BEYOND EXPECTED RETURNS: TIME-AVERAGE REGULARIZED POLICY OPTIMIZATION FOR TRAJECTORY-LEVEL PERFORMANCE

Anonymous authors
Paper under double-blind review

ABSTRACT

Standard reinforcement learning learns policies by maximizing the expected return, which averages performance over many possible trajectories. However, this objective may not fully capture how individual trajectories behave over time. For instance, the expected return can be heavily influenced by rare outcomes and may therefore differ substantially from the behavior observed along most individual trajectories. An alternative perspective is to consider *time-average* growth, which measures how the return of a single trajectory evolves over time. To incorporate such information into policy optimization, we propose a variant of the well-known Box–Cox transformation from statistics. In particular, along each trajectory, we transform returns and use increments between consecutive transformed returns as an auxiliary signal for policy improvement. We implement this growth signal in proximal policy optimization (PPO) via an actor-only update, while the critic, advantage estimation, and value learning remain unchanged. This design preserves conventional value learning while allowing policy improvement to exploit more trajectory-level information to complement conventional expected-return optimization. We provide theoretical conditions under which this transformation yields a well-defined long-run growth rate for a broad class of return dynamics, including additive and multiplicative special cases. Experiments across continuous-control and Atari benchmarks show that suitable transformation and regularization settings can improve learning in several tasks. These results suggest that transformed-growth signals can provide useful complementary information to expected-return optimization in policy learning.

1 INTRODUCTION

Reinforcement learning (RL) provides a general framework for sequential decision-making and has achieved strong empirical performance in domains ranging from scientific control and language-model alignment to real-world robotics (Degrave et al., 2022; Haarnoja et al., 2024; Luo et al., 2024). The standard RL target is to find a policy π that optimizes the expected return, an *ensemble average* over infinitely many rollouts (Puterman, 2014; Sutton & Barto, 2018). However, when returns are highly variable or path-dependent, or when the underlying dynamics are non-stationary (Baumann et al., 2026), this objective may not fully capture individual trajectories and can favor policies whose average performance is dominated by atypical rollouts.

This motivates the *time-average* perspective. While the expected return is an ensemble average over infinitely many parallel rollouts of a policy, the time-average averages along a single but infinitely long trajectory (Price, 2023; Peters & Adamou, 2025). When ensemble-average and time-average behavior differ, we call the dynamics *non-ergodic* (Birkhoff, 1931) and the ensemble average may poorly represent single-trajectory behavior. In RL, this perspective has been explored through ergodicity transformations (Baumann et al., 2025) and path-dependent formulations (Verbruggen et al., 2025). These approaches motivate trajectory-level criteria, but applying them directly to standard continuous-control benchmarks is challenging because they are restricted to simple systems or Monte Carlo-style RL algorithms.

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.

In this work, we introduce a low-dimensional signed Box–Cox transformation for the cumulative return level and use the resulting transformed growth signal as additional information for policy improvement. That is, we augment the standard expected-return objective with a trajectory-level signal motivated by time-average growth. This keeps the conventional ensemble-level learning signal while allowing time-average to influence the policy update. Building on this idea, we propose a time-average-regularized proximal policy optimization (PPO) algorithm, in which the critic remains the conventional return critic, and the growth signal is introduced only in the actor.

Contributions. We make the following contributions: *(i)* we formulate a mixed policy objective that augments the conventional expected-return objective with a trajectory-level time-average growth objective, and introduce a signed Box–Cox transformation for extracting the corresponding growth signal from cumulative returns; *(ii)* we develop an actor-only time-average-regularized PPO method that combines the conventional PPO advantage with the transformed growth signal; *(iii)* we characterize a class of return dynamics compatible with the signed Box–Cox transformation, establish the resulting policy-level time-average growth under an ergodic state–action process, and derive a finite-horizon variance contraction property; *(iv)* we validate the method on a controlled multiplicative problem and across six MuJoCo and four Atari environments, showing improvements observed across multiple parameter settings in several environments.

In the following, we relate our contributions to the literature.

Time-average growth rate. Time-average growth rates have been studied for non-ergodic systems where appropriate transformations can be derived analytically (Peters & Adamou, 2018; Adamou et al., 2021; Peters & Adamou, 2022). Recent work has extended this perspective to RL and developed different approaches; see (Baumann et al., 2026) for an overview. However, existing approaches are restricted to bandit settings (Verbruggen et al., 2025), discrete action spaces Sheng & Baumann (2025), or Monte Carlo-style RL algorithms Baumann et al. (2025). Different from these works, our approach can be combined with a modern deep RL algorithm such as PPO, making the time-average perspective applicable to challenging and high-dimensional RL benchmarks.

Regularized policy optimization. Policy-gradient methods directly optimize parameterized policies (Sutton et al., 1999), while TRPO and PPO stabilize policy improvement through trust-region or clipped surrogate objectives (Schulman et al., 2015; 2017). Other approaches introduce actor-level regularization through, for example, critic temporal-difference errors or action-distribution perturbations (Parisi et al., 2019; Rahman & Xue, 2022). Our method instead uses a trajectory-level growth signal derived from the return process, while leaving the standard PPO critic unchanged.

Risk-sensitive and distributional reinforcement learning. Risk-sensitive RL also modifies expected-return optimization, for example, through utility-transformed returns, conditional value-at-risk (CVaR), coherent risk measures, or other functionals of the return distribution (Shen et al., 2014; Chow & Ghavamzadeh, 2014; Tamar et al., 2015; Markowitz et al., 2023). Distributional RL goes further by learning the return distribution rather than only its expectation (Bellemare et al., 2017), enabling different risk-sensitive decision criteria. Thus, distributional RL asks *what the return distribution is*, and risk-sensitive RL asks *how to evaluate that distribution*. Our work considers a different question: whether ensemble-level return statistics describe the long-run behavior along an individual trajectory. This time–ensemble distinction is separate from risk preference, consistent with recent work distinguishing temporal and risk dimensions in RL (Moghimi et al., 2026).

2 Problem Setup and Motivation

We start from a standard discounted Markov decision process (MDP) with state space $\mathcal{S}$, action space $\mathcal{A}$, transition kernel P, reward r_t, and discount factor $\gamma \in (0, 1)$. We distinguish two reward-related quantities that play different roles in this work. First, let $R_t = R_0 + \sum_{k=0}^{t-1} r_k$ denote the realized return level up to time t, where R_0 is initialized at the beginning of an episode. This quantity summarizes the past reward evolution along a trajectory. Besides, discounted RL algorithms consider the future return-to-go $\sum_{k=0}^{\infty} \gamma^k r_{t+k}$, where rewards are accumulated additively with discounting, and aim to find a parameterized stochastic policy $\pi_\theta(a \mid s)$ maximizing the additive target $J_{\mathrm{add}}(\theta) := \mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{\infty} \gamma^t r_t\right]$, here, $\pi_\theta(a \mid s)$ denotes the probability of taking action a in state s under policy parameters θ.

For simplicity, we write π for π_θ whenever the dependence on θ is not important. Most actor–critic methods, including PPO, estimate a state-value function $V^\pi(s) = \mathbb{E}_\pi\left[\sum_{t=0}^{\infty} \gamma^t r_t \mid S_0 = s\right]$ and the action-value function $Q^\pi(s,a) := \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k} \,\middle|\, S_t = s,\ A_t = a\right]$. Their difference defines the advantage function $A^\pi(s,a) := Q^\pi(s,a) - V^\pi(s)$ to construct the actor update. In addition to this conventional expected, or ensemble-level, objective, we are interested in the long-run evolution of return along individual trajectories.

2.1 Trajectory-level time-average growth rate

Beyond standard additive reward accumulation, RL problems may involve multiplicative or more general level-dependent dynamics (Baier et al., 2025; Xing et al., 2026). For such path-dependent or non-ergodic dynamics, ensemble averages may not directly characterize the long-run behavior of individual trajectories (Peters, 2019; Baumann et al., 2025).

We consider an infinite-horizon stochastic process generated by a fixed stochastic policy π interacting with the environment. Let $\tau = (S_0, A_0, r_0, S_1, A_1, r_1, \ldots) \sim \mathbb{P}^\pi$ denote a trajectory, where $\mathbb{P}^\pi$ is the trajectory law induced jointly by the policy π and the environment dynamics P.

Our interest is in the long-run evolution of this return level along an individual trajectory. In general, the raw increments $R_{t+1} - R_t = r_t$ need not possess stationary or ergodic statistics, so their ensemble behavior need not directly characterize their long-run behavior along a single realization. This motivates applying a transformation u to the return level R_t and studying the transformed increment at time t:

$$g_t^u(\tau) = u(R_{t+1}(\tau)) - u(R_t(\tau)).$$

The goal of such an *ergodicity transformation* u is to make the transformed increments admit stable long-run statistics.

Definition 2.1 (Ergodicity transformation)**.** For a fixed policy π, we call a monotone transformation u an *ergodicity transformation* for the return process if the transformed increment process $\{g_t^u\}_{t\geq 0}$ is stationary, ergodic, and integrable.

