^{\mathrm{GAE}}$ into a random variable. An Upper Confidence Bound (UCB) exploration strategy emerges naturally from this probabilistic treatment, drawing theoretical justification from the exploration-exploitation trade-off in multi-armed bandit theory (Auer et al., 2002). The UCB estimator

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

provides an optimistic estimator that balances expected advantage with uncertainty, where $\kappa > 0$ controls the confidence radius. This ensures that the policy is directed towards state-action regions where the value function exhibits high uncertainty, enabling curiosity-driven exploration with firm theoretical grounding.

The mean estimate for GAE is

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

and remains 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 how the variance term $\operatorname{var}\big[\hat{A}_t^{\mathrm{GAE}}\big]$ in Equation (3) is computed.

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

(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

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

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

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

We use this variance to construct the UCB in Equation (3) and refer to it 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 construct 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

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

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 κ. We use the variance estimate in Equation (5) to construct the UCB in Equation (3).

Practical implementation of EPPO. We provide pseudocode in Algorithm 1 illustrating how to implement EPPO variants by overlaying color-coded modifications on top of a standard PPO implementation, where each color corresponds to a specific EPPO variant. As shown, EPPO variants require only minimal changes to PPO with a clipped objective and GAE-based advantage calculation. The key additions include an evidential value estimator, its update rule, and a probabilistic advantage computation with UCB. All other components remain identical to those in PPO.

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 focus on non-stationarities that affect environment dynamics in a structured manner, based on identifiable patterns of change, and exclude scenarios where changes occur randomly. This design ensures that observed performance improvements stem from enhanced learning capabilities rather than increased robustness to noise. We run our simulations 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. The implementation of the EPPO variants and the full experimental pipeline is available at `anonymous`[3].

[3] Due to the double-blind review process, the link will be revealed upon acceptance.

Table 1: *Performance evaluation on the slippery environments.* Area Under the Learning Curve (AULC) and Final Return (mean±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 based on the ranking of the mean scores.

Metric	Model	decreasing		increasing		Average	
		Ant	HalfCheetah	Ant	HalfCheetah	Score	Ranking
AULC (↑)	PPO	2355±203	2495±201	2237±254	2536±297	2406	5.0
	PFO	$\underline{2522\pm109}$	2300±189	2485±90	1809±430	2279	4.8
	PPO_{DRND}	2475±139	2633±180	2307±348	2021±246	2359	4.5
	$\text{EPPO}_{\text{mean}}$	$\underline{2504\pm127}$	2432±299	$\underline{2875\pm77}$	2822±219	2658	3.5
	EPPO_{cor}	$\underline{2561\pm128}$	$\underline{2699\pm256}$	**2944±80**	**3645±240**	**2962**	**1.5**
	EPPO_{ind}	**2614±138**	**2866±218**	2779±86	3374±220	2908	1.8
FINAL RETURN (↑)	PPO	2357±230	2483±212	2341±270	2720±310	2475	5.3
	PFO	$\underline{2613\pm110}$	2346±214	2620±99	1906±462	2371	5.0
	PPO_{DRND}	2583±141	2672±191	2428±368	2115±259	2449	4.5
	$\text{EPPO}_{\text{mean}}$	$\underline{2660\pm131}$	2522±331	$\underline{3002\pm92}$	2978±227	2790	3.0
	EPPO_{cor}	$\underline{2714\pm128}$	$\underline{2821\pm274}$	**3071±88**	**3872±248**	**3120**	**1.5**
	EPPO_{ind}	**2741±145**	**2970±231**	2941±89	3559±227	3053	1.8

Baselines. To validate our hypothesis that uncertainty-aware value estimation enhances both plasticity preservation and directed exploration, we benchmark against three representative baselines, summarized in Table 2. These baselines are selected to isolate and test the contribution of each hypothesis component. *(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 to represent the state of the art. In particular, 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. *(iii)* $\textit{PPO}_{\textit{DRND}}$ (Yang et al., 2024b): A PPO variant designed for directed exploration using random network distillation, where the distillation signal acts as a pseudo-count to generate intrinsic rewards that guide exploration. We also evaluate the EPPO variant with $\kappa = 0$, which performs evidential value learning without directed exploration. We refer to this model as $\textit{EPPO}_{\textit{mean}}$. Its relative performance highlights the contribution of directed exploration.

Table 2: *Plasticity and exploration.* Comparison of baselines highlighting the presence of plasticity preservation mechanisms and directed exploration.

Model	Plasticity Mechanism	Directed Exploration
PPO	✗	✗
PFO	✓	✗
PPO_{DRND}	✗	✓
$\text{EPPO}_{\text{mean}}$	✓	✗
EPPO_{cor}	✓	✓
EPPO_{ind}	✓	✓

Experimental setup. We propose two experimental setups to assess the ability of the models to adapt to non-stationarity. In both setups, we encourage fast adaptation by limiting task durations to short intervals. We also preserve agent learnability by introducing changes gradually and avoiding abrupt transitions. The setups are as follows:

(i) **Slippery environments.** Inspired by Dohare et al. (2021; 2024), we construct non-stationary environments by varying the friction coefficient of the floor in locomotion tasks using the Ant and HalfCheetah environments. We induce non-stationarity in them by changing friction every 500 000 steps. To create 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 due to slipping or an inability to move forward. We define 15 tasks by changing the friction with a positive or negative offset of 0.25.

Table 3: *Performance evaluation on the paralysis environments.* Area Under the Learning Curve (AULC) and Final Return (mean±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 based on the ranking of the mean scores.

Metric	Environment	Strategy	Model					
			PPO	PFO	PPO$_{\text{DRND}}$	EPPO$_{\text{mean}}$	EPPO$_{\text{cor}}$	EPPO$_{\text{ind}}$
AULC (↑)	Ant	back-one	2009±312	2259±113	2562±124	2455±78	2608±129	**2724±174**
		front-one	2054±260	2098±87	2547±120	2407±92	**2749±112**	2743±121
		back-two	1928±174	2136±57	**2226±92**	2203±79	2099±80	2088±94
		front-two	1975±174	2000±56	2144±100	2259±85	**2294±93**	2275±76
		parallel	2162±175	2298±86	2358±132	2350±103	2348±127	**2558±159**
		cross	1898±185	2161±71	2012±93	2167±91	2197±72	**2281±72**
	Average AULC on Ant		2004	2159	2308	2307	2383	**2445**
	HalfCheetah	back-one	2444±223	2181±282	2131±294	3160±270	3502±173	**3515±131**
		front-one	2076±299	2485±271	2592±319	3384±227	3558±224	**3695±241**
		cross-v1	2311±235	2314±287	2487±245	3002±238	**3205±224**	3120±207
		cross-v2	2477±220	1903±245	2371±225	3039±195	3250±195	**3283±212**
	Average AULC on HalfCheetah		2327	2221	2395	3146	3379	**3403**
Overall Average AULC Score			2133	2184	2343	2643	2781	**2828**
Overall Average Ranking on AULC			5.6	4.8	3.7	3.1	2.1	**1.7**
Final Return (↑)	Ant	back-one	2261±325	2503±114	2784±147	2709±83	2891±129	**2977±169**
		front-one	2253±284	2337±87	2802±128	2649±96	**3020±107**	2956±135
		back-two	2188±205	2454±62	2512±91	**2533±96**	2400±86	2327±112
		front-two	2282±189	2249±61	2425±115	2536±98	**2633±97**	2605±94
		parallel	2397±195	2601±95	2647±145	2649±114	2653±144	**2883±163**
		cross	2144±220	2467±79	2310±104	2495±103	2500±75	**2570±72**
	Average Final Return on Ant		2254	2435	2580	2595	2683	**2720**
	HalfCheetah	back-one	2504±260	2275±303	2204±317	3320±287	3696±178	**3718±133**
		front-one	2115±325	2577±295	2625±362	3540±235	3724±232	**3892±248**
		cross-v1	2405±271	2349±310	2648±246	3159±254	**3420±231**	3341±220
		cross-v2	2550±235	1953±260	2511±250	3217±204	3450±208	**3468±228**
	Average Final Return on HalfCheetah		2394	2288	2497	3309	3573	**3605**
Overall Average Final Return Score			2310	2376	2547	2881	3039	**3074**
Overall Average Ranking on Final Return			5.5	5.0	3.9	3.0	1.9	**1.7**

(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: $[100, 75, 50, 25, 0, 25, 50, 75, 100]$.

