أكاديمية كاوست
KAUST ACADEMY

جامعة الملك عبدالله للعلوم والتقنية
King Abdullah University of Science and Technology

Reliability-Weighted Advantage Estimation for Proximal Policy Optimization

Manahel Altalhan, Raghad Khalid Alshawish, Noura Mushabab Al Mezher, Juri Alshahrani, Abdallah Hammad

445816339@kku.edu.sa, Raghadkh.cs@gmail.com, NouraAlMezher@gmail.com, juri.alshahrani@kaust.edu.sa, abdallah1136@hotmail.com

Abstract

Proximal Policy Optimization (PPO) relies on advantage estimates to guide policy updates, while Generalized Advantage Estimation (GAE) controls the bias–variance trade-off through its λ parameter. However, conventional PPO does not explicitly account for the reliability of individual advantage estimates. In this work, we propose a TD-error-based reliability weighting mechanism that adaptively adjusts the contribution of GAE advantages according to their associated temporal-difference errors. The proposed weighting is applied to the policy learning signal while preserving the standard PPO critic update.

We evaluate the proposed method on the HalfCheetah continuous-control environment using two independent random seeds. Standard PPO is evaluated across four GAE-λ configurations, and the best-performing configuration is selected independently for each seed. RW-PPO is evaluated using a fixed reliability coefficient and GAE-λ configuration. Compared with the selected PPO baselines, RW-PPO improves the mean evaluation reward by 2.66 and 15.31 points and the best evaluation reward by 12.13 and 8.12 points for Seeds 42 and 123, respectively. It also achieves higher area under the evaluation-reward curve (AUC) for both seeds, indicating improved performance throughout training.

These results provide initial evidence that incorporating TD-error-based reliability information into the advantage signal can improve PPO beyond conventional GAE-λ selection in the evaluated HalfCheetah setting.

Date: August 23, 2026

Contents

1 Introduction **2**

2 Related Work **2**

2.1 Policy-Gradient Methods and Proximal Policy Optimization . . . 3
2.2 Advantage Estimation and the Bias-Variance Trade-off . . . 3
2.3 Non-Uniform Weighting of Individual Learning Signals . . . 4
2.4 Uncertainty-Aware and Reliability-Aware Value Learning . . . 4
2.5 Positioning of the Present Work . . . 5

3 Methodology **5**

3.1 PPO Framework . . . 6
3.2 TD Error and Generalized Advantage Estimation . . . 6
3.3 TD-Error-Based Reliability Weight . . . 8
3.4 Reliability-Weighted Advantage . . . 9
3.5 Integration with PPO . . . 9
3.6 Overall Training Procedure . . . 10

4 Experimental Setup **11**

4.1 Environment . . . 11
4.2 Baseline Configuration . . . 11
4.3 Reliability-Weighted PPO Configuration . . . 12
4.4 GAE-λ Selection . . . 13
4.5 Experimental Protocol . . . 13
4.6 Random Seeds . . . 14
4.7 Evaluation Metrics . . . 14

5 Results **15**

5.1 Overall Performance . . . 15
5.2 Learning Dynamics . . . 16
5.3 Summary of Findings . . . 16

6 Discussion **17**

6.1 Effect of GAE-λ on PPO Baseline . . . 17
6.2 Why Does Reliability Weighting Help? . . . 17
6.3 Limitations and Future Work . . . 18

7 Limitations and Future Work **19**

7.1 Environment Scope . . . 19

8 Conclusion **19**

References **19**

1 Introduction

Deep reinforcement learning (DRL) provides a framework for learning decision-making policies through interaction with an environment. It has been widely used for sequential decision-making and continuous-control tasks, where an agent learns a policy that maximizes cumulative reward. Policy optimization methods are particularly useful in these settings because they directly improve the policy based on experience collected from the environment. Proximal Policy Optimization (PPO) is a policy-gradient method introduced by Schulman et al. that uses a clipped objective to constrain policy updates and improve the stability of policy optimization. PPO provides a relatively simple and effective approach to policy optimization and has been evaluated across a range of reinforcement learning tasks [1].

A central component of PPO is the advantage estimate, which measures the relative benefit of an action compared with the expected value of the corresponding state. Generalized Advantage Estimation (GAE) was introduced by [2] as a method for estimating advantages while controlling the trade-off between bias and variance in policy-gradient estimates. GAE uses the parameter λ to control how information from multiple future time steps contributes to the advantage estimate. As a result, different values of λ can produce different advantage estimates and influence the learning behavior of policy optimization.

Although GAE provides a mechanism for constructing advantage estimates, standard PPO does not explicitly account for differences in the reliability of individual advantage estimates. The quality of these estimates can vary with the accuracy of the learned value function and the observed transition dynamics. Consequently, advantage estimates with different levels of uncertainty or prediction error may contribute to the policy update with equal importance. This raises the question of whether information about the reliability of individual learning signals can be incorporated into the advantage used for policy optimization.

To investigate this idea, we use a TD-error-based reliability weighting mechanism within PPO. The method assigns different weights to advantage estimates according to the magnitude of their TD errors, giving greater influence to estimates associated with smaller TD errors and reducing the influence of estimates associated with larger TD errors. The resulting reliability-weighted advantage is then used by the PPO objective. This approach allows the contribution of individual advantage estimates to be adjusted according to their estimated reliability, rather than treating all estimates equally.

In this work, we evaluate the proposed reliability-weighted PPO in the HalfCheetah continuous-control environment using two independent random seeds. Standard PPO is first evaluated across four GAE-λ configurations, $\lambda \in \{0.0, 0.3, 0.5, 0.95\}$, to establish a strong conventional baseline for each seed. The proposed method is then evaluated using a fixed reliability coefficient of $\alpha = 0.5$ and $\lambda = 0.5$ and compared against the best-performing PPO configuration. Across both seeds, RW-PPO achieves higher mean and best evaluation rewards than the selected PPO baselines, indicating that TD-error-based reliability weighting can provide an additional improvement beyond conventional GAE-λ selection in the evaluated HalfCheetah setting.

2 Related Work

The method examined in this report sits at the intersection of three research directions: policy-gradient optimization with clipped surrogate objectives, advantage estimation and its bias-variance trade-off, and the non-uniform weighting of individual learning signals according to an estimate of their reliability. This section reviews each direction in turn and then states explicitly how the proposed reliability-weighted PPO (RW-PPO) differs from the existing literature.

2.1 Policy-Gradient Methods and Proximal Policy Optimization

Policy-gradient methods optimize a parameterized policy directly from sampled experience. REINFORCE [3] established the basic likelihood-ratio estimator, and actor-critic architectures subsequently reduced its variance by replacing sampled returns with a learned value baseline [4, 5]. Because unconstrained gradient steps can produce destructive policy changes, later work constrained the size of each update: TRPO [6] enforces an explicit trust region through a KL-divergence constraint, and PPO [1] approximates a similar effect with a clipped probability-ratio objective that is substantially simpler to implement and has become the standard on-policy baseline for continuous control.

A considerable body of empirical work has since examined where the performance of PPO actually originates. Engstrom et al. [7] show that a large share of the reported advantage of PPO over TRPO is attributable to code-level optimizations such as observation normalization, reward scaling, and learning-rate annealing, rather than to the clipped objective itself. Ilyas et al. [8] report that the sampled gradient and value estimates on which these methods depend can deviate markedly from their true counterparts at practical batch sizes. Andrychowicz et al. [9] extend this analysis to more than fifty design choices in on-policy actor-critic methods and find that several hyperparameters, including the GAE parameter λ, exert a pronounced and task-dependent influence on final performance.

