GA^{2}E: GRADIENT-ALIGNED GENERALIZED ADVANTAGE ESTIMATION FOR ADAPTIVE λ SELECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Generalized Advantage Estimation (GAE) is a widely used advantage estimation method in reinforcement learning, with its performance substantially influenced by the trace parameter λ. Since the effective choice of λ can vary across tasks and stages of training, existing policy optimization methods typically rely on manually fixed values. Prior works generally motivates λ selection through the bias–variance trade-off of advantage estimation, which does not directly characterize the effectiveness of the resulting policy updates. In this work, we revisit λ selection from the perspective of policy optimization and introduce Gradient-Aligned Generalized Advantage Estimation (GA^{2}E), a novel framework that adaptively selects λ by maximizing the alignment of candidate policy gradients with a reference gradient that approximates the true policy gradient direction. We establish a theoretical connection between gradient alignment and policy improvement, and develop a parameter-free algorithm that can be seamlessly integrated into existing policy optimization methods such as PPO. Extensive experiments on MinAtar and MuJoCo benchmarks demonstrate that GA^{2}E consistently achieves strong and robust performance across diverse environments while reducing the dependence on manually tuned fixed λ values. Code is available at `https://anonymous.4open.science/r/PGAE-B4EB`.

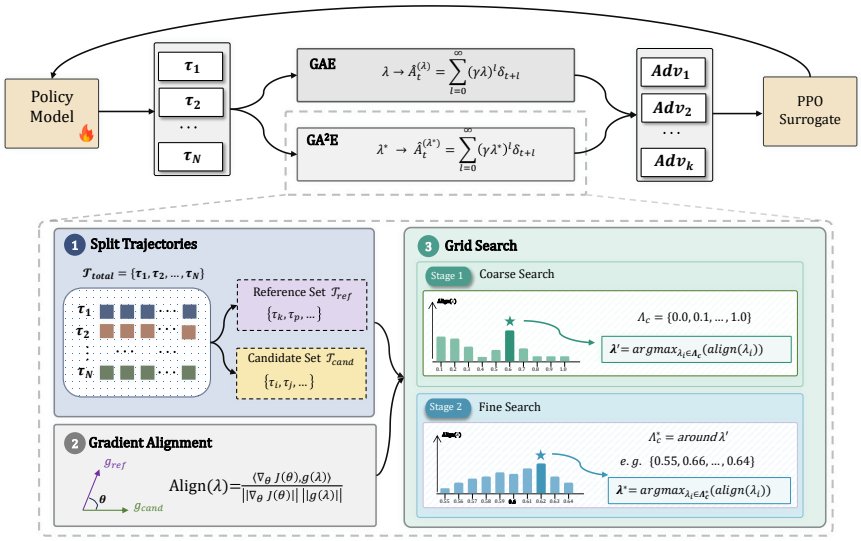


Figure 1: **Comparison between GAE and GA^{2}E on PPO.** Standard GAE computes advantages using a fixed trace parameter λ, whereas GA^{2}E introduces a gradient-alignment based selection module to adaptively determine $\lambda^\star$ according to the consistency between candidate gradients and a reference gradient.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

1 INTRODUCTION

Policy gradient (PG) methods constitute one of the dominant paradigms in modern reinforcement learning (RL), underpinning many state-of-the-art algorithms for continuous control (Makoviychuk et al., 2021; Thalagala et al., 2025), robotics (Han et al., 2023; Tang et al., 2025; Da et al., 2025; Tiwari et al., 2026), and large-scale sequential decision making (Vinyals et al., 2019; OpenAI et al., 2019; Silver et al., 2016; 2017; Unknown, 2025; Xu et al., 2025). A fundamental component of policy gradient methods is advantage estimation, whose quality directly affects the accuracy of policy gradient estimation and, consequently, the stability and effectiveness of policy optimization. Among various approaches, Generalized Advantage Estimation (GAE) (Schulman et al., 2016) has become the de facto standard due to its ability to interpolate between bootstrapped and Monte Carlo returns through the trace parameter λ. As a result, GAE has been widely adopted in representative policy optimization algorithms such as PPO (Schulman et al., 2017) and TRPO (Schulman et al., 2015).

Rather than defining a single advantage estimator, GAE gives rise to a family of estimators parameterized by λ. Different choices of λ produce different advantage estimates and, in turn, induce different policy gradient estimates for policy optimization. In practice, however, most policy gradient methods simply employ a fixed λ throughout training (e.g., $\lambda = 0.95$ in PPO), while comparatively little attention has been devoted to principled adaptive λ selection. As illustrated in Figure 2, the optimal fixed λ varies substantially across environments, suggesting that no universal choice consistently achieves the best optimization performance. This observation naturally leads to a fundamental question: *how should one determine whether one choice of λ is preferable to another?*

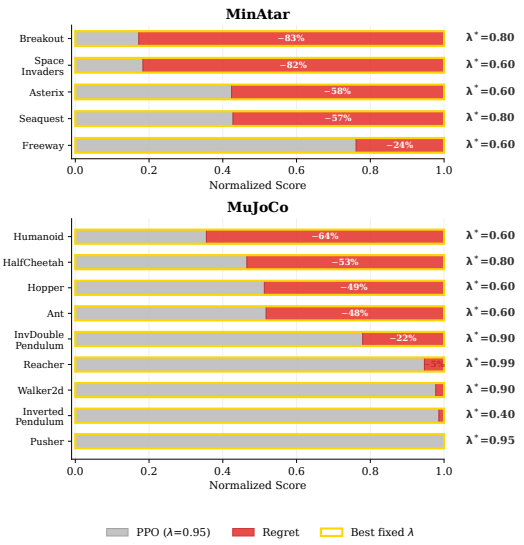


Figure 2: **Motivation for adaptive λ selection.** The optimal fixed λ varies substantially across environments, indicating that no single choice consistently yields the best policy optimization performance.

Existing studies predominantly (Chen et al., 2021; Schulman et al., 2016; Wu et al., 2018) answer this question through the lens of the bias–variance trade-off. Specifically, a smaller λ relies more heavily on bootstrapping, reducing estimation variance at the expense of increased bootstrap bias, whereas a larger λ gradually approaches Monte Carlo returns, alleviating bootstrap bias while introducing higher variance. Such analyses have considerably advanced our understanding of how λ affects the statistical quality of advantage estimation, including its bias and variance characteristics. Nevertheless, they primarily evaluate **the estimator itself** rather than **the effectiveness of the policy updates that it ultimately induces**.

This distinction is fundamental. In policy gradient methods, the advantage estimate is not the optimization objective; instead, it serves as an intermediate quantity for constructing policy gradient estimates. Different choices of λ induce different policy gradients through the resulting advantage estimates, ultimately leading to different policy updates. Consequently, an advantage estimator with more favorable statistical properties does not necessarily produce a policy update that yields greater policy improvement. **Put differently, although the bias–variance trade-off provides an appropriate criterion for characterizing estimator quality, it does not directly answer the question of how λ should be selected from the perspective of policy optimization.**

Motivated by this observation, we revisit λ selection in GAE from the perspective of policy optimization. Instead of evaluating candidate λ values based solely on the statistical properties of the induced advantage estimator, we formulate λ selection according to the quality of the resulting policy updates. We show that this objective admits a tractable first-order characterization, where

adaptive λ selection reduces to maximizing the alignment between candidate policy gradients and the true optimization direction. Since the true policy gradient is unavailable during training, we construct a reference gradient to approximate its direction and develop **Gradient-Alignment Generalized Advantage Estimation (GA^{2}E)**, a parameter-free adaptive advantage estimation method operating at the rollout level. Our contributions are summarized as follows:

- We revisit GAE trace parameter λ selection from the perspective of policy optimization and formulate a principled criterion based on the quality of induced policy updates.
- We establish the theoretical connection between this criterion and gradient alignment, and derive a reference-gradient-based approach for adaptive λ selection.
- We propose **Gradient-Alignment Generalized Advantage Estimation (GA^{2}E)**, a simple adaptive algorithm without additional trainable components or hyperparameter optimization. Experiments on continuous- and discrete-control benchmarks demonstrate consistent improvements over PPO.

2 PRELIMINARIES

2.1 MARKOV DECISION PROCESS

We consider a discounted Markov Decision Process (MDP), defined by the tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{P}, r, \gamma, \rho_0)$, where $\mathcal{S}$ and $\mathcal{A}$ denote the state and action spaces, $\mathcal{P}$ is the transition function, r is the reward function, $\gamma \in [0, 1)$ is the discount factor, and ρ_0 is the initial state distribution. A stochastic policy $\pi_\theta(a|s)$ parameterized by θ induces a trajectory $\tau = (s_0, a_0, s_1, a_1, \ldots)$, and the objective is to maximize the expected discounted return

$$J(\theta) = \mathbb{E}_{\tau\sim\pi_\theta}\left[\sum_{t=0}^{\infty}\gamma^t r(s_t, a_t)\right]. \tag{1}$$

We use $V^\pi(s)$, $Q^\pi(s, a)$, and $A^\pi(s, a) = Q^\pi(s, a) - V^\pi(s)$ to denote the state-value, action-value, and advantage functions, respectively.

2.2 PROXIMAL POLICY OPTIMIZATION

Policy gradient methods optimize the policy parameters by ascending the gradient of the expected return. According to the policy gradient theorem (Sutton et al., 2000), the gradient can be expressed as

$$\nabla_\theta J(\theta) = \mathbb{E}_{\tau\sim\pi_\theta}\left[\sum_{t=0}^{\infty}\nabla_\theta \log \pi_\theta(a_t|s_t) A^{\pi_\theta}(s_t, a_t)\right]. \tag{2}$$

Since the true advantage function is generally unavailable, practical policy gradient algorithms rely on an estimated advantage $\hat{A}_t$. PPO (Schulman et al., 2017) is a widely adopted on-policy policy gradient algorithm that improves optimization stability through a clipped surrogate objective:

$$L^{\mathrm{CLIP}}(\theta) = \mathbb{E}_t\left[\min\left(r_t(\theta)\hat{A}_t(s_t, a_t),\ \mathrm{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\,\hat{A}_t(s_t, a_t)\right)\right], \tag{3}$$

where $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\mathrm{old}}}(a_t|s_t)}$ denotes the probability ratio between the updated and behavior policies, and ϵ controls the clipping range.

2.3 GENERALIZED ADVANTAGE ESTIMATION

GAE (Schulman et al., 2016) constructs low-variance advantage estimates by exponentially weighting multi-step temporal-difference (TD) residuals. The one-step TD residual is defined as

$$\delta_t = r_t + \gamma V^\pi(s_{t+1}) - V^\pi(s_t), \tag{4}$$

where V denotes the learned value function. GAE estimates the advantage as

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

where the trace parameter $\lambda \in [0, 1]$ determines the exponential weighting applied to future TD residuals.

3 METHOD

3.1 GRADIENT ALIGNMENT OBJECTIVE FOR ADAPTIVE λ SELECTION

GAE introduces the trace parameter λ to balance the bias and variance of advantage estimation. Existing analyses of λ mainly focus on the statistical properties of the resulting advantage estimator, where different choices of λ correspond to different bias–variance trade-offs. However, the advantage estimator is not the final objective of policy optimization; it only serves as an intermediate signal for constructing policy gradients.

For a given trace parameter λ, GAE produces an advantage estimate $\hat{A}_t^{(\lambda)}$, which induces the corresponding policy gradient:

$$g(\lambda) = \mathbb{E}\left[\nabla_\theta \log \pi_\theta(a_t|s_t)\hat{A}_t^{(\lambda)}\right]. \tag{6}$$

Therefore, different choices of λ not only produce different advantage estimators, but also lead to different policy update directions. From the perspective of policy optimization, the optimal λ should be determined by the effectiveness of the induced policy update rather than the statistical quality of the advantage estimator itself.