Evaluation metrics. We assess model performance using two metrics: *(i) Area Under the Learning Curve (AULC)* and *(ii) Final Return. AULC* is computed as the average return collected over the entire training trajectory. It captures not only the final performance of the agent but also how quickly and consistently it improves throughout training. In non-stationary environments, where the task dynamics change over time, AULC reflects the agent's ability to continually adapt to new conditions and recover from changes. A higher AULC indicates stronger overall adaptation and learning stability across the full training horizon. *Final Return* is calculated at the completion of each individual task by averaging the returns during the last evaluation steps of that task. These evaluations are averaged over all tasks at the end of training. This metric focuses on how well the agent can adapt and perform on individual tasks after having observed and interacted with them. A higher final return suggests more effective task-specific adaptation. Together, these metrics assess distinct but complementary aspects of adaptation: AULC evaluates an agent's learning efficiency

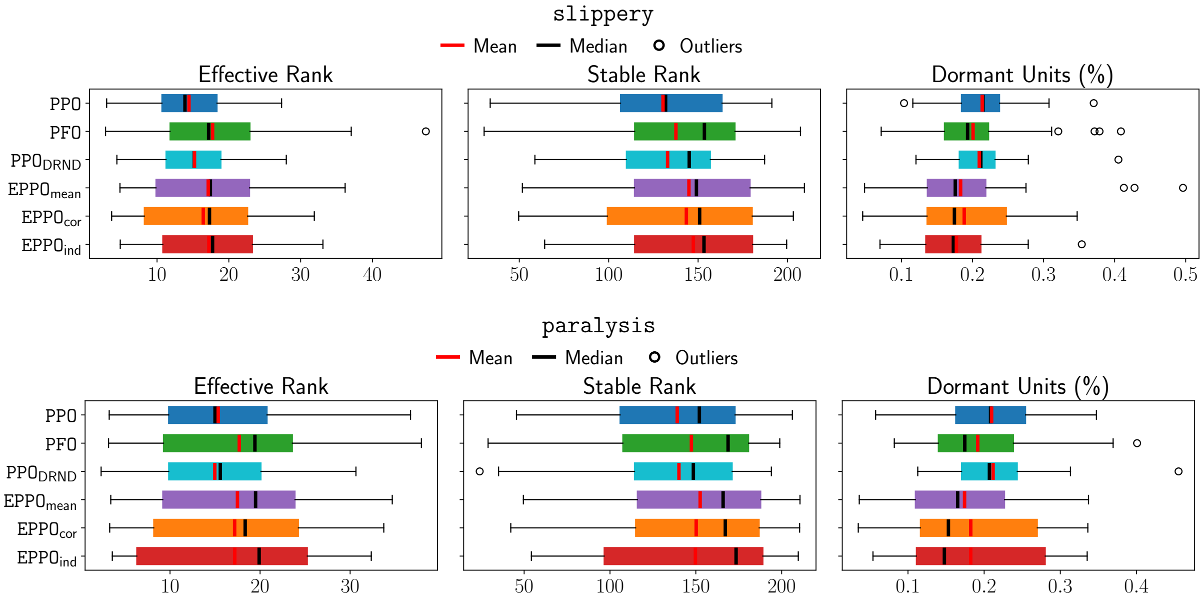

Figure 3: *Plasticity preservation analysis using critic network metrics.* We evaluate three metrics: effective rank, stable rank, and dormant unit percentage, shown from left to right. The top row shows results from the `slippery` environments, and the bottom row shows results from the `paralysis` environments. Each box plot summarizes the distribution of the respective metric across training seeds: the red line indicates the mean, the black line indicates the median, and the individual points represent outliers. These metrics quantify the prediction capacity of the critic networks as learning progresses. EPPO variants consistently preserve plasticity better than PPO variants, as shown by higher ranks and lower dormant unit percentages.

and dynamic adaptability to changing environments over time, while final return assesses the agent's stable performance after task-specific adaptation has occurred.

4.1 Results and Discussion

We present the detailed results of the experiments in Table 1 and Table 3. We also provide experimental 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.

Plasticity preservation analysis. We analyze the plasticity of the agents' critic networks using three metrics: *(i) Effective rank* (Roy & Vetterli, 2007) quantifies the number of significant dimensions in the feature matrix. A high value indicates that most matrix dimensions contribute, suggesting that the network generates diverse features and maintains plasticity. *(ii) Stable rank* (Yang et al., 2020) measures the effective dimensionality of the feature matrix. A low rank suggests limited diversity in the learned representations, implying that the network struggles to preserve plasticity. *(iii) Dormant unit percentage* (Dohare et al., 2024) refers to the proportion of inactive neurons. A high percentage suggests impaired gradient flow and reduced learning capacity, indicating a loss of plasticity. We compute these metrics at the end of each task and report the average over the entire training process. Figure 3 summarizes the results across both experimental setups. As shown, EPPO variants consistently achieve higher effective and stable ranks and exhibit fewer dormant units compared to PPO and its recent variants for handling non-stationarity and directed exploration. Pairwise t-tests on these plasticity metrics confirm that EPPO and its variants significantly outperform plain PPO, with p-values below 0.05 indicating statistical significance. These findings indicate that the evidential value learning framework helps preserve plasticity and that the addition of directed exploration does not compromise it compared to plain PPO. We provide full analysis details in Appendix B.1.3.

Discussion. Our experimental findings are as follows:

(i) Evidential value learning helps preserve plasticity. EPPO variants yield higher effective and stable ranks while creating fewer dormant units, thereby better preserving the plasticity of the critic networks. This capacity to retain plasticity enables EPPO variants to cpmtomie adapting to changing environment dynamics.

(ii) Directed exploration boosts performance. EPPO variants with directed exploration (EPPO_{cor} and EPPO_{ind}) outperform the baselines across all metrics. They also surpass $\text{EPPO}_{\text{mean}}$, which uses the mean value function for policy improvement. These results highlight the unique contribution of directed exploration to performance. Notably, the addition of directed exploration also improves the performance of plain PPO, as demonstrated by PPO_{DRND}.

(iii) Evidential value learning with directed exploration accelerates convergence and improves training stability while preserving plasticity. EPPO variants achieve superior task adaptation compared to the baselines, as demonstrated by the final return scores and learning curves. They also converge more rapidly and improve training stability, as supported by higher AULC scores. By modeling value uncertainty, evidential value learning maintains plasticity throughout training.

(iv) Equipping an agent with a mechanism to quantify the uncertainty of the value function enables it to preserve plasticity and explore effectively in the face of non-stationary dynamics. Our best-performing algorithms, which incorporate uncertainty quantification into the value function, allow agents to maintain plasticity and conduct directed exploration. This facilitates rapid 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 at 74.8 hours for PPO, 75 hours for PFO, 78.1 hours for PPO_{DRND}, 75.3 hours for $\text{EPPO}_{\text{mean}}$, 75.4 hours for EPPO_{cor}, and 75.6 hours for EPPO_{ind}. The total execution time for all experiments reported in this work is approximately 376.1 hours, equivalent to 18.9 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 ξ and the confidence radius κ. 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. We expect that 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 & Szepesvári, 2020) and increasing the Bayesian modeling hierarchy will make EPPO more robust to hyperparameters. As an on-policy policy-gradient algorithm, EPPO shares similar theoretical properties with other PPO variants. The effect of the evidential learning extension on non-asymptotic convergence is a challenging problem; hence, it requires 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 strong 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 those of prior studies. We do not expect extending our results to more tasks to bring any additional insights. We view testing our approach on *physical* robotic systems as 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 co-adaptation is crucial to ensure 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

Zaheer Abbas, Rosie Zhao, Joseph Modayil, Adam White, and Marlos C. Machado. Loss of plasticity in continual deep reinforcement learning. In *Proceedings of The 2nd Conference on Lifelong Learning Agents*, 2023.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Maruan Al-Shedivat, Trapit Bansal, Yuri Burda, Ilya Sutskever, Igor Mordatch, and Pieter Abbeel. Continuous adaptation via meta-learning in nonstationary and competitive environments. In *International Conference on Learning Representations*, 2018.