Whenever the limit exists, we define the trajectory-level *time-average growth rate* as

$$\mathcal{G}_u^\pi(\tau) := \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} g_t^u(\tau).$$

This quantity is obtained by averaging the transformed increments over time along a single realization of the stochastic process. In contrast, for a fixed time t, the corresponding ensemble average is

$$\bar{G}_u^\pi := \mathbb{E}_{\tau\sim\mathbb{P}^\pi}\left[g_t^u(\tau)\right].$$

When $\{g_t^u\}_{t\geq 0}$ is stationary, $\bar{G}_u^\pi$ does not depend on t. If u is an ergodicity transformation, the ergodic theorem (Birkhoff, 1931) implies $\mathcal{G}_u^\pi(\tau) = \bar{G}_u^\pi$ almost surely. Hence, although finite realizations may exhibit different transformed growth, the long–run time average converges almost surely to the same policy-dependent ensemble average. This time–ensemble correspondence under an ergodicity transformation underpins the time-average interpretation by Peters & Adamou (2018) and its application to reinforcement learning by Baumann et al. (2025); see also Baumann et al. (2026, Theorem 4.2) for its relation to ergodic Markov processes in RL.

2.2 A mixed additive-growth objective

The time-average above provides a policy-level description of long-run trajectory behavior. For a parameterized policy π_θ, we denote the corresponding ideal growth objective by $J_{\mathrm{grow}}(\theta) := \bar{G}_u^{\pi_\theta}$. Rather than replacing the conventional additive objective, we consider the mixed objective

$$J_{\lambda_g}(\theta) := (1-\lambda_g) J_{\mathrm{add}}(\theta) + \lambda_g J_{\mathrm{grow}}(\theta), \qquad \lambda_g \in [0,1]. \tag{1}$$

The additive term retains the conventional ensemble-level objective, while the growth term introduces complementary information about long-run trajectory behavior. Thus, $J_{\mathrm{grow}}(\theta)$ can be interpreted as a regularizer. The parameter λ_g controls their relative contribution. We use the subscript g to distinguish this parameter from the generalized advantage estimation (GAE) parameter λ_{GAE} used in PPO. In Section 3, we instantiate this idea by using the signed Box–Cox family and incorporate the resulting growth signal into the PPO actor update.

3 TIME-AVERAGE REGULARIZED REINFORCEMENT LEARNING

We now describe the proposed time-average regularized policy optimization method. We build on PPO (Schulman et al., 2017) and introduce the transformed growth signal only through the actor update, while retaining the standard PPO value-learning components. Section 4 provides the theoretical motivation for this actor-level design; here we focus on the algorithm design. We first summarize the PPO components relevant to our method, then introduce the signed Box–Cox growth signal and finally describe how it is incorporated into the PPO actor update.

Standard PPO components. PPO is an actor–critic method consisting of a policy $\pi_\theta(a \mid s)$ and a value function $V_\phi(s)$. The critic estimates the conventional discounted return, while the actor is updated using an advantage estimate that measures the relative quality of the sampled actions. In standard PPO, this advantage is commonly computed using GAE:

$$\hat{A}_t = \sum_{\ell=0}^{\infty} (\gamma\lambda_{\mathrm{GAE}})^\ell \delta_{t+\ell}, \qquad \delta_t = r_t + \gamma V_\phi(s_{t+1}) - V_\phi(s_t), \tag{2}$$

with the usual truncation at episode and rollout boundaries. The resulting advantage $\hat{A}_t$ provides the learning signal for the policy update. Our method retains this signal and augments the actor signal with the trajectory-level growth information introduced below.

Signed Box–Cox Growth Signal. We compute the growth signal from the evolution of cumulative rewards along each trajectory. We now define the Signed Box–Cox transformation, which lets us estimate the transformed growth increment and, eventually, the time-average growth rate.

Definition 3.1 (Signed Box–Cox transformation)**.** Motivated by the Box–Cox transformation (Box & Cox, 1964), for $\alpha \in \mathbb{R}$, we define the signed Box–Cox transformation with 1-shifting as

$$U_\alpha(x) = \begin{cases} \mathrm{sgn}(x)\dfrac{(1+|x|)^\alpha - 1}{\alpha}, & \alpha \neq 0, \\ \mathrm{sgn}(x)\log(1+|x|), & \alpha = 0, \end{cases} \tag{3}$$

where $\mathrm{sgn}(\cdot)$ is the sign function satisfying $\mathrm{sgn}(x) = 1$ for $x > 0$, $\mathrm{sgn}(x) = -1$ for $x < 0$ and $\mathrm{sgn}(0) = 0$.

Britto (2026) also investigates the Box–Cox transformation as an ergodicity transformation. Different from that, we design it with a unit shift and the sign function such that the transformation can adapt to both positive and negative values with the same scale size, and, thus, to more dynamics. Further, in our method, we restrict $\alpha \in [0, 1]$, thereby covering the dynamics between exponential and linear growth (Pirjol et al., 2017). Given the return sequence R_t and signed Box–Cox transformation, we obtain the transformed growth increment t as $g_t^\alpha = U_\alpha(R_{t+1}) - U_\alpha(R_t)$.

At $\alpha = 1$, $U_1(x) = x$ and therefore $g_t^1 = r_t$, whereas $\alpha = 0$ gives the signed logarithmic transformation. Intermediate values continuously interpolate between these two endpoints. If needed in practice, the growth signal is standardized within each rollout to align its numerical scale with the conventional normalized PPO advantage.

Mixed actor update. The transformed growth signal is incorporated only into the actor update. We define the mixed actor weight as

$$\hat{A}_t^{\mathrm{mix}} := (1-\lambda_g)\hat{A}_t + \lambda_g g_t^\alpha, \qquad \lambda_g \in [0, 1].$$

Here, λ_g controls the relative contribution of the standard additive PPO signal and the transformed growth signal. The actor objective is the standard clipped PPO surrogate with $\hat{A}_t^{\mathrm{mix}}$ replacing the conventional PPO advantage:

$$L_{\lambda_g}^{\mathrm{actor}}(\theta) := \mathbb{E}_t\Big[\min\big(\rho_t(\theta)\hat{A}_t^{\mathrm{mix}}, \mathrm{clip}(\rho_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t^{\mathrm{mix}}\big)\Big], \tag{4}$$

where $\rho_t(\theta) := \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\mathrm{old}}}(a_t|s_t)}$ is the probability ratio measuring the probabilistic difference between old and new policy for an action. When $\lambda_g = 0$, the method recovers the baseline PPO actor update, so

Algorithm 1 PPO with signed Box–Cox growth regularization.

Initialize actor parameters θ and critic parameters ϕ
for iteration $= 1, 2, \ldots$ **do**
  Collect a rollout of length T_{roll} using $\pi_{\theta_{\text{old}}}$
  Update cumulative raw returns $R_{t+1} = R_t + r_t$, resetting R_t at episode termination
  Compute transformed growth increments $g_t^{\alpha} = U_{\alpha}(R_{t+1}) - U_{\alpha}(R_t)$
  Compute the conventional GAE advantages $\hat{A}_t$ using Eq. (2)
  for each PPO minibatch **do**
    Form $\hat{A}_t^{\text{mix}} = (1 - \lambda_g)\hat{A}_t + \lambda_g g_t^{\alpha}$
    Update the actor with new actor loss $L_{\lambda_g}^{\text{actor}}$ in Eq. (4)
    Update the critic using the standard PPO value loss

the growth signal has no effect. For $\lambda_g > 0$, the transformed growth signal provides an additional trajectory-dependent contribution to policy improvement.

Algorithm 1 summarizes the above method. The standard PPO critic continues to estimate the additive return and compute the conventional GAE advantage. The signed Box–Cox transformation incorporates information from the cumulative reward process, and we deliberately adopt an actor-only design to isolate the effect of the time-average growth signal. We use this conservative design, since the standard PPO objective still provides useful ensemble-level information, while individual trajectories should not be treated as representative of the entire policy-induced trajectory distribution. At the same time, these trajectories are natural realizations generated by the policy and contain additional information about their realized growth. Therefore, we combine the two signals through λ_g rather than replacing the standard PPO objective.

4 Theoretical Analysis

We now provide the theoretical motivation for the proposed method. In Section 4.1, we first show how to incorporate the growth signal into the actor update when it is directly available. We then characterize return dynamics that allow us to extract the growth rate through our designed transformation in Section 4.2 Finally, we show that the resulting transformed dynamics admit an almost-sure policy-level time-average limit and satisfy a finite-horizon variance contraction property in Section 4.3.

4.1 Actor-level incorporation of the growth signal

Consider a policy π_θ and an ergodicity transformation u. As discussed in Sec. 2, under a fixed policy the time-average growth rate satisfies $\mathcal{G}_u^{\pi_\theta}(\tau) = \bar{G}_u^{\pi_\theta} := \mathbb{E}\left[g_t^u\right]$ almost surely. We therefore take $J_{\text{grow}}(\theta) := \bar{G}_u^{\pi_\theta}$ as the ideal growth objective. Maximizing time-average growth is consistent with prior work on non-ergodic decision problems (Peters & Adamou, 2025; 2018).

