KLip-PPO: A per-sample KL perspective on PPO-Clip

Riccardo Colletti*
University of California, Berkeley
riccardo_colletti [at] berkeley.edu

Robin Holzinger*
University of California, Berkeley
robin.holzinger [at] berkeley.edu

Abstract

Proximal Policy Optimization (PPO) is the standard policy-gradient algorithm for on-policy reinforcement learning. The literature presents it in two forms, a clipped surrogate that bounds the importance ratio between successive policies and a Kullback–Leibler penalty between them. These forms are treated as separate algorithms with their own gradients, their own hyperparameters, and their own reference implementations, and a sizeable body of empirical work compares them. We show that the gradient of the clipped surrogate is reproduced exactly by a Kullback–Leibler surrogate whose coefficient varies per sample, with closed-form dependence on the importance ratio and the advantage. The identity holds at every minibatch step and across the entire inner loop, and on five MuJoCo continuous-control benchmarks the two losses produce indistinguishable training curves. The reformulation exposes a structural feature of the clipped surrogate that the $\min$ notation hides. PPO-Clip's implicit per-sample penalty is a step function at the boundary of the trust region, and the shape of this coefficient is the natural design axis for generalising the algorithm. We sketch the resulting follow-up directions in the discussion.

1 Introduction

Proximal Policy Optimization [15] is the default policy-gradient algorithm for on-policy reinforcement learning. The method approximates the trust-region step of TRPO [9, 13] by maximising a surrogate objective that keeps the new policy close to the rollout policy. The original work proposes two surrogates. The first clips the importance ratio between the new and old policies inside a fixed band. The second adds an adaptive Kullback–Leibler penalty between them. Schulman et al. [15] report that the clipped variant outperforms the penalty variant on MuJoCo continuous-control benchmarks and recommend it as the default. The community has followed the recommendation. PPO-Clip is the de facto choice in modern open-source implementations [7, 12], and the clipped surrogate is the building block of token-level extensions such as GRPO [16] that underpin recent reasoning models [2].

Subsequent empirical work has scrutinised PPO from many angles. Engstrom et al. [4] show that "code-level" optimisations explain most of PPO's gain over TRPO and that the clip mechanism itself is not load-bearing for performance. Ilyas et al. [8] demonstrate that auxiliary optimisations, rather than the clip term, are what actually maintain the trust region. Andrychowicz et al. [1] train over 2.5×10^5 agents to compare design choices and recommend PPO-Clip among five policy losses, though they do not run a standalone PPO-KL variant. Hsu et al. [6] report that KL-regularised PPO matches or outperforms the clipped variant outside MuJoCo with Gaussian policies, and Sun et al.

*Equal contribution.
Code: https://github.com/learning-mechanisms/KLip-PPO
Project page: https://klip-ppo.org
Public W&B artifacts: https://wandb.ai/KLip-PPO/KLip-PPO

Preprint.

[17] argue that ratio clipping is not necessary in PPO at all. Across this body of work the clip and KL forms are treated as alternative algorithmic choices to be compared empirically.

We show that this treatment misunderstands the relationship between the two surrogates. The per-sample gradient of PPO-Clip is reproduced exactly by a Kullback–Leibler surrogate whose coefficient varies per sample, with closed-form dependence on the importance ratio and the advantage. The identity holds at every minibatch step and across the entire inner loop. On HalfCheetah, Hopper, Walker2d, Ant, and Humanoid [20] the two losses produce indistinguishable training curves. Where the original PPO paper notes that L^{CLIP} and the unclipped surrogate agree to first order around θ_{old} [15], we strengthen the statement to a per-sample identity that holds at every parameter configuration. Recent work has analysed gradient-level relationships between different KL formulations [10] and proposed unified clip-plus-KL design frameworks [24], but to our knowledge no prior work establishes the per-sample equivalence between PPO-Clip and PPO-KL itself.

Making the implicit coefficient explicit clarifies the position of PPO-Clip in the policy-optimisation landscape. The clipped surrogate is a per-sample KL penalty whose coefficient is a step function on the trust-region boundary, and the **shape of this coefficient** is the natural design axis for generalising the algorithm. Soft relaxations of the step, asymmetric and position-conditioned penalties, and off-policy extensions all fit inside the same template; we sketch these directions in Section 6.

The paper is organised as follows. Section 2 reviews PPO-Clip, PPO-KL, and the existing comparisons between them. Section 3 states and proves the per-sample gradient identity. Section 4 validates the identity empirically on five MuJoCo continuous-control benchmarks. Section 6 surveys natural extensions of the framework. Section 5 concludes.

2 Background

We adopt the standard on-policy actor-critic setting [18, 19]. A behaviour policy π_θ collects a rollout of N trajectories of horizon H in a Markov decision process. For each sample (i, t) the rollout records a state $s_t^{(i)}$, an action $a_t^{(i)}$, a reward $r_t^{(i)}$, and an advantage estimate $\hat{A}_t^{(i)}$ produced by generalized advantage estimation [14]. An update step lifts the rollout policy π_θ to a new policy $\pi_{\theta'}$ by repeated stochastic gradient steps on a surrogate objective [9, 13]; the importance ratio $w_t^{(i)} = \pi_{\theta'}(a_t^{(i)} \mid s_t^{(i)})/\pi_\theta(a_t^{(i)} \mid s_t^{(i)})$ tracks how much $\pi_{\theta'}$ has moved away from π_θ on the sampled actions. The two surrogate objectives we study are the proximal-policy [15] pair: PPO-Clip and PPO-KL.

2.1 The surrogate objective

Let π_θ denote the policy parameterised by θ. PPO collects a rollout under the behaviour policy π_θ and updates the parameters to θ' by repeated stochastic gradient steps on a surrogate objective. The starting point is the importance sampled return [9, 13], written for a batch of N episodes of horizon H as

$$L_{\mathrm{IS}}(\theta') \;=\; \frac{1}{N}\sum_{i=1}^{N}\sum_{t=1}^{H} w_t^{(i)}\,\hat{A}_t^{(i)}, \qquad w_t^{(i)} \;=\; \frac{\pi_{\theta'}\Big(a_t^{(i)} \mid s_t^{(i)}\Big)}{\pi_\theta\Big(a_t^{(i)} \mid s_t^{(i)}\Big)},$$

where $\hat{A}_t^{(i)}$ is an estimator of the advantage at the transition $(s_t^{(i)}, a_t^{(i)})$. Direct optimisation of L_{IS} is unstable because the importance ratios $w_t^{(i)}$ can become large when θ' drifts away from θ. Schulman et al. [13] address the instability by constraining the KL divergence between $\pi_{\theta'}$ and π_θ. Schulman et al. [15] replace the constrained optimisation by one of two penalised first-order surrogates.

2.2 PPO-Clip

The clipped surrogate of Schulman et al. [15] is

$$L_{\mathrm{CLIP}}(\theta') \;=\; \frac{1}{N}\sum_{i=1}^{N}\sum_{t=1}^{H} \min\Big\{ w_t^{(i)}\,\hat{A}_t^{(i)},\;\; \mathrm{clip}\Big(w_t^{(i)},\,1-\epsilon,\,1+\epsilon\Big)\,\hat{A}_t^{(i)} \Big\}, \tag{1}$$

with $\epsilon \in (0, 1)$ a fixed hyperparameter (typically $\epsilon = 0.2$). The $\min$ acts as a one-sided trust region. When $\hat{A}_t^{(i)} > 0$, increasing $w_t^{(i)}$ beyond $1+\epsilon$ no longer increases the loss; when $\hat{A}_t^{(i)} < 0$, decreasing

$w_t^{(i)}$ below $1-\epsilon$ no longer increases the loss. This prevents the optimiser from exploiting large positive surrogate values that would correspond to large policy changes.

2.3 PPO-KL

The KL-penalised surrogate of Schulman et al. [15] adds an explicit divergence term,

$$\mathcal{L}_{\mathrm{KL}}(\theta') \;=\; L_{\mathrm{IS}}(\theta') \;-\; \beta\, \widehat{D}_{\mathrm{KL}}\big[\pi_\theta \,\|\, \pi_{\theta'}\big],$$

where $\widehat{D}_{\mathrm{KL}}$ is an empirical KL estimate and $\beta > 0$ is a scalar coefficient that is held fixed or adapted at the end of each outer update to keep the empirical KL near a target [5, 15]. In the fixed-β form, β is shared across samples and across training. In the adaptive form, β is multiplied or divided by a constant whenever the empirical KL exceeds or falls below the target.

In both forms, β is a single scalar applied uniformly to all samples in the batch. The comparison of clipping against scalar-KL penalisation has been studied extensively [1, 4, 6, 8, 15, 17]. The two surrogates are taken throughout this literature as algorithmically distinct; the gradient identity we establish in Section 3 shows that, at the per-sample level, they are not.

3 The per-sample KL view of PPO-Clip

3.1 Partition of the rollout

The gradient of L_{CLIP} depends on which of the two arguments of the inner $\min$ is active for each sample. Fix θ' and define the importance ratio $w_t^{(i)}$ as in Section 2. The pairs (i,t) partition into three disjoint index sets,

$$\begin{aligned}
\mathcal{I}_{\mathrm{in}} &= \big\{(i,t) \;:\; w_t^{(i)} \in [1-\epsilon,\, 1+\epsilon]\big\} \;\cup\; \big\{(i,t) \;:\; \hat{A}_t^{(i)} = 0\big\},\\
\mathcal{I}_{\mathrm{kill}} &= \big\{(i,t) \;:\; w_t^{(i)} > 1+\epsilon \text{ and } \hat{A}_t^{(i)} > 0\big\} \;\cup\; \big\{(i,t) \;:\; w_t^{(i)} < 1-\epsilon \text{ and } \hat{A}_t^{(i)} < 0\big\},\\
\mathcal{I}_{\mathrm{pass}} &= \big\{(i,t) \;:\; w_t^{(i)} > 1+\epsilon \text{ and } \hat{A}_t^{(i)} < 0\big\} \;\cup\; \big\{(i,t) \;:\; w_t^{(i)} < 1-\epsilon \text{ and } \hat{A}_t^{(i)} > 0\big\}.
\end{aligned}$$

Intuitively, $\mathcal{I}_{\mathrm{in}}$ contains the samples for which the clip is inactive (together with the zero-advantage samples, which contribute no gradient to either surrogate), $\mathcal{I}_{\mathrm{kill}}$ the samples for which the clip suppresses the gradient because the policy is already moving in the advantage-improving direction, and $\mathcal{I}_{\mathrm{pass}}$ the samples for which the clip leaves the unclipped term active because the policy is moving against the advantage.