Alexander Amini, Wilko Schwarting, Ava Soleimany, and Daniela Rus. Deep evidential regression. In *Advances in Neural Information Processing Systems*, 2020.

Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 2002.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.

Marc G Bellemare, Will Dabney, and Mark Rowland. *Distributional Reinforcement Learning*. MIT Press, 2023.

Glen Berseth, Zhiwei Zhang, Grace Zhang, Chelsea Finn, and Sergey Levine. CoMPS: Continual meta policy search. *arXiv preprint arXiv:2112.04467*, 2021.

Omar Besbes, Yonatan Gur, and Assaf Zeevi. Stochastic multi-armed-bandit problem with non-stationary rewards. In *Advances in Neural Information Processing Systems*, 2014.

Zhenshan Bing, David Lerch, Kai Huang, and Alois Knoll. Meta-reinforcement learning in non-stationary and dynamic environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

Christopher Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. Exploration by random network distillation. In *International Conference on Learning Representations*, 2019.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *Advances in Neural Information Processing Systems*, 2017.

Wesley Chung, Lynn Cherif, Doina Precup, and David Meger. Parseval regularization for continual reinforcement learning. In *Advances in Neural Information Processing Systems*, 2024.

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

Shibhansh Dohare, Richard S Sutton, and A Rupam Mahmood. Continual backprop: Stochastic gradient descent with persistent randomness. *arXiv preprint arXiv:2108.06325*, 2021.

Shibhansh Dohare, Qingfeng Lan, and A. Rupam Mahmood. Overcoming policy collapse in deep reinforcement learning. In *Sixteenth European Workshop on Reinforcement Learning*, 2023.

Shibhansh Dohare, J Fernando Hernandez-Garcia, Qingfeng Lan, Parash Rahman, A Rupam Mahmood, and Richard S Sutton. Loss of plasticity in deep continual learning. *Nature*, 2024.

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

Junyu Gao, Mengyuan Chen, Liangyu Xiang, and Changsheng Xu. A comprehensive survey on evidential deep learning and its applications. *arXiv preprint arXiv:2409.04720*, 2024.

Andrew Gelman, John B. Carlin, Hal S. Stern, David B. Dunson, Aki Vehtari, and Donald B. Rubin. *Bayesian Data Analysis.* Chapman and Hall/CRC, 2013.

Mohammad Ghavamzadeh, Shie Mannor, Joelle Pineau, Aviv Tamar, et al. Bayesian reinforcement learning: A survey. *Foundations and Trends® in Machine Learning*, 2015.

Sham Kakade and John Langford. Approximately optimal approximate reinforcement learning. In *International Conference on Machine Learning*, 2002.

Melih Kandemir, Abdullah Akgül, Manuel Haussmann, and Gozde Unal. Evidential Turing processes. In *International Conference on Learning Representations*, 2022.

Christos Kaplanis, Murray Shanahan, and Claudia Clopath. Policy consolidation for continual reinforcement learning. In *International Conference on Machine Learning*, 2019.

Emilie Kaufmann, Olivier Cappe, and Aurelien Garivier. On Bayesian upper confidence bounds for bandit problems. In *International Conference on Artificial Intelligence and Statistics*, 2012.

Ramtin Keramati, Christoph Dann, Alex Tamkin, and Emma Brunskill. Being optimistic to be conservative: Quickly learning a CVaR policy. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020.

Khimya Khetarpal, Matthew Riemer, Irina Rish, and Doina Precup. Towards continual reinforcement learning: A review and perspectives. *Journal of Artificial Intelligence Research*, 2020.

Diederik P Kingma and Jimmy Lei Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 2017.

Saurabh Kumar, Henrik Marklund, and Benjamin Van Roy. Maintaining plasticity in continual learning via regenerative regularization. In *Proceedings of The 3rd Conference on Lifelong Learning Agents*, 2025.

Tor Lattimore and Csaba Szepesvári. *Bandit Algorithms.* Cambridge University Press, 2020.

Hojoon Lee, Hyeonseo Cho, Hyunseung Kim, Donghu Kim, Dugki Min, Jaegul Choo, and Clare Lyle. Slow and steady wins the race: Maintaining plasticity with Hare and Tortoise networks. In *International Conference on Machine Learning*, 2024.

Lihong Li, Wei Chu, John Langford, and Robert E. Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th International Conference on World Wide Web*, 2010.

Guilherme Cano Lopes, Murillo Ferreira, Alexandre da Silva Simões, and Esther Luna Colombini. 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)*, 2018.

Carlos E. Luis, Alessandro G. Bottero, Julia Vinogradska, Felix Berkenkamp, and Jan Peters. Value-distributional model-based reinforcement learning. *Journal of Machine Learning Research*, 2024.

Clare Lyle, Mark Rowland, and Will Dabney. Understanding and preventing capacity loss in reinforcement learning. In *International Conference on Learning Representations*, 2022.

Clare Lyle, Zeyu Zheng, Evgenii Nikishin, Bernardo Avila Pires, Razvan Pascanu, and Will Dabney. Understanding plasticity in neural networks. In *International Conference on Machine Learning*, 2023.

Clare Lyle, Zeyu Zheng, Khimya Khetarpal, Hado van Hasselt, Razvan Pascanu, James Martens, and Will Dabney. Disentangling the causes of plasticity loss in neural networks. In *Proceedings of The 3rd Conference on Lifelong Learning Agents*, 2025.

Nis Meinert, Jakob Gawlikowski, and Alexander Lavin. The unreasonable effectiveness of deep evidential regression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2023.

Luckeciano Carvalho Melo and Marcos Ricardo Omena Albuquerque Máximo. 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)*, 2019.

Skander Moalla, Andrea Miele, Daniil Pyatko, Razvan Pascanu, and Caglar Gulcehre. No representation, no trust: Connecting representation, collapse, and trust issues in PPO. In *Advances in Neural Information Processing Systems*, 2024.

Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted Boltzmann machines. In *International Conference on Machine Learning*, 2010.

Evgenii Nikishin, Max Schwarzer, Pierluca D'Oro, Pierre-Luc Bacon, and Aaron Courville. The primacy bias in deep reinforcement learning. In *International Conference on Machine Learning*, 2022.

Ian Osband, Benjamin Van Roy, Daniel J Russo, and Zheng Wen. Deep exploration via randomized value functions. *Journal of Machine Learning Research*, 2019.

Olivier Roy and Martin Vetterli. The effective rank: A measure of effective dimensionality. In *2007 15th European signal processing conference*. IEEE, 2007.

Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International Conference on Machine Learning*, 2015.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *International Conference on Learning Representations*, 2016.

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

Murat Sensoy, Lance Kaplan, and Melih Kandemir. Evidential deep learning to quantify classification uncertainty. In *Advances in Neural Information Processing Systems*, 2018.

Niranjan Srinivas, Andreas Krause, Sham Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. In *International Conference on Machine Learning*, 2010.

Christian Alexander Steinparz, Thomas Schmied, Fabian Paischer, Marius-constantin Dinu, Vihang Prakash Patil, Angela Bitto-nemling, Hamid Eghbal-zadeh, and Sepp Hochreiter. Reactive exploration to cope with non-stationarity in lifelong reinforcement learning. In *Proceedings of The 1st Conference on Lifelong Learning Agents*, 2022.

Richard S Sutton, Anna Koop, and David Silver. On the role of tracking in stationary environments. In *International Conference on Machine Learning*, 2007.

Sebastian Thrun. Lifelong learning algorithms. In *Learning to learn*. Springer, 1998.

Emanuel Todorov, Tom Erez, and Yuval Tassa. MuJoCo: A physics engine for model-based control. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

René Traoré, Hugo Caselles-Dupré, Timothée Lesort, Te Sun, Guanghang Cai, Natalia Díaz-Rodríguez, and David Filliat. DisCoRL: Continual reinforcement learning via policy distillation. *arXiv preprint arXiv:1907.05855*, 2019.

Dingrong Wang, Deep Shankar Pandey, Krishna Prasad Neupane, Zhiwei Yu, Ervine Zheng, Zhi Zheng, and Qi Yu. Deep temporal sets with evidential reinforced attentions for unique behavioral pattern discovery. In *International Conference on Machine Learning*, 2023.