To connect this long-run objective to finite sampled trajectories in practice, consider the finite-horizon growth statistic defined as $G_{u,T}(\tau) := \frac{1}{T}\sum_{t=0}^{T-1} g_t^u(\tau)$. Under stationary sampling after transformation u, we naturally obtain

$$\mathbb{E}_{\tau\sim p_\theta}\left[G_{u,T}(\tau)\right] = \bar{G}_u^{\pi_\theta} = J_{\text{grow}}(\theta), \tag{5}$$

where τ is a trajectory statistics and $G_{u,T}(\tau)$ can be regarded as its induced random variable. Hence, for a parameterized policy π_θ, the finite estimate of time-average growth can enter the policy-gradient update directly through the actor, without requiring a Bellman value function for the growth objective. In particular, the likelihood-ratio identity gives the gradient of (5),

$$\nabla_\theta J_{\text{grow}}(\theta) = \mathbb{E}_{\tau\sim p_\theta}\left[G_{u,T}(\tau)\nabla_\theta \log p_\theta(\tau)\right] = \mathbb{E}_{\tau\sim p_\theta}\left[\sum_t \nabla_\theta \log \pi_\theta(A_t \mid S_t) G_{u,T}(\tau)\right],$$

where the environment dynamics are independent of θ. Therefore, although the growth term does not define a standard Bellman advantage, it can still provide a valid policy-gradient signal.

For the additive part, the policy gradient can be written using the standard additive advantage:

$$\nabla_\theta J_{\text{add}}(\theta) = \mathbb{E}_{\tau\sim p_\theta}\left[\sum_t \nabla_\theta \log \pi_\theta(A_t \mid S_t) A_{\text{add}}^{\pi}(S_t, A_t)\right].$$

Here, $A^{\pi}_{\mathrm{add}}(S_t, A_t) = Q^{\pi}_{\mathrm{add}}(S_t, A_t) - V^{\pi}_{\mathrm{add}}(S_t)$ is the standard additive advantage. Therefore, the mixed objective has the formal gradient

$$\begin{aligned}\nabla_\theta J_{\lambda_g}(\theta) &= (1-\lambda_g)\nabla_\theta J_{\mathrm{add}}(\theta) + \lambda_g \nabla_\theta J_{\mathrm{grow}}(\theta) \\ &= \mathbb{E}_{\tau\sim p_\theta}\left[\sum_t \nabla_\theta \log \pi_\theta(a_t \mid s_t)\big((1-\lambda_g)A^{\pi}_{\mathrm{add}}(S_t, A_t) + \lambda_g G_{u,T}(\tau)\big)\right].\end{aligned}$$

This expression $(1-\lambda_g)A^{\pi}_{\mathrm{add}}(S_t, A_t) + \lambda_g G_{u,T}(\tau)$ shows that the growth term can be embedded as an additional trajectory-level weight in the actor update, without requiring a Bellman advantage or a separate growth critic. Our practical PPO update uses the transformed increment $g^u_t(\tau)$ as a local surrogate for this trajectory-level growth signal, rather than explicitly estimating $G_{u,T}(\tau)$.

4.2 Signed Box–Cox Transformation and Compatible Return Dynamics

Having incorporated the growth signal into the actor, we need to see how large a family of dynamics the designed signed Box–Cox transformation can adapt to. We model the return as a one-dimensional continuous Itô process $\mathrm{d}R_t = \mu(R_t, t)\,\mathrm{d}t + \sigma(R_t, t)\,\mathrm{d}W_t$, where μ and σ denote the drift and diffusion coefficients, respectively. In particular, we ask when the transformation can remove the explicit dependence of the drift and diffusion coefficients on the return level, leaving dynamics determined only by the (S_t, A_t) process.

Recall Definition 3.1, To analyze the transformed return dynamics, we first note the regularity properties of U_α required for the generalized Itô formula. For any $\alpha \in \mathbb{R}$, the signed Box–Cox transformation U_α is odd, belongs to $C^1(\mathbb{R})$, and satisfies $U'_\alpha(x) = (1+|x|)^{\alpha-1}$ which is locally absolutely continuous, as shown in Lemma A.1. Its second derivative therefore exists Lebesgue-almost everywhere and, for $x \neq 0$, is given by

$$U''_\alpha(x) = (\alpha-1)\,\mathrm{sgn}(x)(1+|x|)^{\alpha-2}, \qquad x \neq 0. \tag{6}$$

Although U''_α is generally undefined at $x = 0$, its value at this single point is immaterial; we set $U''_\alpha(0) = 0$ by convention. This regularity is sufficient for the Meyer–Itô formula (Protter, 2005, Thm. 70). The full regularity statement in Lemma A.2 are given in Appendix A.1. Given these properties, we now analyze for which return dynamics the signed Box-Cox transformation can removes the explicit dependence on the return level and yields an additive transformed process.

Theorem 4.1 (Return dynamics compatible with the signed Box–Cox transformation)**.** *Let $(R_t)_{t\geq 0}$ be a one-dimensional continuous Itô process satisfying $\mathrm{d}R_t = \mu(R_t, t)\,\mathrm{d}t + \sigma(R_t, t)\,\mathrm{d}W_t$. For $\alpha \in [0, 1]$, define $Y_t = U_\alpha(R_t)$. Then*

$$\begin{aligned}\mathrm{d}Y_t = &\Big[(1+|R_t|)^{\alpha-1}\mu(R_t, t) + \frac{1}{2}(\alpha-1)\,\mathrm{sgn}(R_t)(1+|R_t|)^{\alpha-2}\sigma^2(R_t, t)\Big]\mathrm{d}t \\ &+ (1+|R_t|)^{\alpha-1}\sigma(R_t, t)\,\mathrm{d}W_t.\end{aligned} \tag{7}$$

In particular, let shift and diffusion coefficients depend on the state–action pair, denoted by $Z_t := (S_t, A_t)$, and suppose that $a(Z_t)$ and $b(Z_t)$ are measurable functions. If

$$\mu(R_t, Z_t) = a(Z_t)(1+|R_t|)^{1-\alpha} + \frac{1-\alpha}{2}\,\mathrm{sgn}(R_t)b^2(Z_t)(1+|R_t|)^{1-2\alpha}, \tag{8}$$

$$\sigma(R_t, Z_t) = b(Z_t)(1+|R_t|)^{1-\alpha}, \tag{9}$$

then

$$\mathrm{d}U_\alpha(R_t) = a(Z_t)\,\mathrm{d}t + b(Z_t)\,\mathrm{d}W_t. \tag{10}$$

Thus, return dynamics with coefficients given by (8)–(9) are transformed into the additive dynamics in (10), whose coefficients depend only on the state–action process. The proof is provided in Appendix A.2.

Remark 4.2 (Relation to return dynamics in RL). A natural question is how restrictive this return dynamic family is for RL. Standard RL environments specify transition and reward mechanisms rather than a closed-form stochastic dynamic for the cumulative return, and we are not aware of a general characterization of the induced return-level drift and diffusion coefficients. Therefore, exact compatibility with (8) and (9) is typically difficult to verify a priori. Nevertheless, the compatible

family contains several familiar forms. For $\alpha = 1$, $U_1(x) = x$, and the dynamics reduce to $\mathrm{d}X_t = a\,\mathrm{d}t + b\,\mathrm{d}W_t$, corresponding to additive return dynamics. For $\alpha = 0$, the dynamics become

$$\mathrm{d}X_t = \left[a + \frac{1}{2}b^2 \operatorname{sgn}(X_t)\right](1 + |X_t|)\,\mathrm{d}t + b(1 + |X_t|)\,\mathrm{d}W_t, \tag{11}$$

which is a shifted multiplicative dynamic. Multiplicative reward criteria have also been studied in Markov chains and MDPs (Baier et al., 2025). For intermediate α, the diffusion coefficient scales as $(1 + |X_t|)^{1-\alpha}$, matching the power-law structure of constant-elasticity-of-variance(CEV)-type diffusions, which have also appeared in continuous-time RL (Xing et al., 2026). More generally, continuous-time RL permits state- and action-dependent drift and diffusion coefficients (Jia & Zhou, 2023). Our experiments in Section 5 therefore evaluate the transformation empirically in standard RL environments where the exact return dynamics are not known in closed form.

This characterization also motivates our actor-only implementation. Since we generally do not know the reward dynamics in RL and, hence, also not whether they are of the form defined in Theorem 4.1, directly replacing the environment reward r_t by g_t^U would propagate the potentially misspecified transformation through value estimation, bootstrapping, and advantage estimation. Instead, we use transformed growth only as an auxiliary actor-level signal, with $\lambda_g \in [0, 1]$ controlling its contribution to policy improvement. Next, we move from this local dynamic characterization to the policy level and study whether the resulting transformed growth admits a well-defined long-run time average under a fixed stochastic policy.

4.3 Policy-Level Time-Average Growth Rate

We now consider the long-run behavior of the transformed return dynamics under a fixed stochastic policy π. We assume that the induced state–action process $Z_t = (S_t, A_t)$ is an ergodic Markov chain and therefore admits a stationary state–action distribution $\nu^\pi(s, a) = d^\pi(s)\pi(a \mid s)$, where d^π denotes the stationary state distribution induced by policy π. This assumption ensures that the empirical state–action frequencies converge to their stationary probabilities, allowing the local transformed growth rate to be aggregated into a policy-level time-average growth rate along the trajectory. Under this condition, the compatible dynamics in Theorem 4.1 yield the following result.