To formalize this idea, let $J : \mathcal{P} \to \mathbb{R}$ denote the policy performance objective, where $\mathcal{P}$ is the policy space. Given a current policy parameterized by θ and a policy update induced by λ, we define the optimal trace parameter as the one that achieves the largest local improvement in policy performance:

Definition 1 (Policy-Oriented λ Selection Principle)**.** *For a candidate trace parameter λ, let $g(\lambda)$ denote the corresponding policy gradient. The optimal trace parameter is defined as*

$$\lambda^\star = \arg\max_\lambda J(\theta + \alpha g(\lambda)). \tag{7}$$

where α denotes the policy update step size. Definition 1 reformulates λ selection from an advantage estimation problem into a policy optimization problem. Since the true policy improvement after an update is unavailable during training, the selection objective can be equivalently transformed into the following gradient alignment criterion (derivation in Appendix B.1):

$$\lambda^\star = \arg\max_\lambda \langle \nabla_\theta J(\theta), g(\lambda)\rangle. \tag{8}$$

To mitigate the influence of varying gradient scales across optimization settings, we focus on gradient directions and define the alignment objective as

$$\mathrm{Align}(\lambda) = \frac{\langle \nabla_\theta J(\theta), g(\lambda)\rangle}{\|\nabla_\theta J(\theta)\|\|g(\lambda)\|}. \tag{9}$$

Thus, adaptive λ selection can be viewed as identifying the candidate advantage estimator whose induced policy gradient is most consistent with the true policy improvement direction.

3.2 REFERENCE GRADIENT CONSTRUCTION AND THEORETICAL JUSTIFICATION

The ideal gradient alignment objective in equation 9 requires the true policy gradient $\nabla_\theta J(\theta)$, which is unavailable during policy optimization. Therefore, we introduce a reference gradient g_{ref} to approximate the true policy improvement direction.

Specifically, we show that the error of the reference gradient is controlled by the underlying advantage estimation error, and the resulting directional deviation leads to a bounded error in the gradient alignment objective. Let

$$g^\star = \nabla_\theta J(\theta) \tag{10}$$

denote the true policy gradient, and consider the policy gradient induced by an arbitrary advantage estimator $\hat{A}(s,a)$:

$$g(\hat{A}) = \mathbb{E}\left[\nabla_\theta \log \pi_\theta(a|s)\hat{A}(s,a)\right]. \tag{11}$$

We first characterize the relationship between advantage estimation error and the induced policy gradient error.

Lemma 1 (Stability of Policy Gradient Estimation)**.** *Assume that the score function satisfies* $\mathbb{E}\left[\|\nabla_\theta \log \pi_\theta(a|s)\|^2\right] < \infty$*. Then, for any square-integrable advantage estimator* $\hat{A}$*,*

$$\|g(\hat{A}) - g^\star\| \le C\|\hat{A} - A^\pi\|_{L^2}, \tag{12}$$

where

$$C = \left(\mathbb{E}\left[\|\nabla_\theta \log \pi_\theta(a|s)\|^2\right]\right)^{\frac{1}{2}}. \tag{13}$$

Lemma 1 (Proof in Appendix B.2) shows that the policy gradient error is bounded by the corresponding advantage estimation error. This provides the theoretical motivation for constructing the reference gradient with a low-bias advantage estimator.

However, since the alignment objective (Eq equation 9) depends on gradient direction rather than magnitude, we further analyze the directional deviation introduced by approximating the true gradient $g^\star$ with the reference gradient g_{ref}.

Lemma 2 (Gradient Direction Perturbation Bound)**.** *Suppose* $g^\star \neq 0$*, and let a reference gradient be expressed as* $g_r = g^\star + e$ *with* $|e| < |g^\star|$*. Then the angular deviation between* g_r *and* $g^\star$ *satisfies*

$$\sin \angle(g_r, g^\star) \le \frac{\|e\|}{\|g^\star\| - \|e\|}. \tag{14}$$

Lemma 2 (Proof in Appendix B.3) shows that when the reference gradient error is small compared with the true gradient magnitude, the reference direction remains close to the true policy improvement direction. Finally, we quantify how such directional deviation affects the alignment objective. In GA^{2}E, the reference gradient g_{ref} plays the role of g_r in Lemma 2.

Theorem 1 (Alignment Perturbation Bound)**.** *Let* $\varepsilon = \angle(g_r, g^\star)$*, and for any non-zero candidate gradient* g*, the difference between the alignment measured using the reference gradient and the ideal alignment satisfies*

$$|\cos(g, g_r) - \cos(g, g^\star)| \le 2\sin\frac{\varepsilon}{2}. \tag{15}$$

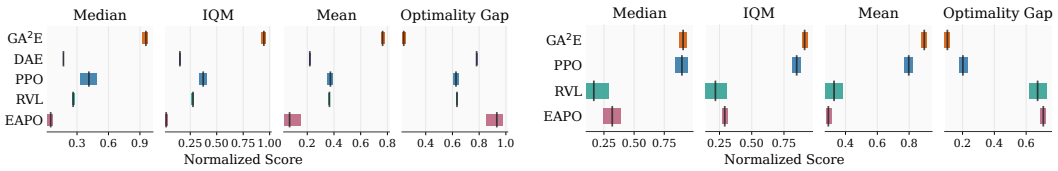
Theorem 1 (Proof in Appendix B.4) shows that the reference-based alignment objective remains a reliable approximation to the ideal objective when the reference gradient is close to the true policy gradient. Therefore, GA^{2}E constructs the reference gradient using a low-bias GAE estimator with a large trace parameter λ_{ref}:

$$g_{\text{ref}} = \mathbb{E}\left[\nabla_\theta \log \pi_\theta(a_t|s_t)\hat{A}_t^{\lambda_{\text{ref}}}\right]. \tag{16}$$

Here, λ_{ref} is only used to approximate the true policy gradient direction and is not applied to the final policy update, which always uses the adaptively selected parameter $\lambda^\star$.

Moreover, To avoid trivial alignment caused by shared sampling noise, we compute the reference gradient and candidate gradients using disjoint trajectory subsets. Specifically, the rollout batch is partitioned as

$$\mathcal{D} = \mathcal{D}_{\text{ref}} \cup \mathcal{D}_{\text{cand}}, \qquad \mathcal{D}_{\text{ref}} \cap \mathcal{D}_{\text{cand}} = \emptyset. \tag{17}$$



(a) **MinAtar.** Aggregate performance metrics

(b) **MuJoCo.** Aggregate performance metrics

Figure 3: **Aggregate performance comparison with representative RL baselines.** Results are reported using IQM, mean, median, and optimality gap with 95% stratified bootstrap confidence intervals. Higher is better for IQM, Mean, and Median, while lower is better for Optimality Gap. Training curves are provided in Figure 12.

The reference gradient is computed as

$$g_{\text{ref}} = \text{PolicyGrad}(\mathcal{D}_{\text{ref}}, \lambda_{\text{ref}}), \tag{18}$$

while each candidate gradient is evaluated independently:

$$g(\lambda) = \text{PolicyGrad}(\mathcal{D}_{\text{cand}}, \lambda). \tag{19}$$

This separation ensures that the measured alignment reflects the consistency between different gradient estimators rather than correlations introduced by the same trajectory samples, providing a more faithful approximation to the reference-based alignment objective analyzed in Theorem 1.

3.3 The GA^{2}E Algorithm

Given the reference gradient g_{ref}, GA^{2}E evaluates each candidate trace parameter according to

$$S(\lambda) = \frac{g_{\text{ref}}^\top g(\lambda)}{\|g_{\text{ref}}^\top\| \|g(\lambda)\|}. \tag{20}$$

Since λ is a one-dimensional continuous variable, we optimize this objective using a **two-stage grid search**. The coarse stage identifies the promising region, while the refinement stage performs a local search around the best candidate. The selected $\lambda^\star$ is then used to compute the final GAE estimate for policy optimization:

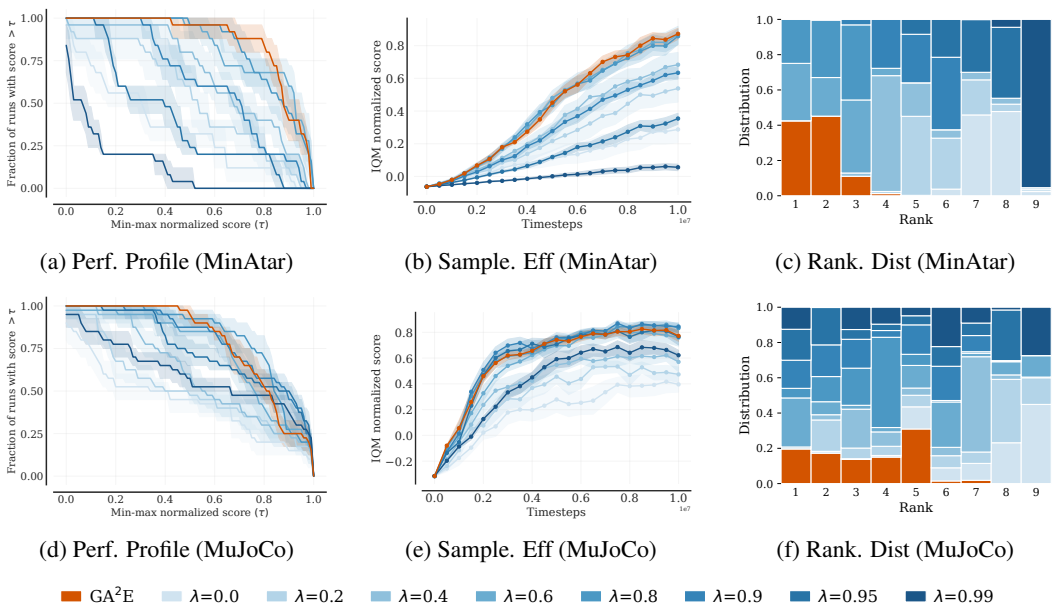
$$\hat{A}_t^{(\lambda^\star)} = \sum_{l=0}^{T-t-1} (\gamma\lambda^\star)^l \delta_{t+l}, \tag{21}$$

which is subsequently used for the policy update. In practice, the frequency of λ selection can be adjusted according to the optimization schedule, including rollout-level or epoch-level adaptation. We adopt rollout-level adaptation as the default configuration, where $\lambda^\star$ is selected after each rollout collection and remains fixed during subsequent PPO optimization epochs. Accordingly, the complete procedure of GA^{2}E is summarized in Algorithm 1.

4 Experiments

We evaluate GA^{2}E integrated with PPO on both discrete and continuous control benchmarks (see Appendix A.3 for the complete experimental setup). Unless otherwise specified, all PPO baselines use the default GAE estimator with $\lambda = 0.95$. Our experiments are designed to answer the following research questions:

- **RQ1: Can GA^{2}E improve policy optimization performance compared with existing baselines?** We compare GA^{2}E with standard PPO (Schulman et al., 2017) and



(a) Perf. Profile (MinAtar) (b) Sample. Eff (MinAtar) (c) Rank. Dist (MinAtar)

(d) Perf. Profile (MuJoCo) (e) Sample. Eff (MuJoCo) (f) Rank. Dist (MuJoCo)

Figure 4: **Performance, sample efficiency, and rank distribution on MinAtar and MuJoCo.** performance profiles, IQM-based sample efficiency and rank distribution are reported following (Agarwal et al., 2021). The corresponding training curves are shown in Figure 13.

several representative advantage estimation methods, including DAE (Pan et al., 2022), RVL (Höftmann et al., 2026), and EAPO (Choe & Kim, 2025). Detailed descriptions of the baselines are provided in Appendix A.5.

- **RQ2: Can GA^{2}E achieve robust performance across different λ configurations without requiring manual tuning?** We compare GA^{2}E with fixed-λ PPO variants across different λ configurations to assess the robustness of our adaptive λ selection.
- **RQ3: How Does the Estimation Granularity Affect GA^{2}E?** We conduct ablation studies on estimation granularity and adaptive λ dynamics, and measure the computational overhead of GA^{2}E relative to PPO.