Dingrong Wang, Krishna Prasad Neupane, Ervine Zheng, and Qi Yu. Evidential conservative Q-learning for dynamic recommendations, 2024.

Yuhui Wang, Hao He, Xiaoyang Tan, and Yaozhong Gan. Trust region-guided proximal policy optimization. In *Advances in Neural Information Processing Systems*, 2019.

Hongzheng Yang, Cheng Chen, Yueyao Chen, Markus Scheppach, Hon Chi Yip, and Qi Dou. Uncertainty estimation for safety-critical scene segmentation via fine-grained reward maximization. In *Advances in Neural Information Processing Systems*, 2024a.

Kai Yang, Jian Tao, Jiafei Lyu, and Xiu Li. Exploration and anti-exploration with distributional random network distillation. In *International Conference on Machine Learning*, 2024b.

Yuzhe Yang, Guo Zhang, Zhi Xu, and Dina Katabi. Harnessing structures for value-based planning and reinforcement learning. In *International Conference on Learning Representations*, 2020.

Junwei Zhang, Zhenghao Zhang, Shuai Han, and Shuai Lü. Proximal policy optimization via enhanced exploration efficiency. *Information Sciences*, 2022.

Peng Zhao, Lijun Zhang, Yuan Jiang, and Zhi-Hua Zhou. A simple approach for non-stationary linear bandits. In *International Conference on Machine Learning*, 2020.

Xujiang Zhao, Shu Hu, Jin-Hee Cho, and Feng Chen. Uncertainty-based decision making using deep reinforcement learning. In *2019 22th International Conference on Information Fusion (FUSION)*, 2019.

Rui Zheng, Shihan Dou, Songyang Gao, Yuan Hua, Wei Shen, Binghai Wang, Yan Liu, Senjie Jin, Qin Liu, Yuhao Zhou, et al. Secrets of RLHF in large language models part i: PPO. *arXiv preprint arXiv:2307.04964*, 2023.

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 $\nu, \alpha, \beta > 0$. The mean, mode, and variance are given by

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

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

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

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

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

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

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

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

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

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

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

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

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

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 becomes infeasible or yields 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[4] [5] to adjust the floor friction coefficients.

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.1.3 Plasticity preservation analysis

We calculate the metrics as follows:

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

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

- *Effective rank* is calculated using the feature matrix $\Phi \in \mathbb{R}^{n\times m}$ from the penultimate layer, with singular values σ_k for $k = 1, 2, \ldots, q$, where $q = \max(n, m)$. We define $p_k = \dfrac{\sigma_k}{||\boldsymbol{\sigma}||_1}$, where $||\boldsymbol{\sigma}||_1 = \sum_k |\sigma_k|$. The effective rank of Φ is given by:

$$\text{effective rank}(\Phi) \triangleq \exp H(p_1, p_2, \ldots, p_q),$$

where the entropy $H(p_1, p_2, \ldots, p_q) = -\sum_k p_k \log p_k$.

- *Stable rank* is also computed using the feature matrix Φ from the penultimate layer, with:

$$\text{stable rank}(\Phi) \triangleq \min_k \left\{ \frac{\sum_i^k \sigma_i^2}{\sum_j^q \sigma_j^2} > 1 - \delta \right\},$$

where δ is a threshold. We set $\delta = 0.01$, meaning that the selected rank captures at least 99% of the total variance.

- *Dormant unit percentage* measures the portion of neurons that remain consistently inactive across a batch of inputs. We compute activations immediately after applying the nonlinearity and consider a neuron dormant if its output remains below a small threshold (0.01) for all samples in the batch.

We measure these metrics at the end of each task in the evaluation environment and report their averages over the entire training process. The results for each environment are presented in Figures 5 to 7.

Significance test. We conduct pairwise significance tests between all methods using one-sided paired t-tests. For each experimental setup, we aggregate results across seeds and tasks and apply the t-test to assess significant improvements between these pairs. Figure 4 presents the p-value matrices. Only the lower triangular part of each matrix is shown, where each entry corresponds to the p-value from comparing the row and column methods. The results show that our EPPO variants preserve plasticity significantly better than plain PPO at a 0.05 significance level. Note that each pairwise test is evaluated and reported independently, i.e., no Bonferroni correction was applied.

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 10.

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 & 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 & 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.

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.

Table 4: p-values for effective rank in `slippery`.

	PPO	PFO	PPO$_{\texttt{DRND}}$	EPPO$_{\texttt{mean}}$	EPPO$_{\texttt{cor}}$	EPPO$_{\texttt{ind}}$
PPO						
PFO	0.020					
PPO$_{\texttt{DRND}}$	0.127	0.994				
EPPO$_{\texttt{mean}}$	0.000	0.690	0.019			
EPPO$_{\texttt{cor}}$	0.010	0.877	0.070	0.825		
EPPO$_{\texttt{ind}}$	0.001	0.683	0.011	0.457	0.135	

Table 5: p-values for effective rank in `paralysis`.

	PPO	PFO	PPO$_{\texttt{DRND}}$	EPPO$_{\texttt{mean}}$	EPPO$_{\texttt{cor}}$	EPPO$_{\texttt{ind}}$
PPO						
PFO	0.000					
PPO$_{\texttt{DRND}}$	0.747	1.000				
EPPO$_{\texttt{mean}}$	0.000	0.621	0.000			
EPPO$_{\texttt{cor}}$	0.001	0.809	0.000	0.751		
EPPO$_{\texttt{ind}}$	0.003	0.789	0.000	0.736	0.486	

Table 6: p-values for stable rank in `slippery`.

	PPO	PFO	PPO$_{\texttt{DRND}}$	EPPO$_{\texttt{mean}}$	EPPO$_{\texttt{cor}}$	EPPO$_{\texttt{ind}}$
PPO						
PFO	0.041					
PPO$_{\texttt{DRND}}$	0.211	0.884				
EPPO$_{\texttt{mean}}$	0.000	0.044	0.002			
EPPO$_{\texttt{cor}}$	0.000	0.045	0.002	0.668		
EPPO$_{\texttt{ind}}$	0.000	0.004	0.000	0.248	0.051	

Table 7: p-values for stable rank in `paralysis`.

	PPO	PFO	PPO$_{\texttt{DRND}}$	EPPO$_{\texttt{mean}}$	EPPO$_{\texttt{cor}}$	EPPO$_{\texttt{ind}}$
PPO						
PFO	0.000					
PPO$_{\texttt{DRND}}$	0.323	0.999				
EPPO$_{\texttt{mean}}$	0.000	0.016	0.000			
EPPO$_{\texttt{cor}}$	0.000	0.081	0.000	0.909		
EPPO$_{\texttt{ind}}$	0.000	0.124	0.000	0.924	0.606	

Table 8: p-values for dormant unit percentage in `slippery`.

	PPO	PFO	PPO$_{\texttt{DRND}}$	EPPO$_{\texttt{mean}}$	EPPO$_{\texttt{cor}}$	EPPO$_{\texttt{ind}}$
PPO						
PFO	0.097					
PPO$_{\texttt{DRND}}$	0.278	0.847				
EPPO$_{\texttt{mean}}$	0.003	0.071	0.010			
EPPO$_{\texttt{cor}}$	0.003	0.089	0.010	0.698		
EPPO$_{\texttt{ind}}$	0.000	0.006	0.000	0.286	0.044	

Table 9: p-values for dormant unit percentage in `paralysis`.

	PPO	PFO	PPO$_{\texttt{DRND}}$	EPPO$_{\texttt{mean}}$	EPPO$_{\texttt{cor}}$	EPPO$_{\texttt{ind}}$
PPO						
PFO	0.000					
PPO$_{\texttt{DRND}}$	0.663	1.000				
EPPO$_{\texttt{mean}}$	0.000	0.001	0.000			
EPPO$_{\texttt{cor}}$	0.000	0.043	0.000	0.962		
EPPO$_{\texttt{ind}}$	0.000	0.044	0.000	0.951	0.494	

Figure 4: Pairwise significance test results evaluating whether one method performed significantly better than another. Each lower-triangular entry shows the p-value of a one-tailed paired t-test testing whether the method in the row significantly outperformed the method in the column. Smaller p-values indicate stronger evidence in favor of the row method's superior performance.