Theorem 4.3 (Policy-level time-average growth)**.** *Let π be a fixed stochastic policy and let Z_t be ergodic with stationary state–action distribution $\nu^\pi(s, a) = d^\pi(s)\pi(a \mid s)$. Consider the return-level dynamic $\mathrm{d}R_t = \mu(R_t, t)\,\mathrm{d}t + \sigma(R_t, t)\,\mathrm{d}W_t$. where $\mu(R_t, t)$ and $\sigma(R_t, t)$ satisfy* (8) *and* (9) *respectively. Assume that $a(s, a)$ and $b(s, a)$ are finite for all $(s, a) \in \mathcal{S} \times \mathcal{A}$. Applying the signed Box–Cox transformation for $\alpha \in [0, 1]$, there exists*

$$\frac{U_\alpha(R_T) - U_\alpha(R_0)}{T} \xrightarrow[T\to\infty]{\text{a.s.}} \bar{G}_\alpha^\pi, \quad \textit{with} \quad \bar{G}_\alpha^\pi = \sum_{s\in\mathcal{S}} d^\pi(s) \sum_{a\in\mathcal{A}} \pi(a \mid s) a(s, a).$$

Theorem 4.3 shows that, for a fixed policy, the long-run transformed growth is determined by the stationary state–action occupancy and the corresponding local transformed drifts $a(s, a)$.

Note that when $\alpha = 0$, the compatible dynamics reduce branch-wise to a shifted multiplicative form. On the positive branch, with $X_t = 1 + R_t$, the dynamics are a geometric Brownian motion (GBM, as studied by Sheng & Baumann (2025, Proposition 1), until the process leaves that branch. The signed formulation considered here extends the approach by Sheng & Baumann (2025) to return levels of either sign, applying the same symmetric rescaling to positive and negative values without requiring the process to remain globally positive.

The preceding result is asymptotic. For the finite-time case, we treat the trajectory τ as a random variable; thus, R_t is also a random variable correlated with time t. We then show that the signed Box–Cox transformation controls the return-level variance.

Proposition 4.4 (Finite-horizon variance contraction)**.** *Let $\alpha \in [0, 1]$ and R_0 be deterministic. At time T, there exists*

$$\operatorname{Var}\left(\frac{1}{T}\sum_{t=0}^{T-1} g_t^\alpha\right) = \operatorname{Var}\left(\frac{U_\alpha(R_T) - U_\alpha(R_0)}{T}\right) \le \operatorname{Var}\left(\frac{R_T - R_0}{T}\right).$$

Thus, before taking the long-run limit, the transformed time-average estimator has variance no larger than that of the corresponding raw return-level average. The proofs of Theorem 4.3 and Proposition 4.4 are provided in Appendix A.3.

Overall, the above results provide three complementary properties supporting our method. First, the policy gradient shows that transformed trajectory growth can enter policy optimization directly through the actor, without requiring a Bellman advantage or a separate growth critic. Second, Theorem 4.1 characterizes a family of return dynamics for which the signed Box–Cox transformation removes the explicit dependence of the drift and diffusion coefficients on the return level, yielding an additive transformed process. Third, under an ergodic state–action process induced by a fixed policy, Theorem 4.3 establishes an almost-sure policy-level time-average growth rate, while Proposition 4.4 provides a complementary finite-horizon variance contraction property. Together, these results motivate using the signed Box–Cox growth signal as an auxiliary actor-level signal while retaining the standard additive critic and value-learning mechanism.

5 Simulation Results

We evaluate our method in a controlled multiplicative coin-toss game (Baumann et al., 2025), on six MuJoCo (Todorov et al., 2012) continuous-control, and four Atari (Bellemare et al., 2013) environments. For the coin-toss experiments, we use the PPO implementation in Stable-Baselines3 (Raffin et al., 2021), while we use the ClearnRL (Huang et al., 2022) implementation for the MuJoCo and Atari experiments. For all experiments, we train each configuration with five random seeds. We sweep the transformation parameter α and the growth-regularization weight λ_g and present representative results below, while reporting the full sweeps and their results in the corresponding appendices. For the MuJoCo and Atari benchmarks, we evaluate each trained policy on the same 100 held-out environment seeds. We average returns across the 100 evaluation episodes for each training seed, and then aggregate the resulting scores across the five training seeds.

Coin-toss. We first consider a controlled multiplicative coin-toss environment adapted from Sheng & Baumann (2025). The environment has two states $s \in \{0, 1\}$ for heads and tails, each occurring with probability 0.5, with the policy choosing a state-dependent betting fraction $a(s) \in [0, 1]$. The return evolves as $R_{t+1} = 1.5a(1)R_t$ for $s = 1$ and $R_{t+1} = 0.6a(0)R_t$ for $s = 0$. Thus, the environment exhibits multiplicative return dynamics. The time-average growth-optimal action is $a^\star = 0.25$ in both states. The PPO baseline learns substantially larger betting fractions, which lead to a negative growth rate and, thus, $R_t \to 0$ almost surely, whereas our method with $(\alpha, \lambda_g) = (0.25, 0.75)$ learns $a(0) = 0.260 \pm 0.033$ and $a(1) = 0.241 \pm 0.051$, close to $a^\star$. Thus, while the baseline shows negative growth, our method learns an a that yields a winning strategy with R_t growing exponentially. These results show that the nonlinear transformed growth signal shifts the learned policy toward the behavior favored by the time-average criterion. We report full coin-toss results, including the parameter sweep, learned actions, and rollout trajectories illustrating the behavior of different policies in Appendix B.

Standard RL benchmarks. We further evaluate the method across ten standard deep RL environments: six MuJoCo continuous-control tasks and four Atari environments. Together, these benchmarks cover continuous-action control and discrete-action decision making from high-dimensional observations. Alongside conventional mean and median returns, we report the interquartile mean (IQM) and 95% confidence intervals (CIs), following Agarwal et al. (2021), providing a robust performance summary together with an estimate of uncertainty.

Table 1 reports results for two MuJoCo and two Atari environments, with two selected (α, λ_g) configurations for each task. The remaining results and training curves are collected in Appendix C. Across these four environments, the selected growth-regularized configurations (α, λ) improve the mean, median, and IQM over the PPO baseline. For Beamrider, our method's median return is even double the baseline's. The complete parameter sweeps in Tables 3–13 in Appendix C show that these improvements are not confined to isolated hyperparameter choices: in several environments, multiple (α, λ_g) configurations outperform baseline PPO. The training and periodic evaluation curves in Figures 2–3 further show that improvements are consistent throughout training. Ant, HalfCheetah, Reacher, BeamRider, and Seaquest show sustained improvement over PPO; only Hopper and Walker2d improve mainly at later stages of training.

Table 1: Final evaluation performance comparison between the PPO baseline and two selected configurations on two MuJoCo and two Atari environments. We evaluate each training seed on 100 held-out environment seeds, and compute all reported statistics from the resulting five per-seed mean returns. The best values per environment are highlighted in bold. We see that, for all environments, our algorithm performs better with reduced stochasticity.

Environment	Configuration	Mean	Median	Std	IQM	95% CI
Ant	Baseline	2675.6	2660.7	360.9	2561.1	[2388.4, 3132.3]
	$\alpha = 0.00,\ \lambda_g = 0.03$	3361.6	3511.5	**524.0**	3394.0	[2728.7, 3938.1]
	$\alpha = 0.25,\ \lambda_g = 0.10$	**3638.7**	**3862.0**	582.8	**3597.9**	**[2960.0, 4298.9]**
Hopper	Baseline	2915.8	3343.4	580.7	2904.5	[2231.7, 3539.5]
	$\alpha = 0.25,\ \lambda_g = 0.05$	3463.6	**3526.8**	**101.8**	**3494.3**	**[3331.1, 3550.3]**
	$\alpha = 0.25,\ \lambda_g = 0.10$	**3476.6**	3500.1	142.6	3493.6	[3297.7, 3630.0]
BeamRider	Baseline	2919.4	2608.0	568.8	3144.6	[2219.4, 3335.2]
	$\alpha = 0.00,\ \lambda_g = 0.70$	**5565.0**	**5268.0**	**464.7**	**5605.8**	**[5037.2, 6097.6]**
	$\alpha = 0.50,\ \lambda_g = 0.50$	4956.3	5053.0	487.6	4974.4	[4380.4, 5522.3]
Seaquest	Baseline	1638.5	1820.0	359.9	1792.1	[1200.6, 1875.4]
	$\alpha = 0.75,\ \lambda_g = 0.05$	**2009.1**	**1840.0**	338.2	**1841.5**	**[1836.9, 2405.1]**
	$\alpha = 0.75,\ \lambda_g = 0.50$	1967.7	1830.0	**303.2**	1821.3	[1805.1, 2325.6]

At the same time, the benefit of our method is task-dependent. For Swimmer, Breakout, and Qbert, the returns remain largely comparable to the baseline. This task dependence appears related to the effective horizon of the return process. In most MuJoCo tasks, a successful policy is expected to sustain good behavior over long trajectories, and we observe a similar pattern in BeamRider and Seaquest under our training and evaluation setup. Such return processes align very well with our time perspective. In contrast, Breakout and Qbert exhibit more pronounced game-ending boundaries, which shorten the horizon over which cumulative return can evolve. For such dynamics, an average over an infinitely long trajectory is less meaningful. Swimmer is an exception: although it is also a long-horizon locomotion task, its reward process may become relatively stable once a consistent swimming behavior is learned, leaving less additional structure for the transformed growth signal to exploit. These observations suggest that the benefit of the growth signal may depend on the structure of the cumulative-return process. Crucially, depending on the environment, incorporating the growth signal can significantly improve performance or have a modest impact—but, with proper hyperparameter tuning, it will not hamper performance.