3.2 Gradient of PPO-Clip

The gradient of a sum is the sum of gradients, and the gradient of the inner $\min$ on each sample depends on which of its two arguments is active.

Case $\mathcal{I}_{\mathrm{in}}$: when $w_t^{(i)} \in [1-\epsilon, 1+\epsilon]$, the clipping operator leaves $w_t^{(i)}$ unchanged, so $\mathrm{clip}(w_t^{(i)}, 1-\epsilon, 1+\epsilon) = w_t^{(i)}$ and both arguments of the $\min$ coincide:

$$\min\big\{w_t^{(i)}\hat{A}_t^{(i)},\;\; \mathrm{clip}(w_t^{(i)},\, 1-\epsilon,\, 1+\epsilon)\hat{A}_t^{(i)}\big\} \;=\; w_t^{(i)}\hat{A}_t^{(i)}.$$

Its gradient is $\hat{A}_t^{(i)}\,\nabla_{\theta'} w_t^{(i)}$.

Case $\mathcal{I}_{\mathrm{kill}}$: consider first the subcase $w_t^{(i)} > 1+\epsilon$ and $\hat{A}_t^{(i)} > 0$. The clipped value $\mathrm{clip}(w_t^{(i)}, 1-\epsilon, 1+\epsilon)$ saturates at $1+\epsilon$, so the clipped term equals $(1+\epsilon)\hat{A}_t^{(i)}$, while the unclipped term equals $w_t^{(i)}\hat{A}_t^{(i)}$. Because $\hat{A}_t^{(i)} > 0$ and $w_t^{(i)} > 1+\epsilon$, the clipped term is the smaller of the two and the $\min$ selects it. The clipped value is constant in θ', so its gradient vanishes. The symmetric subcase $w_t^{(i)} < 1-\epsilon$ and $\hat{A}_t^{(i)} < 0$ is analogous: the clipped term saturates at $(1-\epsilon)\hat{A}_t^{(i)}$, which is smaller (more negative) than the unclipped term $w_t^{(i)}\hat{A}_t^{(i)}$, and the gradient vanishes again. In both subcases the per-sample gradient is zero.

Case $\mathcal{I}_{\text{pass}}$: consider the subcase $w_t^{(i)} > 1+\epsilon$ and $\hat{A}_t^{(i)} < 0$. The clipped term equals $(1+\epsilon)\hat{A}_t^{(i)}$ and the unclipped term equals $w_t^{(i)}\hat{A}_t^{(i)}$. Because $\hat{A}_t^{(i)} < 0$ and $w_t^{(i)} > 1+\epsilon$, the unclipped term is more negative and the min selects it. The symmetric subcase $w_t^{(i)} < 1-\epsilon$ and $\hat{A}_t^{(i)} > 0$ is analogous. In both subcases the active term is the unclipped $w_t^{(i)}\hat{A}_t^{(i)}$ and its gradient is $\hat{A}_t^{(i)}\nabla_{\theta'} w_t^{(i)}$.

Combining the three cases,

$$\nabla_{\theta'} L_{\text{CLIP}} \;=\; \frac{1}{N} \sum_{(i,t)\,\in\,\mathcal{I}_{\text{in}}\,\cup\,\mathcal{I}_{\text{pass}}} \hat{A}_t^{(i)}\,\nabla_{\theta'} w_t^{(i)},$$

where samples in $\mathcal{I}_{\text{kill}}$ contribute zero. To make the dependence on the policy explicit, observe that $\pi_\theta(a_t^{(i)} \mid s_t^{(i)})$ does not depend on θ', so

$$\nabla_{\theta'} w_t^{(i)} \;=\; \nabla_{\theta'}\left[\frac{\pi_{\theta'}(a_t^{(i)} \mid s_t^{(i)})}{\pi_\theta(a_t^{(i)} \mid s_t^{(i)})}\right] \;=\; \frac{\nabla_{\theta'}\pi_{\theta'}\left(a_t^{(i)} \mid s_t^{(i)}\right)}{\pi_\theta\left(a_t^{(i)} \mid s_t^{(i)}\right)}.$$

Substituting yields

$$\nabla_{\theta'} L_{\text{CLIP}} \;=\; \frac{1}{N} \sum_{(i,t)\,\in\,\mathcal{I}_{\text{in}}\,\cup\,\mathcal{I}_{\text{pass}}} \hat{A}_t^{(i)}\,\frac{\nabla_{\theta'}\pi_{\theta'}\left(a_t^{(i)} \mid s_t^{(i)}\right)}{\pi_\theta\left(a_t^{(i)} \mid s_t^{(i)}\right)}. \tag{2}$$

Table 1 summarises the per-sample gradient contribution on each index set.

Index set	**Condition**	**Active term in** min	**Gradient**
$\mathcal{I}_{\text{in}}$	$w_t^{(i)} \in [1-\epsilon,\ 1+\epsilon]$ *or* $\hat{A}_t^{(i)} = 0$	$w_t^{(i)}\hat{A}_t^{(i)}$ (both equal)	$\hat{A}_t^{(i)}\nabla_{\theta'} w_t^{(i)}$
$\mathcal{I}_{\text{kill}}$	$w_t^{(i)} > 1+\epsilon,\ \hat{A}_t^{(i)} > 0$ *or* $w_t^{(i)} < 1-\epsilon,\ \hat{A}_t^{(i)} < 0$	clipped term (constant)	**0** (killed)
$\mathcal{I}_{\text{pass}}$	$w_t^{(i)} > 1+\epsilon,\ \hat{A}_t^{(i)} < 0$ *or* $w_t^{(i)} < 1-\epsilon,\ \hat{A}_t^{(i)} > 0$	unclipped term $w_t^{(i)}\hat{A}_t^{(i)}$	$\hat{A}_t^{(i)}\nabla_{\theta'} w_t^{(i)}$

Table 1: Per-sample contribution of the PPO-Clip objective on the three index sets. The gradient is killed only on $\mathcal{I}_{\text{kill}}$, where further policy change would push the importance ratio further outside the trust region.

3.3 Gradient of PPO-KL

The empirical KL estimate $\widehat{D}_{\text{KL}}\big[\pi_\theta \,\|\, \pi_{\theta'}\big]$ of Section 2 can be written, up to a θ'-independent additive constant that drops out of the gradient, as a sum of negative log-probabilities of the sampled actions under the new policy. After absorbing the constant and allowing the penalty coefficient to vary per sample, the surrogate becomes

$$\mathcal{L}_{\text{KL}}(\theta') \;=\; \frac{1}{N}\sum_{i=1}^{N}\sum_{t=1}^{H}\Big[w_t^{(i)}\,\hat{A}_t^{(i)} + \beta_t^{(i)}\,\log\pi_{\theta'}\left(a_t^{(i)} \mid s_t^{(i)}\right)\Big].$$

The standard fixed and adaptive PPO-KL variants of Schulman et al. [15] correspond to the choice $\beta_t^{(i)} \equiv \beta$, with β either fixed or updated between outer iterations.

For the importance-sampled return term,

$$\nabla_{\theta'}\left[w_t^{(i)}\,\hat{A}_t^{(i)}\right] \;=\; \hat{A}_t^{(i)}\,\nabla_{\theta'} w_t^{(i)} \;=\; \hat{A}_t^{(i)}\,\frac{\nabla_{\theta'}\pi_{\theta'}\left(a_t^{(i)} \mid s_t^{(i)}\right)}{\pi_\theta\left(a_t^{(i)} \mid s_t^{(i)}\right)},$$

using the same calculation as in the PPO-Clip derivation. For the penalty term, the coefficient $\beta_t^{(i)}$ is a *stop-gradient* (detached) coefficient: it is evaluated at the current θ' and held fixed when differentiating, as is standard for the penalty coefficient of a KL-penalised policy gradient and as the implementation does. Its derivative in θ' vanishes by convention, so

$$\nabla_{\theta'}\Big[\beta_t^{(i)} \log \pi_{\theta'}\Big(a_t^{(i)} \mid s_t^{(i)}\Big)\Big] \;=\; \beta_t^{(i)} \,\frac{\nabla_{\theta'}\pi_{\theta'}\Big(a_t^{(i)} \mid s_t^{(i)}\Big)}{\pi_{\theta'}\Big(a_t^{(i)} \mid s_t^{(i)}\Big)}.$$

Multiplying numerator and denominator of the right-hand side by $\pi_\theta(a_t^{(i)} \mid s_t^{(i)})$ converts the rollout-policy denominator to the same form as the importance-sampled term,

$$\beta_t^{(i)} \,\frac{\nabla_{\theta'}\pi_{\theta'}\Big(a_t^{(i)} \mid s_t^{(i)}\Big)}{\pi_{\theta'}\Big(a_t^{(i)} \mid s_t^{(i)}\Big)} \;=\; \beta_t^{(i)} \,\frac{\nabla_{\theta'}\pi_{\theta'}\Big(a_t^{(i)} \mid s_t^{(i)}\Big)}{\pi_{\theta}\Big(a_t^{(i)} \mid s_t^{(i)}\Big)} \cdot \frac{\pi_{\theta}\Big(a_t^{(i)} \mid s_t^{(i)}\Big)}{\pi_{\theta'}\Big(a_t^{(i)} \mid s_t^{(i)}\Big)} \;=\; \frac{\beta_t^{(i)}}{w_t^{(i)}} \,\frac{\nabla_{\theta'}\pi_{\theta'}\Big(a_t^{(i)} \mid s_t^{(i)}\Big)}{\pi_{\theta}\Big(a_t^{(i)} \mid s_t^{(i)}\Big)}.$$

Summing the two contributions and factoring the common $\nabla_{\theta'}\pi_{\theta'}\Big(a_t^{(i)} \mid s_t^{(i)}\Big)\,/\pi_\theta\Big(a_t^{(i)} \mid s_t^{(i)}\Big)$,

$$\nabla_{\theta'}\mathcal{L}_{\mathrm{KL}} \;=\; \frac{1}{N}\sum_{i=1}^{N}\sum_{t=1}^{H} \frac{\nabla_{\theta'}\pi_{\theta'}\Big(a_t^{(i)} \mid s_t^{(i)}\Big)}{\pi_{\theta}\Big(a_t^{(i)} \mid s_t^{(i)}\Big)} \left[\, \hat{A}_t^{(i)} \;+\; \frac{\beta_t^{(i)}}{w_t^{(i)}} \,\right]. \tag{3}$$

Every sample contributes to the gradient. The penalty does not zero out any term; it shifts the effective advantage of sample (i,t) from $\hat{A}_t^{(i)}$ to $\hat{A}_t^{(i)} + \beta_t^{(i)}/w_t^{(i)}$.