Table 10: 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 ϵ	0.2
GAE parameter λ	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
Optimizer for actor	Adam
Optimizer for critic	Adam
Discount factor γ	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 (ξ)	0.01
Hyperprior distribution of w	$\mathcal{N}\left(\omega \vert 0, 100^2\right)$
Hyperprior distribution of ν	$\mathcal{G}am\left(\nu \vert 5, 1\right)$
Hyperprior distribution of α	$\mathcal{G}am\left(\alpha \vert 5, 1\right) + 1^{\dagger}$
Hyperprior distribution of β	$\mathcal{G}am\left(\beta \vert 5, 1\right)$
Grid Search-related	
Seeds	$[1001, 1002, 1003]$
Radius parameter κ for $\texttt{EPPO}_{\text{cor}}$	$[0.01, 0.1, 0.25]$
Radius parameter κ for $\texttt{EPPO}_{\text{ind}}$	$[0.01, 0.05, 0.1]$
PPO$_{\text{DRND}}$-related	
Hidden dimensions of bonus ensemble	$[256, 256, 256, 32]$
Activation functions of bonus ensemble	ReLU
Normalization layers of bonus ensemble	None
Learning rate for bonus ensemble	0.0003
Optimizer for bonus ensemble	Adam
Number of ensemble elements	10
Bonus scaling factor	0.9

†The $+1$ ensures a finite mean for α.

Table 11: Radius parameters (κ) of EPPO.

Experiment	Environment	Strategy	Confidence radius parameter (κ)	
			EPPO$_{\text{cor}}$	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

EPPO details. We set the regularization coefficient (ξ) 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 ω, though a positively skewed distribution may further improve performance. For ν, α, and β, we use a gamma distribution $\mathcal{G}am(5, 1)$ to ensure positivity. Additionally, we shift the hyperprior distribution of α by $+1$ to ensure a finite mean.

PPO$_{\text{DRND}}$ details. We use a bonus ensemble with 10 neural networks, each composed of a 4-layer feedforward architecture with hidden dimensions $[256, 256, 256, 32]$. Each network takes a concatenated state-action pair as input. We apply ReLU activations after each layer and do not include normalization layers. We optimize the ensemble using Adam (Kingma & Ba, 2015) with a learning rate of 0.0003. We scale the output of the ensemble by a bonus factor of 0.9 to guide exploration during training. We adopt the architecture and hyperparameters from the original implementation by Yang et al. (2024b).

Grid search details for κ of EPPO. We introduce a confidence radius parameter (κ) 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 EPPO$_{\text{ind}}$ and $\kappa \in [0.01, 0.1, 0.25]$ for 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 κ values and use them for EPPO's final evaluation. Table 11 presents the κ values selected for training.

Practical implementation of EPPO. We provide pseudocode in Algorithm 1 illustrating how to implement EPPO variants by overlaying color-coded modifications on top of a standard PPO implementation, where each color corresponds to a specific EPPO variant. All lines are shared across EPPO variants and PPO unless otherwise indicated in the comment section.

B.3 Result visualizations

The learning curves across environment steps are illustrated in Figures 8 to 10. In these figures, the thick curve (dashed, dotted, dash-dotted, or solid) 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

Algorithm 1 Evidential Proximal Policy Optimization variants ($\texttt{EPPO}_{\texttt{mean}}$, $\texttt{EPPO}_{\texttt{cor}}$, $\texttt{EPPO}_{\texttt{ind}}$) over $\texttt{PPO}$

```
Input: Initial policy parameters $\theta$, value function parameters $\phi$, clipping threshold $\epsilon$, minibatch size $M$, number of update epochs $K$, trajectory horizon $T$, discount factor $\gamma$, GAE parameter $\lambda$, learning rates $\lambda_\pi$, $\lambda_V$, radius $\kappa$ for $\texttt{EPPO}_{\texttt{cor}}$ and $\texttt{EPPO}_{\texttt{ind}}$, regularization coefficient $\xi$
for each epoch do
    Roll out policy in the environment and fill the buffer $D$ with $(s_t, a_t, r_t, s_{t+1}, d_t)$
    $V_t \leftarrow V_\phi(s_t)$ and $V_{t+1} \leftarrow V_\phi(s_{t+1})$    ▷ Value estimates for $\texttt{PPO}$
    $\omega_t, \nu_t, \alpha_t, \beta_t \leftarrow V_\phi(s_t)$ and $\omega_{t+1}, \nu_{t+1}, \alpha_{t+1}, \beta_{t+1} \leftarrow V_\phi(s_{t+1})$    ▷ For $\texttt{EPPO}$s
    $V_t \leftarrow \omega_t, \quad V_{t+1} \leftarrow \omega_{t+1}$    ▷ Mean value estimates of $\texttt{EPPO}$s
    $\text{var}\left[V_t\right] \leftarrow \frac{\beta_t}{\alpha_t - 1}\left(1 + \frac{1}{\nu_t}\right), \quad \text{var}\left[V_{t+1}\right] \leftarrow \frac{\beta_{t+1}}{\alpha_{t+1} - 1}\left(1 + \frac{1}{\nu_{t+1}}\right)$    ▷ Variance value estimates of $\texttt{EPPO}$s
    Compute deltas: $\delta_t \leftarrow r_t + \gamma V_{t+1}(1 - d_t) - V_t$
    Initialize accumulator $A \leftarrow 0$ and $\hat{A}$ as empty list of size $T$ for mean
    Initialize accumulator $\text{var}\left[A\right] \leftarrow 0$ and $\text{var}\left[\hat{A}\right]$ as empty list of size $T$ for variance ▷ $\texttt{EPPO}_{\texttt{cor}}$, $\texttt{EPPO}_{\texttt{ind}}$
    for $t = T - 1$ to $0$ by $-1$ do
        $A \leftarrow \delta_t + \gamma\lambda A(1 - d_t)$
        $\hat{A}_t \leftarrow A$
        $\text{var}\left[A\right] \leftarrow (\gamma\lambda)^2 \left(\text{var}\left[V_{t+1}\right] + (1 - d_t)\text{var}\left[A\right]\right)$    ▷ $\texttt{EPPO}_{\texttt{cor}}$, $\texttt{EPPO}_{\texttt{ind}}$
        $\text{var}\left[\hat{A}_t\right] \leftarrow \text{var}\left[V_t\right] + \left(\frac{1 - \lambda}{\lambda}\right)^2 \text{var}\left[A\right]$    ▷ Equation (4) $\texttt{EPPO}_{\texttt{cor}}$
        $\text{var}\left[\hat{A}_t\right] \leftarrow \frac{1 - \lambda}{1 + \lambda}\text{var}\left[V_t\right] + \left(\frac{1 - \lambda}{\lambda}\right)^2 \text{var}\left[A\right]$    ▷ Equation (5) $\texttt{EPPO}_{\texttt{ind}}$
    end for
    $\hat{A} \leftarrow \hat{A} + \kappa\sqrt{\text{var}\left[\hat{A}\right]}$ for all samples    ▷ UCB with Equation (3) $\texttt{EPPO}_{\texttt{cor}}$, $\texttt{EPPO}_{\texttt{ind}}$
    Compute returns: $\hat{R}_t \leftarrow \hat{A}_t + V_t$ and normalize advantages
    for $k = 1$ to $K$ do
        Shuffle $D$ and split into minibatches of size $M$
        for each minibatch do
            Compute importance ratio: $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}$
            Compute clipped objective: $\mathcal{L}_{\text{clip}}(\theta) = \frac{1}{M}\sum \min\left(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1 - \epsilon, 1 + \epsilon)\hat{A}_t\right)$
            Update policy with gradient clipping: $\theta \leftarrow \theta + \lambda_\pi \nabla_\theta \mathcal{L}_{\text{clip}}(\theta)$
            Compute value loss: $\mathcal{L}_{VF}(\phi) = \frac{1}{T}\sum\left(V_\phi(s_t)\hat{R}_t\right)^2$    ▷ $\texttt{PPO}$
            Estimate $\omega_t, \nu_t, \alpha_t, \beta_t \leftarrow V_\phi(s_t)$    ▷ $\texttt{EPPO}$s
            Compute model-fit loss: where $\Omega_t = 2\beta_t(1 + \nu_t)$    ▷ Equation (2) $\texttt{EPPO}$s
                $\mathcal{L}_{\text{NLL}}(\phi) = \frac{1}{T}\sum \frac{1}{2}\log\left(\frac{\pi}{\nu_t}\right) - \alpha_t \log\left(\Omega_t\right) + \left(\alpha_t + \frac{1}{2}\right)\log\left(\left(\hat{R}_t - \omega_t\right)^2 \nu_t + \Omega_t\right) + \log\left(\frac{\Gamma\left(\alpha_t\right)}{\Gamma\left(\alpha_t + \frac{1}{2}\right)}\right)$
            Compute regularization: $\mathcal{L}_{reg}(\phi) = \frac{1}{T}\sum \log p(\omega_t) + \log p(\nu_t) + \log p(\alpha_t) + \log p(\nu_t)$    ▷ $\texttt{EPPO}$s
            Compute value loss: $\mathcal{L}_{VF}(\phi) = \mathcal{L}_{\text{NLL}}(\phi) - \xi\mathcal{L}_{reg}(\phi)$    ▷ $\texttt{EPPO}$s
            Update value function with gradient clipping: $\phi \leftarrow \phi - \lambda_V \nabla_\phi \mathcal{L}_{VF}(\phi)$
        end for
    end for
end for
```