4.1 RQ1: Performance Comparison

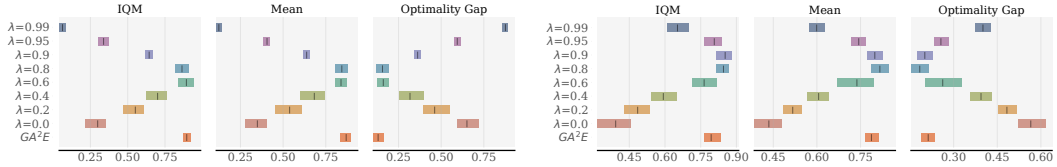
We first evaluate GA^{2}E against existing advantage estimation methods on MinAtar and MuJoCo benchmarks. Specifically, we compare GA^{2}E with representative approaches that improve policy optimization through alternative advantage refinement strategies. To provide a comprehensive comparison across diverse tasks, we adopt aggregate metrics following the evaluation protocol of (Agarwal et al., 2021), including Interquartile Mean (IQM), Mean, Median, and Optimality Gap.

Figure 3 shows that GA^{2}E consistently outperforms existing methods across both benchmarks, achieving higher IQM, Mean, and Median while reducing the Optimality Gap. These results demonstrate that our GA^{2}E achieves stronger overall performance and better robustness than existing advantage estimation methods across diverse reinforcement learning tasks.

4.2 RQ2: Robust Performance Across Diverse λ Choices

We further investigate whether GA^{2}E can achieve competitive performance without task-specific λ tuning. We compare a unified GA^{2}E configuration with fixed-λ PPO using $\lambda \in \{0.0, 0.2, 0.4, 0.6, 0.8, 0.9, 0.95, 0.99\}$. Here, best fixed λ denotes the best-performing setting among the evaluated candidates rather than the theoretical optimum over the continuous range $[0, 1]$.

Figure 5 compares GA^{2}E with fixed-λ PPO variants using IQM, Mean, and Optimality Gap. GA^{2}E achieves better aggregate performance on both benchmarks, obtaining higher IQM and Mean while

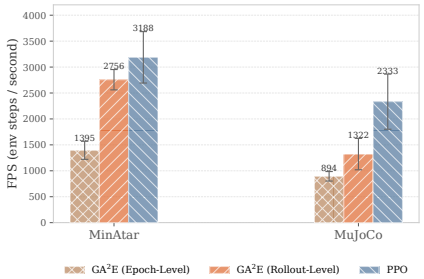


(a) **MinAtar.** Aggregate performance metrics (b) **MuJoCo.** Aggregate performance metrics

Figure 5: **Aggregate performance of GA^{2}E and fixed-λ PPO variants.** Median is omitted because it is less informative for evaluating robustness to suboptimal λ choices .

	MinAtar			MuJoCo		
Method	Perf.	Var.	Worst	Perf.	Var.	Worst
GA^{2}E	**1.8**	3.2	**3**	**3.4**	4.0	5
PPO(λ=0.0)	7.6	5.6	8	8.1	7.6	9
PPO(λ=0.2)	6.0	7.0	7	6.1	6.3	9
PPO(λ=0.4)	4.6	6.4	6	6.0	5.9	8
PPO(λ=0.6)	2.2	**2.4**	**3**	4.8	5.6	9
PPO(λ=0.8)	2.0	5.0	**3**	3.6	4.3	**4**
PPO(λ=0.9)	5.0	4.8	6	3.7	**3.4**	7
PPO(λ=0.95)	6.8	5.0	8	4.3	4.1	8
PPO(λ=0.99)	9.0	5.6	9	5.0	3.8	9

(a) Performance, stability, and worst-case ranks.



(b) Training throughput (FPS).

Figure 6: **Performance summary and training efficiency of GA^{2}E. (a)** Performance, stability, and worst-case ranks of GA^{2}E and fixed-λ PPO. Perf. and Var. ranks are computed over the final 10% of training, while Worst denotes the maximum per-environment rank across all environments. Lower is better. **(b)** Training throughput in FPS. Rollout-level GA^{2}E achieves higher throughput than epoch-level GA^{2}E due to its coarser-grained λ adaptation, while both GA^{2}E variants incur additional computation compared with vanilla PPO.

reducing the Optimality Gap. These results indicate that adaptive λ selection approaches the performance of strong fixed-λ configurations without task-specific tuning.

Beyond aggregate performance, Figures 4a and 4d further analyze performance distribution, while Figures 4b and 4e evaluate sample efficiency. GA^{2}E maintains higher performance profiles across a wide range of thresholds and achieves competitive learning efficiency throughout training, demonstrating consistent improvements beyond individual tasks. Table 6a, along with Figures 4c and 4f summarize the performance and variance rankings across tasks. GA^{2}E achieves the best average performance rank and favorable variance rank on both benchmarks, indicating that its improvements are consistent and not limited to specific environments.

Overall, GA^{2}E does not rely on manually identifying a globally optimal fixed λ. Instead, it adaptively selects appropriate advantage estimation parameters according to the current policy optimization state without task-specific tuning. The empirical results demonstrate that GA^{2}E effectively avoids the performance degradation of fixed-λ approaches and provides robust and competitive policy optimization across diverse reinforcement learning environments.

4.3 RQ3: Ablation Study

We investigate how the granularity of gradient consistency estimation affects GA^{2}E. Specifically, we consider two variants: (i) **epoch-level GA^{2}E**, which updates λ^* after each PPO optimization epoch, and (ii) **rollout-level GA^{2}E**, which updates λ^* after each rollout collection and keeps it fixed during subsequent optimization epochs.

Table 1 compares the final performance of the two variants. Both approaches achieve competitive results across diverse environments, with small performance gaps in most tasks, suggesting that GA^{2}E is robust to different estimation granularities. The two variants each achieve the best performance on 7 environments, indicating that more frequent λ^* updates do not consistently bring additional gains.

Table 1: **Ablation study of the update granularity in GA^{2}E with** $\lambda_{\text{ref}} = 0.97$. Results are mean±std episodic returns over the final 10% of training. **Avg. CV** is computed as the average of environment-wise $\text{CV} = \sigma/|\mu|$ values, where lower values indicate better stability.

Environment	Epoch-Level	Rollout-Level
	MinAtar	
Asterix	**28.3±1.0**	27.2±1.4
Breakout	85.6±11.1	**97.4±21.2**
Freeway	60.8±0.8	**62.1±1.2**
Seaquest	**89.5±7.7**	84.3±6.7
SpaceInvaders	477.8±19.8	**533.3±37.2**
Avg. CV ↓	**0.061**	0.087
	MuJoCo	
InvertedPendulum	999.1±2.1	**1000.0±0.0**
InvertedDoublePendulum	9043.2±419.6	**9249.9±83.9**
Reacher	**-3.6±0.0**	-4.0±0.2
Pusher	**-25.9±0.6**	-30.4±1.9
HalfCheetah	8108.8±2363.9	**10936.3±3319.2**
Hopper	**2265.7±746.1**	1724.1±270.5
Walker2d	5283.1±1420.2	**5522.2±715.7**
Ant	**5523.4±1040.7**	5321.6±1080.8
Humanoid	**8737.6±765.0**	8214.4±724.2
Avg. CV ↓	0.139	**0.111**

Figure 9 further shows similar λ^* adaptation patterns, while rollout-level GA^{2}E produces smoother trajectories.

To evaluate the computational efficiency of different gradient consistency estimation granularities, we compare the training throughput of the two variants (Figure 6b). Rollout-level GA^{2}E achieves nearly twice the throughput of epoch-level GA^{2}E by reducing gradient consistency estimation from per-epoch to per-rollout updates. Considering its comparable performance and lower computational overhead, rollout-level GA^{2}E is adopted as the default configuration.

4.4 Additional Experiments

We provide additional experiments in the appendix to further evaluate the robustness and general applicability of GA^{2}E. These experiments include additional Atari benchmark results (Appendix C.4), sensitivity analysis of key hyperparameters, including the reference trace parameter λ_{ref} and the ratio between reference (Appendix C.2) and candidate trajectories (Appendix C.1), as well as evaluations on sparse-reward environments (Appendix C.3).

5 Conclusion

In this work, we propose GA^{2}E, an adaptive advantage estimation framework that selects the GAE trace parameter λ through gradient alignment. By measuring the consistency between induced policy gradients and the desired optimization direction, GA^{2}E provides a principled alternative to manually tuned fixed-λ strategies. Experiments with PPO across diverse benchmarks demonstrate that GA^{2}E achieves robust and competitive performance while reducing the need for task-specific λ tuning, highlighting gradient consistency as an effective principle for designing adaptive learning signals in policy optimization.

Beyond the effectiveness of adaptive λ selection, our study reveals that the granularity of adaptation is also an important design factor. rollout-level adaptation provides a favorable performance-efficiency trade-off in most environments, while epoch-level can be beneficial when the optimal trace parameter changes rapidly during optimization. These findings suggest that future adaptive optimization methods should consider not only which parameters to adapt, but also the appropriate timescale for adaptation.

REFERENCES

Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron C Courville, and Marc Bellemare. Deep reinforcement learning at the edge of the statistical precipice. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, pp. 29304–29320, 2021.

Yurou Chen, Fengyi Zhang, and Zhiyong Liu. Adaptive advantage estimation for actor-critic algorithms. In *International Joint Conference on Neural Networks*, pp. 1–8. IEEE, 2021. doi: 10.1109/IJCNN52387.2021.9534005. URL https://ieeexplore.ieee.org/document/9534005.

Maxime Chevalier-Boisvert, Lucas Willems, and Suman Pal. Minimalistic gridworld environment for openai gym. https://github.com/Farama-Foundation/Minigrid, 2018.

Jean Seong Bjorn Choe and Jong-Kook Kim. Maximum Entropy Softmax Policy Gradient via Entropy Advantage Estimation. In *International Joint Conference on Artificial Intelligence*, pp. 4958–4966, 2025. doi: 10.24963/IJCAI.2025/552. URL https://mlanthology.org/ijcai/2025/choe2025ijcai-maximum/.

Karl Cobbe, Jacob Hilton, Oleg Klimov, and John Schulman. Phasic policy gradient. *Proceedings of the 38th International Conference on Machine Learning*, 139:2034–2045, 2021. URL https://arxiv.org/abs/2009.04416.

Longchao Da, Justin Turnau, Thirulogasankar Pranav Kutralingam, Alvaro Velasquez, Paulo Shakarian, and Hua Wei. A survey of sim-to-real methods in rl: Progress, prospects and challenges with foundation models. *arXiv preprint arXiv:2502.13187*, 2025. URL https://arxiv.org/abs/2502.13187.

Logan Engstrom, Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Firdaus Janoos, Larry Rudolph, and Aleksander Madry. Implementation matters in deep policy gradients: A case study on ppo and trpo. In *International Conference on Learning Representations*, 2020. URL https://arxiv.org/abs/2005.12729.

Dong Han, Beni Mulyana, Vladimir Stankovic, and Samuel Cheng. A survey on deep reinforcement learning algorithms for robotic manipulation. *Sensors*, 23(7):3762, 2023.

Philipp Hauf et al. Relative value learning. *Proceedings of the Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=ulTRUwrzt9. ICLR 2026.

Marc Höftmann, Jan Robine, and Stefan Harmeling. Relative value learning. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=ulTRUwrzt9.

Nai-Chieh Huang, Ping-Chun Hsieh, Kuo-Hao Ho, and I-Chen Wu. Ppo-clip attains global optimality: Towards deeper understandings of clipping. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(11):11635–11643, 2024. doi: 10.1609/aaai.v38i11.29154. URL https://arxiv.org/abs/2312.12065.

Shengyi Huang, Rousslan Fernand Julien Dossa, Antonin Raffin, Anssi Kanervisto, and Weixun Wang. The 37 implementation details of proximal policy optimization. In *First Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=Hl6jCqIp2j.

Tianying Ji, Yongyuan Liang, Yan Zeng, Yu Luo, Guowei Xu, Jiawei Guo, Ruijie Zheng, Furong Huang, Fuchun Sun, and Huazhe Xu. Ace: Off-policy actor-critic with causality-aware entropy regularization. *arXiv preprint arXiv:2402.14528*, 2024. URL https://arxiv.org/abs/2402.14528. ICML 2024 Oral.