3.4 Gradient identity

Comparison of (2) and (3) yields the main result.

Theorem 1 (Per-sample gradient identity)**.** *Let L_{CLIP} be the PPO-Clip surrogate of Schulman et al. [15] and let $\mathcal{L}_{\mathrm{KL}}$ be the PPO-KL surrogate with per-sample stop-gradient coefficients $\{\beta_t^{(i)}\}$, each evaluated at the current θ' and held fixed under differentiation. Define*

$$\beta_t^{(i)} \;=\; \begin{cases} 0, & (i,t) \in \mathcal{I}_{\mathrm{in}} \cup \mathcal{I}_{\mathrm{pass}}, \\ -\,w_t^{(i)}\,\hat{A}_t^{(i)}, & (i,t) \in \mathcal{I}_{\mathrm{kill}}. \end{cases} \tag{4}$$

Then

$$\nabla_{\theta'} L_{\mathrm{CLIP}} \;=\; \nabla_{\theta'}\mathcal{L}_{\mathrm{KL}}$$

at every θ' where L_{CLIP} is differentiable, namely wherever no sample lies exactly on a clip boundary $w_t^{(i)} = 1 \pm \epsilon$; this excludes only a measure-zero set, on which L_{CLIP} has a kink.

Proof. Fix a sample (i,t) and write $g_t^{(i)}(\theta')$ for its per-sample gradient contribution under either surrogate. From (2) and (3),

$$g_t^{(i)}\big(L_{\mathrm{CLIP}}\big) \;=\; \begin{cases} \hat{A}_t^{(i)} \,\dfrac{\nabla_{\theta'}\pi_{\theta'}\Big(a_t^{(i)} \mid s_t^{(i)}\Big)}{\pi_{\theta}\Big(a_t^{(i)} \mid s_t^{(i)}\Big)}, & (i,t) \in \mathcal{I}_{\mathrm{in}} \cup \mathcal{I}_{\mathrm{pass}}, \\ 0, & (i,t) \in \mathcal{I}_{\mathrm{kill}}, \end{cases}$$

and

$$g_t^{(i)}\big(\mathcal{L}_{\mathrm{KL}}\big) \;=\; \frac{\nabla_{\theta'}\pi_{\theta'}\Big(a_t^{(i)} \mid s_t^{(i)}\Big)}{\pi_{\theta}\Big(a_t^{(i)} \mid s_t^{(i)}\Big)} \left[\, \hat{A}_t^{(i)} \;+\; \frac{\beta_t^{(i)}}{w_t^{(i)}} \,\right].$$

Under the choice (4), the bracket of $g_t^{(i)}(\mathcal{L}_{\mathrm{KL}})$ takes two values depending on the index set.

Case $(i,t) \in \mathcal{I}_{\mathrm{in}} \cup \mathcal{I}_{\mathrm{pass}}$. The definition (4) gives $\beta_t^{(i)} = 0$, so the bracket reduces to $\hat{A}_t^{(i)}$. Hence

$$g_t^{(i)}\big(\mathcal{L}_{\mathrm{KL}}\big) \;=\; \hat{A}_t^{(i)} \,\frac{\nabla_{\theta'}\pi_{\theta'}\Big(a_t^{(i)} \mid s_t^{(i)}\Big)}{\pi_{\theta}\Big(a_t^{(i)} \mid s_t^{(i)}\Big)} \;=\; g_t^{(i)}\big(L_{\mathrm{CLIP}}\big).$$

Case $(i,t) \in \mathcal{I}_{\text{kill}}$. The definition (4) gives $\beta_t^{(i)} = -w_t^{(i)} \hat{A}_t^{(i)}$. The bracket evaluates to

$$\hat{A}_t^{(i)} \;+\; \frac{\beta_t^{(i)}}{w_t^{(i)}} \;=\; \hat{A}_t^{(i)} \;+\; \frac{-w_t^{(i)} \hat{A}_t^{(i)}}{w_t^{(i)}} \;=\; \hat{A}_t^{(i)} - \hat{A}_t^{(i)} \;=\; 0,$$

where the cancellation uses $w_t^{(i)} > 0$ (since $\pi_{\theta'}\left(a_t^{(i)} \mid s_t^{(i)}\right) > 0$ and $\pi_{\theta}\left(a_t^{(i)} \mid s_t^{(i)}\right) > 0$ for any sampled action). Therefore

$$g_t^{(i)}\big(\mathcal{L}_{\text{KL}}\big) \;=\; 0 \;=\; g_t^{(i)}\big(L_{\text{CLIP}}\big).$$

The per-sample contributions agree on every (i,t), so the two batch gradients agree:

$$\nabla_{\theta'} L_{\text{CLIP}} \;=\; \frac{1}{N} \sum_{i=1}^{N} \sum_{t=1}^{H} g_t^{(i)}\big(L_{\text{CLIP}}\big) \;=\; \frac{1}{N} \sum_{i=1}^{N} \sum_{t=1}^{H} g_t^{(i)}\big(\mathcal{L}_{\text{KL}}\big) \;=\; \nabla_{\theta'} \mathcal{L}_{\text{KL}}.$$

□

3.5 Interpretation

Figure 1 reads the equivalence row by row: for each combination of $w_t^{(i)}$ and $\hat{A}_t^{(i)}$, the PPO-Clip gradient and the value of $\beta_t^{(i)}$ that reproduces it under the KL surrogate are listed side by side.

Region	**PPO-Clip gradient**	**Equivalent β_t**
$w_t^{(i)} \in [1-\epsilon,\, 1+\epsilon]$	$\hat{A}_t^{(i)} \nabla w_t^{(i)}$	$\beta_t = 0$
$w_t^{(i)} > 1+\epsilon,\, \hat{A}_t^{(i)} > 0$	0 (killed)	$\beta_t = -w_t^{(i)} \hat{A}_t^{(i)}$
$w_t^{(i)} > 1+\epsilon,\, \hat{A}_t^{(i)} < 0$	$\hat{A}_t^{(i)} \nabla w_t^{(i)}$	$\beta_t = 0$
$w_t^{(i)} < 1-\epsilon,\, \hat{A}_t^{(i)} > 0$	$\hat{A}_t^{(i)} \nabla w_t^{(i)}$	$\beta_t = 0$
$w_t^{(i)} < 1-\epsilon,\, \hat{A}_t^{(i)} < 0$	0 (killed)	$\beta_t = -w_t^{(i)} \hat{A}_t^{(i)}$

Figure 1: Summary of the per-sample equivalence. In the green rows the PPO-Clip gradient is reproduced by a PPO-KL surrogate with $\beta_t = 0$. In the red rows the PPO-Clip gradient is reproduced by a PPO-KL surrogate with $\beta_t = -w_t \hat{A}_t$, which is the value that kills the corresponding term of the per-sample gradient.

The coefficient (4) makes explicit what PPO-Clip implicitly applies. The clip is a Kullback–Leibler penalty whose strength is zero everywhere except on the killed region, where it takes the value $-w_t^{(i)} \hat{A}_t^{(i)}$. The sign of the coefficient is consistent with the trust-region intuition. When $\hat{A}_t^{(i)} > 0$ and $w_t^{(i)} > 1 + \epsilon$, the policy is already over-weighting a beneficial action, and a negative $\beta_t^{(i)}$ pulls $\log \pi_{\theta'}$ away from the current direction; when $\hat{A}_t^{(i)} < 0$ and $w_t^{(i)} < 1 - \epsilon$, the policy is already under-weighting a harmful action, and a positive $\beta_t^{(i)}$ stabilises it. In both cases the effective advantage of the bracket in (3) is driven exactly to zero, reproducing the clip's behaviour.

The identity in Theorem 1 is stronger than the first-order observation of Schulman et al. [15] that L_{CLIP} and the unclipped surrogate agree around $w = 1$. It holds at every θ' off the clip boundary, every minibatch step, and across the entire inner loop. Two closely related lines of recent work have approached the clip and the KL term from related but distinct angles: Liu et al. [10] establish a gradient equivalence between three KL estimators inside RLHF objectives, and Zhang et al. [24] propose a unified design framework for KL-regularised policy gradient. Neither identifies the per-sample coefficient (4) that turns PPO-Clip itself into a KL penalty.

4 Experiments

4.1 Setup

We evaluate four objectives that share the surrogate of Section 2 and differ only in the policy loss. These are PPO-Clip, fixed-β PPO-KL, adaptive-β PPO-KL, and the per-sample PPO-KL of (4); the first three are the variants of Schulman et al. [15] and the fourth is the construction of Section 3. We train each on the five MuJoCo locomotion tasks [20] HalfCheetah-v4, Hopper-v4, Walker2d-v4, Ant-v4, and Humanoid-v4 for 10^6 environment steps over five seeds, and include CartPole-v1 and LunarLander-v3 as a low-dimensional and a discrete-action check.

All four variants share the trainer, the rollout collector, and the value head, and use the standard PPO configuration of CleanRL [7] and Stable-Baselines3 [12]. This configuration is a two-hidden-layer (64-64, $\tanh$) actor-critic with orthogonal initialisation and a diagonal Gaussian policy, GAE [14] with $\gamma = 0.99$ and $\lambda = 0.95$, Adam at $3 \cdot 10^{-4}$ with linear annealing, gradient clipping at norm 0.5, value-loss clipping at 0.2, observation and reward normalisation, and an inner loop of $K = 10$ epochs over size-64 minibatches of each 2048-step rollout.

The variants differ in the trust-region knob. PPO-Clip uses $\epsilon = 0.2$, fixed-β PPO-KL uses $\beta = 1$, and adaptive-β PPO-KL follows Schulman et al. [15], updating β once per rollout by a factor of two toward a target $D_{\mathrm{KL}} = 0.02$, the second-order KL $\epsilon^2/2$ of the clip radius. The fixed and adaptive variants penalise the analytic Gaussian KL, while the per-sample variant penalises the sampled log-ratio $-\log w_t^{(i)}$, the estimator for which Theorem 1 holds. A knob sweep on CartPole-v1, HalfCheetah-v4, and Hopper-v4 over $\epsilon \in \{0.1, 0.2, 0.3\}$, $\beta \in \{0.1, 0.3, 1, 3, 10\}$, and $D_{\mathrm{KL}} \in \{0.003, 0.01, 0.02, 0.03, 0.1\}$ fixes each baseline at its best value.