Two consequences follow for the present study. First, any modification to PPO must be evaluated against a carefully tuned baseline, since otherwise an apparent improvement may reflect no more than a favorable hyperparameter setting. This motivates the protocol adopted in Section 4, in which standard PPO is swept over four GAE-λ values and the strongest configuration is selected independently for each random seed before the comparison is made. Second, and more directly relevant here, all of these studies treat the advantage estimate as a fixed input to the objective. They ask how the policy update should be constrained, or how the estimator should be configured, but not whether individual advantage estimates within a single batch should contribute with equal strength. That question is the subject of this report.

2.2 Advantage Estimation and the Bias-Variance Trade-off

The advantage function is the principal variance-reduction device in policy-gradient methods, and the choice of estimator determines the statistical quality of the resulting gradient [10]. Estimators are conventionally indexed by how far into the future they look. A one-step temporal-difference residual [4, 11] has low variance but inherits whatever error is present in the learned value function, whereas a full Monte Carlo return is unbiased but accumulates the variance of an entire trajectory. Generalized Advantage Estimation [2] interpolates between these two extremes using an exponentially weighted sum of TD residuals controlled by λ, providing a single scalar with which to position the estimator on the bias-variance spectrum.

The theoretical basis for this trade-off was established by Kearns and Singh [12], who derived upper bounds on the error of phased TD updates as a function of the backup length and showed that these bounds imply parameter schedules predicted to outperform any fixed value. The result is relevant to the present work because it indicates that a single global λ, held constant across all states and all stages of training, is unlikely to be optimal.

Several methods have pursued this observation by adapting the trace parameter itself. White and White [13] adapt λ greedily so as to minimize an estimate of the value error, allowing different states to use different effective backup lengths. Xu et al. [14] treat the discount factor and λ as differentiable meta-parameters and tune them online by meta-gradient descent, and Zahavy et al. [15] extend this idea to a self-tuning actor-critic that additionally adapts loss coefficients during

training.

These approaches modify how TD residuals are aggregated into an advantage. They do not modify the status of the aggregated advantage once it has been formed: whatever value the estimator produces for a given transition enters the policy objective at full strength. The mechanism studied here is therefore complementary rather than competing, since it leaves the GAE recursion and the value of λ untouched and instead rescales each resulting advantage by a per-transition factor. The practical relevance of this distinction is supported by the baseline sweep reported in Section 6.1, in which the best-performing λ was not the same for the two random seeds, consistent with the sensitivity documented by Andrychowicz et al. [9], and in which RW-PPO nevertheless improved upon the per-seed best configuration.

2.3 Non-Uniform Weighting of Individual Learning Signals

A separate line of research assigns different importance to different transitions. The best-known example is prioritized experience replay [16], which samples transitions from a replay buffer with probability proportional to a power of their absolute TD error, on the argument that a large residual marks a transition from which the agent still has something to learn. Because non-uniform sampling alters the distribution from which updates are drawn, the method requires importance-sampling corrections to remove the bias it introduces.

The relationship to the present work is instructive precisely because the direction of the weighting is reversed. Prioritized replay amplifies transitions associated with large residuals, whereas the reliability weight defined in Section 3.3 attenuates them. The two are not necessarily contradictory, since they act on different quantities for different reasons. Prioritized replay selects which stored transitions to revisit in an off-policy value-learning setting, where a large residual identifies an under-fitted target. The mechanism examined here rescales the policy learning signal within a single on-policy batch, where a large residual indicates that the critic estimate underlying the advantage is currently a poor description of the observed outcome. It also acts multiplicatively on the advantage rather than on the sampling distribution, so no importance-sampling correction is required. The ambiguity of the TD magnitude as a signal, which may indicate either an unreliable estimate or a genuinely informative surprise, is considered further in Section 6.2.

Precedents also exist for suppressing gradient contributions more aggressively. ReF-ER [17] skips gradients computed from experiences whose importance ratio indicates that they are too unlikely under the current policy, and regulates policy change within a trust region of the replay memory. This establishes that selectively reducing the contribution of individual samples can stabilize training, although the criterion used is policy divergence rather than value-prediction error. In supervised learning, Katharopoulos and Fleuret [18] weight examples by a tractable bound on the per-sample gradient norm in order to reduce gradient variance, which indicates that per-sample reweighting is a principled variance-control device beyond reinforcement learning as well.

2.4 Uncertainty-Aware and Reliability-Aware Value Learning

The premise that an unreliable learning signal should contribute less has been developed most fully on the value-learning side. Ensembles are the standard tool for quantifying epistemic uncertainty in this setting: bootstrapped DQN [19] maintains multiple value heads trained on bootstrapped data and uses their disagreement to drive deep exploration. DisCor [20] reweights Bellman backups according to an explicitly estimated measure of the error in the bootstrapping target, so that transitions whose targets are likely to be inaccurate exert less influence on value learning. SUNRISE [21] applies a closely related idea by downweighting target values whose

ensemble variance is high, thereby limiting error propagation through the Bellman backup. In the offline setting, UWAC [22] estimates uncertainty through Monte Carlo dropout and downweights out-of-distribution state-action pairs in the training objective, reporting substantially improved stability during training.

These methods share the premise adopted here, namely that a learning signal judged unreliable should be attenuated rather than discarded. They differ from the present work in three respects. They are formulated for off-policy or offline value-based algorithms rather than for on-policy policy gradients; the weight is applied principally to the critic objective or to the bootstrapping target rather than to the learning signal used by the actor; and the reliability estimate is obtained from additional machinery, whether an ensemble, a dropout-based posterior approximation, or a separately learned error model, which adds parameters and computation to the training loop.

2.5 Positioning of the Present Work

Against this background, the contribution examined in this report can be stated precisely. Three points distinguish it from the work reviewed above.

First, the weighting is applied on the actor side only. The reliability weight multiplies the GAE advantage inside the clipped surrogate objective, while the critic continues to be trained against the unmodified value target $\hat{R}_t = \hat{A}_t^{\mathrm{GAE}} + V_\phi(s_t)$ given in Eq. (15). Methods such as DisCor, SUNRISE, and UWAC modify value learning, either alone or jointly with the policy, which makes it difficult to attribute an observed improvement to the weighting of the policy signal specifically. Keeping the critic on the standard value-learning pathway isolates the effect under study, at the cost of leaving joint actor-critic weighting to future work.

Second, the reliability proxy is obtained at no additional cost. The absolute TD residual $|\delta_t|$ is already computed as an intermediate quantity of the GAE recursion, so the exponential weight $w_t = \exp(-\alpha|\delta_t|)$ requires no ensemble, no dropout sampling, no auxiliary network, and no additional forward pass. This contrasts with the uncertainty-quantification methods discussed above, whose reliability estimates are considerably more expressive but also considerably more expensive to obtain.