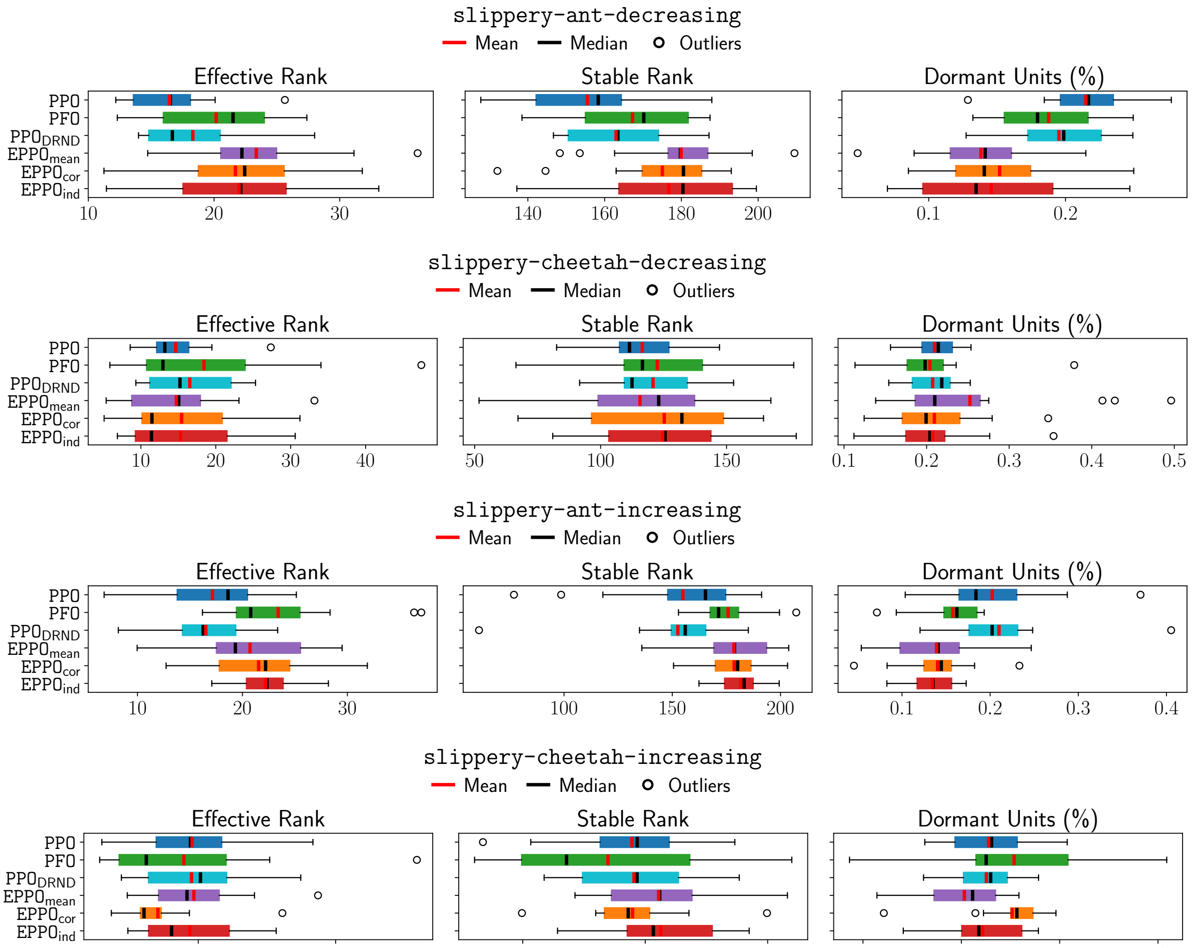

Figure 5: Plasticity preservation analysis for the slippery experiment.

mean and standard error for the AULC and final return scores, listed in this order. The vertical black dotted lines mark the task changes.

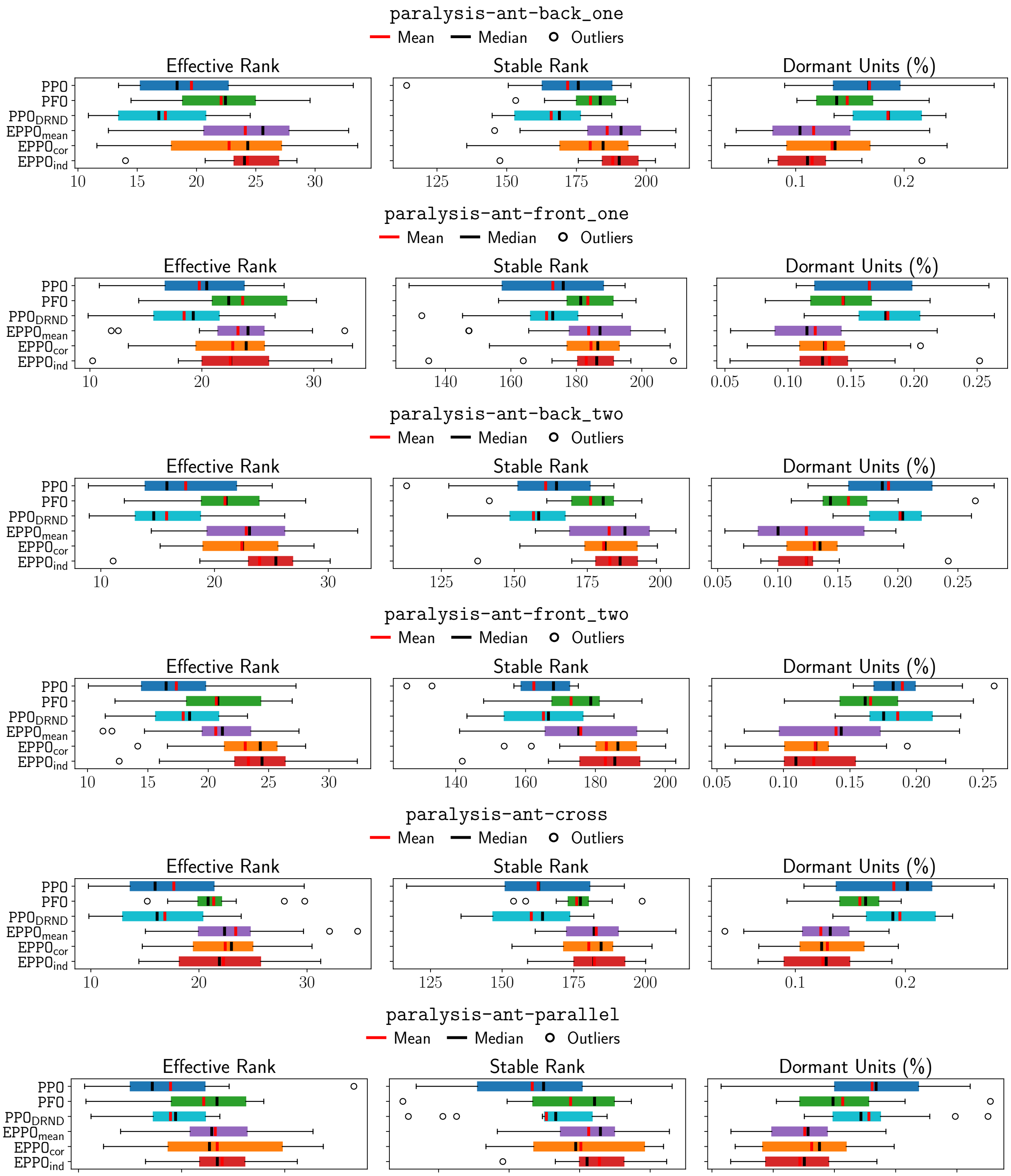

Figure 6: Plasticity preservation analysis for the paralysis experiment on `Ant` environment.