All public run histories and per-run reproducibility artifacts are available in the Weights & Biases project at `https://wandb.ai/KLip-PPO/KLip-PPO`.

4.2 Results

By Theorem 1, PPO-Clip and the per-sample PPO-KL surrogate share a gradient at every step. Their learning curves coincide on all five MuJoCo tasks (Figure 2), and their final returns agree on every task (Table 2). Schulman et al. [15] showed only that the clipped objective and the unclipped surrogate ($\beta = 0$) agree to first order when the new policy is close to the one that collected the rollout. The per-sample identity is sharper, because with the coefficient β_t of (4) in place the equality becomes exact and holds at every θ', which is why the curves stay together over the whole run, however far training moves the policy.

The literature has consistently found clipping to outperform a KL penalty on continuous control. The original PPO study reports this on MuJoCo [15], and subsequent benchmarks repeat it [1, 4]. Our experiments reproduce the same ordering. PPO-KL with a fixed or an adaptively tuned β matches PPO-Clip on the easier tasks but falls behind on the high-dimensional ones (Ant-v4 and Humanoid-v4), where the policy must travel far from its initialisation and the trust region does real work.

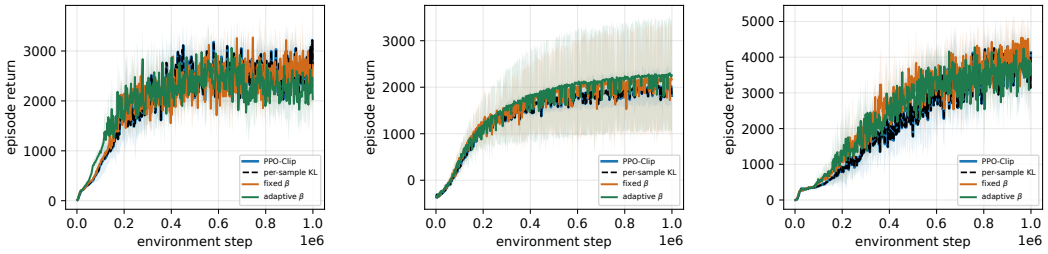
The per-sample identity explains the shortfall. Clipping constrains each sample on its own terms, turning the penalty on only for the transitions whose ratio has left the band and scaling it by that sample's ratio and advantage. The scalar β of PPO-KL, fixed or adaptive, instead applies one value to every sample, so a β large enough to restrain the few runaway transitions over-penalises the many well-behaved ones, and no single value reproduces what the clip does pointwise [6]. The per-sample coefficient of (4) removes that limitation, setting the penalty separately for each sample, and so reproduces the clip exactly.

5 Discussion

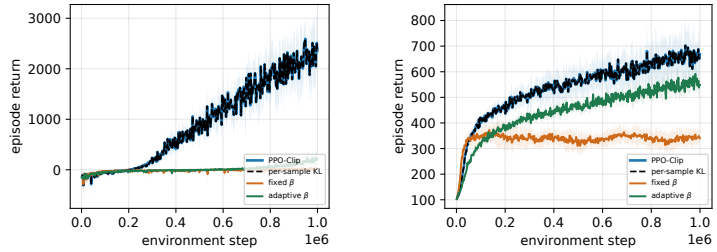
Theorem 1 makes PPO-Clip's per-sample gradient exactly the gradient of a Kullback–Leibler penalty,

$$\nabla_{\theta'} L_{\mathrm{CLIP}} = \nabla_{\theta'} \, \mathbb{E}_t\big[\, w_t \hat{A}_t + \beta_t \log \pi_{\theta'}(a_t \mid s_t) \,\big], \qquad \beta_t = -\, w_t \hat{A}_t \, \mathbf{1}[(i,t) \in \mathcal{I}_{\mathrm{kill}}] \,,$$

whose coefficient is set for each sample by its own importance ratio and advantage. The clip is in this sense a trust region acting in the space of policy distributions, applied to the samples in $\mathcal{I}_{\mathrm{kill}}$ that



(a) Hopper-v4 (b) HalfCheetah-v4 (c) Walker2d-v4



(d) Ant-v4 (e) Humanoid-v4

Figure 2: Episode return on the five MuJoCo tasks (mean over 5 seeds, $\pm$ std band). PPO-Clip and per-sample PPO-KL are indistinguishable on every task, while the fixed- and adaptive-β penalties fall behind on Ant-v4 (d) and Humanoid-v4 (e).

Task	PPO-Clip	Per-sample	Fixed β	Adaptive β
Ant	2193 ± 361	2193 ± 361	16 ± 16	159 ± 86
CartPole	478 ± 18	478 ± 18	484 ± 23	492 ± 9
HalfCheetah	1955 ± 312	1955 ± 312	2111 ± 1066	2207 ± 1117
Hopper	2598 ± 278	2598 ± 278	2616 ± 208	2236 ± 365
Humanoid	660 ± 70	660 ± 70	342 ± 31	553 ± 22
LunarLander	107 ± 8	107 ± 8	121 ± 12	125 ± 20
Walker2d	3717 ± 425	3717 ± 425	3995 ± 402	3639 ± 374

Table 2: Final return (mean $\pm$ std over 5 seeds, last 10% of training). PPO-Clip and per-sample PPO-KL are identical on every task; the scalar-β variants fall behind on the high-dimensional tasks.

it would otherwise discard. We show in Appendix A that the same identity has a matching form in importance-weight space, so the clipped objective (1), the weight-space penalty, and the per-sample KL penalty are **three equivalent expressions of a single per-sample gradient**.

This reading has two consequences. First, it makes precise a question the field has answered only empirically. Clipping and KL penalisation have been compared through benchmark scores [1, 4, 6, 8, 15, 17], with no theory relating the two objectives. The identity reframes that comparison. Since clipping is itself a KL penalty, the methods here differ only in how they set the penalty's coefficient. PPO-KL uses one scalar β for the whole batch, while PPO-Clip, by Theorem 1, uses the per-sample β_t of (4). The benchmark gap read as clipping versus KL is thus a gap between a scalar and a per-sample coefficient, which is where our experiments locate it. Second, it gives the penalty a **flexibility** the clipped form hides. Once β_t is written out, its step shape is one choice among many, and replacing it with another, such as one that softens the boundary or that varies with the individual sample, stays within the same surrogate family and defines new algorithms (Section 6).

What governs the update is therefore the per-sample gradient coefficient and not the surface choice between a clip and a KL term, a principle that recent analyses of KL regularisation in language-model training independently corroborate. Liu et al. [10] find that a KL term written as a loss and its associated per-sample reward coefficient induce the same gradient, and Zhang et al. [24] report that clipped and KL objectives agree once their per-sample importance weights are aligned. Neither

isolates the exact coefficient $\beta_t = -w_t \hat{A}_t$ that makes PPO-Clip itself a KL penalty, which is the identity established here; that their gradient-level findings point the same way is evidence that the per-sample view this identity rests on is the correct one.

6 Future work

The per-sample reformulation turns the implicit penalty coefficient $\beta_t^{(i)}$ of PPO-Clip into an explicit object. Its functional form, a step function on the trust-region boundary in the present case, can be modified by design without leaving the surrogate-loss family of Section 2. We outline five extensions that follow from making different choices for $\beta_t^{(i)}$. Each defines a new policy-optimisation algorithm and is left as the subject of a separate study.

Soft relaxations of the boundary. The coefficient defined in (4) is discontinuous at $w_t^{(i)} = 1 \pm \epsilon$. Several proposals in the literature soften this discontinuity from different angles. Trust Region-Guided PPO [21] keeps the hard clip but lets its width depend on the local KL of the policy. Truly PPO [22] keeps the clip and adds a rollback term that drags the policy back when the ratio exits the trust region. ESPO [17] removes ratio clipping altogether and controls the inner loop by early stopping. Simple Policy Optimization [23] substitutes clipping with a regulariser on the ratio that admits a tighter trust region. Probability Smoothing Policy Optimisation [3], in the language-model setting, replaces the hard ratio with a soft mixture of old and new policies so that the gradient is non-zero everywhere.

The per-sample form unifies these proposals under one quantity: each is a choice of $\beta_t^{(i)}$ in the template $\mathbb{E}_t[\, w_t \hat{A}_t + \beta_t \log \pi_{\theta'}(a_t \mid s_t)\,]$, with the soft variants replacing the step of (4) by a continuous shape that agrees with it far inside and far outside the trust region. The linear ramp of width $\delta \geq 0$, which interpolates between 0 and $-w_t^{(i)} \hat{A}_t^{(i)}$ across the boundary, is one explicit member: it recovers the PPO-Clip step as $\delta \to 0$ and the unconstrained surrogate as $\delta \to \infty$. The right shape, as a function of the task and of the inner-loop epoch count K, is left as an empirical question.

Position-aware coefficient for sequence models. On token-level applications such as language-model fine-tuning, each sample corresponds to one token of a generated completion. The per-sample coefficient $\beta_t^{(i)}$ can then be allowed to depend on the token's position within the sequence. A coefficient that is sharper near the answer span and softer in the reasoning prefix, for instance, is a specific position-conditioned $\beta_t^{(i)}$ and is not expressible in standard PPO-Clip, which uses the same trust-region radius for every token. The construction and empirical study of position-aware variants is a direct use of the framework and is left to follow-up work.

Age-conditioned coefficient for off-policy learning. The trust-region argument behind PPO-Clip assumes that the rollout was sampled under a behaviour policy close to the current one. With a replay buffer this assumption fails. The importance ratio $w_t^{(i)}$ on a sample drawn from an older policy can be arbitrarily large, and PPO-Clip discards all such samples by the step coefficient. Letting $\beta_t^{(i)}$ depend on the age of the sample at the time of the update, so that older samples are weighted differently from fresh ones, defines an off-policy variant of the algorithm whose properties can be analysed within the same formulation. The appropriate functional form of the age dependence, and its interaction with the bias-variance tradeoff of off-policy estimation, is an open question.

Asymmetric trust regions. The coefficient (4) is symmetric under the swap $(w_t^{(i)} - 1) \mapsto -(w_t^{(i)} - 1)$, in the sense that the same step shape applies on both sides of the trust region. An asymmetric variant softens the coefficient on the side that pulls the policy back toward the rollout distribution and keeps it sharp on the side that pushes it away. This modification is a single change to the per-sample form and is not expressible inside the original $\min$ formulation of Schulman et al. [15]. Its analysis is a natural follow-up.