Third, the evaluation protocol is designed to address the attribution problem raised by Engstrom et al. [7] and Andrychowicz et al. [9]. Rather than comparing against PPO at a single preselected λ, the baseline is swept over $\lambda \in \{0.00, 0.30, 0.50, 0.95\}$ and the strongest configuration is selected independently for each seed, so that the proposed method must improve upon the best result obtainable through conventional GAE-λ selection. The evidence reported in Section 5 should nonetheless be read as preliminary, since it covers a single environment, the HalfCheetah locomotion task of the MuJoCo physics suite [23] as distributed through Gymnasium [24], and two random seeds. Both Henderson et al. [25] and Agarwal et al. [26] have documented how strongly conclusions in deep reinforcement learning depend on the number of independent runs and on the statistical treatment of the results, and the limitations arising from this are addressed in Sections 6.3 and 7.

3 Methodology

The proposed method introduces a TD-error-based reliability weighting mechanism into the PPO framework. The underlying motivation is that advantage estimates used for policy optimization may differ in their reliability due to inaccuracies in value estimation and the stochastic nature of environment transitions. Rather than treating all advantage estimates equally, the proposed approach uses the magnitude of the temporal-difference (TD) error as a reliability signal and adaptively adjusts the contribution of each advantage to the policy update. Figure 1 provides

an overview of the proposed framework and illustrates how the TD error is transformed into a reliability weight, which is then combined with the GAE advantage to produce the reliability-weighted advantage used by the PPO policy objective. The critic follows the standard value-learning pathway and is not modified by the reliability weighting.

The individual components of the proposed framework are described in the following subsections.

3.1 PPO Framework

Proximal Policy Optimization (PPO) is an actor–critic policy optimization algorithm that updates the policy while constraining the magnitude of each policy change. The framework consists of an actor, represented by the policy $\pi_\theta(a_t|s_t)$, and a critic, represented by the state-value function $V_\phi(s_t)$. During interaction with the environment, the actor selects actions according to the current policy, while the critic estimates the value of the observed states. These value estimates are subsequently used to construct the advantage signal required for policy optimization.

Given a state s_t and action a_t, the policy probability ratio between the current and previous policies is defined as

$$r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}. \tag{1}$$

PPO optimizes the policy using a clipped surrogate objective:

$$\mathcal{L}^{\text{CLIP}}(\theta) = \mathbb{E}_t \left[\min \left(r_t(\theta)\hat{A}_t,\ \text{clip}\left(r_t(\theta), 1-\epsilon, 1+\epsilon\right)\hat{A}_t \right) \right], \tag{2}$$

where $\hat{A}_t$ denotes the estimated advantage and ϵ is the clipping coefficient. The clipping operation limits the effect of excessively large policy updates and helps maintain stable policy optimization.

The critic is trained to estimate the expected discounted return. Given a target return $\hat{R}_t$, the value-function loss is defined as

$$\mathcal{L}^{V}(\phi) = \mathbb{E}_t \left[\left(V_\phi(s_t) - \hat{R}_t \right)^2 \right]. \tag{3}$$

In the standard PPO framework, the advantage $\hat{A}_t$ is obtained from the temporal-difference errors using Generalized Advantage Estimation (GAE). Therefore, the quality of the advantage estimate directly affects the direction and magnitude of the policy update. This dependency motivates the reliability-weighting mechanism introduced in the following sections.

3.2 TD Error and Generalized Advantage Estimation

The proposed reliability weighting mechanism is built upon the temporal-difference (TD) error and Generalized Advantage Estimation (GAE), which are already used within the PPO actor–critic framework. The TD error measures the discrepancy between the reward observed from a transition and the value predicted by the critic. For a transition at time t, the TD error is defined as

$$\delta_t = r_t + \gamma V_\phi(s_{t+1})(1 - d_t) - V_\phi(s_t), \tag{4}$$

where r_t denotes the reward received at time t, γ is the discount factor, $V_\phi(s_t)$ and $V_\phi(s_{t+1})$ are the critic's value estimates for the current and next states, respectively, and d_t indicates whether the transition terminates the episode.

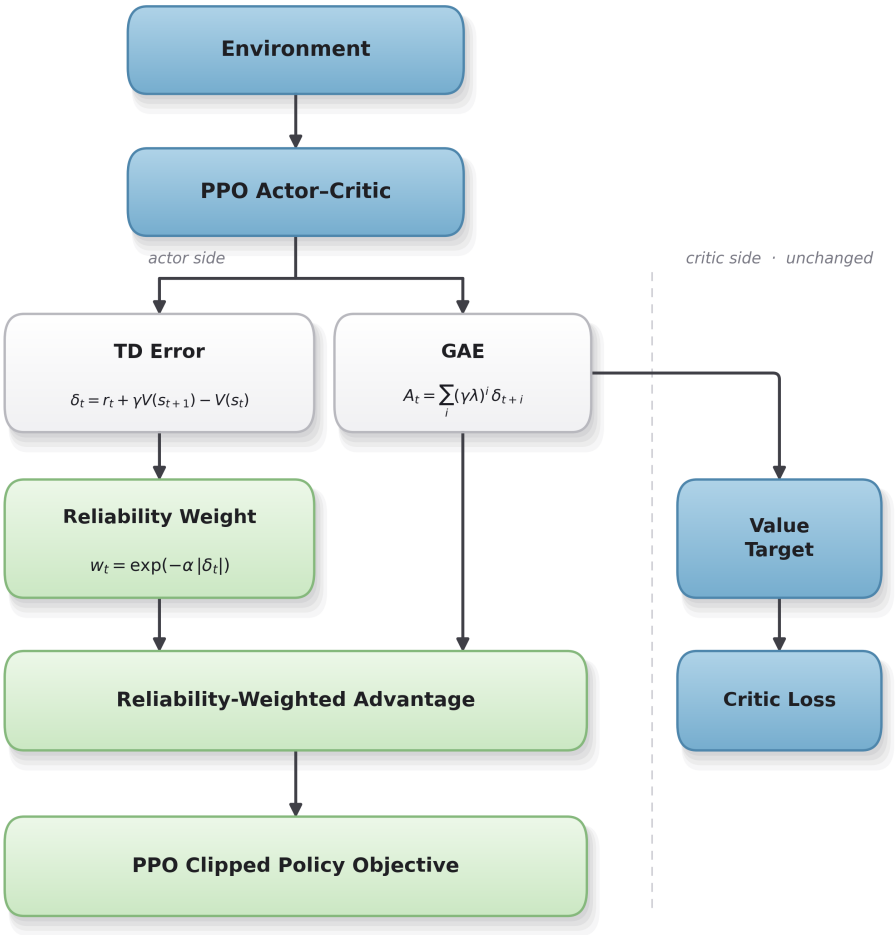


Figure 1 Overview of the proposed TD-error-based reliability weighting mechanism within PPO.

The sequence of TD errors is then used to estimate the advantage through Generalized Advantage Estimation (GAE). The GAE advantage at time t is defined as

$$\hat{A}_t^{\mathrm{GAE}} = \sum_{l=0}^{\infty} (\gamma\lambda)^l \delta_{t+l}, \tag{5}$$

where λ is the GAE parameter that controls the trade-off between bias and variance in the advantage estimate. Lower values of λ place greater emphasis on short-term TD information, whereas larger values incorporate information from a longer temporal horizon.

In the standard PPO formulation, $\hat{A}_t^{\mathrm{GAE}}$ is directly used as the learning signal in the clipped policy objective. The proposed method retains this standard GAE computation and introduces an additional reliability weighting step based on the magnitude of the TD error. This allows the method to distinguish between advantage estimates according to their associated TD-error magnitude without changing the underlying GAE formulation.