Taken together, the coin-toss experiment shows that the transformed growth signal can steer policy learning toward time-average behavior. The MuJoCo and Atari results further show that it can improve performance in standard RL tasks, although the benefit varies across environments.

6 Conclusion

We introduced a time-average regularized policy optimization framework that augments the conventional expected-return objective with a trajectory-level growth signal. Using a low-dimensional signed Box–Cox transformation, the proposed method incorporates the transformed growth signal directly into the PPO actor while leaving the conventional critic unchanged. Our analysis characterizes a family of return dynamics for which the transformation removes the explicit dependence on the accumulated return level and yields additive transformed dynamics, and establishes the corresponding policy-level time-average growth rate.

The controlled coin-toss experiment illustrates the effect of the transformed growth signal, while experiments on MuJoCo and Atari show improvements over PPO in several standard environments. The effective transformation and regularization strength remain environment-dependent, motivating future work on automatically selecting α and λ_g within this low-dimensional framework. Another direction is to investigate extensions to history-dependent policies and partially observable settings using appropriate information states. More broadly, this work highlights return-level dynamics as a useful perspective for designing RL objectives that explicitly account for trajectory-level behavior beyond conventional expected-return optimization.

AI USE STATEMENT

In this work, we used generative AI tools to assist with language editing, to discuss the feasibility of author-generated research ideas, and to support software implementation, code editing, and debugging. We additionally used generative AI tools for literature search and discovery and for improving the presentation and readability of the manuscript.

All research ideas and methodological decisions were made by the authors. References identified with the assistance of generative AI were checked against the original sources. AI-assisted code was manually reviewed and validated through our own testing and experiments. Mathematical claims and proofs were independently developed and verified by the authors. Generative AI tools were also used to assist with the LaTeX typesetting and presentation of author-provided mathematical proofs. We take responsibility for the final content of this work, including text, claims, code, and other artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We provide the algorithmic details, implementation choices, hyperparameters, and experimental settings in the main text and appendix. Complete proofs of the theoretical results are provided in the appendix. Additional experimental results, ablations, and implementation details are also included in the appendix. All experiments are conducted over multiple random seeds, and the reported results are aggregated across seeds.

The supplementary material contains the code used for all CoinToss, MuJoCo, and Atari experiments, together with dependency information and reproduction instructions. The coin-toss experiments build on Stable-Baselines3 PPO (Raffin et al., 2021), while the MuJoCo and Atari experiments build on CleanRL (Huang et al., 2022) the corresponding standard PPO implementations described in the experimental setup. All modifications introduced by our method are included in the supplementary code.

REFERENCES

Alexander Adamou, Yonatan Berman, Diomides Mavroyiannis, and Ole Peters. Microfoundations of discounting. *Decision Analysis*, 18(4):257–272, 2021.

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

Christel Baier, Krishnendu Chatterjee, Tobias Meggendorfer, and Jakob Piribauer. Multiplicative rewards in Markovian models. In *ACM/IEEE Symposium on Logic in Computer Science*, pp. 499–512, 2025.

Dominik Baumann, Erfaun Noorani, James Price, Ole Peters, Colm Connaughton, and Thomas B Schön. Reinforcement learning with non-ergodic reward increments: robustness via ergodicity transformations. *Transactions on Machine Learning Research*, 2025.

Dominik Baumann, Erfaun Noorani, Arsenii Mustafin, Xinyi Sheng, Bert Verbruggen, Arne Vanhoyweghen, Vincent Ginis, and Thomas B Schön. Ergodicity in reinforcement learning. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 2026.

Marc G Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47: 253–279, 2013.

Marc G Bellemare, Will Dabney, and Rémi Munos. A distributional perspective on reinforcement learning. In *International Conference on Machine Learning*, pp. 449–458. PMLR, 2017.

George D Birkhoff. Proof of the ergodic theorem. *Proceedings of the National Academy of Sciences*, 1931.

George EP Box and David R Cox. An analysis of transformations. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 26(2):211–243, 1964.

Anthony Britto. Likelihood-based ergodicity transformations in time series analysis. *arXiv preprint arXiv:2601.11237*, 2026.

Yinlam Chow and Mohammad Ghavamzadeh. Algorithms for CVaR optimization in MDPs. *Advances in Neural Information Processing Systems*, 27, 2014.

Jonas Degrave, Federico Felici, Jonas Buchli, Michael Neunert, Brendan Tracey, Francesco Carpanese, Timo Ewalds, Roland Hafner, Abbas Abdolmaleki, Diego de Las Casas, et al. Magnetic control of tokamak plasmas through deep reinforcement learning. *Nature*, 602(7897):414–419, 2022.

Tuomas Haarnoja, Ben Moran, Guy Lever, Sandy H Huang, Dhruva Tirumala, Jan Humplik, Markus Wulfmeier, Saran Tunyasuvunakool, Noah Y Siegel, Roland Hafner, et al. Learning agile soccer skills for a bipedal robot with deep reinforcement learning. *Science Robotics*, 9(89):eadi8022, 2024.

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

Oliver Hulme, Arne Vanhoyweghen, Colm Connaughton, Ole Peters, Simon Steinkamp, Alexander Adamou, Dominik Baumann, Vincent Ginis, Bert Verbruggen, James Price, et al. Reply to "the limitations of growth-optimal approaches to decision making under uncertainty". *Econ Journal Watch*, 20(2):335–348, 2023.

Yanwei Jia and Xun Yu Zhou. Q-learning in continuous time. *Journal of Machine Learning Research*, 24(161):1–61, 2023.

Shuzhen Luo, Menghan Jiang, Sainan Zhang, Junxi Zhu, Shuangyue Yu, Israel Dominguez Silva, Tian Wang, Elliott Rouse, Bolei Zhou, Hyunwoo Yuk, et al. Experiment-free exoskeleton assistance via learning in simulation. *Nature*, 630(8016):353–359, 2024.

Jared Markowitz, Ryan W Gardner, Ashley Llorens, Raman Arora, and I-Jeng Wang. A risk-sensitive approach to policy optimization. In *AAAI Conference on Artificial Intelligence*, volume 37, pp. 15019–15027, 2023.

Mehrdad Moghimi, Anthony Coache, and Hyejin Ku. Decoupling time and risk: risk-sensitive reinforcement learning with general discounting. *arXiv preprint arXiv:2602.04131*, 2026.

Simone Parisi, Voot Tangkaratt, Jan Peters, and Mohammad Emtiyaz Khan. TD-regularized actor-critic methods. *Machine Learning*, 108(8):1467–1501, 2019.

Ole Peters. The ergodicity problem in economics. *Nature Physics*, 15(12):1216–1221, 2019.

Ole Peters and Alexander Adamou. The time interpretation of expected utility theory. *arXiv preprint arXiv:1801.03680*, 2018.

Ole Peters and Alexander Adamou. The ergodicity solution of the cooperation puzzle. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 380(2227), 2022.

Ole Peters and Alexander Adamou. *An Introduction to Ergodicity Economics*. LML Press, 2025.

Dan Pirjol, Farshid Jafarpour, and Srividya Iyer-Biswas. Phenomenology of stochastic exponential growth. *Physical Review E*, 95(6):062406, 2017.

James Price. *The dynamics of decision making with uncertain outcomes*. PhD thesis, University of Warwick, 2023.

Philip E Protter. Stochastic integration and differential equations. In *Stochastic Integration and Differential Equations*. Springer, 2005.

Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

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.

Md Masudur Rahman and Yexiang Xue. Robust policy optimization in deep reinforcement learning. *arXiv preprint arXiv:2212.07536*, 2022.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International Conference on Machine Learning*, pp. 1889–1897. PMLR, 2015.

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

Yun Shen, Michael J Tobia, Tobias Sommer, and Klaus Obermayer. Risk-sensitive reinforcement learning. *Neural Computation*, 26(7):1298–1328, 2014.

Xinyi Sheng and Dominik Baumann. Beyond expected value: geometric mean optimization for long-term policy performance in reinforcement learning. In *IEEE Conference on Decision and Control (CDC)*, pp. 4154–4160, 2025.

Richard S Sutton and Andrew G Barto. *Reinforcement Learning: An Introduction*. MIT Press, 2018.

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*, volume 12, 1999.

Aviv Tamar, Yinlam Chow, Mohammad Ghavamzadeh, and Shie Mannor. Policy gradient for coherent risk measures. In *Advances in Neural Information Processing Systems*, volume 28, 2015.

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.

B Verbruggen, A Vanhoyweghen, and V Ginis. Reinforcement learning in path-dependent, non-ergodic contexts. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 481(2314), 2025.

Xiaoyu Xing, Xingtian Zhang, and Jiarou Luo. Exploratory mean-variance portfolio selection with constant elasticity of variance models in regime-switching markets. *Journal of Industrial and Management Optimization*, 22(2):1087–1111, 2026.