A unified per-sample template. The per-sample form is not specific to PPO-Clip. The template

$$\mathcal{L}_{\mathrm{tmpl}}(\theta') \;=\; \mathbb{E}_t\big[\, w_t \hat{A}_t + \beta_t \log \pi_{\theta'}(a_t \mid s_t)\,\big] \tag{5}$$

recovers several existing on-policy algorithms once the dependence of $\beta_t^{(i)}$ on the sample is specified, and Table 3 summarises this view. The unconstrained importance-sampled surrogate is the case $\beta_t \equiv 0$. PPO-KL with a fixed scalar [15] corresponds to $\beta_t \equiv \beta \in \mathbb{R}$. Adaptive PPO-KL [15] replaces the constant by a quantity $\beta(t)$ that is updated once per rollout according to the measured KL. PPO-Clip is the per-sample step identified in Theorem 1. Token-level PPO-Clip and GRPO [2, 16] apply the same step independently to each token of a generated sequence. The directions in this section correspond to making different choices of $\beta_t^{(i)}$ within the same template.

Algorithm	$\boldsymbol{\beta_t^{(i)}}$ **form**	**Depends on**	**Reference**
Unconstrained surrogate	0	—	—
PPO-KL (fixed)	$\beta \in \mathbb{R}$	none	Schulman et al. [15]
PPO-KL (adaptive)	$\beta(t) \in \mathbb{R}$	training time	Schulman et al. [15]
PPO-Clip	$-w_t \hat{A}_t \cdot \mathbf{1}_{\mathcal{I}_{\text{kill}}}$	$(w, \hat{A})$	Schulman et al. [15]
Soft-clip (linear ramp)	$-w_t \hat{A}_t \cdot g_\delta(w, \hat{A})$	$(w, \hat{A})$, δ	this paper, Sec. 6
Token-level / GRPO	step shape per token	token $(w, \hat{A})$	DeepSeek-AI [2], Shao et al. [16]
Position-aware	per-token shape, conditioned on position	position $+(w, \hat{A})$	future
Off-policy	per-sample shape, conditioned on age	age $+(w, \hat{A})$	future
Asymmetric	non-symmetric in $\operatorname{sign}(w-1)$	$(w, \hat{A})$	future

Table 3: Instances of the per-sample template (5). Each row is a particular choice of the dependence of $\beta_t^{(i)}$ on the sample; the rest of the loss is shared.

Several research questions follow from the per-sample template. The published algorithms in the upper part of the table can be trained under identical pipelines and compared on the same metrics, which isolates the effect of the shape of $\beta_t^{(i)}$ from the implementation choices that the literature has shown matter strongly [1, 4]. The soft, position-aware, age-conditioned and asymmetric directions described in the previous paragraphs enrich the arguments of $\beta_t^{(i)}$, and the corresponding algorithms can be implemented and evaluated inside the same pipeline. On the theory side, the boundedness and monotone-improvement guarantees available for fixed and scheduled scalars in the original PPO analysis [9, 13] have direct analogues at the per-sample level, and characterising which shapes of $\beta_t^{(i)}$ preserve them is an open problem. The template also crosses domains: the same form covers MuJoCo locomotion, language-model fine-tuning [11, 16] and off-policy regimes, and $\beta_t^{(i)}$ is the shared design variable across the three.

References

[1] Marcin Andrychowicz, Anton Raichuk, Piotr Stańczyk, Manu Orsini, Sertan Girgin, Raphael Marinier, Léonard Hussenot, Matthieu Geist, Olivier Pietquin, Marcin Michalski, Sylvain Gelly, and Olivier Bachem. What matters in on-policy reinforcement learning? a large-scale empirical study. In *International Conference on Learning Representations (ICLR)*, 2021.

[2] DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

[3] Madeleine Dwyer, Adam Sobey, and Adriane Chapman. It's not you, it's clipping: A soft trust-region via probability smoothing for LLM RL. *arXiv preprint arXiv:2509.21282*, 2025.

[4] Logan Engstrom, Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Firdaus Janoos, Larry Rudolph, and Aleksander Madry. Implementation matters in deep RL: A case study on PPO and TRPO. In *International Conference on Learning Representations (ICLR)*, 2020.

[5] Nicolas Heess, Dhruva TB, Srinivasan Sriram, Jay Lemmon, Josh Merel, Greg Wayne, Yuval Tassa, Tom Erez, Ziyu Wang, S. M. Ali Eslami, Martin Riedmiller, and David Silver. Emergence of locomotion behaviours in rich environments. *arXiv preprint arXiv:1707.02286*, 2017.

[6] Chloe Ching-Yun Hsu, Celestine Mendler-Dünner, and Moritz Hardt. Revisiting design choices in proximal policy optimization. *arXiv preprint arXiv:2009.10897*, 2020.

[7] Shengyi Huang, Rousslan Fernand Julien Dossa, Chang Ye, Jeff Braga, Dipam Chakraborty, Kinal Mehta, and João G. M. Araújo. CleanRL: High-quality single-file implementations of deep reinforcement learning algorithms. *Journal of Machine Learning Research*, 23(274):1–18, 2022.

[8] Andrew Ilyas, Logan Engstrom, Shibani Santurkar, Dimitris Tsipras, Firdaus Janoos, Larry Rudolph, and Aleksander Madry. A closer look at deep policy gradients. In *International Conference on Learning Representations (ICLR)*, 2020.

[9] Sham Kakade and John Langford. Approximately optimal approximate reinforcement learning. In *Proceedings of the 19th International Conference on Machine Learning (ICML)*, pages 267–274, 2002.

[10] Kezhao Liu, Jason Klein Liu, Mingtao Chen, and Yiming Liu. Rethinking KL regularization in RLHF: From value estimation to gradient optimization. *arXiv preprint arXiv:2510.01555*, 2025.

[11] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022.

[12] Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22(268):1–8, 2021.

[13] John Schulman, Sergey Levine, Pieter Abbeel, Michael I. Jordan, and Philipp Moritz. Trust region policy optimization. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, pages 1889–1897, 2015.

[14] John Schulman, Philipp Moritz, Sergey Levine, Michael I. Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. In *International Conference on Learning Representations (ICLR)*, 2016.

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

[16] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

[17] Mingfei Sun, Vitaly Kurin, Guoqing Liu, Sam Devlin, Tao Qin, Katja Hofmann, and Shimon Whiteson. You may not need ratio clipping in PPO. *arXiv preprint arXiv:2202.00079*, 2022.

[18] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 2nd edition, 2018.

[19] Richard S. Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems (NIPS)*, 2000.

[20] 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)*, pages 5026–5033, 2012.

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

[22] Yuhui Wang, Hao He, and Xiaoyang Tan. Truly proximal policy optimization. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2020.

[23] Zhengpeng Xie, Qiang Zhang, Fan Yang, Marco Hutter, and Renjing Xu. Simple policy optimization. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, pages 68813–68824, 2025.

[24] Yifan Zhang, Yifeng Liu, Huizhuo Yuan, Yang Yuan, Quanquan Gu, and Andrew Chi-Chih Yao. On the design of KL-regularized policy gradient algorithms for LLM reasoning. In *International Conference on Learning Representations (ICLR)*, 2026.

A Weight-space dual: PPO-Clip as a Φ penalty

The per-sample β_t identity of the main text places PPO-Clip in distribution space: the penalty is $\beta_t \log \pi_{\theta'}(a_t \mid s_t)$, a per-sample KL contribution. PPO-Clip also admits a dual formulation in weight space, where the penalty acts directly on the deviation of w_t from the trust region $[1-\epsilon,\, 1+\epsilon]$.

Theorem 2 (Weight-space form of PPO-Clip)**.** *The PPO-Clip objective can be written as*

$$L_{\mathrm{CLIP}}(\theta') \;=\; \frac{1}{N}\sum_{i=1}^{N}\sum_{t=1}^{H}\Big[\, w_t^{(i)}\hat{A}_t^{(i)} \;-\; \Phi\Big(w_t^{(i)}, \hat{A}_t^{(i)}\Big)\Big]$$

with the per-sample weight-space penalty

$$\Phi(w,\hat{A}) \;=\; \begin{cases} \big(w-(1+\epsilon)\big)\,\hat{A} & \text{if } w > 1+\epsilon \text{ and } \hat{A} > 0, \\ \big(w-(1-\epsilon)\big)\,\hat{A} & \text{if } w < 1-\epsilon \text{ and } \hat{A} < 0, \\ 0 & \text{otherwise.} \end{cases}$$

Proof. Fix a sample (i,t) and write $\ell_t^{(i)} = \min\big(w_t^{(i)}\hat{A}_t^{(i)},\, \mathrm{clip}(w_t^{(i)})\hat{A}_t^{(i)}\big)$ for its PPO-Clip per-sample contribution.

Case $w_t^{(i)} > 1+\epsilon$ and $\hat{A}_t^{(i)} > 0$. The clipped weight is $1+\epsilon$ and the clipped product is smaller than the unclipped one, so

$$\ell_t^{(i)} \;=\; (1+\epsilon)\hat{A}_t^{(i)} \;=\; w_t^{(i)}\hat{A}_t^{(i)} - \big(w_t^{(i)} - (1+\epsilon)\big)\hat{A}_t^{(i)} \;=\; w_t^{(i)}\hat{A}_t^{(i)} - \Phi(w_t^{(i)}, \hat{A}_t^{(i)}).$$

Case $w_t^{(i)} < 1-\epsilon$ and $\hat{A}_t^{(i)} < 0$. The clipped weight is $1-\epsilon$ and again the clipped product is the smaller, so

$$\ell_t^{(i)} \;=\; (1-\epsilon)\hat{A}_t^{(i)} \;=\; w_t^{(i)}\hat{A}_t^{(i)} - \big(w_t^{(i)} - (1-\epsilon)\big)\hat{A}_t^{(i)} \;=\; w_t^{(i)}\hat{A}_t^{(i)} - \Phi(w_t^{(i)}, \hat{A}_t^{(i)}).$$

Otherwise. The unclipped term is the minimum, so $\ell_t^{(i)} = w_t^{(i)}\hat{A}_t^{(i)} = w_t^{(i)}\hat{A}_t^{(i)} - 0$, and the definition of Φ gives $\Phi(w_t^{(i)}, \hat{A}_t^{(i)}) = 0$.

Summing over (i,t) yields the result. □

The penalty Φ is non-negative on $\mathcal{I}_{\mathrm{kill}}$: when $w > 1+\epsilon$ and $\hat{A} > 0$ the factor $w-(1+\epsilon)$ is positive and so is $\hat{A}$; when $w < 1-\epsilon$ and $\hat{A} < 0$ both factors are negative and their product is positive again. PPO-Clip therefore subtracts a non-negative penalty from the unconstrained surrogate, proportional to how far $w_t^{(i)}$ exceeds the trust-region boundary, on exactly the samples in $\mathcal{I}_{\mathrm{kill}}$.