Martin Kaloev and Georgi Krastev. Exploring advantage estimation with higher-order summation of state values in deep reinforcement learning: Experimental insights for online temporal difference approach. In *International Conference on Artificial Intelligence Algorithm Design for Systems*, Lecture Notes in Networks and Systems, pp. 181–190. Springer, 2024. doi: 10.1007/978-3-031-70518-2_16. URL https://link.springer.com/chapter/10.1007/978-3-031-70518-2_16.

Junbo Li, Peng Zhou, Rui Meng, Meet P Vadera, Lihong Li, and Yang Li. Turn-ppo: Turn-level advantage estimation with ppo for improved multi-turn rl in agentic llms. *arXiv preprint arXiv:2512.17008*, 2026. URL https://arxiv.org/abs/2512.17008. Accepted to EACL 2026.

Andrei Lixandru. Proximal policy optimization with adaptive exploration. *arXiv preprint arXiv:2405.04664*, 2024. URL https://arxiv.org/abs/2405.04664.

Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, et al. Isaac gym: High performance gpu-based physics simulation for robot learning. *arXiv preprint arXiv:2108.10470*, 2021.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015. doi: 10.1038/nature14236.

OpenAI, Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemysław Debiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, Rafal Józefowicz, Scott Gray, Catherine Olsson, Jakub Pachocki, Michael Petrov, Henrique P. d. O. Pinto, Jonathan Raiman, Tim Salimans, Jeremy Schlatter, Jonas Schneider, Szymon Sidor, Ilya Sutskever, Jie Tang, Filip Wolski, and Susan Zhang. Dota 2 with large scale deep reinforcement learning, 2019. URL https://arxiv.org/abs/1912.06680.

Hsiao-Ru Pan and Bernhard Schölkopf. Direct advantage estimation for scalable and sample-efficient deep reinforcement learning. *arXiv preprint arXiv:2606.20411*, 2026. URL https://arxiv.org/abs/2606.20411. Accepted at RLC 2026.

Hsiao-Ru Pan, Nico Gürtler, Alexander Neitz, and Bernhard Schölkopf. Direct advantage estimation. In *Advances in Neural Information Processing Systems*, volume 35, pp. 11869–11880, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/4d893f766ab60e5337659b9e71883af4-Abstract.html.

John Schulman, Sergey Levine, Philipp Moritz, Michael Jordan, and Pieter Abbeel. Trust region policy optimization. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 1889–1897, 2015. URL https://proceedings.mlr.press/v37/schulman15.html.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. In *International Conference on Learning Representations*, 2016. URL https://arxiv.org/abs/1506.02438.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. In *arXiv preprint arXiv:1707.06347*, 2017. URL https://arxiv.org/abs/1707.06347.

David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *nature*, 550(7676):354–359, 2017.

Sanghyun Son, Laura Yu Zheng, Ryan Sullivan, Yi-Ling Qiao, and Ming C Lin. Gradient informed proximal policy optimization. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://arxiv.org/abs/2312.08710.

Xiulei Song, Yizhao Jin, Gregory Slabaugh, and Simon Lucas. Partial advantage estimator for proximal policy optimization. *IEEE Transactions on Games*, 17(1):158–166, 2025. doi: 10.1109/TG.2024.3408298. URL https://arxiv.org/abs/2301.10920.

Ryan Sullivan, Akarsh Kumar, Shengyi Huang, John P Dickerson, and Joseph Suarez. Reward scale robustness for proximal policy optimization via dreamerv3 tricks. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://arxiv.org/abs/2310.17805.

Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. *Advances in Neural Information Processing Systems (NeurIPS)*, 12, 2000.

Charlie B Tan, Edan Toledo, Benjamin Ellis, Jakob N Foerster, and Ferenc Huszár. Beyond the boundaries of proximal policy optimization. *arXiv preprint arXiv:2411.00666*, 2024. URL https://arxiv.org/abs/2411.00666.

Chen Tang, Ben Abbatematteo, Jiaheng Hu, Rohan Chandra, Roberto Martín-Martín, and Peter Stone. Deep Reinforcement Learning for Robotics: A Survey of Real-World Successes. 39(27): 28698–28699, 2025. doi: 10.1609/aaai.v39i27.35095. URL https://mlanthology.org/aaai/2025/tang2025aaai-deep/.

Shiron Thalagala, Pak Kin Wong, Xiaozheng Wang, and Tianang Sun. Broad critic deep actor reinforcement learning for continuous control. *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–8, 2025. doi: 10.1109/TNNLS.2025.3554082. URL https://arxiv.org/abs/2411.15806.

Rajesh Tiwari, Shailesh Khapre, and Avantika Singh. Reinforcement learning in robotic systems: A review on sim-to-real transfer. *Robotics and Autonomous Systems*, 198:105327, 2026. doi: https://doi.org/10.1016/j.robot.2025.105327. URL https://www.sciencedirect.com/science/article/abs/pii/S0921889025004245.

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*, pp. 5026–5033, 2012.

George Tucker, Surya Bhupatiraju, Shixiang Gu, Richard Turner, Zoubin Ghahramani, and Sergey Levine. The mirage of action-dependent baselines in reinforcement learning. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pp. 5015–5024. PMLR, 2018. URL https://proceedings.mlr.press/v80/tucker18a.html.

Unknown. A closer look at deep policy gradients. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=ryxdEkHtPS.

Unknown. Fast proximal policy optimization. In *International Conference on Intelligent Computing*, Lecture Notes in Networks and Systems, pp. 55–65. Springer, 2022. doi: 10.1007/978-3-031-02444-3_6. URL https://link.springer.com/chapter/10.1007/978-3-031-02444-3_6.

Unknown. Adaptive bias-variance trade-off in advantage estimator for actor-critic methods. *Neural Networks*, 169:106–119, 2024. doi: https://doi.org/10.1016/j.neunet.2023.10.023. URL https://www.sciencedirect.com/science/article/pii/S0893608023005774.

Unknown. Enhancing autonomous driving decision: A hybrid deep reinforcement learning framework. *IEEE Transactions on Intelligent Transportation Systems*, 2025. doi: 10.1109/TITS.2025.3529838. URL https://dl.acm.org/doi/10.1109/TITS.2025.3529838.

Unknown. Dynamic proximal policy optimization: Enhancing ppo with adaptive entropy and smooth clipping. *Neurocomputing*, 2026. doi: https://doi.org/10.1016/j.neucom.2026.132861. URL https://www.sciencedirect.com/science/article/pii/S0925231226002584.

Oriol Vinyals, Igor Babuschkin, Wojciech M Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H Choi, Richard Powell, Timo Ewalds, Petko Georgiev, et al. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *nature*, 575(7782):350–354, 2019.

Yinuo Wang, Likun Wang, Yuxuan Jiang, Wenjun Zou, Tong Liu, Xujie Song, Wenxuan Wang, Liming Xiao, Jiang Wu, Jingliang Duan, and Shengbo Eben Li. Diffusion actor-critic with entropy regulator. *arXiv preprint arXiv:2405.15177*, 2024. URL https://arxiv.org/abs/2405.15177. NeurIPS 2024.

Yuhui Wang, Hao He, Chao Wen, and Xiaoyang Tan. Truly proximal policy optimization. *Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence*, pp. 113–122, 2020. URL https://arxiv.org/abs/1903.07940.

Cathy Wu, Aravind Rajeswaran, Yan Duan, Vikash Kumar, Alexandre M Bayen, Sham Kakade, Igor Mordatch, and Pieter Abbeel. Variance reduction for policy gradient with action-dependent factorized baselines. In *International Conference on Learning Representations (ICLR)*, 2018.

Yaqi Wu, Yingying Liu, et al. Trust region policy optimization with optimal transport discrepancies. *Advances in Neural Information Processing Systems*, 35:35274–35286, 2022. URL https://arxiv.org/abs/2210.11137.

Zhengpeng Xie, Qiang Zhang, Fan Yang, Marco Hutter, and Renjing Xu. Simple policy optimization. *arXiv preprint arXiv:2401.16025*, 2024. URL https://arxiv.org/abs/2401.16025.

Xi Xu et al. Agentgym-rl: Training llm agents for long-horizon decision making through multi-turn reinforcement learning. *arXiv preprint arXiv:2509.08755*, 2025. URL https://arxiv.org/abs/2509.08755.

Kenny Young and Tian Tian. Minatar: An atari-inspired testbed for thorough and reproducible reinforcement learning experiments. In *arXiv preprint arXiv:1903.03176*, 2019.

Ziqi Zhang, Jingzehua Xu, Zifeng Zhuang, Hongyin Zhang, Jinxin Liu, Donglin Wang, and Shuai Zhang. A dynamical clipping approach with task feedback for proximal policy optimization. *arXiv preprint arXiv:2312.07624*, 2024. URL https://arxiv.org/abs/2312.07624.

Hanyang Zhao, Wenpin Tang, and David Yao. Policy optimization for continuous reinforcement learning. *Advances in Neural Information Processing Systems*, 36, 2023. URL https://arxiv.org/abs/2305.18901.

A Appendix

A.1 Related Works

PPO Variants. Proximal Policy Optimization (PPO) (Schulman et al., 2017) has become a standard on-policy reinforcement learning algorithm due to its simplicity and effectiveness. Numerous studies have investigated improvements from different perspectives, including adaptive and smoothed clipping mechanisms (Zhang et al., 2024; Unknown, 2026; Huang et al., 2024; Unknown, 2022), alternative objectives and training paradigms with stronger stability or theoretical properties (Xie et al., 2024; Wang et al., 2020; Cobbe et al., 2021), and structural or exploration-oriented extensions (Tan et al., 2024; Li et al., 2026; Lixandru, 2024). Other works further enhance PPO through improved entropy regularization (Ji et al., 2024; Wang et al., 2024; Choe & Kim, 2025), implementation refinements (Engstrom et al., 2020; Huang et al., 2022), and additional optimization techniques such as DreamerV3-inspired tricks, differentiable gradient information, and alternative trust-region constraints (Sullivan et al., 2023; Son et al., 2023; Wu et al., 2022).

Advantage Estimation. Accurate advantage estimation is critical for reducing gradient variance and improving the sample efficiency of actor-critic algorithms. Generalized Advantage Estimation (GAE) (Schulman et al., 2016) provides a widely adopted bias-variance trade-off through a manually selected coefficient λ. Beyond standard value-based estimation, early studies investigated variance reduction through improved baselines and control variates. Wu et al. (Wu et al., 2018) proposed a bias-free action-dependent factorized baseline for stochastic policies, while Tucker et al. (Tucker et al., 2018) analyzed the limitations of learned state-action baselines and their potential impact on variance reduction. More recent works have attempted to replace the fixed λ in GAE with adaptive

Algorithm 1 GA^{2}E: Gradient-Aligned Generalized Advantage Estimation

Input: rollout batch $\mathcal{D}$, reference coefficient λ_{ref},
coarse search set Λ_c, and trajectory ratio $N_{\text{ref}} : N_{\text{cand}}$
Output: selected coefficient $\lambda^\star$ and advantage estimates $\{\hat{A}_t^{\lambda^\star}\}$

(I) Reference-gradient estimation
Split the trajectories in $\mathcal{D}$ into disjoint sets
$\mathcal{D}_{\text{ref}}$ and $\mathcal{D}_{\text{cand}}$ with ratio $N_{\text{ref}} : N_{\text{cand}}$
Compute the reference gradient g_{ref} using λ_{ref}

(II) Coarse search
for $\lambda \in \Lambda_c$ **do**
  Compute candidate advantages $\hat{A}^\lambda$ on $\mathcal{D}_{\text{cand}}$
  Compute the corresponding policy gradient $g(\lambda)$
  Compute the alignment score via (20)
end for
Set $\lambda_c = \arg\max_{\lambda \in \Lambda_c} S(\lambda)$

(III) Local refinement and advantage construction
Construct a local search set $\Lambda_f(\lambda_c)$ around λ_c
for $\lambda \in \Lambda_f(\lambda_c)$ **do**
  Compute $g(\lambda)$ and $S(\lambda)$
end for
Select $\lambda^\star = \arg\max_{\lambda \in \Lambda_f} S(\lambda)$

Return: $\lambda^\star$

mechanisms. AAE (Chen et al., 2021) introduced online bias and variance indicators to adjust the estimation strategy, and subsequent studies further analyzed adaptive advantage estimation methods in continuous control tasks (Unknown, 2024). Other directions investigate alternative advantage formulations, including higher-order return-based estimators (Kaloev & Krastev, 2024) and methods addressing trajectory truncation bias, such as PAE (Song et al., 2025). Meanwhile, direct advantage estimation methods aim to eliminate the dependency on value functions. DAE (Pan & Schölkopf, 2026) learns the advantage function directly and extends the framework to high-dimensional and partially observable settings, while Relative Value Learning (Hauf et al., 2026) models value differences through pairwise representations and derives unbiased advantage estimators. Additional theoretical analyses have examined practical discrepancies in advantage estimation (Unknown, 2020) and the relationship between policy gradient, trust-region optimization, and PPO (Zhao et al., 2023).