A Appendix A

A.1 Properties of the signed Box–Cox transformation

Lemma A.1 (Basic properties)**.** *For $\alpha \in \mathbb{R}$, recall the signed Box–Cox transformation*

$$U_\alpha(x) = \begin{cases} \operatorname{sgn}(x)\dfrac{(1+|x|)^\alpha - 1}{\alpha}, & \alpha \neq 0, \\ \operatorname{sgn}(x)\log(1+|x|), & \alpha = 0. \end{cases}$$

Then the following properties hold.

1. ***Linear and logarithmic endpoints.***

$$U_1(x) = x, \qquad U_0(x) = \operatorname{sgn}(x)\log(1+|x|),$$

where

$$\lim_{\alpha \to 0} \operatorname{sgn}(x)\frac{(1+|x|)^\alpha - 1}{\alpha} = U_0(x).$$

2. ***Symmetry and monotonicity.*** *The transformation is odd,*

$$U_\alpha(-x) = -U_\alpha(x),$$

and strictly increasing. In particular,

$$U'_\alpha(x) = (1+|x|)^{\alpha-1} > 0,$$

with $U'_\alpha(0) = 1$ defined by continuity.

3. ***Ordering and growth-direction preservation.*** *For any $x, y \in \mathbb{R}$,*

$$x < y \quad \Longleftrightarrow \quad U_\alpha(x) < U_\alpha(y).$$

4. ***1-Lipschitz continuity.*** *For $\alpha \in [0,1]$,*

$$|U_\alpha(x) - U_\alpha(y)| \le |x-y|, \qquad x, y \in \mathbb{R}.$$

Proof. 1. For $\alpha = 1$, $U_1(x) = \operatorname{sgn}(x)|x| = x$. For the logarithmic endpoint, using

$$\lim_{\alpha\to 0} \frac{z^\alpha - 1}{\alpha} = \log z, \qquad z > 0,$$

with $z = 1 + |x|$ gives

$$\lim_{\alpha\to 0} U_\alpha(x) = \operatorname{sgn}(x)\log(1+|x|) = U_0(x).$$

2. Oddness follows directly from the dependence on $|x|$ together with the sign function $\operatorname{sgn}(x)$. Besides, since for $x > 0$, there is

$$U'_\alpha(x) = (1+x)^{\alpha-1},$$

while for $x < 0$,

$$U'_\alpha(x) = (1-x)^{\alpha-1}.$$

Hence, for $x \neq 0$,

$$U'_\alpha(x) = (1+|x|)^{\alpha-1}.$$

Moreover,

$$\lim_{x\to 0^-} U'_\alpha(x) = \lim_{x\to 0^+} U'_\alpha(x) = 1,$$

so that $U'_\alpha(0) = 1$. Since $U'_\alpha(x) > 0$ for all x, U_α is strictly increasing.

3. The ordering property then follows from monotonicity immediately.

4. Finally, for $\alpha \in [0,1]$,

$$0 < U'_\alpha(x) = (1+|x|)^{\alpha-1} \le 1.$$

By the mean-value theorem, for any $x, y \in \mathbb{R}$,

$$|U_\alpha(x) - U_\alpha(y)| \le \sup_{z\in[x,y]} |U'_\alpha(z)|\,|x-y| \le |x-y|.$$

Thus U_α is globally 1-Lipschitz. □

Lemma A.2 (Regularity of the signed Box–Cox transformation)**.** *For any $\alpha \in \mathbb{R}$, $U_\alpha \in C^1(\mathbb{R})$ and U'_α is locally absolutely continuous. For $x \neq 0$,*

$$U''_\alpha(x) = (\alpha-1)\operatorname{sgn}(x)(1+|x|)^{\alpha-2}. \tag{12}$$

For $\alpha = 1$, $U_1(x) = x$ and hence $U''_1(x) = 0$ for all x. For $\alpha \neq 1$, the second derivative does not exist at the origin, since the one-sided limits satisfy

$$\lim_{x\to 0^-} U''_\alpha(x) = 1-\alpha, \qquad \lim_{x\to 0^+} U''_\alpha(x) = \alpha - 1.$$

Nevertheless, U''_α exists Lebesgue-almost everywhere.

Proof. From Lemma A.1,

$$U'_\alpha(x) = (1 + |x|)^{\alpha-1},$$

with $U'_\alpha(0) = 1$ by continuity. Therefore $U'_\alpha \in C(\mathbb{R})$ and $U_\alpha \in C^1(\mathbb{R})$.

For $x \neq 0$, differentiating once more gives

$$U''_\alpha(x) = (\alpha - 1)\,\mathrm{sgn}(x)(1 + |x|)^{\alpha-2}.$$

Since this expression is locally integrable and

$$U'_\alpha(x) = U'_\alpha(0) + \int_0^x U''_\alpha(y)\,\mathrm{d}y,$$

U'_α is locally absolutely continuous.

For $\alpha = 1$, $U_1(x) = x$, and hence $U''_1(x) = 0$ for all x. For $\alpha \neq 1$,

$$\lim_{x\to 0^-} U''_\alpha(x) = 1 - \alpha, \qquad \lim_{x\to 0^+} U''_\alpha(x) = \alpha - 1.$$

which are unequal. Thus the classical second derivative does not exist at the origin, while U''_α exists Lebesgue-almost everywhere. □

A.2 Proof of Theorem 4.1

Proof. By Lemma A.2, U_α satisfies the regularity conditions required by the generalized Itô formula. We use the almost-everywhere representative $U''_\alpha(0) = 0$ at the origin. Applying the generalized Itô formula to $Y_t = U_\alpha(R_t)$ gives

$$\mathrm{d}Y_t = \left[U'_\alpha(R_t)\mu(R_t, Z_t) + \frac{1}{2}U''_\alpha(R_t)\sigma^2(R_t, Z_t)\right]\mathrm{d}t + U'_\alpha(R_t)\sigma(R_t, Z_t)\,\mathrm{d}W_t. \tag{13}$$

Using

$$U'_\alpha(R_t) = (1 + |R_t|)^{\alpha-1}$$

and

$$U''_\alpha(R_t) = (\alpha - 1)\,\mathrm{sgn}(R_t)(1 + |R_t|)^{\alpha-2},$$

gives Eq. (7).

We now ask when the transformed drift and diffusion depend only on Z_t, i.e.,

$$\mathrm{d}U_\alpha(R_t) = a(Z_t)\,\mathrm{d}t + b(Z_t)\,\mathrm{d}W_t.$$

Matching the diffusion coefficients in Eqs. (7) and (10) gives

$$(1 + |R_t|)^{\alpha-1}\sigma(R_t, Z_t) = b(Z_t),$$

and therefore we have

$$\sigma(R_t, Z_t) = b(Z_t)(1 + |R_t|)^{1-\alpha}. \tag{14}$$

Similarly, matching the drift coefficients requires

$$a(Z_t) = (1 + |R_t|)^{\alpha-1}\mu(R_t, Z_t) + \frac{1}{2}(\alpha - 1)\,\mathrm{sgn}(R_t)(1 + |R_t|)^{\alpha-2}\sigma^2(R_t, Z_t).$$

Substituting $\sigma^2(R_t, Z_t) = b^2(Z_t)(1 + |R_t|)^{2-2\alpha}$ gives

$$a(Z_t) = (1 + |R_t|)^{\alpha-1}\mu(R_t, Z_t) + \frac{1}{2}(\alpha - 1)\,\mathrm{sgn}(R_t)b^2(Z_t)(1 + |R_t|)^{-\alpha}.$$

Solving for $\mu(R_t, Z_t)$ yields

$$\mu(R_t, Z_t) = a(Z_t)(1 + |R_t|)^{1-\alpha} + \frac{1 - \alpha}{2}\,\mathrm{sgn}(R_t)b^2(Z_t)(1 + |R_t|)^{1-2\alpha}. \tag{15}$$

Conversely, substituting Eq. (14)–Eq. (15) into Eq. (7) gives

$$\mathrm{d}U_\alpha(R_t) = a(Z_t)\,\mathrm{d}t + b(Z_t)\,\mathrm{d}W_t.$$

Therefore, the stated form characterizes the return dynamics whose explicit dependence on R_t is removed by the signed Box–Cox transformation. □

A.3 PROOFS OF SECTION 4.3

Proof of Theorem 4.3. Consider the interval $[0, T]$ and partition it into consecutive state–action segments

$$[\tau_{k-1}, \tau_k), \qquad k = 1, \ldots, n_T,$$

where on the k-th segment, all state-action pairs within it have $(S_t, A_t) = (s_k, a_k)$ and $t_k = \tau_k - \tau_{k-1}$ denotes its duration. We have $T = \tau_{n_T} = \sum_{k=1}^{n_T} t_k$, then $\tau_0 = 0$ and $\tau_k = \sum_{j=1}^{k} t_j$, Let Z_t denote (S_t, A_t). By Theorem 4.1, throughout the k-th segment, the transformed process satisfies

$$\mathrm{d}U_\alpha(R_t) = a(z_k)\,\mathrm{d}t + b(z_k)\,\mathrm{d}W_t.$$

Integrating over $[\tau_{k-1}, \tau_k]$ gives

$$U_\alpha(R_{\tau_k}) - U_\alpha(R_{\tau_{k-1}}) = a(z_k)t_k + b(z_k)\Delta W_k,$$

where $\Delta W_k = W_{\tau_k} - W_{\tau_{k-1}} \sim \mathcal{N}(0, t_k)$ because W_t is a Brownian process. Summing over all n_T segments, the intermediate terms telescope and we obtain

$$U_\alpha(R_T) - U_\alpha(R_0) = \sum_{k=1}^{n_T} a(z_k)t_k + \sum_{k=1}^{n_T} b(z_k)\Delta W_k.$$