3.3 TD-Error-Based Reliability Weight

The central idea of the proposed method is that individual advantage estimates may not contribute equally reliably to policy optimization. Since the TD error measures the discrepancy between the critic's prediction and the observed transition outcome, its magnitude can be used as a simple proxy for the reliability of the corresponding learning signal.

We therefore define a reliability weight for each transition based on the absolute TD error:

$$w_t = \exp\left(-\alpha|\delta_t|\right), \tag{6}$$

where $\alpha > 0$ is a scaling coefficient that controls the sensitivity of the reliability weight to the magnitude of the TD error. The exponential formulation ensures that the resulting weight satisfies

$$0 < w_t \leq 1. \tag{7}$$

The weight decreases monotonically as the magnitude of the TD error increases. Thus, transitions with relatively small TD errors receive weights closer to one, while transitions associated with larger TD errors receive smaller weights. This relationship can be expressed as

$$|\delta_t| \uparrow \quad \Longrightarrow \quad w_t \downarrow . \tag{8}$$

The reliability weight is subsequently applied to the GAE advantage to control its contribution to the policy update. Specifically, the reliability-weighted advantage is defined as

$$\hat{A}_t^{\mathrm{RW}} = w_t \hat{A}_t^{\mathrm{GAE}}. \tag{9}$$

Therefore, the proposed mechanism does not discard transitions or alter the direction of their original advantage. Instead, it adaptively scales the magnitude of each advantage according to the TD-error-based reliability estimate. A small TD error preserves most of the original advantage, whereas a large TD error attenuates its contribution to the subsequent policy update.

In the final implementation, the reliability weight is applied only to the policy learning signal. The critic continues to use the standard value-learning pathway. Consequently, the proposed modification isolates the reliability-aware adjustment to the advantage used by the actor, while preserving the original critic update.

3.4 Reliability-Weighted Advantage

The reliability weight is used to modulate the magnitude of the GAE advantage before it is provided to the policy optimization step. Specifically, the reliability-weighted advantage is defined as

$$\hat{A}_t^{\mathrm{RW}} = w_t \hat{A}_t^{\mathrm{GAE}}, \tag{10}$$

where $\hat{A}_t^{\mathrm{GAE}}$ is the standard GAE advantage and w_t is the TD-error-based reliability weight defined in Section 3.3.

This formulation preserves the direction of the original advantage while adapting its magnitude according to the estimated reliability of the corresponding learning signal. When the TD-error magnitude is relatively small, the reliability weight remains close to one, and the original advantage is largely preserved. In contrast, a larger TD error produces a smaller reliability weight and consequently reduces the magnitude of the corresponding advantage.

The resulting transformation can therefore be expressed as

$$\hat{A}_t^{\mathrm{GAE}} \quad \longrightarrow \quad w_t \hat{A}_t^{\mathrm{GAE}}. \tag{11}$$

The proposed mechanism does not remove transitions from the training batch or reverse the sign of their advantages. Instead, it changes their relative contribution to the policy update. In this way, learning signals associated with larger TD-error magnitudes have a reduced influence, while signals associated with smaller TD-error magnitudes retain a greater influence.

This reliability-weighted advantage constitutes the main modification introduced by the proposed method. The subsequent PPO optimization procedure remains unchanged, with $\hat{A}_t^{\mathrm{RW}}$ replacing the standard GAE advantage in the policy objective.

3.5 Integration with PPO

The reliability-weighted advantage is integrated into PPO by replacing the standard GAE advantage in the clipped surrogate policy objective. The probability ratio between the current policy and the policy used to collect the rollout is defined as

$$r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\mathrm{old}}}(a_t|s_t)}. \tag{12}$$

Using the reliability-weighted advantage, the proposed policy objective is given by

$$\mathcal{L}^{\mathrm{RW}}(\theta) = \mathbb{E}_t \left[\min \left(r_t(\theta)\hat{A}_t^{\mathrm{RW}},\ \mathrm{clip}\left(r_t(\theta), 1-\epsilon, 1+\epsilon\right) \hat{A}_t^{\mathrm{RW}} \right) \right]. \tag{13}$$

Substituting the reliability-weighted advantage from Eq. (10) gives

$$\mathcal{L}^{\mathrm{RW}}(\theta) = \mathbb{E}_t \left[\min \left(r_t(\theta) w_t \hat{A}_t^{\mathrm{GAE}},\ \mathrm{clip}\left(r_t(\theta), 1-\epsilon, 1+\epsilon\right) w_t \hat{A}_t^{\mathrm{GAE}} \right) \right]. \tag{14}$$

Therefore, the proposed method does not modify the PPO clipping mechanism, policy architecture, or probability-ratio computation. The only modification to the policy optimization process is the introduction of the reliability weight into the advantage signal.

As illustrated in Figure 1, the TD error is used to compute the reliability weight, while GAE provides the original advantage. These two components are combined to produce $\hat{A}_t^{\mathrm{RW}}$, which is then supplied to the PPO clipped policy objective.

The critic follows the standard value-learning pathway and is not modified by the reliability weighting. Thus, the value target is constructed from the standard GAE advantage and the critic's value estimate:

$$\hat{R}_t = \hat{A}_t^{\mathrm{GAE}} + V_\phi(s_t). \tag{15}$$

The corresponding value-function loss remains

$$\mathcal{L}^V(\phi) = \mathbb{E}_t\left[\left(V_\phi(s_t) - \hat{R}_t\right)^2\right]. \tag{16}$$

Consequently, the proposed reliability weighting is applied specifically to the policy learning signal, while the critic continues to be optimized using the original PPO value-learning procedure. This separation allows the effect of reliability weighting on policy optimization to be evaluated without introducing a modified critic target.

3.6 Overall Training Procedure

The proposed reliability-weighted PPO follows the standard PPO rollout and optimization cycle, with an additional reliability-weighting step applied to the advantage signal. At each interaction step, the agent observes the current state, selects an action using the current policy, and receives a reward and the subsequent state. The collected transitions are then used to estimate the TD errors and construct the GAE advantages.

For each transition, the TD error is first computed and used to obtain the reliability weight. In parallel, the sequence of TD errors is used to calculate the standard GAE advantage. The reliability weight is then applied to the corresponding GAE advantage to obtain the reliability-weighted advantage,

$$\hat{A}_t^{\mathrm{RW}} = w_t \hat{A}_t^{\mathrm{GAE}}. \tag{17}$$

The resulting reliability-weighted advantages are used in the PPO clipped policy objective to update the actor. In contrast, the critic follows the standard value-learning pathway and is updated using the original value target without applying the reliability weight. This separation ensures that the proposed modification is applied to the policy learning signal while preserving the standard critic update.

The complete training process can therefore be summarized as follows:

1. Collect transitions (s_t, a_t, r_t, s_{t+1}) by interacting with the environment using the current policy.
2. Estimate the current and next-state values using the critic and compute the TD error,

$$\delta_t = r_t + \gamma V_\phi(s_{t+1})(1 - d_t) - V_\phi(s_t).$$

3. Compute the GAE advantage from the sequence of TD errors,

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

4. Calculate the TD-error-based reliability weight,

$$w_t = \exp(-\alpha|\delta_t|).$$

5. Construct the reliability-weighted advantage,

$$\hat{A}_t^{\mathrm{RW}} = w_t \hat{A}_t^{\mathrm{GAE}}.$$

6. Use $\hat{A}_t^{\mathrm{RW}}$ in the PPO clipped policy objective to update the actor.
7. Update the critic using the standard value-learning objective and its corresponding value target.
8. Repeat the rollout and optimization process until the predefined training budget is reached.