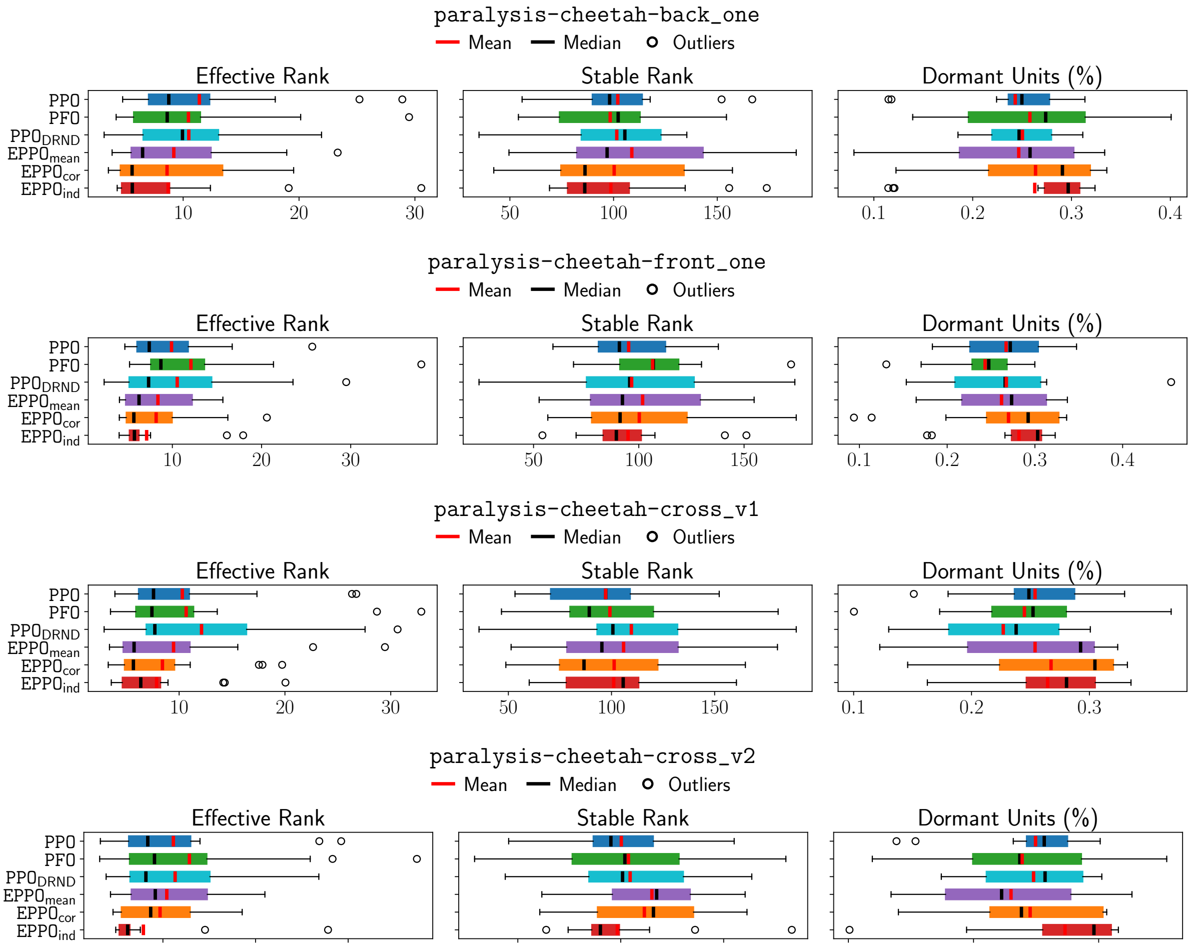

Figure 7: Plasticity preservation analysis for the paralysis experiment on `HalfCheetah` environment.

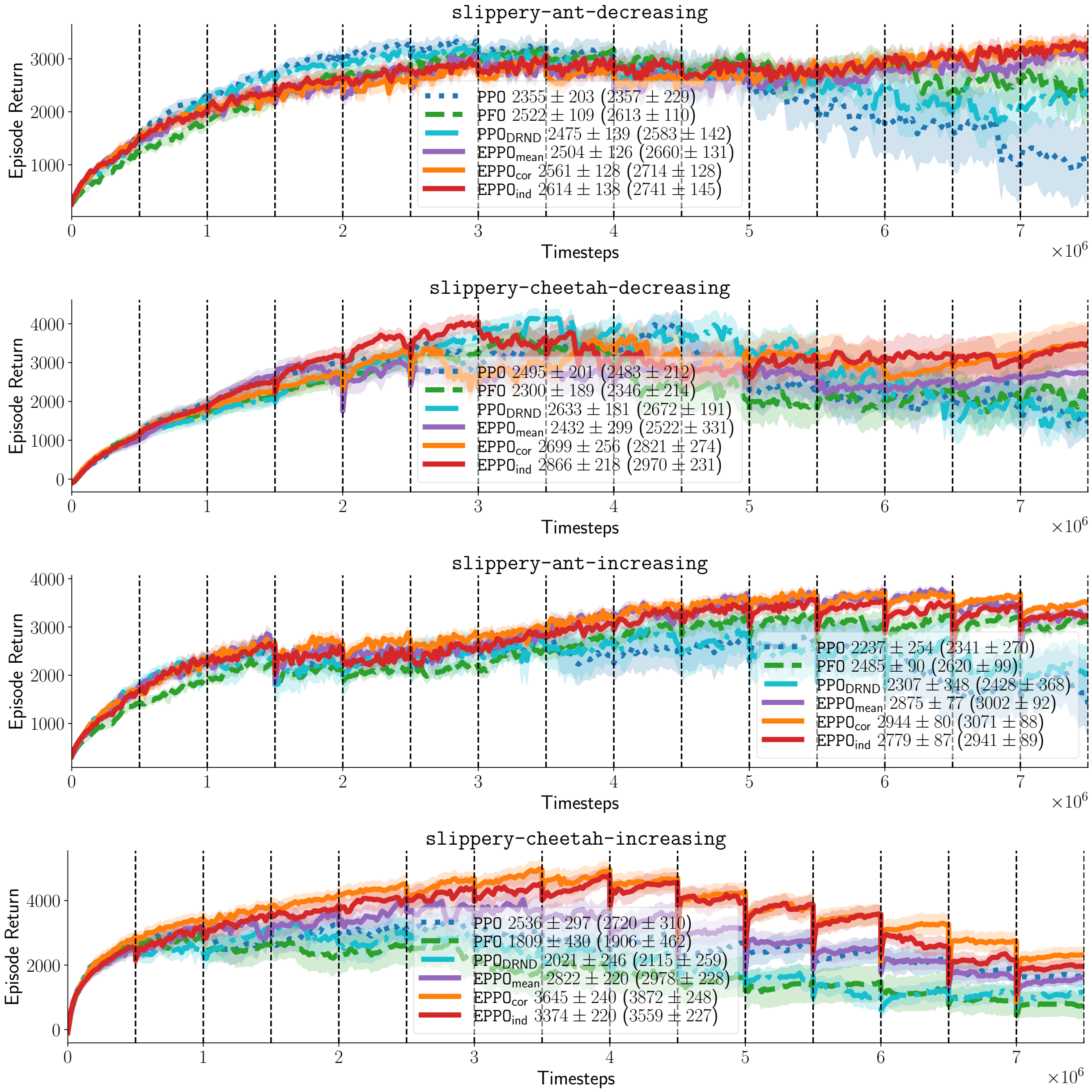

Figure 8: Learning curves for the slippery experiment.