Dividing by T yields

$$\frac{U_\alpha(R_T) - U_\alpha(R_0)}{T} = \frac{1}{T}\sum_{k=1}^{n_T} a(z_k)t_k + \frac{1}{T}\sum_{k=1}^{n_T} b(z_k)\Delta W_k. \tag{16}$$

Consider first the drift term. It can be regrouped according to state-action pairs as

$$\frac{1}{T}\sum_{k=1}^{n_T} a(z_k)t_k = \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} a(z)\left[\frac{1}{T}\sum_{k=1}^{n_T} t_k \mathbf{1}_{\{(z_k)=(z)\}}\right]. \tag{17}$$

By the assumed ergodicity of the policy-induced state-action process, the fraction of time spent in each state-action pair converges almost surely to its stationary occupation probability:

$$\frac{1}{T}\sum_{k=1}^{n_T} t_k \mathbf{1}_{\{(z_k)=(z)\}} \xrightarrow[T\to\infty]{\text{a.s.}} \nu^\pi(z).$$

Therefore,

$$\frac{1}{T}\sum_{k=1}^{n_T} a(z_k)t_k \xrightarrow[T\to\infty]{\text{a.s.}} \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} \nu^\pi(z)a(z). \tag{18}$$

Now consider the stochastic term. Let

$$M_T := \sum_{k=1}^{n_T} b(z_k)\Delta W_k \sim \mathcal{N}\big(0, \sum_{k=1}^{n_T} b(z_k)^2\, t_k\big).$$

Since $\mathcal{S}$ and $\mathcal{A}$ are finite and $b(s,a)$ is finite for every state-action pair, there exists $b_{\max} = \max_{s\in\mathcal{S},\, a\in\mathcal{A}} b(s,a) < \infty$. Hence,

$$\frac{1}{T}M_T \sim \mathcal{N}\big(0, \frac{\sum_{k=1}^{n_T} b(z_k)^2\, t_k}{T^2}\big),$$

where the variance term has

$$\frac{\sum_{k=1}^{n_T} b(z_k)^2\, t_k}{T^2} \le \frac{b_{\max}^2 \sum_{k=1}^{n_T} t_k}{T^2} = \frac{b_{\max}^2}{T}$$

The strong law for continuous martingales therefore gives

$$\frac{M_T}{T} \xrightarrow[T\to\infty]{\text{a.s.}} 0. \tag{19}$$

Combining Equ. (18) and Equ.(19) gives

$$\frac{U_\alpha(R_T) - U_\alpha(R_0)}{T} \xrightarrow[T\to\infty]{\text{a.s.}} \sum_{s,a} \nu^\pi(z)(z) = \sum_{s,a} d^\pi(s)\pi(a \mid s)a(s,a).$$

The proof is completed by denoting $\bar{G}_\alpha^\pi = \sum_s d^\pi(s) \sum_a \pi(a \mid s)a(s,a)$. □

Proof of Proposition 4.4. By the definition of the transformed increment $g_t^\alpha = U_\alpha(R_{t+1}) - U_\alpha(R_t)$, taking telescoping gives

$$\frac{1}{T}\sum_{t=0}^{T-1} g_t^\alpha = \frac{U_\alpha(R_T) - U_\alpha(R_0)}{T}.$$

Since R_0 is deterministic,

$$\mathrm{Var}\left(\frac{1}{T}\sum_{t=0}^{T-1} g_t^\alpha\right) = \frac{1}{T^2}\,\mathrm{Var}\left(U_\alpha(R_T)\right). \tag{20}$$

Since for a random variable X and its independent copy X', there exists $\mathrm{Var}(R_T) = \frac{1}{2}\mathbb{E}[(X - X')^2]$. Let R'_T be an independent copy of R_T, then $U_\alpha(R'_T)$ is also an independent copy of $U_\alpha(R_T)$. We obtain

$$\begin{aligned}\mathrm{Var}\left(U_\alpha(R_T)\right) &= \frac{1}{2}\mathbb{E}\left[\left(U_\alpha(R_T) - U_\alpha(R'_T)\right)^2\right] \\ &\le \frac{1}{2}\mathbb{E}\left[(R_T - R'_T)^2\right] \\ &= \mathrm{Var}(R_T).\end{aligned} \tag{21}$$

where the inequality comes from 1-Lipschitz continuity of U_α in Lemma A.1.

Therefore,

$$\mathrm{Var}\left(\frac{1}{T}\sum_{t=0}^{T-1} g_t^\alpha\right) \le \frac{1}{T^2}\,\mathrm{Var}(R_T) = \mathrm{Var}\left(\frac{R_T - R_0}{T}\right), \tag{22}$$

which proves the result. □

B Optimal action in coin-toss and experiments

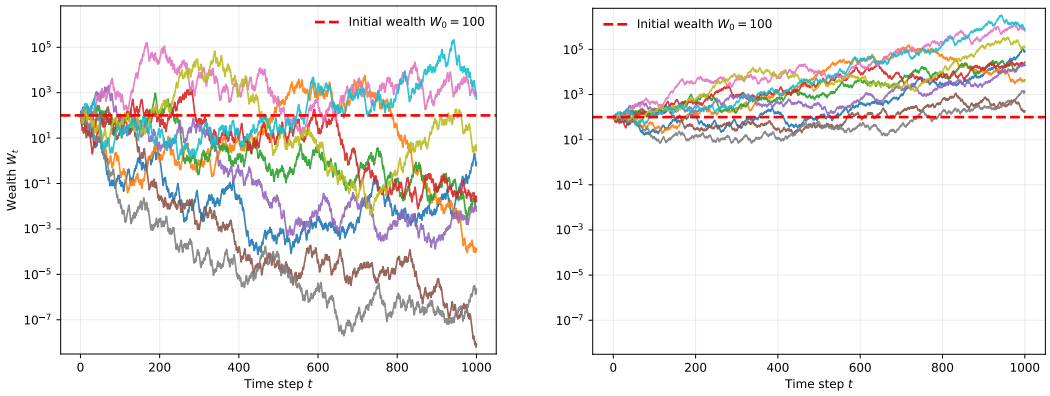
We also evaluate the proposed actor-only regularization on the coin-toss environment considered by Sheng & Baumann (2025), which has purely multiplicative return dynamics. Following the standard PPO implementation used by Baumann et al. (2025), we use PPO from Stable-Baselines3 (Raffin et al., 2021) as the baseline and keep the policy architecture, critic, GAE computation, value loss, and optimization settings unchanged. We represent the two outcomes of the coin toss as two discrete states, $s \in \{0, 1\}$, and the policy selects a state-dependent betting fraction $a(s) \in [0, 1]$. Each configuration is trained using five independent random seeds. Table 2 reports the resulting deterministic actions as mean $\pm$ standard deviation across training seeds.

For a fixed betting fraction a, the asymptotic time-average log-growth rate is

$$g(a) = \frac{1}{2}\log(1 + 0.5a) + \frac{1}{2}\log(1 - 0.4a).$$

This analysis follows Hulme et al. (2023). Hence, $g(a) > 0$ for $0 < a < 0.5$, which is the action region resulting in return increase. Within it, the maximum can be attained at $a^\star = 0.25$, whereas $a > 0.5$ yields negative long-run growth. The standard PPO baseline learns actions that are above this positive-growth region on average. In contrast, several configurations of our method move both state-dependent actions into the positive-growth region. In particular, $(\alpha, \lambda_g) = (0.25, 0.50)$ and $(\alpha, \lambda_g) = (0.25, 0.75)$ both produce actions concentrated near the growth-optimal value $a^\star = 0.25$, with the latter exhibiting lower across-seed variability. To illustrate the resulting trajectory-level behavior, Figure 1 compares rollout trajectories from one same trained seed of the PPO baseline and our method.

The results also illustrate the role of the transformation parameter. For $\alpha = 0.25$, intermediate-to-large values of λ_g tend to produce actions close to the growth-optimal region. Strong regularization with $\alpha = 0$ instead drives the betting fraction toward zero, while larger values of α increasingly favor larger actions. In particular, $\alpha = 1$, for which the transformation reduces to the identity, frequently recovers actions close to the upper action bound.



(a) Standard PPO baseline.

(b) Box–Cox PPO with $(\alpha, \lambda_g) = (0.25, 0.75)$.

Figure 1: Coin-toss wealth rollouts under learned policies. Both policies are trained with seed 1, and each panel shows 10 rollout trajectories generated from the same 10 distinct evaluation seeds. The x-axis denotes the rollout time step t, and the y-axis shows wealth W_t on a logarithmic scale. The red dashed line denotes the initial wealth $W_0 = 100$.

Table 2: Learned deterministic actions in the coin-toss environment. Results are reported as mean $\pm$ standard deviation over five training seeds. Bold indicates the mean closest to the optimal action $a^\star = 0.25$.