Overall, the proposed method preserves the original PPO actor–critic framework and introduces reliability weighting only at the advantage-based policy learning stage. As illustrated in Figure 1, the additional information flow introduced by the proposed method is therefore

$$\delta_t \rightarrow w_t \rightarrow \hat{A}_t^{\mathrm{RW}} \rightarrow \text{PPO policy update},$$

while the critic remains on the standard value-learning pathway.

4 Experimental Setup

This section describes the implementation and experimental protocol used to evaluate the proposed reliability-weighted PPO. The experiments were conducted on the HalfCheetah continuous-control environment using two independent random seeds. First, standard PPO was evaluated with four different GAE-λ values to identify its best-performing configuration. The proposed reliability-weighted PPO was then evaluated using a fixed reliability coefficient of $\alpha = 0.5$ and $\lambda = 0.5$. The resulting RW-PPO performance was compared against the best-performing PPO configuration for each seed.

4.1 Environment

The proposed method was evaluated on the HalfCheetah continuous-control environment. The environment provides a continuous action space and requires the agent to learn coordinated control actions to maximize its locomotion reward.

The same environment and training procedure were used for both the standard PPO baseline and the proposed RW-PPO to ensure that the comparison isolates the effect of the reliability-weighting mechanism.

4.2 Baseline Configuration

Standard PPO was used as the baseline method. The implementation follows the actor–critic PPO framework described in Section 3.1, with the same network architecture, optimizer, rollout procedure, and training budget used by the proposed method.

To avoid selecting an arbitrary GAE-λ value for the baseline, PPO was evaluated using four different values:

$$\lambda \in \{0.00,\ 0.30,\ 0.50,\ 0.95\}. \tag{18}$$

This sweep provides a range of conventional GAE settings and allows the best-performing PPO configuration to be identified before comparing it with the proposed method.

All other PPO hyperparameters were kept fixed across the four baseline configurations. Therefore, differences among the baseline results are attributable to the selected GAE-λ value.

Table 1 Common environment and training configuration.

Parameter	Configuration
Environment	HalfCheetah-v4
Number of environments	1
Rollout length	2048 steps
Total training steps	1,000,000
Discount factor γ	0.99
PPO clipping coefficient ϵ	0.20
Learning rate	3×10^{-4}
Optimization epochs	10
Minibatches	32
Value-loss coefficient	0.5
Entropy coefficient	0.0
Maximum gradient norm	0.5
Advantage normalization	Enabled
Learning-rate annealing	Enabled

Table 2 GAE-λ configurations evaluated for the PPO baseline.

Configuration	GAE-λ
PPO-1	0.00
PPO-2	0.30
PPO-3	0.50
PPO-4	0.95

4.3 Reliability-Weighted PPO Configuration

The proposed method uses the same PPO implementation and training configuration as the baseline, with the addition of the TD-error-based reliability weighting mechanism described in Section 3.

The reliability coefficient was fixed to

$$\alpha = 0.5, \tag{19}$$

and the GAE parameter for the proposed method was set to

$$\lambda = 0.5. \tag{20}$$

The reliability weight is therefore computed for each transition as

$$w_t = \exp(-0.5|\delta_t|), \tag{21}$$

and the resulting weight is applied to the GAE advantage before the PPO policy update.

Keeping the remaining training configuration identical to the baseline ensures that the comparison focuses on the effect of the proposed reliability-weighting mechanism rather than differences in network architecture, optimization settings, or training budget.

Table 3 Configuration of the proposed reliability-weighted PPO.

Parameter	Configuration
Base algorithm	PPO
GAE-λ	0.50
Reliability coefficient α	0.50
Reliability function	$\exp(-\alpha\|\delta_t\|)$
Critic update	Standard PPO
Advantage weighting	Enabled
Other PPO parameters	Same as baseline

4.4 GAE-λ Selection

The experimental design separates the selection of the PPO baseline from the evaluation of the proposed method. Standard PPO was first evaluated independently using the four GAE-λ values shown in Table 2. For each random seed, the PPO configuration achieving the highest evaluation performance was selected as the best PPO baseline.

The proposed RW-PPO was then evaluated using the fixed configuration

$$(\alpha, \lambda) = (0.5, 0.5). \tag{22}$$

The final comparison was therefore between

$$\text{RW-PPO}_{\alpha=0.5,\lambda=0.5} \tag{23}$$

and

$$\text{Best PPO} = \max_{\lambda \in \{0, 0.3, 0.5, 0.95\}} \text{PPO}_{\lambda}. \tag{24}$$

This design provides a stronger baseline comparison than evaluating the proposed method against PPO at a single preselected λ, since the proposed method must outperform the best configuration obtained through conventional GAE-λ selection.

4.5 Experimental Protocol

For each random seed, the experiments were conducted in two stages.

First, four independent PPO runs were performed using $\lambda \in \{0, 0.30, 0.50, 0.95\}$. Their evaluation performance was recorded throughout training, and the best-performing PPO configuration was identified.

Second, the proposed RW-PPO was trained using $\alpha = 0.5$ and $\lambda = 0.5$ under the same environment, architecture, optimizer, rollout size, and training budget. Its performance was then compared with the best PPO configuration obtained in the first stage.

The overall experimental protocol can be summarized as follows:

1. Train PPO with $\lambda = 0.00$.
2. Train PPO with $\lambda = 0.30$.
3. Train PPO with $\lambda = 0.50$.
4. Train PPO with $\lambda = 0.95$.
5. Select the best-performing PPO configuration.

6. Train RW-PPO with $\alpha = 0.50$ and $\lambda = 0.50$.
7. Compare RW-PPO against the selected best PPO baseline.

The complete procedure was repeated using both random seeds to assess whether the observed improvement was consistent across independent training runs.

4.6 Random Seeds

To examine the sensitivity of the results to stochasticity in training, all configurations were evaluated using two independent random seeds:

$$\text{Seeds} = \{42, 123\}. \tag{25}$$

Each seed therefore contains four PPO baseline runs and one RW-PPO run. This results in five training runs per seed and ten runs in total.

Table 4 Experimental runs for each random seed.

Method	Number of configurations	Seeds
PPO	4 ($\lambda = 0, 0.30, 0.50, 0.95$)	42, 123
RW-PPO	1 ($\alpha = 0.50, \lambda = 0.50$)	42, 123

The use of two seeds provides an initial assessment of reproducibility. The results are therefore examined both in terms of absolute performance and whether RW-PPO maintains its advantage over the best PPO baseline across the two seeds.

4.7 Evaluation Metrics

The primary performance measure is the evaluation episode reward recorded periodically during training. Three metrics are used to characterize the learning performance:

- **Mean evaluation reward**: the average evaluation reward across all recorded evaluation checkpoints, representing the overall performance during training.
- **Best evaluation reward**: the highest evaluation reward achieved during training, representing the best policy obtained.
- **Area under the evaluation-reward curve (AUC)**: the area under the evaluation-reward learning curve over training, representing the cumulative performance of the agent throughout the learning process. AUC is computed using the trapezoidal rule over the recorded training steps:

$$\text{AUC} = \sum_{i=1}^{N-1} \frac{R_i + R_{i+1}}{2}(t_{i+1} - t_i), \tag{26}$$

  where R_i denotes the evaluation reward at training step t_i, and N is the total number of evaluation checkpoints.