Combining the weight-space identity above with the per-sample β_t gradient identity of the main text, PPO-Clip admits three forms on every minibatch, the first two equal in value and the third equal in gradient:

1. the $\min$ formulation of Schulman et al. [15]:
$$L_{\mathrm{CLIP}} \;=\; \mathbb{E}_t\big[\min\big(w_t\hat{A}_t,\, \mathrm{clip}(w_t)\hat{A}_t\big)\big];$$
2. the weight-space form of Theorem 2:
$$L_{\mathrm{CLIP}} \;=\; \mathbb{E}_t\big[\, w_t\hat{A}_t \;-\; \Phi(w_t, \hat{A}_t)\big],$$
with penalty acting on $|w_t - 1|$ in importance-weight space;
3. the per-sample KL form of the main theorem, which matches PPO-Clip in gradient:
$$\nabla_{\theta'} L_{\mathrm{CLIP}} \;=\; \nabla_{\theta'}\,\mathbb{E}_t\big[\, w_t\hat{A}_t \;+\; \beta_t \log \pi_{\theta'}(a_t \mid s_t)\big], \qquad \beta_t = -w_t\hat{A}_t \cdot \mathbf{1}[(i,t) \in \mathcal{I}_{\mathrm{kill}}]\,,$$
with penalty acting in distribution space.

The first two forms are equal as functions and differ only in surface notation; the third matches them in per-sample gradient and places PPO-Clip inside the PPO-KL family. The space in which the trust region is expressed (importance-weight space for Φ, distribution space for β_t) changes the notation but not the per-sample gradient, which is the same in all three forms.

Form	Per-sample loss term	Penalty acts in	Reference
min	$\min\big(w_t\hat{A}_t,\ \mathrm{clip}(w_t)\hat{A}_t\big)$	—	[15]
Φ	$w_t\hat{A}_t - \Phi(w_t, \hat{A}_t)$	importance-weight space	Theorem 2
β_t	$w_t\hat{A}_t + \beta_t \log \pi_{\theta'}(a_t \mid s_t)$	distribution space	main theorem

Table 4: Three forms of the PPO-Clip per-sample loss. All three produce the same per-sample gradient on every (i, t).

B Supplementary Figures

This appendix collects supplementary figures that illustrate the geometry of PPO-Clip and PPO-KL and the per-sample equivalence between them. The notation follows the main text, with w_t the importance ratio, $\hat{A}_t$ the GAE advantage estimate, and $\mathcal{I}_{\text{in}}, \mathcal{I}_{\text{kill}}, \mathcal{I}_{\text{pass}}$ the partition of the minibatch.

B.1 The clipping function

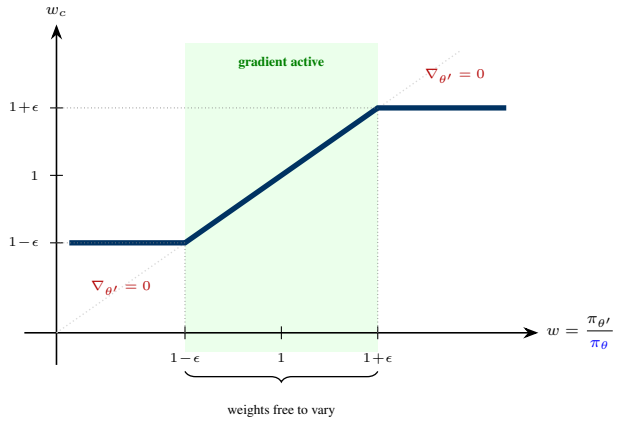


Figure 3: The clipping function $w_c = \text{clip}(w, 1-\epsilon, 1+\epsilon)$. Inside the band $[1-\epsilon, 1+\epsilon]$ the clipped weight equals the true importance ratio and the gradient flows normally; outside the band the clipped weight is constant and the gradient with respect to θ' vanishes.

B.2 The PPO-Clip surrogate

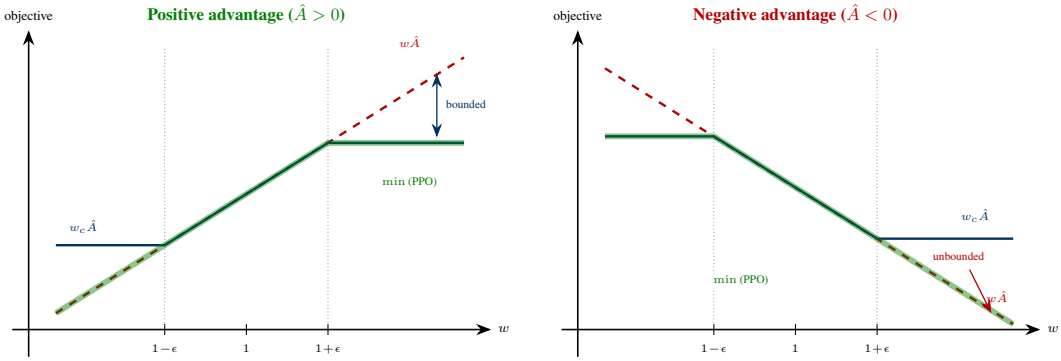


Figure 4: The PPO-Clip surrogate, split by the sign of $\hat{A}$. Left panel ($\hat{A} > 0$): for $w > 1+\epsilon$ the minimum follows the clipped term, bounding the upside. Right panel ($\hat{A} < 0$): for $w > 1+\epsilon$ the minimum follows the unclipped term, so the penalty grows without bound. The asymmetry between the two panels is what makes PPO-Clip a one-sided trust region.

B.3 PPO-Clip flowchart

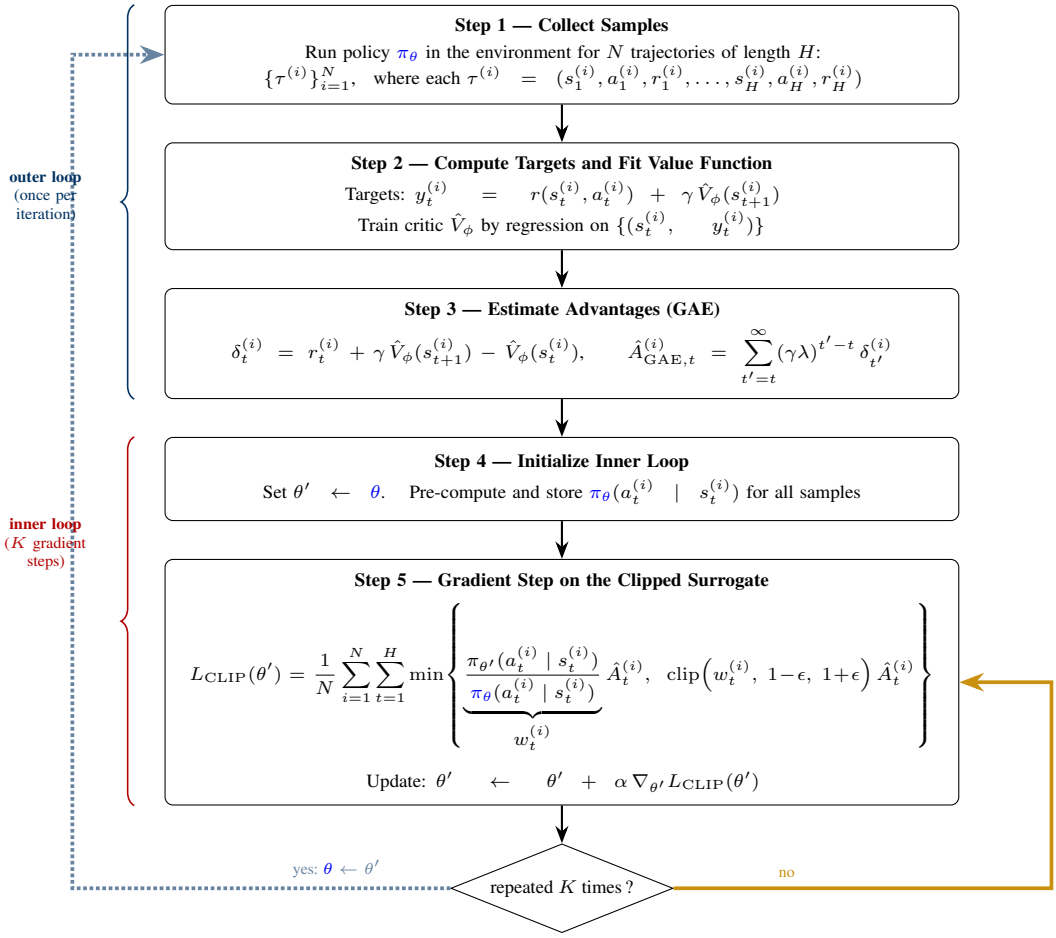


Figure 5: PPO-Clip. The outer loop collects rollouts and fits the critic; the inner loop takes K gradient steps on the clipped surrogate L_{CLIP} before refreshing the behaviour policy.