A.2 AI Use Statement

In this work, we used generative AI tools to assist with mathematical analysis and proof development, as well as to draft and polish parts of the manuscript. In particular, we used AI tools to help organize and refine the theoretical analysis and mathematical arguments, which we subsequently reviewed, verified, and revised. We also used AI tools to improve the clarity, grammar, and readability of the manuscript and to assist in drafting selected sections.

The research ideas, methodology, implementation, experiments, experimental execution, and analysis of experimental results were developed and conducted by us. We carefully reviewed and verified all AI-assisted content and take full responsibility for the final content, claims, methodology, experiments, and conclusions of the paper.

A.3 Experimental Setup

We evaluate GA^{2}E integrated with PPO on both discrete- and continuous-control benchmarks. For discrete control, we use the MinAtar benchmark, including five environments: Asterix, Breakout, Freeway, Seaquest, and SpaceInvaders. For continuous control, we evaluate on the MuJoCo bench-

Table 2: **Hyperparameter configurations used for PPO and GA^{2}E experiments.** Shared PPO parameters are identical for both methods, while GA^{2}E-specific parameters are listed separately.

Category	Parameter	MinAtar	MuJoCo	Atari
Shared PPO	Batch size	256	512	256
	Rollout length	128	128	128
	PPO epochs	6	10	6
	Number of environments	512	128	1024
	Learning rate	$2.5 \times 10^{-4} \rightarrow 0$ (linear decay)		
	Clip range	$0.1 \rightarrow 0$ (linear decay)		
	Value loss coefficient	0.5		
	Entropy coefficient	0.01	0.0	0.01
GA^{2}E	Reference λ (λ_{ref})	0.99	0.99	0.99
	λ update granularity	Rollout-level	Rollout-level	Rollout-level
	Trajectory ratio ($N_{\text{ref}} : N_{\text{cand}}$)	1:4	1:4	1:4

mark with 9 standard locomotion and manipulation tasks. All methods are trained for 10M environment steps using five independent random seeds under an identical interaction budget.

GA^{2}E is implemented as a plug-in advantage estimation module for PPO, requiring no modification to the original policy optimization objective and introducing no additional learnable parameters. Unless otherwise specified, we adopt the rollout-level gradient consistency estimation variant, where the trace parameter λ is selected after each rollout and kept fixed throughout the subsequent PPO optimization epochs. The selected λ is further smoothed using an exponential moving average (EMA) to improve robustness against non-stationary training dynamics. The epoch-level variant, which updates λ after every PPO optimization epoch, is evaluated in the ablation study.

We compare GA^{2}E against representative advantage estimation methods, including PPO with fixed GAE ($\lambda = 0.95$), DAE, RVL, and EAPO. Except for the advantage estimation module, all methods share identical training protocols, optimization hyperparameters, and environment settings to ensure a fair comparison. Detailed hyperparameters are provided in Table 2. For GA^{2}E, we construct two trajectory sets during rollout: a reference set $\mathcal{D}_{\text{ref}}$ and a candidate set $\mathcal{D}_{\text{cand}}$. The reference set is used to provide stable target estimates, while the candidate set is used for estimating the generalized advantage. We denote their trajectory numbers as N_{ref} and N_{cand}, respectively. In all experiments, we set the trajectory ratio to $N_{\text{ref}} : N_{\text{cand}} = 1 : 4$. Our implementation is built upon the Stable-Baselines3 framework, and the source code is available at `https://anonymous.4open.science/r/PGAE-B4EB`.

A.4 Benchmarks

MinAtar. We use MinAtar (Young & Tian, 2019) to evaluate GA^{2}E in a compact visual discrete-control setting. This benchmark is selected because its small, Atari-inspired environments retain the temporal credit-assignment and exploration challenges of arcade-style tasks while allowing controlled and reproducible comparisons across advantage estimators. We consider five environments: *Asterix*, *Breakout*, *Freeway*, *Seaquest*, and *SpaceInvaders*. These tasks use discrete action spaces and provide diverse interaction patterns, including navigation, object avoidance, target pursuit, and long-horizon score accumulation.

MuJoCo. We use MuJoCo (Todorov et al., 2012) to test whether the benefits of adaptive advantage estimation extend from discrete to continuous control. MuJoCo is a physics-based benchmark suite that exposes the optimizer to varied transition dynamics, action dimensionalities, and reward scales, making it suitable for assessing robustness beyond a single task family. Our experiments include nine locomotion and manipulation tasks: *InvertedPendulum*, *InvertedDoublePendulum*, *Reacher*, *Pusher*, *HalfCheetah*, *Hopper*, *Walker2d*, *Ant*, and *Humanoid*. All of these environments have continuous action spaces and differ substantially in control difficulty and state–action structure.

MiniGrid. We additionally use MiniGrid (Chevalier-Boisvert et al., 2018) to examine GA^{2}E under sparse-reward and exploration-limited conditions. MiniGrid provides lightweight gridworlds with

discrete actions and compact observations, enabling controlled evaluation of long-horizon planning, navigation, retrieval, and key–door manipulation. The sparse-reward study covers ten environments; representative tasks reported in this appendix include *Empty-16x16*, *FourRooms*, *DoorKey-16x16*, *KeyCorridorS6R3*, and *Fetch-8x8-N3*. This benchmark complements MinAtar and MuJoCo by isolating the interaction between advantage estimation and delayed or sparse reward feedback.

Atari. Finally, we report an additional evaluation on the standard 55-game Atari suite (Mnih et al., 2015) to assess generalization to large-scale visual control. Atari provides high-dimensional image observations, long training horizons, and heterogeneous game dynamics, thereby testing whether the proposed trace-parameter adaptation remains effective beyond the compact benchmark settings. The Atari results are presented separately in Appendix C.4. For the primary MinAtar and MuJoCo experiments, all methods are trained for 10M environment steps using five independent random seeds under an identical interaction budget.

A.5 Baselines

PPO. Standard Proximal Policy Optimization (PPO) (Schulman et al., 2017) serves as the primary on-policy actor–critic baseline. PPO stabilizes policy updates through a clipped surrogate objective that constrains the change in the policy probability ratio, while its actor update is driven by an estimated advantage function. We use Generalized Advantage Estimation (GAE) (Schulman et al., 2016) with a fixed $\lambda = 0.95$, following the standard PPO configuration. This baseline therefore provides a direct reference for measuring the effect of replacing a fixed trace parameter with alternative advantage estimation mechanisms.

DAE. Direct Advantage Estimation (DAE) (Pan et al., 2022) learns the advantage function directly from rollout data rather than obtaining it by combining bootstrapped value predictions with multi-step temporal difference residuals. It consequently provides a complementary estimator that reduces reliance on the conventional state-value-based construction of GAE. The DAE implementation used in our experiments supports only discrete action spaces; therefore, DAE is evaluated on the five MinAtar environments and is omitted from the continuous-action MuJoCo experiments.

RVL. Relative Value Learning (RVL) (Höftmann et al., 2026) constructs advantage estimates from relative value differences instead of requiring an absolute value estimate for each state. In particular, RVL models value differences through pairwise representations, making the resulting advantage signal directly reflect the relative quality of competing decisions. This provides a distinct comparison point for GA^{2}E, which selects the trace parameter by aligning the induced policy gradient with a reference direction. RVL supports both discrete and continuous action spaces and is evaluated on MinAtar and MuJoCo.

EAPO. EAPO, or Maximum Entropy Softmax Policy Gradient via Entropy Advantage Estimation, incorporates an entropy-aware advantage signal into policy optimization (Choe & Kim, 2025). Rather than using entropy only as an auxiliary regularization term, EAPO uses entropy-related information in the advantage estimation process, thereby coupling exploration with the direction and magnitude of policy updates. This baseline tests GA^{2}E against an estimator that modifies policy optimization through entropy-aware credit assignment rather than through adaptive selection of the GAE trace parameter. EAPO supports both discrete and continuous action spaces and is evaluated on MinAtar and MuJoCo.

B Proof

B.1 Proof of the First-Order Policy-Oriented Selection Principle

Proof. We first denote the policy update induced by a candidate λ as

$$\Delta\theta(\lambda) = \alpha g(\lambda), \tag{22}$$

where $g(\lambda)$ represents the policy gradient constructed using the advantage estimator parameterized by λ.

Applying the first-order Taylor expansion of the policy objective around the current parameters θ yields

$$\begin{aligned} J(\theta + \Delta\theta(\lambda)) =& J(\theta) + \langle \nabla_\theta J(\theta), \Delta\theta(\lambda) \rangle \\ &+ \mathcal{O}(\|\Delta\theta(\lambda)\|^2). \end{aligned} \tag{23}$$

Substituting $\Delta\theta(\lambda) = \alpha g(\lambda)$ gives

$$\begin{aligned} J(\theta + \alpha g(\lambda)) =& J(\theta) + \alpha \langle \nabla_\theta J(\theta), g(\lambda) \rangle \\ &+ \mathcal{O}(\alpha^2). \end{aligned} \tag{24}$$

Since $J(\theta)$ does not depend on λ, and the step size α is sufficiently small such that the higher-order term can be neglected, maximizing the original selection objective becomes equivalent to maximizing the first-order improvement term:

$$\begin{aligned} \arg\max_\lambda J(\theta + \alpha g(\lambda)) &= \arg\max_\lambda \left(\alpha \langle \nabla_\theta J(\theta), g(\lambda) \rangle\right) \\ &= \arg\max_\lambda \langle \nabla_\theta J(\theta), g(\lambda) \rangle . \end{aligned}$$

Therefore, the proposed policy-oriented selection principle admits the gradient alignment criterion in Eq. (8). □

B.2 Proof of Lemma 1

Proof of Lemma 1. According to the policy gradient formulation,

$$g^\star = \mathbb{E}\left[\nabla_\theta \log \pi_\theta(a|s) A^\pi(s,a)\right].$$

Therefore,

$$g(\hat{A}) - g^\star = \mathbb{E}\left[\nabla_\theta \log \pi_\theta(a|s)(\hat{A}(s,a) - A^\pi(s,a))\right].$$

Taking the Euclidean norm and applying Jensen's inequality gives

$$\|g(\hat{A}) - g^\star\| \le \mathbb{E}\left[\|\nabla_\theta \log \pi_\theta(a|s)\| |\hat{A} - A^\pi|\right].$$

Applying the Cauchy–Schwarz inequality,

$$\|g(\hat{A}) - g^\star\| \le \left(\mathbb{E}[\|\nabla_\theta \log \pi_\theta(a|s)\|^2]\right)^{1/2} \left(\mathbb{E}[(\hat{A} - A^\pi)^2]\right)^{1/2}.$$

By the definition of C, we obtain

$$\|g(\hat{A}) - g^\star\| \le C \|\hat{A} - A^\pi\|_{L^2}.$$

Hence, the lemma follows.

□

B.3 Proof of Lemma 2

Proof of Lemma 2. Let

$$u = g^\star, \qquad v = g_r = u + e.$$

Decompose v into the component parallel to u and the orthogonal component:

$$v = P_u(v) + v_\perp,$$

where

$$P_u(v) = \frac{u^\top v}{\|u\|^2} u.$$

By the definition of the angle between two vectors,

$$\sin\theta = \frac{\|v_\perp\|}{\|v\|}.$$

Since

$$v = u + e,$$

the orthogonal component of v with respect to u originates only from the perturbation term e. Therefore,

$$\|v_\perp\| \le \|e\|.$$

Moreover, the triangle inequality gives

$$\|v\| = \|u + e\| \ge \|u\| - \|e\|.$$

Combining the above inequalities,

$$\sin\theta \le \frac{\|e\|}{\|u\| - \|e\|}.$$

Substituting $u = g^\star$ completes the proof.