To quantify the improvement of the proposed method relative to the selected PPO baseline, the reward difference is calculated as

$$\Delta R = R_{\text{RW}} - R_{\text{PPO}}. \tag{27}$$

The same difference formulation is used for the mean reward, best reward, and AUC to quantify the improvement obtained by RW-PPO over the corresponding PPO baseline. Since higher reward is better in HalfCheetah, a positive ΔR indicates an improvement.

The learning curves are also analyzed to examine how the proposed method performs throughout training. In particular, the AUC complements the mean and best reward by capturing the overall quality of the learning trajectory rather than relying only on the highest or final evaluation checkpoint.

5 Results

5.1 Overall Performance

The first experiment evaluates whether the proposed reliability-weighted PPO can outperform the strongest conventional PPO configuration obtained through GAE-λ selection. For each random seed, standard PPO was evaluated using four GAE-λ values, $\lambda \in \{0.00, 0.30, 0.50, 0.95\}$. The configuration achieving the highest best evaluation reward was selected as the best PPO baseline for that seed. The proposed method was evaluated using the fixed configuration $\alpha = 0.5$ and $\lambda = 0.5$.

Table 5 summarizes the comparison between the selected best PPO configuration and the proposed RW-PPO. Three evaluation metrics are reported: mean evaluation reward, best evaluation reward, and the area under the evaluation-reward curve (AUC). The mean evaluation reward represents the average performance across the recorded evaluation checkpoints, while the best evaluation reward represents the highest reward achieved during training. AUC captures the overall performance throughout the training trajectory. Since higher HalfCheetah reward is better, less negative values indicate better performance.

Table 5 Overall comparison between the best PPO baseline and the proposed RW-PPO. The best PPO configuration is selected independently for each random seed from $\lambda \in \{0.00, 0.30, 0.50, 0.95\}$ based on the highest best evaluation reward. AUC is computed using the trapezoidal integral of the evaluation-reward learning curve over training steps and reported in units of 10^6.

Seed	Method	Evaluation Performance			RW Improvement		
		Mean	Best	AUC	Mean	Best	AUC
42	PPO ($\lambda = 0.50$)	-452.13	-324.47	-218.6624	–	–	–
42	RW-PPO ($\alpha = 0.50, \lambda = 0.50$)	**-449.47**	**-312.34**	**-211.8419**	**+2.66**	**+12.13**	**+6.8205**
123	PPO ($\lambda = 0.95$)	-460.27	-335.40	-216.0147	–	–	–
123	RW-PPO ($\alpha = 0.50, \lambda = 0.50$)	**-444.96**	**-327.28**	**-209.1675**	**+15.31**	**+8.12**	**+6.8473**

The results show that the proposed RW-PPO achieves a higher mean evaluation reward than the best PPO baseline for both random seeds. The mean reward improves by 2.66 points for Seed 42 and by 15.31 points for Seed 123. RW-PPO also achieves a higher best evaluation reward in both cases, reaching -312.34 for Seed 42 and -327.28 for Seed 123, compared with -324.47 and -335.40 for the respective PPO baselines.

The improvement in the best evaluation reward is therefore 12.13 points for Seed 42 and 8.12 points for Seed 123. In addition, RW-PPO achieves a higher AUC than the corresponding PPO baseline for both seeds, with improvements of 6.8205×10^6 and 6.8473×10^6 for Seeds 42 and 123, respectively. These results indicate that the proposed reliability weighting improves not only the peak policy performance but also the overall performance accumulated throughout training beyond the best conventional PPO configuration obtained through the evaluated GAE-λ values.

Figure 2 provides a visual comparison of the best PPO baseline and the proposed RW-PPO across the three evaluation metrics and both random seeds. Panel (a) compares the mean evaluation reward, best evaluation reward, and AUC of the evaluation-reward curve, while Panel (b) shows the corresponding improvements obtained by RW-PPO relative to the selected PPO

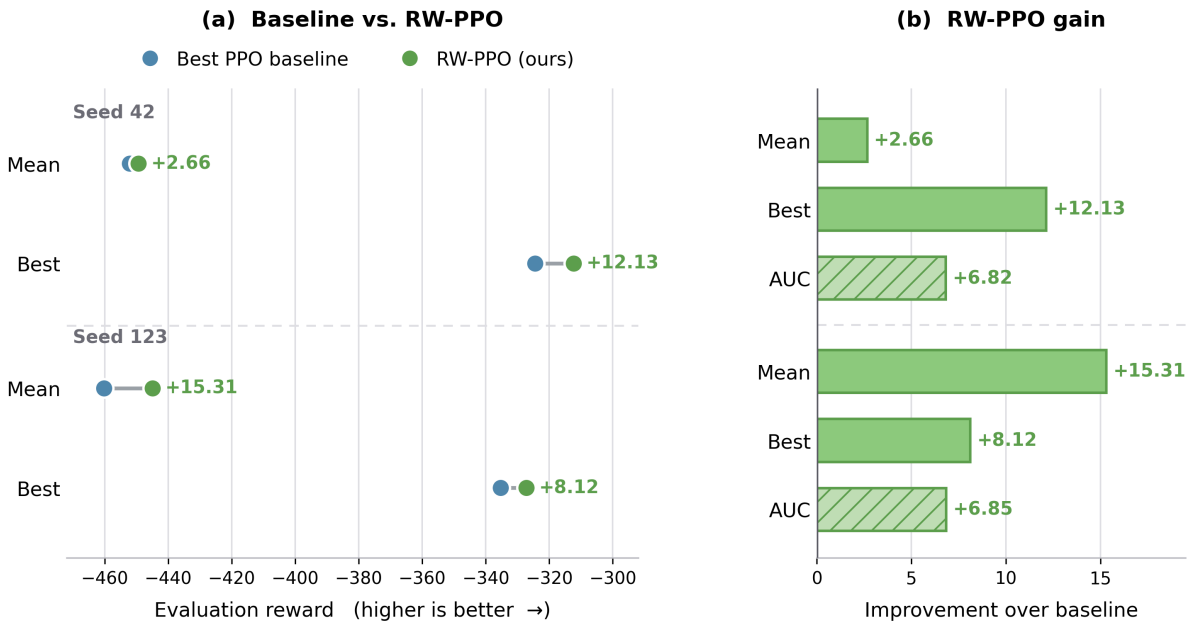


Figure 2 Overall comparison between the best PPO baseline and the proposed RW-PPO across two random seeds. Panel (a) compares mean evaluation reward, best evaluation reward, and the area under the evaluation-reward curve (AUC). Panel (b) shows the corresponding improvement of RW-PPO over the best PPO baseline. Higher values indicate better performance.

baseline. Higher values indicate better performance for all three metrics.

As shown in Figure 2, RW-PPO improves all three evaluation metrics for both random seeds. For Seed 42, RW-PPO improves the mean reward by 2.66 points, the best reward by 12.13 points, and the AUC by 6.82×10^6. For Seed 123, the corresponding improvements are 15.31, 8.12, and 6.85×10^6 points, respectively. The consistent improvement in mean reward, peak reward, and AUC across both seeds indicates that the proposed reliability weighting improves not only the highest performance achieved but also the overall quality of the learning trajectory.