B.4 PPO-KL flowchart

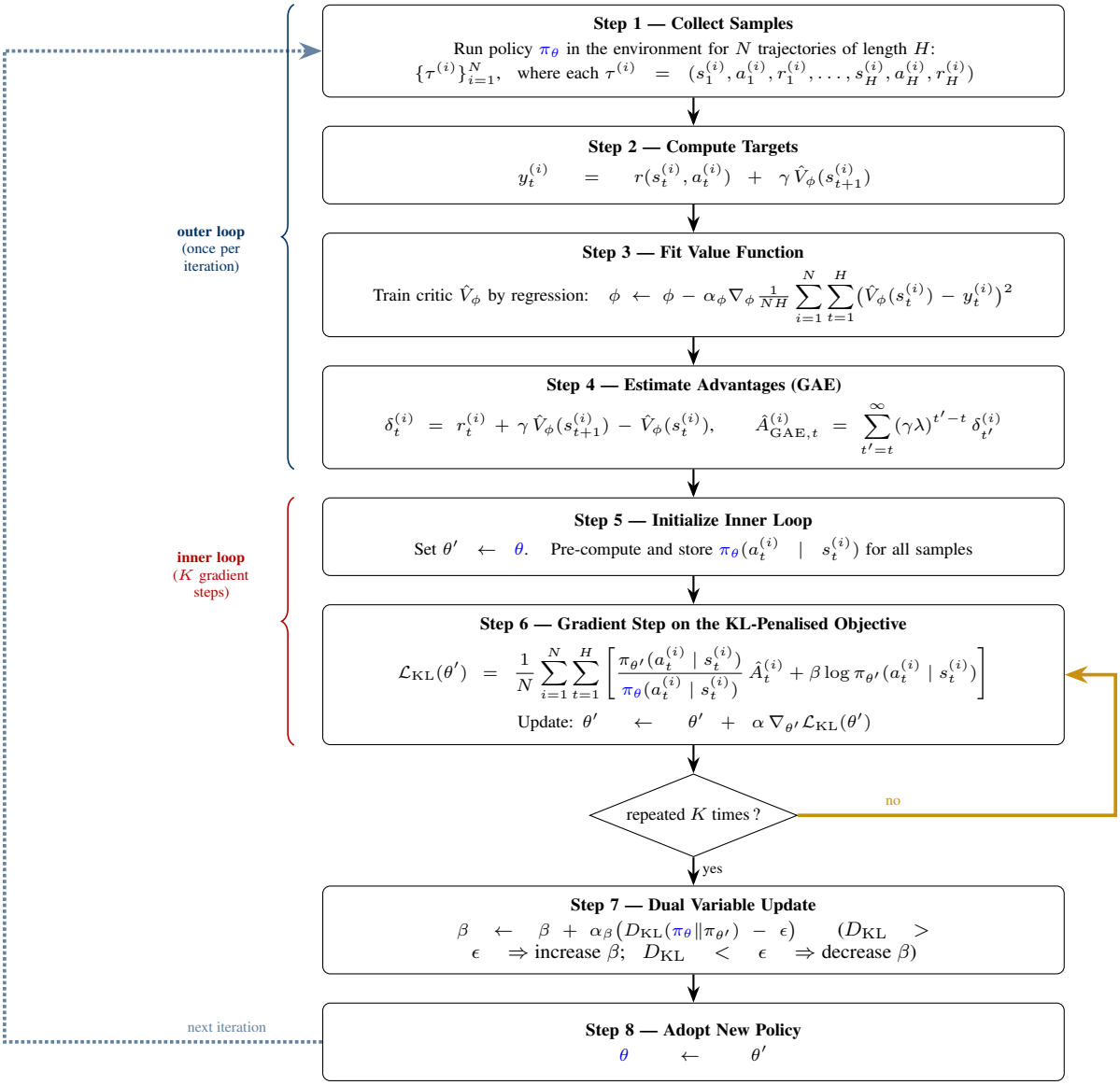


Figure 6: PPO-KL. The objective is the unclipped importance-weighted advantage plus a KL penalty with a scalar coefficient β; after the inner loop β is adjusted by a dual gradient step that targets a desired KL.

B.5 Effective gradient multiplier as a function of w

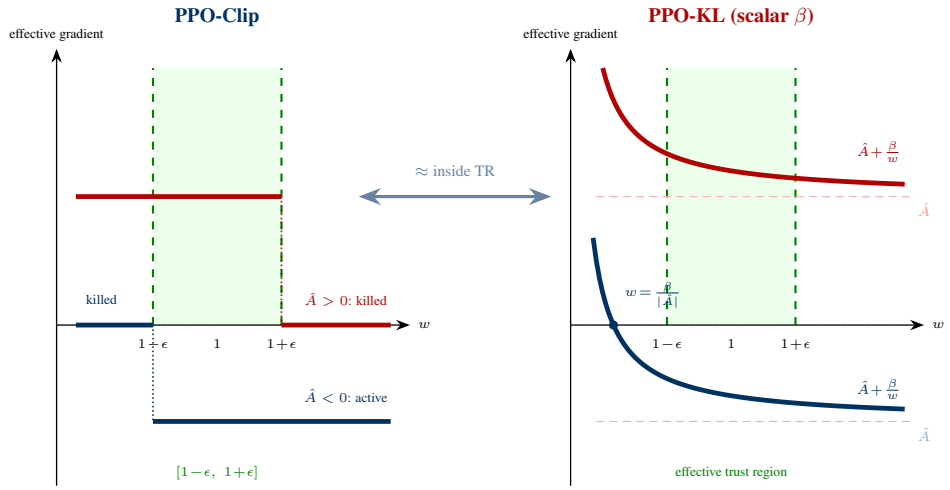


Figure 7: Per-sample effective gradient multiplier as a function of the importance ratio w. Left: PPO-Clip is a hard step. The multiplier equals $\hat{A}$ inside $[1-\epsilon, 1+\epsilon]$ and on $\mathcal{I}_{\text{pass}}$, and zero on $\mathcal{I}_{\text{kill}}$. Right: a scalar PPO-KL produces the smooth curve $\hat{A}+\beta/w$ (the asymptote is $\hat{A}$). The two coincide inside the trust region; outside, PPO-Clip is a step function while scalar PPO-KL is a smooth one. The per-sample β_t construction of the main text is the choice that recovers the step function.

B.6 Morphing from scalar to per-sample β

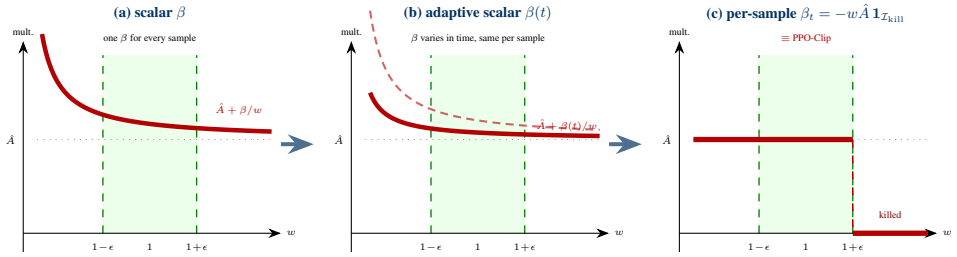


Figure 8: Morphing from scalar to per-sample β, shown for the case $\hat{A} > 0$. (a) A constant scalar β produces the smooth curve $\hat{A}+\beta/w$, which can never coincide with the PPO-Clip step. (b) Letting β adapt in time but stay scalar shifts the curve along its family but cannot reshape it. (c) Letting β become per-sample, with $\beta_t = -w_t\hat{A}_t$ exactly on $\mathcal{I}_{\text{kill}}$ and zero elsewhere, snaps the curve onto the PPO-Clip step function. The clip is the per-sample limit of the KL surrogate; scalar penalties never reach it.

B.7 The β_t map in the $(w, \hat{A})$ plane

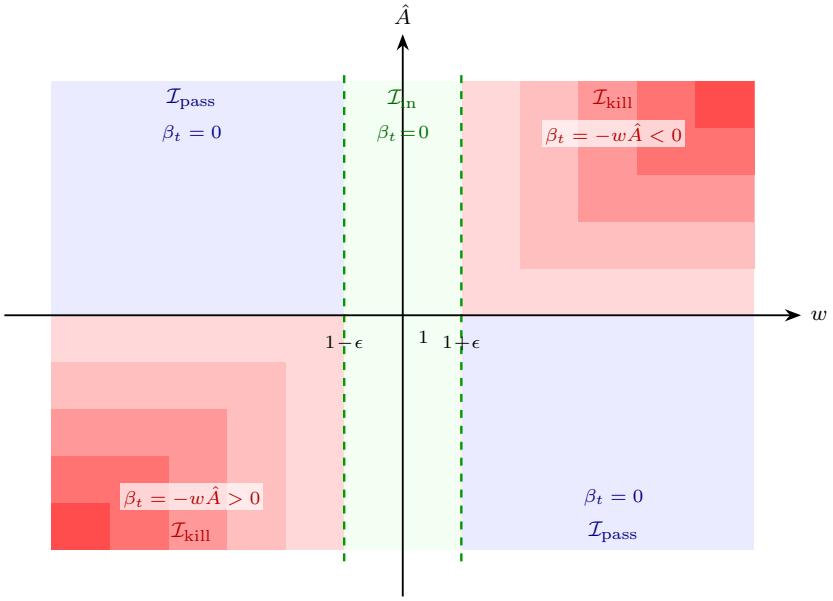


Figure 9: The β_t landscape in $(w, \hat{A})$ space. Vertical dashed lines mark the trust region $[1-\epsilon, 1+\epsilon]$. Two corners (top-right $w > 1+\epsilon, \hat{A} > 0$ and bottom-left $w < 1-\epsilon, \hat{A} < 0$) carry the per-sample coefficient $\beta_t = -w\hat{A}$, with shading indicating $|\beta_t|$. Everywhere else $\beta_t = 0$. The support and value of β_t reproduce the PPO-Clip gradient sample by sample.