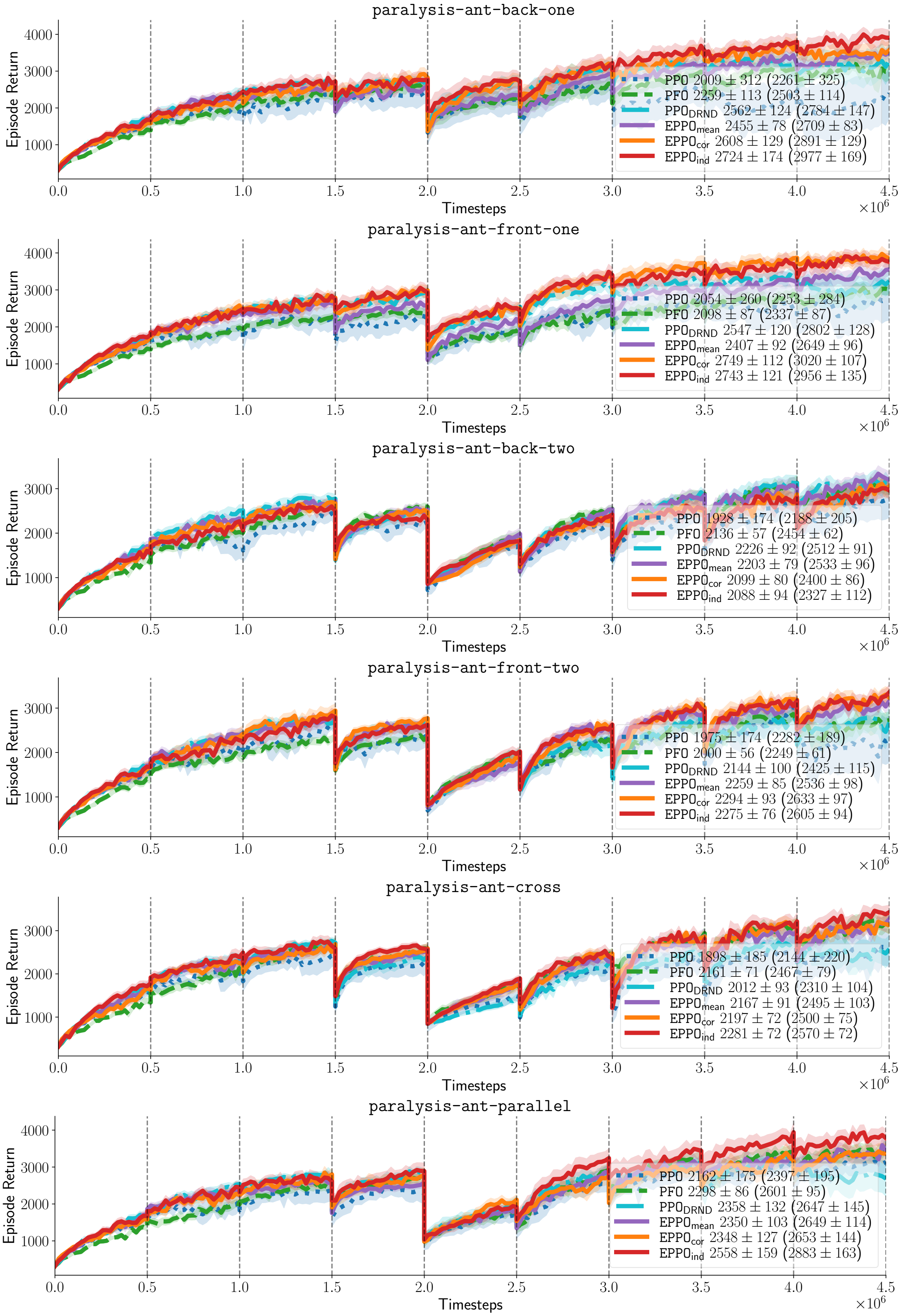

Figure 9: Learning curves for the paralysis experiment on Ant environment.

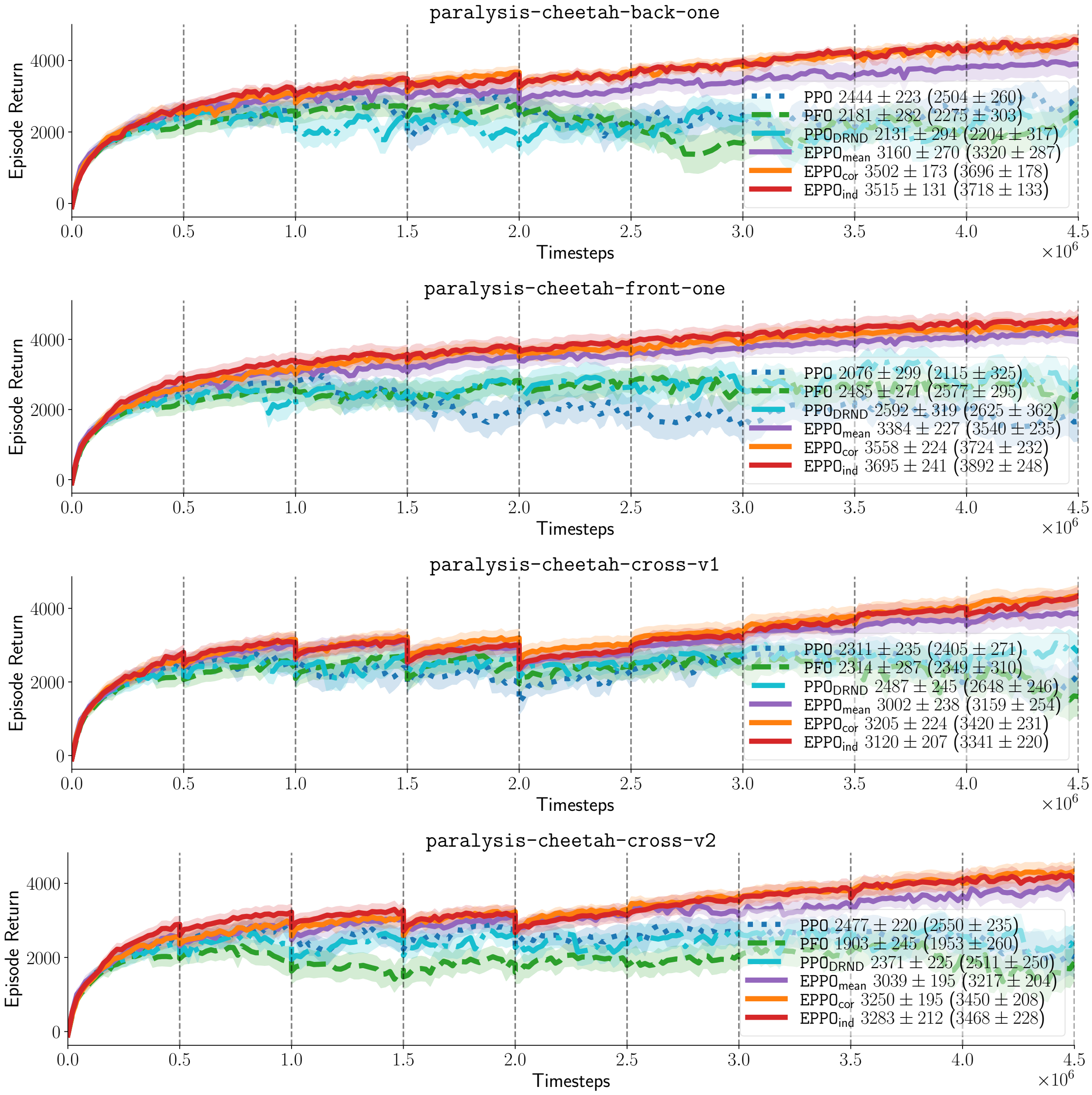

Figure 10: Learning curves for the paralysis experiment on `HalfCheetah` environment.