□

B.4 Proof of Theorem 1

Proof of Theorem 1. Define normalized vectors

$$\hat{r} = \frac{g_r}{\|g_r\|}, \qquad \hat{t} = \frac{g^\star}{\|g^\star\|}, \qquad \hat{g} = \frac{g}{\|g\|}.$$

Then,

$$\cos(g, g_r) = \hat{g}^\top \hat{r},$$

and

$$\cos(g, g^\star) = \hat{g}^\top \hat{t}.$$

Therefore,

$$|\cos(g, g_r) - \cos(g, g^\star)| = \left|\hat{g}^\top (\hat{r} - \hat{t})\right|.$$

Using the Cauchy–Schwarz inequality,

$$\left|\hat{g}^\top(\hat{r}-\hat{t})\right| \le \|\hat{r}-\hat{t}\|.$$

Because $\hat{r}$ and $\hat{t}$ are unit vectors separated by angle ε,

$$\|\hat{r}-\hat{t}\|^2 = \|\hat{r}\|^2 + \|\hat{t}\|^2 - 2\hat{r}^\top\hat{t}.$$

Since

$$\hat{r}^\top\hat{t} = \cos\varepsilon,$$

we have

$$\|\hat{r}-\hat{t}\|^2 = 2 - 2\cos\varepsilon.$$

Using the trigonometric identity

$$1-\cos\varepsilon = 2\sin^2\frac{\varepsilon}{2},$$

we obtain

$$\|\hat{r}-\hat{t}\| = 2\sin\frac{\varepsilon}{2}.$$

Therefore,

$$|\cos(g,g_r) - \cos(g,g^\star)| \le 2\sin\frac{\varepsilon}{2}.$$

The theorem follows.

□

C Additional Experiments

C.1 Sensitivity to the Reference-Candidate Trajectory Ratio

We investigate how the allocation ratio between reference and candidate trajectories sampled from the rollout buffer affects both final performance and run-to-run stability across 14 environments spanning MinAtar and MuJoCo. Several observations emerge.

Overall, **GA^{2}E exhibits robust performance across different reference-candidate trajectory allocations**. Across the combined benchmark suite, all four configurations achieve competitive performance, without any allocation ratio causing systematic degradation. The performance differences remain limited in most environments, indicating that GA^{2}E does not rely on a narrowly tuned trajectory allocation strategy. For example, near-saturated tasks such as Freeway and Reacher exhibit less than 2% performance variation across all configurations. Furthermore, the average CV on MinAtar remains within a narrow range (0.079–0.116), demonstrating stable optimization behavior under different reference-candidate ratios. These results suggest that GA^{2}E is robust to moderate variations in trajectory allocation.

Furthermore, **the influence of trajectory allocation on optimization stability differs across discrete and continuous control domains**. In MinAtar, all configurations achieve comparable stability, with only minor differences in average CV. In contrast, MuJoCo environments are more sensitive to the trajectory allocation ratio. The 1:4 configuration achieves the lowest average CV (0.106), outperforming 1:2 and 1:3, which obtain higher CV values of 0.151. This behavior can be attributed

Table 3: **Sensitivity to the reference-candidate trajectory ratio in GA^{2}E.** The ratio denotes the number of reference trajectories relative to candidate trajectories sampled from the rollout buffer. Here, IDP and IP denote InvertedDoublePendulum and InvertedPendulum, respectively.

Environment	1:1	1:2	1:3	1:4
		MinAtar		
Asterix	27.4±2.0	28.1±1.0	**28.5±1.1**	27.2±1.4
Breakout	78.8±6.3	89.9±4.2	81.8±16.8	**97.5±21.1**
Freeway	62.4±0.7	61.7±1.3	**62.5±1.3**	62.1±1.2
Seaquest	82.5±8.8	**85.3±7.0**	74.1±16.0	84.4±6.7
SpaceInv.	493.1±15.8	519.6±41.2	494.6±29.8	**534.7±38.6**
Avg. CV ↓	0.060	**0.053**	0.108	0.088
		MuJoCo		
Ant	5570.9±1364.1	5791.0±1214.5	**5999.0±734.1**	5321.6±1080.8
HalfCheetah	9464.6±3103.7	8530.8±4977.4	9466.2±4303.9	**10936.3±3319.2**
Hopper	**2050.7±193.1**	1734.5±177.5	1859.2±309.5	1724.1±270.5
Humanoid	8344.0±1172.6	8619.0±1045.5	**8733.2±476.5**	8214.4±724.2
IDP	9225.2±135.6	9025.0±402.4	7803.7±2489.2	**9249.9±83.9**
IP	834.4±370.2	**1000.0±0.0**	986.7±29.7	**1000.0±0.0**
Pusher	-29.7±0.8	-29.6±1.5	**-29.5±1.1**	-30.4±1.9
Reacher	-4.0±0.1	-4.0±0.1	**-4.0±0.1**	-4.0±0.2
Walker2d	5214.0±1095.0	5046.2±1515.8	5272.7±817.2	**5522.2±715.7**
Avg. CV ↓	0.171	0.160	0.151	**0.111**

to the characteristics of continuous-control optimization, where trajectory-level statistics used by GA^{2}E can be more sensitive to sampling noise. Allocating more trajectories to the candidate set provides a more reliable estimate of the optimization statistics, thereby stabilizing the adaptive horizon selection.

Finally, **the 1:4 configuration provides the best overall performance-stability trade-off**. Although different ratios achieve the best return on individual tasks, the 1:4 configuration consistently maintains strong performance while avoiding large variance across the benchmark suite. For instance, on InvertedDoublePendulum, the 1:3 configuration achieves a comparable final return but exhibits substantially larger variance than 1:4 (2389 vs. 116 standard deviation). Meanwhile, 1:4 achieves strong performance on challenging tasks such as Breakout and HalfCheetah while maintaining the most stable optimization behavior among MuJoCo configurations. Therefore, we adopt the 1:4 reference-candidate trajectory ratio as the default configuration in the main experiments. This choice provides an effective balance between optimization reliability and trajectory efficiency without requiring environment-specific tuning.

C.2 Sensitivity to the Reference Trace Parameter

GA^{2}E constructs the reference gradient using a large trace parameter λ_{ref}. To evaluate the sensitivity of this parameter, we compare three representative configurations, $\lambda_{\text{ref}} \in \{0.97, 0.99, 1.0\}$, while keeping all other hyperparameters unchanged.

Table 4 reports the final performance and stability metrics on MinAtar and MuJoCo. Overall, **GA^{2}E demonstrates robustness to the choice of** λ_{ref}, with all three configurations achieving competitive performance across diverse environments. The performance differences among configurations remain moderate in most tasks, indicating that the reference gradient only requires a reliable approximation of the optimization direction rather than an exact Monte Carlo estimate.

Moreover, $\lambda_{\text{ref}} = 0.99$ consistently provides the most stable optimization behavior. It achieves the lowest average CV on both MinAtar (0.066) and MuJoCo (0.067), substantially reducing variance compared with $\lambda_{\text{ref}} = 1.0$, especially in continuous-control environments. This suggests that using

Table 4: **Sensitivity to the reference trace parameter λ_{ref}.** GA^{2}E is evaluated with $\lambda_{\text{ref}} \in \{0.97, 0.98, 0.99, 1.0\}$ while keeping all other hyperparameters unchanged. The last row of each benchmark reports the average coefficient of variation (CV), where lower values indicate more stable performance. Here, IDP and IP denote InvertedDoublePendulum and InvertedPendulum, respectively.

Environment	0.97	0.98	0.99	1.00
		MinAtar		
Asterix	27.2±1.4	28.1±0.9	27.8±1.3	**28.7±0.7**
Breakout	97.4±21.2	**99.1±30.3**	93.2±17.3	77.1±7.7
Freeway	62.1±1.2	62.5±0.8	**62.8±0.2**	62.3±0.7
Seaquest	84.3±6.7	84.5±6.0	86.6±5.8	**89.8±9.9**
SpaceInv.	**533.3±37.2**	518.2±32.5	506.2±16.9	523.2±39.5
Avg. CV ↓	0.087	0.097	0.067	**0.064**
		MuJoCo		
Ant	5321.6±1080.8	**6063.7±1192.8**	5393.3±1081.9	5987.7±1041.2
HalfCheetah	10936.3±3319.2	8994.8±5928.6	**11461.2±649.4**	8517.6±4312.3
Hopper	1724.1±270.5	1923.2±304.7	**2033.7±139.0**	1937.9±384.8
Humanoid	8214.4±724.2	**9347.2±509.0**	8743.8±519.4	8747.8±815.7
IDP	**9249.9±83.9**	8977.1±698.3	8850.6±472.9	7819.6±2726.0
IP	**1000.0±0.0**	**1000.0±0.0**	**1000.0±0.0**	**1000.0±0.0**
Pusher	-30.4±1.9	-29.6±2.5	**-28.4±1.0**	-28.6±1.1
Reacher	-4.0±0.2	**-3.9±0.1**	-4.0±0.1	-3.9±0.1
Walker2d	5522.2±715.7	5382.9±705.8	**5746.0±533.1**	5121.2±995.0
Avg. CV ↓	0.111	0.154	**0.067**	0.175

a trace parameter slightly smaller than the Monte Carlo limit can effectively balance the accuracy of the reference gradient and the stability of gradient estimation.

Therefore, we adopt $\lambda_{\text{ref}} = 0.99$ as the default configuration throughout the main experiments. This choice provides a stable reference direction while avoiding the increased variance associated with the full Monte Carlo setting.

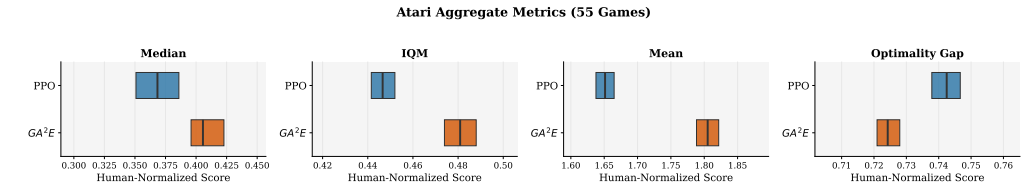
C.3 Generalization to Sparse-Reward Tasks

To evaluate GA^{2}E under sparse-reward settings, we conduct additional experiments on ten MiniGrid environments covering navigation, retrieval, and key-door manipulation tasks. All experiments use rollout-level GA^{2}E with $\lambda_{\text{ref}} = 0.99$, and results are averaged over five random seeds(see Figure 8).

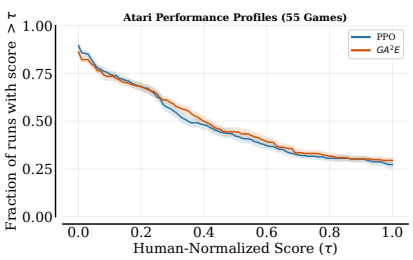
GA^{2}E consistently improves or matches PPO across the evaluated tasks, with the largest gain observed on *Fetch-8x8-N3*, where it improves the final return by approximately 62% and the area under the learning curve (AUC) by about 40%. This demonstrates that adaptive λ selection can provide more effective advantage estimation when rewards are sparse but informative.

On *Empty-16x16*, GA^{2}E achieves performance comparable to PPO, showing that it preserves performance when a fixed advantage estimator is already sufficient. For exploration-limited tasks such as *FourRooms*, *DoorKey-16x16*, and *KeyCorridorS6R3*, both methods exhibit similar performance, indicating that the benefits of adaptive advantage estimation are constrained when learning is dominated by exploration difficulty.