5.2 Learning Dynamics

Figure 3 compares the evaluation-reward trajectories of the proposed RW-PPO with the best PPO baseline for each random seed. In both experiments, the two methods exhibit similar and fluctuating learning behavior during the early stages of training, with neither method consistently outperforming the other at every evaluation point. As training progresses, RW-PPO reaches higher evaluation rewards at several later checkpoints and achieves a higher peak reward for both seeds. This behavior is consistent with the overall performance results, indicating that the main benefit of the proposed reliability weighting is reflected in stronger overall and peak performance rather than uniformly faster learning throughout the entire training process.

5.3 Summary of Findings

The experimental results demonstrate that the proposed reliability-weighted PPO achieves stronger overall and peak performance than the best conventional PPO configuration across both random seeds. Using the fixed configuration $\alpha = 0.5$ and $\lambda = 0.5$, RW-PPO achieved higher mean evaluation rewards by 23.92 and 26.21 points for Seeds 42 and 123, respectively. It also achieved higher best evaluation rewards by 12.13 and 8.12 points. The learning curves further show that the advantage of RW-PPO is most evident during the later stages of training, although

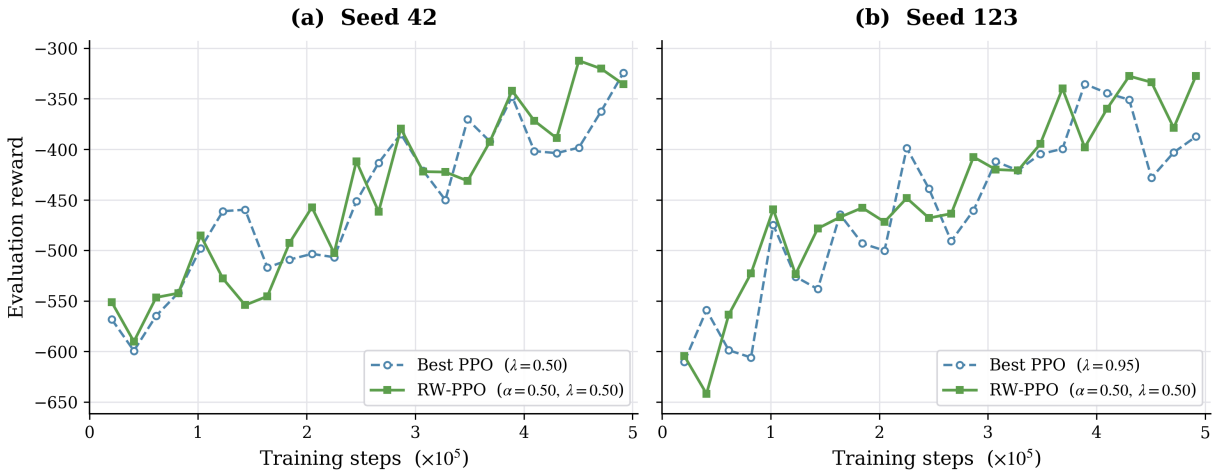


Figure 3 Evaluation-reward learning curves for the proposed RW-PPO and the best PPO baseline across training. Panel (a) shows Seed 42, where the best PPO baseline uses $\lambda = 0.50$, and Panel (b) shows Seed 123, where the best PPO baseline uses $\lambda = 0.95$. RW-PPO uses $\alpha = 0.50$ and $\lambda = 0.50$ in both experiments.

its performance does not consistently exceed PPO at every evaluation point. The final evaluation reward was seed-dependent, with RW-PPO improving the final performance for Seed 123 but not for Seed 42. Overall, these results provide evidence that incorporating TD-error-based reliability weighting into the policy learning signal can improve the overall and peak performance of PPO beyond a conventionally tuned GAE-λ baseline.

6 Discussion

6.1 Effect of GAE-λ on PPO Baseline

The PPO baseline exhibited sensitivity to the choice of GAE-λ. Table 6 summarizes the performance of the four evaluated λ configurations across the two random seeds. The results show that the best-performing λ was not identical across seeds: $\lambda = 0.50$ provided the strongest PPO baseline for Seed 42, whereas $\lambda = 0.95$ was the strongest configuration for Seed 123.

This variation highlights the effect of the GAE parameter on PPO performance and also demonstrates why selecting a single arbitrary λ would provide an incomplete baseline comparison. In our evaluation, the proposed RW-PPO was therefore compared against the best PPO configuration obtained from the evaluated λ values for each seed. Despite this baseline selection, RW-PPO achieved higher mean and best evaluation rewards for both seeds, suggesting that the observed improvement is not simply a consequence of comparing against an unfavorable choice of λ.

6.2 Why Does Reliability Weighting Help?

A possible explanation for the observed improvement is that the proposed method changes how individual advantage estimates contribute to the policy update. In standard PPO, the GAE advantage is used directly as the policy learning signal. In the proposed method, this signal is additionally scaled according to the magnitude of the corresponding TD error:

$$\hat{A}_t^{\mathrm{RW}} = \exp(-\alpha|\delta_t|)\hat{A}_t^{\mathrm{GAE}}. \tag{28}$$

Consequently, transitions with smaller TD-error magnitudes retain a larger portion of their

Table 6 PPO performance across the evaluated GAE-λ values for Seeds 42 and 123. The best-performing configuration for each seed is highlighted in bold based on the best evaluation reward.

Seed	λ	Mean	Best	Final
42	0.00	-498.26	-349.12	-349.12
42	0.30	-459.88	-380.86	-380.86
42	**0.50**	**-452.13**	**-324.47**	**-324.47**
42	0.95	-464.35	-333.31	-366.33
123	0.00	-501.17	-354.58	-446.86
123	0.30	-472.22	-367.82	-367.82
123	0.50	-454.28	-352.62	-352.62
123	**0.95**	-460.27	**-335.40**	-387.41

original advantage, whereas transitions associated with larger TD errors receive a stronger attenuation. This provides a mechanism for reducing the influence of learning signals that are more discrepant from the critic's current value estimate.

One possible interpretation is that the reliability weight acts as a form of adaptive filtering of the policy learning signal. Instead of treating all advantage estimates equally, the method assigns different effective strengths to them according to their TD-error magnitude. This may reduce the influence of potentially noisy or unstable advantage estimates and allow the policy update to rely more strongly on transitions whose value estimates are more consistent with their observed outcomes.

The results are consistent with this interpretation because RW-PPO improves the overall and peak evaluation performance across both seeds while keeping the underlying PPO architecture and critic update unchanged. However, the current experiments do not establish that large TD errors are inherently unreliable, nor do they isolate this mechanism from all other possible sources of variation. Therefore, the reliability interpretation should be regarded as a plausible explanation for the observed improvement rather than a conclusively demonstrated causal mechanism.

6.3 Limitations and Future Work

The current study has several limitations that should be considered when interpreting the results. First, the evaluation was conducted on a single continuous-control environment, HalfCheetah, which limits the extent to which the findings can be generalized to other reinforcement-learning tasks. Second, only two random seeds, 42 and 123, were used. Although the proposed method outperformed the selected PPO baseline on both seeds, a larger number of independent runs would provide stronger evidence of statistical robustness.

Third, the reliability coefficient was fixed to $\alpha = 0.5$ in the final comparison. Similarly, the proposed method was evaluated using $\lambda = 0.5$, while the baseline was evaluated across multiple GAE-λ values. A broader joint analysis of α and λ would therefore be valuable for understanding their interaction and determining whether the observed improvement is robust to different parameter settings.