B.8 The same per-sample gradient via two surrogates

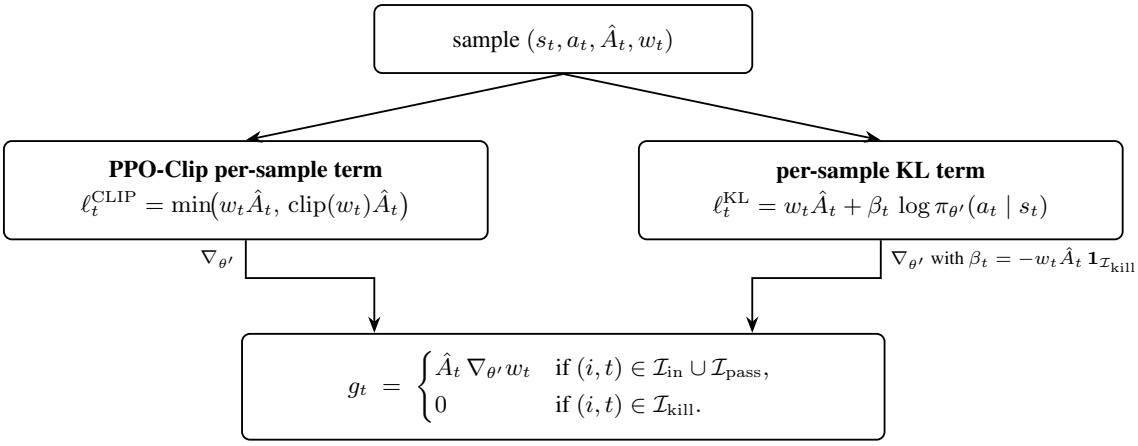
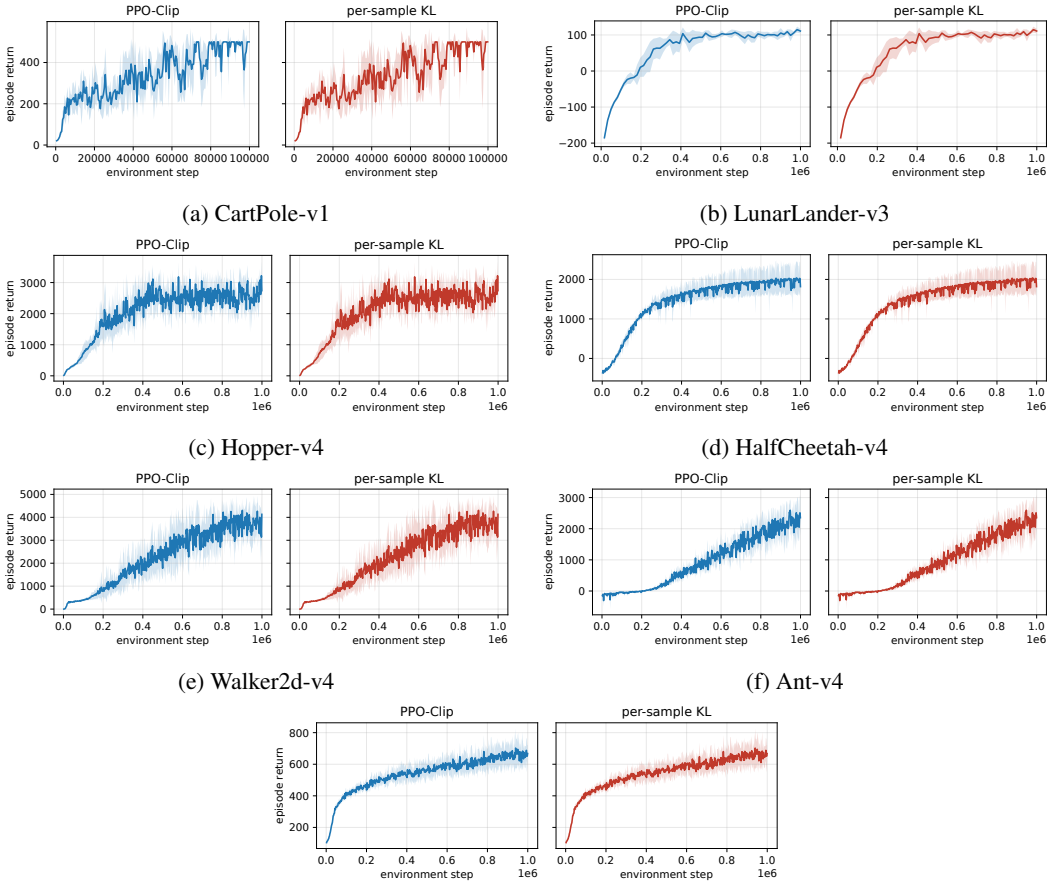


Figure 10: The per-sample gradient under the two surrogates. The same sample feeds both per-sample terms; differentiating each in θ' yields the same g_t. PPO-Clip selects the active branch of the min to kill the gradient on $\mathcal{I}_{\text{kill}}$; the per-sample KL surrogate sets $\beta_t = -w_t\hat{A}_t$ on $\mathcal{I}_{\text{kill}}$ and 0 elsewhere, which makes the bracket $\hat{A}_t + \beta_t/w_t$ vanish on those samples.

C Supplementary experimental results

C.1 Per-task equivalence

Figure 11 reports PPO-Clip and the per-sample PPO-KL surrogate on each of the seven tasks separately, namely CartPole-v1, LunarLander-v3, Hopper-v4, HalfCheetah-v4, Walker2d-v4, Ant-v4, and Humanoid-v4. The two are indistinguishable on every environment. Each curve is a mean over 5 seeds with a $\pm$ std band.



(a) CartPole-v1

(b) LunarLander-v3

(c) Hopper-v4

(d) HalfCheetah-v4

(e) Walker2d-v4

(f) Ant-v4

(g) Humanoid-v4

Figure 11: PPO-Clip and per-sample PPO-KL on each task, mean over 5 seeds with a $\pm$ std band. The two coincide on every environment.

C.2 Trust-region knob sweeps

To place each scalar-β baseline at a defensible operating point we sweep the trust-region knob of each variant; the main-text baselines use the best value of each knob.

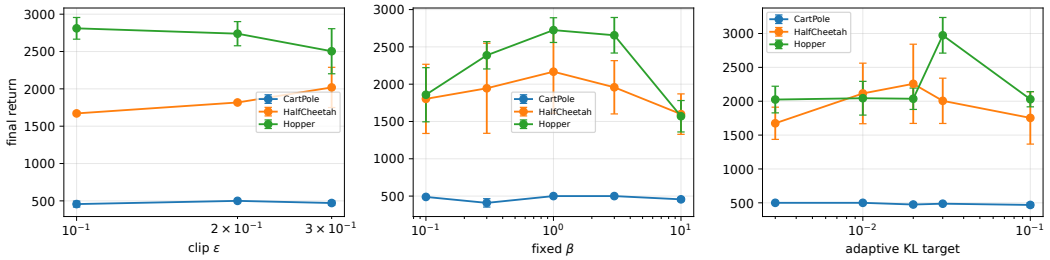
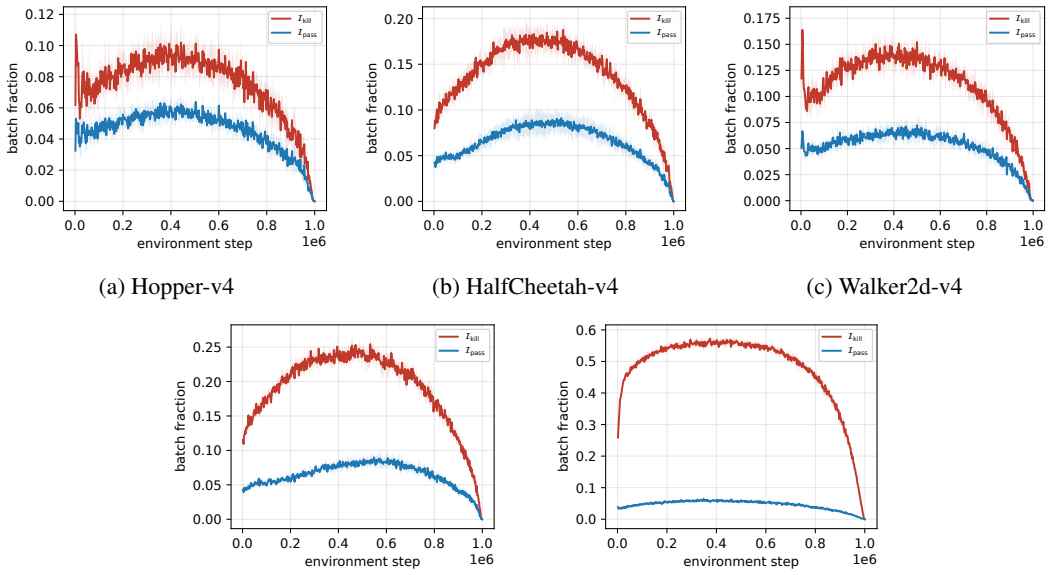


Figure 12: Final return (mean $\pm$ standard error over 5 seeds) on CartPole-v1, HalfCheetah-v4, and Hopper-v4 as a function of each trust-region knob: clip ϵ for PPO-Clip, fixed β for PPO-KL, and the KL target for adaptive PPO-KL.

C.3 Clipping partition

Figure 13 shows, for PPO-Clip, the fraction of each minibatch that falls in $\mathcal{I}_{\text{kill}}$ and $\mathcal{I}_{\text{pass}}$ over training. This is the empirical view of the partition on which the identity rests: the penalty β_t is non-zero only on $\mathcal{I}_{\text{kill}}$, and its reach grows with task difficulty, exceeding half the batch on Humanoid-v4.



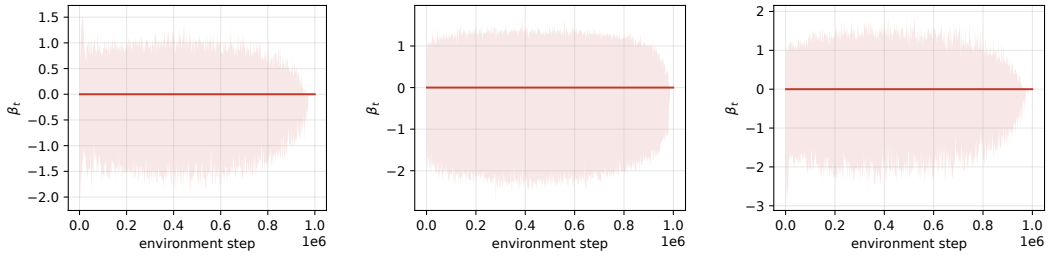
(a) Hopper-v4 (b) HalfCheetah-v4 (c) Walker2d-v4

(d) Ant-v4 (e) Humanoid-v4

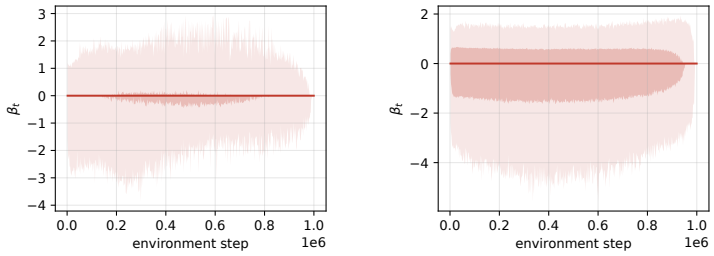
Figure 13: Fraction of the PPO-Clip minibatch in $\mathcal{I}_{\text{kill}}$ and $\mathcal{I}_{\text{pass}}$ over training (mean over 5 seeds, $\pm$ std band). The per-sample coefficient β_t acts only on $\mathcal{I}_{\text{kill}}$.

C.4 Per-sample coefficient

Figure 14 plots the per-sample coefficient β_t over training for the per-sample variant. Its median is zero, since most samples lie outside $\mathcal{I}_{\text{kill}}$, while the tails carry the active penalty $-w_t\hat{A}_t$ and widen on the harder tasks.



(a) Hopper-v4 (b) HalfCheetah-v4 (c) Walker2d-v4



(d) Ant-v4 (e) Humanoid-v4

Figure 14: Per-sample coefficient β_t over training for the per-sample variant. The median is zero because β_t vanishes on every sample outside the kill region; the shaded bands show how far the active coefficient $-w_t\hat{A}_t$ reaches on the samples in $\mathcal{I}_{\text{kill}}$, widening on the high-dimensional tasks.