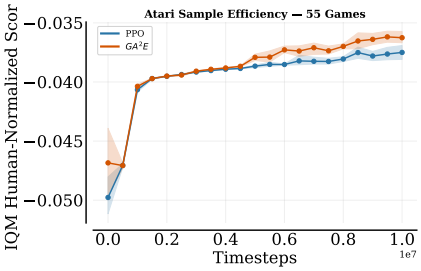
Overall, these results suggest that GA^{2}E improves optimization efficiency in sparse-reward scenarios by adapting advantage estimation, while remaining compatible with environments where exploration is the primary challenge.



(a) Aggregate Metrics

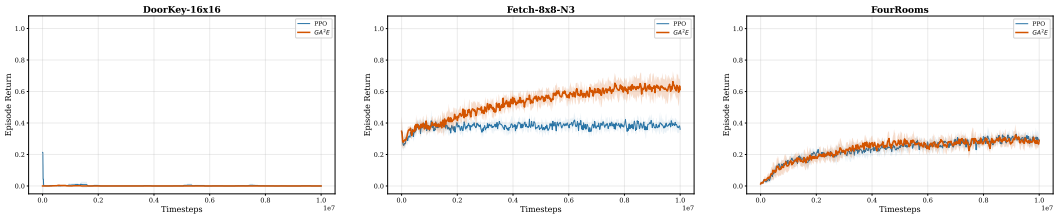


(b) Performance Profile

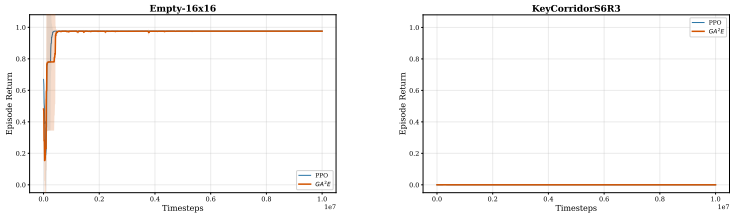


(c) Sample Efficiency

Figure 7: **Comprehensive evaluation on 55 Atari games.** The aggregate metrics are computed using the final evaluation performance of each method. Following the standard Atari evaluation protocol, raw scores are converted into human-normalized scores before calculating IQM, mean, median, and optimality gap. The performance profile compares the distribution of human-normalized scores across games, while the sample efficiency curves characterize the learning progress with respect to environment interactions.



(a) DoorKey-16x16 (b) Fetch-8x8-N3 (c) FourRooms



(d) Empty-16x16 (e) KeyCorridorS6R3

Figure 8: **Learning curves on representative MiniGrid environments.** GA^{2}E consistently achieves competitive performance across diverse sparse-reward tasks. Notably, it substantially improves learning efficiency on *Fetch-8x8-N3*, while maintaining performance comparable to PPO on exploration-dominated tasks such as *DoorKey-16x16* and *FourRooms*.

C.4 Atari Benchmark Experiment Results

To further evaluate the generality of GA^{2}E, we additionally conduct experiments on the Atari benchmark. Table 5 reports the final performance across all 55 Atari games. Compared with PPO, GA^{2}E achieves higher scores on 29 out of 55 environments and obtains comparable performance on another 4 environments (), demonstrating consistent improvements across a broad range of visual control tasks. These results indicate that gradient-alignment-based adaptive λ selection can effectively

Environment	PPO	GA^{2}E	Environment	PPO	GA^{2}E
Alien	1450.1±470.2	**1615.9±420.4**	Montezuma	0.0±0.0	0.0±0.0
Amidar	124.3±15.9	**180.2±66.6**	MsPacman	**1872.7±282.6**	1853.6±281.9
Assault	1474.8±187.8	**1883.0±180.2**	NameThisGame	**5507.5±936.4**	5299.1±958.6
Asterix	1636.6±66.0	**2213.8±205.7**	Phoenix	5093.1±168.1	**5093.4±247.2**
Asteroids	**1051.6±71.9**	931.1±59.7	Pitfall	0.0±0.0	0.0±0.0
Atlantis	**934277.6±16183.6**	901253.1±24640.4	Pong	21.0±0.0	21.0±0.0
BankHeist	237.8±15.0	**304.5±7.4**	PrivateEye	80.0±44.7	80.0±44.7
BattleZone	**20560.0±2676.5**	20487.5±2848.4	Qbert	**4731.4±354.7**	4456.4±226.2
BeamRider	4496.0±181.9	**4825.6±456.5**	Riverraid	5688.2±525.9	**6542.4±496.9**
Berzerk	**798.8±57.8**	709.6±77.0	RoadRunner	12931.0±3569.0	**16602.5±5233.9**
Bowling	**51.7±10.6**	36.9±13.0	Robotank	13.4±1.7	**16.1±2.5**
Boxing	19.2±2.4	**31.2±6.2**	Seaquest	2167.7±243.8	**3318.1±1217.5**
Breakout	121.8±66.3	**160.0±88.3**	Skiing	**-18351.0±10664.7**	-22605.3±9888.3
Centipede	**2868.8±458.3**	2809.1±666.5	Solaris	**692.1±340.2**	435.4±346.1
ChopperCmd	2718.6±699.9	**3550.6±240.8**	SpaceInvaders	655.3±109.6	**700.3±33.2**
CrazyClimber	62546.0±7847.2	**97326.2±6992.5**	StarGunner	4354.4±1529.6	**11918.8±3684.8**
DemonAttack	7145.9±931.9	**10562.1±1238.3**	Tennis	**-0.2±0.7**	-4.5±5.3
DoubleDunk	-3.4±2.2	**-0.8±1.2**	TimePilot	3494.2±636.5	**3901.2±612.7**
Enduro	**0.0±0.1**	0.0±0.0	Tutankham	147.1±55.3	**235.1±22.1**
FishingDerby	-31.6±15.5	**1.8±13.8**	UpNDown	**20951.2±12176.7**	20421.0±5993.1
Freeway	**26.7±3.0**	11.7±16.1	Venture	**10.4±23.3**	0.0±0.0
Frostbite	**338.8±153.9**	336.7±120.1	VideoPinball	215.6±482.0	**964.2±734.1**
Gopher	1257.4±438.2	**1451.6±365.6**	WizardOfWor	**2730.4±1666.0**	2256.9±636.1
Gravitar	**317.8±58.7**	152.2±22.5	YarsRevenge	8567.3±759.3	**8846.0±1359.9**
Hero	12490.6±2192.1	**13737.2±254.9**	Zaxxon	4654.4±612.0	**6022.5±930.9**
IceHockey	**-4.5±0.5**	-5.6±1.3			
Jamesbond	**396.1±39.9**	359.1±99.3			
Kangaroo	1477.2±293.4	**1690.0±227.5**			
Krull	5531.5±316.8	**6056.2±874.8**			
KungFuMaster	**17855.0±4414.9**	16782.5±2411.8			
Avg. Return ↑	20862.9	**21196.9**	**Avg. CV ↓**	**0.254**	0.392

Table 5: **Performance comparison between PPO and GA^{2}E on the 55 Atari benchmark games.** Both methods are evaluated under identical training protocols. Results are reported as the mean episode return over 5 random seeds (± standard deviation). For each seed, the reported score is obtained by evaluating the final policy over 100 episodes. The best result for each game is highlighted in bold. For games with negative rewards (e.g., Skiing, Tennis, and IceHockey), higher (less negative) returns indicate better performance. Overall, GA^{2}E achieves higher returns than PPO on 29 games and matches PPO on 4 additional games (Montezuma, Pitfall, Pong, and PrivateEye), demonstrating competitive performance across a diverse set of Atari environments. The bottom rows report the average return over all 55 games and the average coefficient of variation (CV $= \sigma/|\mu|$ per game; lower is more stable), computed over the 49 games whose mean return exceeds 1 point for both methods.

generalize beyond low-dimensional control tasks and improve advantage estimation in large-scale Atari domains.

Beyond the final scores, we further analyze the overall performance and learning efficiency using the aggregate metrics(Figure 7a), performance profiles(Figure 7b), and sample efficiency curves (Figure 7c). As shown in Figures 7a, GA^{2}E achieves stronger aggregated performance across the Atari suite, indicating consistent improvements rather than gains from a small subset of games. The performance profiles in Figure 7b further demonstrate that GA^{2}E outperforms competing methods over a wide range of normalized scores, while the sample efficiency comparison in Figure 7c shows that GA^{2}E reaches higher performance with fewer environment interactions.

We also investigate the behavior of the dynamically selected λ values. Figure 9 presents the distribution of selected λ values across different Atari environments. The diverse distributions suggest that GA^{2}E does not collapse to a fixed trace coefficient, but instead adapts λ according to task-specific optimization dynamics. This analysis helps illustrate how GA^{2}E dynamically adapts the trace coefficient according to different optimization conditions.

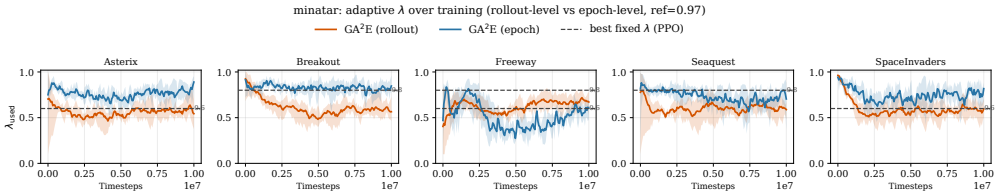


Figure 9: **Dynamic adaptation of the trace parameter λ on the MinAtar benchmark.** The learned λ trajectories are shown for both epoch-level and rollout-level adaptation. The dashed line denotes the best fixed λ for PPO selected from $\{0.0, 0.2, 0.4, 0.6, 0.8, 0.9, 0.95, 0.99\}$ via grid search. Unlike PPO, which relies on a manually selected constant λ, GA^{2}E adapts λ online during training. The corresponding results on the MuJoCo benchmark are shown in Figure 10.

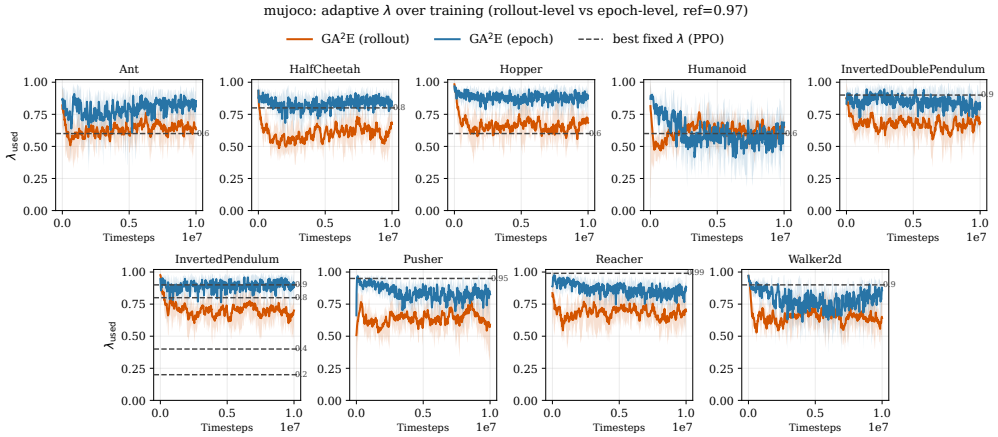


Figure 10: **Dynamic adaptation of the trace parameter λ on the MuJoCo benchmark.** The learned λ trajectories are shown for both epoch-level and rollout-level adaptation. The dashed line denotes the best fixed λ for PPO selected by grid search over $\{0.0, 0.2, 0.4, 0.6, 0.8, 0.9, 0.95, 0.99\}$. Unlike PPO, which relies on a manually selected constant λ, GA^{2}E adapts λ online during training. The corresponding results on the MinAtar benchmark are shown in Figure 9.