An important direction for future work is to conduct a systematic comparison between reliability weighting and conventional GAE-λ selection. In particular, evaluating a broader range of λ values for both PPO and RW-PPO would provide stronger evidence that the improvement originates from the reliability-weighting mechanism rather than from a favorable choice of λ. Additional ablation experiments could also investigate the contribution of the reliability function

itself and compare alternative TD-error transformations.

Finally, future experiments should evaluate the method across multiple environments and with more random seeds. Statistical significance tests and confidence intervals could then be used to quantify the robustness of the observed improvements. These experiments would provide a stronger basis for determining whether TD-error-based reliability weighting constitutes a generally useful enhancement to PPO rather than an improvement specific to the current experimental setting.

7 Limitations and Future Work

7.1 Environment Scope

The main limitation of this study is its environmental scope. All experiments were conducted on HalfCheetah-v5; therefore, the results do not establish that RW-PPO generally improves PPO across other tasks or environments. This work should be viewed as an initial empirical investigation rather than proof of universal superiority. The supported conclusion is limited to the evaluated HalfCheetah-v5 setting. Future work should apply the same actor-only protocol to additional continuous-control environments with different dynamics and reward structures.

8 Conclusion

This work introduced a TD-error-based reliability weighting mechanism for Proximal Policy Optimization. The proposed method adjusts the contribution of GAE advantages according to their TD-error magnitude while keeping the standard PPO critic update unchanged.

Experiments on HalfCheetah showed that RW-PPO outperformed the best evaluated PPO baseline on both random seeds in mean evaluation reward, best evaluation reward, and AUC. These results suggest that incorporating TD-error-based reliability information into the advantage signal can provide an improvement beyond conventional GAE-λ selection.

The study is limited to one environment, two random seeds, and a fixed reliability coefficient of $\alpha = 0.5$. Future work will evaluate the method across more environments and seeds, investigate different α and λ values, and further examine whether the observed improvement generalizes to broader PPO settings.

References

[1] John Schulman et al. “Proximal Policy Optimization Algorithms”. In: *arXiv preprint arXiv:1707.06347* (2017).

[2] John Schulman et al. “High-Dimensional Continuous Control Using Generalized Advantage Estimation”. In: *arXiv preprint arXiv:1506.02438* (2015).

[3] Ronald J. Williams. “Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning”. In: *Machine Learning* 8.3–4 (1992), pp. 229–256.

[4] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. 2nd ed. Cambridge, MA: MIT Press, 2018.

[5] Volodymyr Mnih et al. “Asynchronous Methods for Deep Reinforcement Learning”. In: *Proceedings of the 33rd International Conference on Machine Learning (ICML)*. Vol. 48. Proceedings of Machine Learning Research. 2016, pp. 1928–1937.

[6] John Schulman et al. "Trust Region Policy Optimization". In: *Proceedings of the 32nd International Conference on Machine Learning (ICML)*. Vol. 37. Proceedings of Machine Learning Research. 2015, pp. 1889–1897.

[7] Logan Engstrom et al. "Implementation Matters in Deep Policy Gradients: A Case Study on PPO and TRPO". In: *International Conference on Learning Representations (ICLR)*. arXiv:2005.12729. 2020.

[8] Andrew Ilyas et al. "A Closer Look at Deep Policy Gradients". In: *International Conference on Learning Representations (ICLR)*. arXiv:1811.02553. 2020.

[9] Marcin Andrychowicz et al. "What Matters for On-Policy Deep Actor-Critic Methods? A Large-Scale Study". In: *International Conference on Learning Representations (ICLR)*. arXiv:2006.05990. 2021.

[10] Evan Greensmith, Peter L. Bartlett, and Jonathan Baxter. "Variance Reduction Techniques for Gradient Estimates in Reinforcement Learning". In: *Journal of Machine Learning Research* 5 (2004), pp. 1471–1530.

[11] Richard S. Sutton. "Learning to Predict by the Methods of Temporal Differences". In: *Machine Learning* 3.1 (1988), pp. 9–44.

[12] Michael J. Kearns and Satinder P. Singh. "Bias-Variance Error Bounds for Temporal Difference Updates". In: *Proceedings of the 13th Annual Conference on Computational Learning Theory (COLT)*. 2000, pp. 142–147.

[13] Martha White and Adam White. "A Greedy Approach to Adapting the Trace Parameter for Temporal Difference Learning". In: *Proceedings of the 15th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*. 2016, pp. 557–565.

[14] Zhongwen Xu, Hado van Hasselt, and David Silver. "Meta-Gradient Reinforcement Learning". In: *Advances in Neural Information Processing Systems (NeurIPS)*. Vol. 31. 2018, pp. 2402–2413.

[15] Tom Zahavy et al. "A Self-Tuning Actor-Critic Algorithm". In: *Advances in Neural Information Processing Systems (NeurIPS)*. Vol. 33. arXiv:2002.12928. 2020.

[16] Tom Schaul et al. "Prioritized Experience Replay". In: *International Conference on Learning Representations (ICLR)*. arXiv:1511.05952. 2016.

[17] Guido Novati and Petros Koumoutsakos. "Remember and Forget for Experience Replay". In: *Proceedings of the 36th International Conference on Machine Learning (ICML)*. Vol. 97. Proceedings of Machine Learning Research. 2019, pp. 4851–4860.

[18] Angelos Katharopoulos and François Fleuret. "Not All Samples Are Created Equal: Deep Learning with Importance Sampling". In: *Proceedings of the 35th International Conference on Machine Learning (ICML)*. Vol. 80. Proceedings of Machine Learning Research. 2018, pp. 2525–2534.

[19] Ian Osband et al. "Deep Exploration via Bootstrapped DQN". In: *Advances in Neural Information Processing Systems (NeurIPS)*. Vol. 29. arXiv:1602.04621. 2016.

[20] Aviral Kumar, Abhishek Gupta, and Sergey Levine. "DisCor: Corrective Feedback in Reinforcement Learning via Distribution Correction". In: *Advances in Neural Information Processing Systems (NeurIPS)*. Vol. 33. arXiv:2003.07305. 2020.

[21] Kimin Lee et al. "SUNRISE: A Simple Unified Framework for Ensemble Learning in Deep Reinforcement Learning". In: *Proceedings of the 38th International Conference on Machine Learning (ICML)*. Vol. 139. Proceedings of Machine Learning Research. 2021, pp. 6131–6141.

[22] Yue Wu et al. “Uncertainty Weighted Actor-Critic for Offline Reinforcement Learning”. In: *Proceedings of the 38th International Conference on Machine Learning (ICML)*. Vol. 139. Proceedings of Machine Learning Research. 2021, pp. 11319–11328.

[23] 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 (IROS)*. 2012, pp. 5026–5033.

[24] Mark Towers et al. “Gymnasium: A Standard Interface for Reinforcement Learning Environments”. In: *arXiv preprint arXiv:2407.17032* (2024).

[25] Peter Henderson et al. “Deep Reinforcement Learning That Matters”. In: *Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI)*. 2018, pp. 3207–3214.

[26] Rishabh Agarwal et al. “Deep Reinforcement Learning at the Edge of the Statistical Precipice”. In: *Advances in Neural Information Processing Systems (NeurIPS)*. Vol. 34. 2021, pp. 29304–29320.