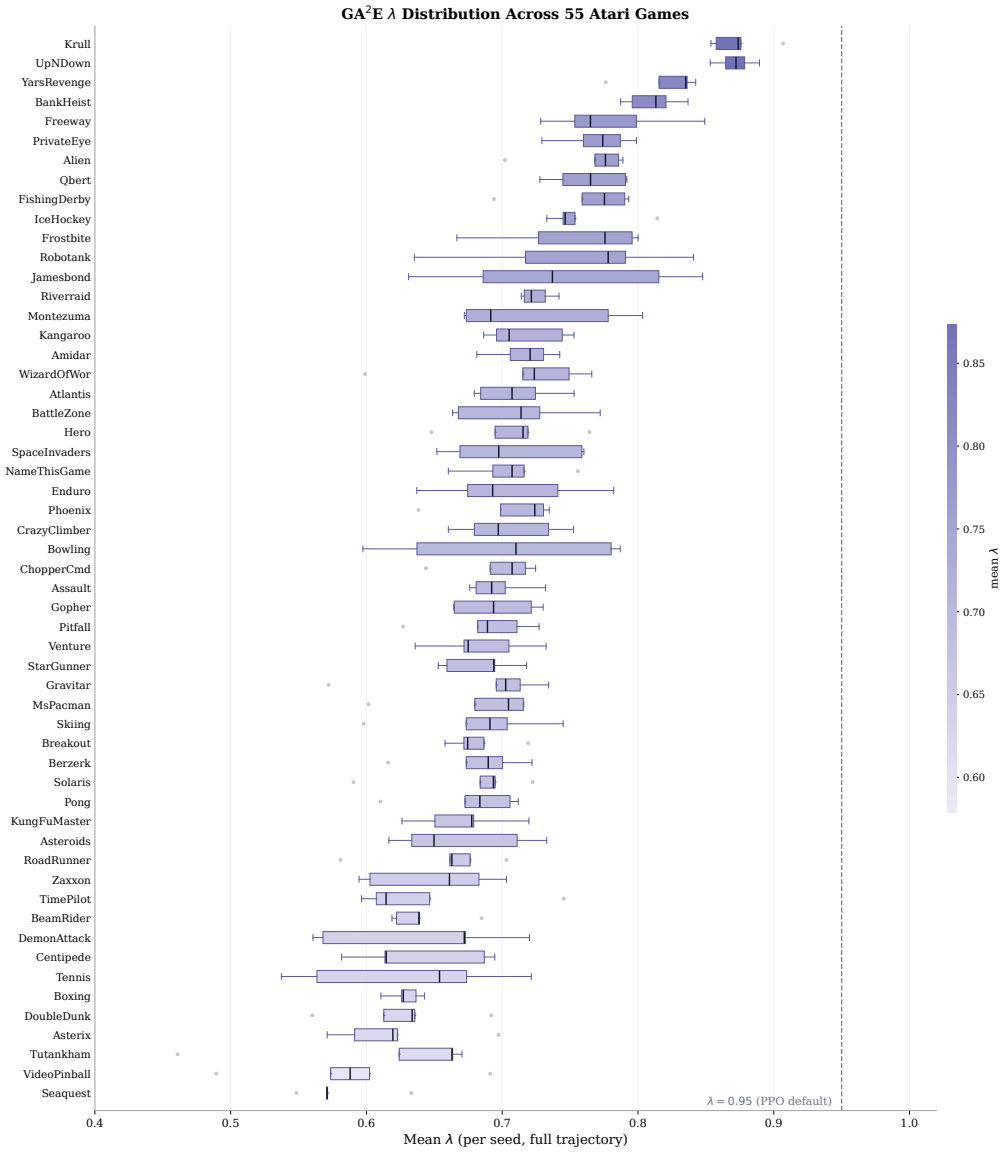
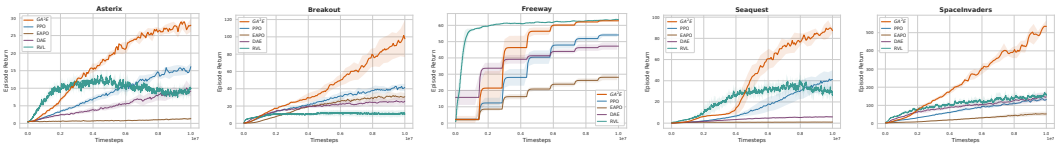
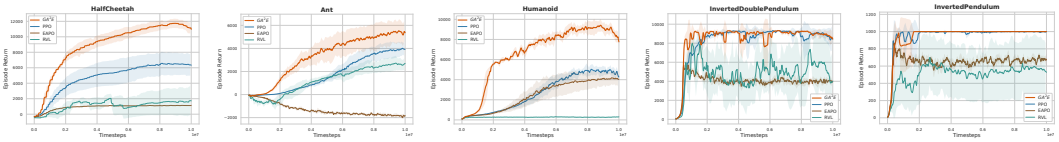


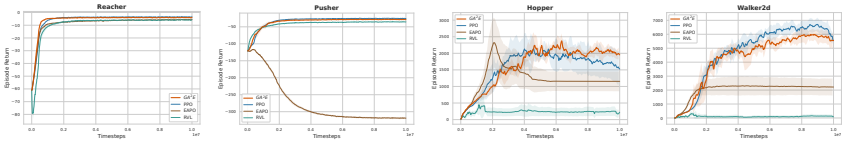
Figure 11: **Empirical distributions of the EMA-smoothed λ values selected by rollout-level GA^{2}E across 55 Atari games.** The distributions are computed from the EMA-smoothed λ trajectories throughout training over five random seeds. The considerable variation across environments indicates that different tasks favor different trace parameters, rather than converging to a common fixed λ.



(a) Asterix (b) Breakout (c) Freeway (d) Seaquest (e) SpaceInvaders

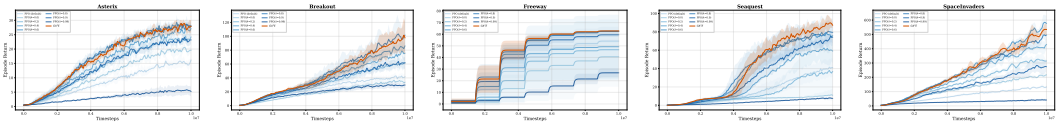


(f) HalfCheetah (g) Ant (h) Humanoid (i) IDP (j) IP

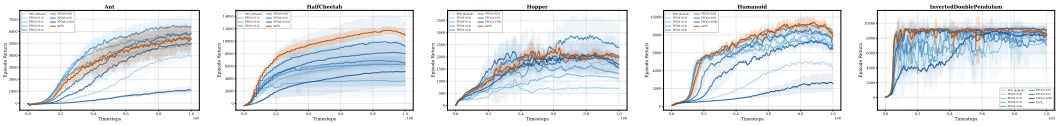


(k) Reacher (l) Pusher (m) Hopper (n) Walker2d

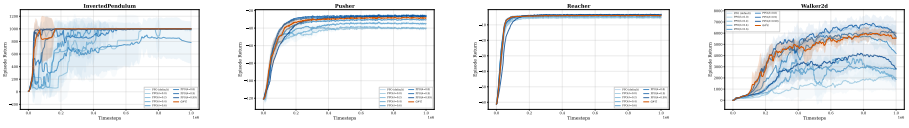
Figure 12: **Learning curves of GA^{2}E and baselines a cross discrete and continuous control benchmarks.** The top row reports results on MinAtar environments, while the second and third rows present results on MuJoCo tasks. Each curve shows the mean episodic return over training, with shaded regions indicating the standard deviation across 5 random seeds. The results demonstrate the effectiveness and robustness of GA^{2}E under diverse environments with different state and action spaces. Here, IDP and IP denote InvertedDoublePendulum and InvertedPendulum, respectively.



(a) Asterix (b) Breakout (c) Freeway (d) Seaquest (e) SpaceInvaders



(f) Ant (g) HalfCheetah (h) Hopper (i) Humanoid (j) IDP



(k) IP (l) Pusher (m) Reacher (n) Walker2d

Figure 13: **Learning curves of PPO and GA^{2}E on the MinAtar and MuJoCo benchmarks.** The first row reports results on the five MinAtar environments, while the second and third rows present nine MuJoCo continuous-control tasks (excluding *HumanoidStandup* and *Swimmer*). GA^{2}E achieves consistently competitive or superior learning performance across a diverse range of discrete and continuous control environments. Here, IDP and IP denote InvertedDoublePendulum and InvertedPendulum, respectively.

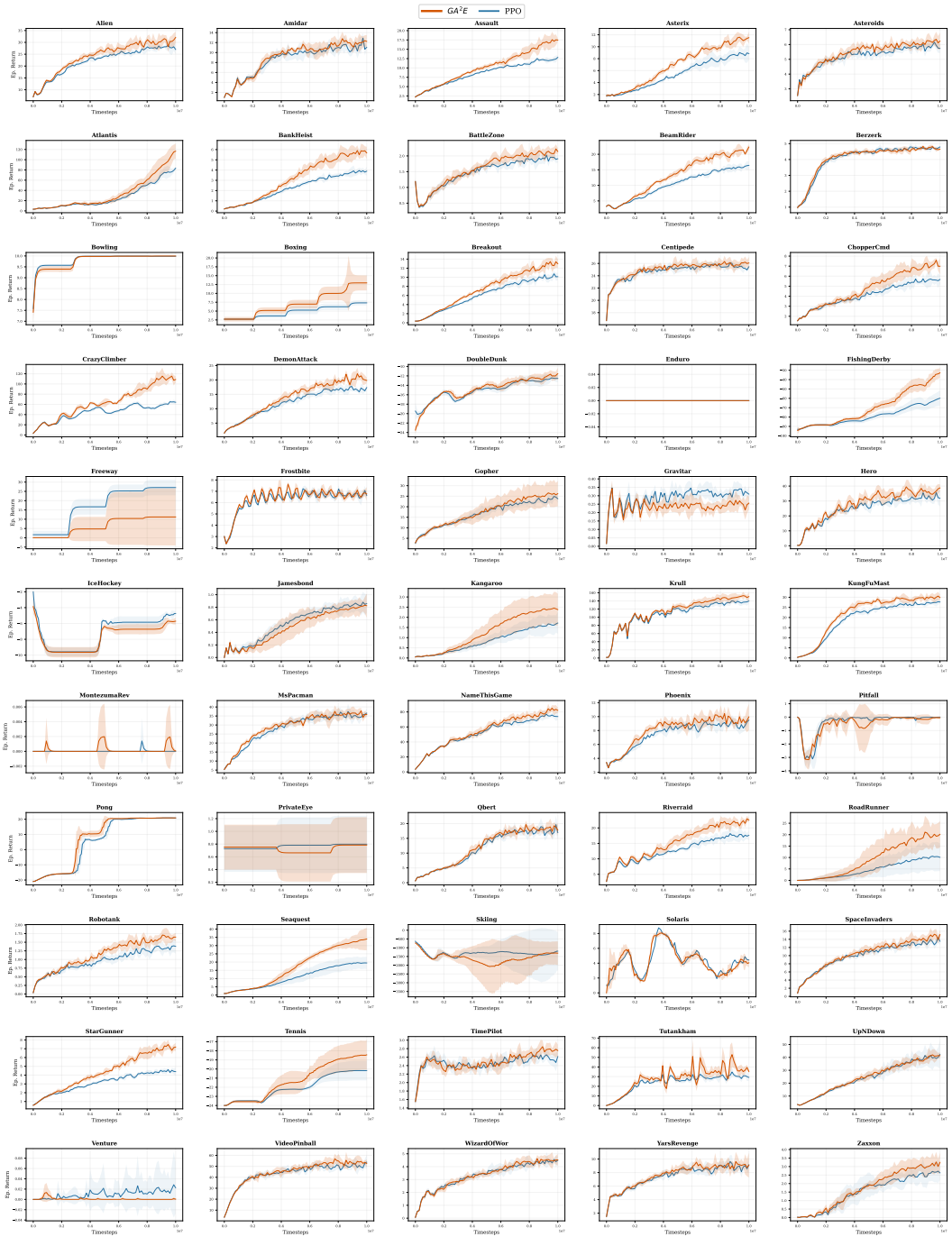


Figure 14: **Training curves of GA^{2}E and PPO across all 55 Atari games.** Each subplot shows the episodic return (EMA-smoothed, $\alpha = 0.7$) over 10M environment steps, averaged over 5 seeds with shaded standard deviation. GA^{2}E (orange) matches or exceeds PPO (blue) on the majority of games (29 out of 55 with a win–tie–loss record of 29–4–22), demonstrating that adaptive λ selection generalizes robustly to large-scale visual control tasks without per-game hyperparameter tuning.