α	λ_g	$a(0)$	$a(1)$
Baseline	—	0.676 ± 0.352	0.520 ± 0.342
0.00	0.25	0.204 ± 0.091	0.216 ± 0.128
	0.50	0.121 ± 0.113	0.131 ± 0.099
	0.75	0.000 ± 0.000	0.000 ± 0.000
	1.00	0.000 ± 0.000	0.000 ± 0.000
0.25	0.25	0.281 ± 0.159	0.294 ± 0.111
	0.50	$\mathbf{0.259 \pm 0.148}$	$\mathbf{0.248 \pm 0.148}$
	0.75	$\mathbf{0.260 \pm 0.033}$	$\mathbf{0.241 \pm 0.051}$
	1.00	0.057 ± 0.126	0.047 ± 0.080
0.50	0.25	0.526 ± 0.077	0.473 ± 0.156
	0.50	0.503 ± 0.048	0.470 ± 0.088
	0.75	0.452 ± 0.025	0.444 ± 0.067
	1.00	0.444 ± 0.042	0.408 ± 0.040
0.75	0.25	0.808 ± 0.207	0.801 ± 0.220
	0.50	0.922 ± 0.143	0.699 ± 0.227
	0.75	0.838 ± 0.206	0.820 ± 0.138
	1.00	0.829 ± 0.196	0.768 ± 0.102
1.00	0.25	0.859 ± 0.190	0.827 ± 0.204
	0.50	1.000 ± 0.000	0.950 ± 0.111
	0.75	1.000 ± 0.000	1.000 ± 0.000
	1.00	1.000 ± 0.000	0.958 ± 0.094

C Standard Benchmark Experiments

C.1 Training Plots

Figures 2 shows the episodic return during training for each environment, averaged across 5 seeds with ± 1 standard deviation shading. Curves are smoothed with a sliding window over uniformly spaced step bins to reduce noise from on-policy rollouts and exploration. Two selected configurations per environment are displayed alongside the baseline. Ant and HalfCheetah show a clear and sustained improvement over the baseline throughout training, with the selected configurations converging to markedly higher returns. Reacher exhibits a distinct separation from the baseline early in training and maintains it, achieving better return consistently. Hopper and Walker2d display more comparable trajectories, yet the selected configurations still converge to a higher final return. Swimmer presents the configurations that are the closest-to-baseline with one configuration better than the baseline, In Atari, BeamRider and Seaquest exhibit pronounced improvement: both selected configurations

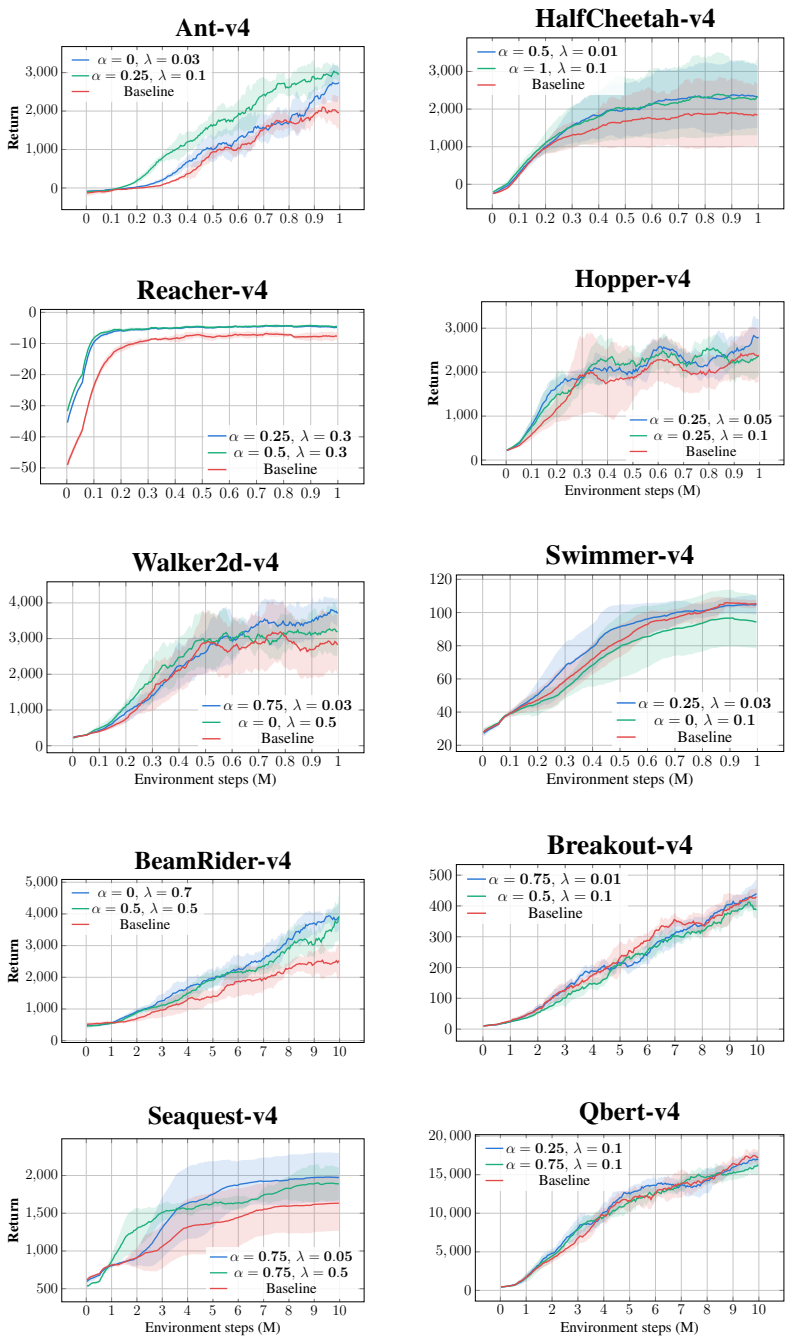


Figure 2: Training performance across the six MuJoCo and four Atari environments. Solid lines show the mean episodic return across five training seeds, and shaded regions indicate the standard deviation. The blue line shows the first configuration (α, λ_g) in Table 3, while the second is in green. Red represents the baseline, vanilla PPO. The x-axis represents environment steps on a million-step scale and the y-axis shows the mean episodic return averaged over five training seeds.

separate clearly from the baselines early in training and converge to substantially higher returns, with the gap widening and stabilising as training progresses. Qbert displays a more modest separation; the selected configurations track above the baseline for much of training but the gap narrows toward the end. In Breakout, all configurations converge to comparable final returns and no configuration is clearly better than the baseline; the displayed configurations are the closest-to-baseline by IQM.

C.2 EVALUATION RESULTS

In addition to the final evaluation results reported, we track policy performance at regular checkpoints throughout training via periodic evaluation. Figure 3 shows these curves for the Mujoco and Atari baselines and two selected configurations per environment (same configurations as shown in the training plots). The consistent separation visible in Ant, HalfCheetah, and Reacher across the full training horizon strengthens confidence that the observed gains reflect a distinct improvement in the learning dynamics. Hopper and Walker2d track the baseline closely for much of training before diverging toward higher returns in later stages. Swimmer shows overlapping curves across the configurations throughout. BeamRider and Seaquest show a clear and sustained separation from the baseline throughout training, confirming that the large performance gains at the final checkpoint emerge early and persist, rather than arising as a late-stage artifact. Qbert and Breakout display overlapping curves for much of training, with the selected configurations tracking close to the baseline, and maintains an inconclusive separation at the final checkpoint.

Table 3 shows for each environment in Mujoco and Atari the final evaluation results of the 2 selected configurations that were selected from the parameter sweep. These configurations chosen perform well in comparison to the Baseline (Vanilla PPO) in terms of higher mean, median and IQM.

Mean and Median are the arithmetic average and middle value of the five per-seed means, respectively. The median is less sensitive to a single outlier seed. IQM (interquartile mean) (Agarwal et al., 2021) sorts the five values, drops the lowest and highest (trimming 25% from each tail), and averages the remaining three, balancing robustness and statistical efficiency. Std is the standard deviation across the five per-seed means, measuring cross-seed consistency. The 95% CI on the IQM is obtained by stratified bootstrap resampling (2,000 resamples) of the five means: the 2.5th and 97.5th percentiles of the bootstrap IQM distribution form the interval. The CI is essential with only five seeds, point estimates alone can reflect variation rather than a distinct effect and a CI lower bound above the baseline IQM provides statistical evidence of the improvement.

Tables 4–9 present the complete evaluation results for all growth-rate parameter configurations (α, λ_g) across the six MuJoCo continuous-control environments, and Tables 10–13 present the corresponding results for the four Atari discrete-control environments. Each table reports, for every tested (α, λ_g) pair, the mean, median, standard deviation, IQM, and 95% confidence interval of the per-seed mean returns obtained at the final training checkpoint. All statistics are computed from the 5 per-seed means, where each per-seed mean is the average return over 100 deterministic evaluation episodes run after training completes. The configuration $\lambda_g = 0$ is omitted from all tables as it reduces to the unmodified PPO baseline. Within each table, the highest mean, median, and IQM are typeset in **bold** for ease of comparison. These tables complement the summary results in Tables 3 by providing full transparency over the sweep, allowing the selection decisions made.

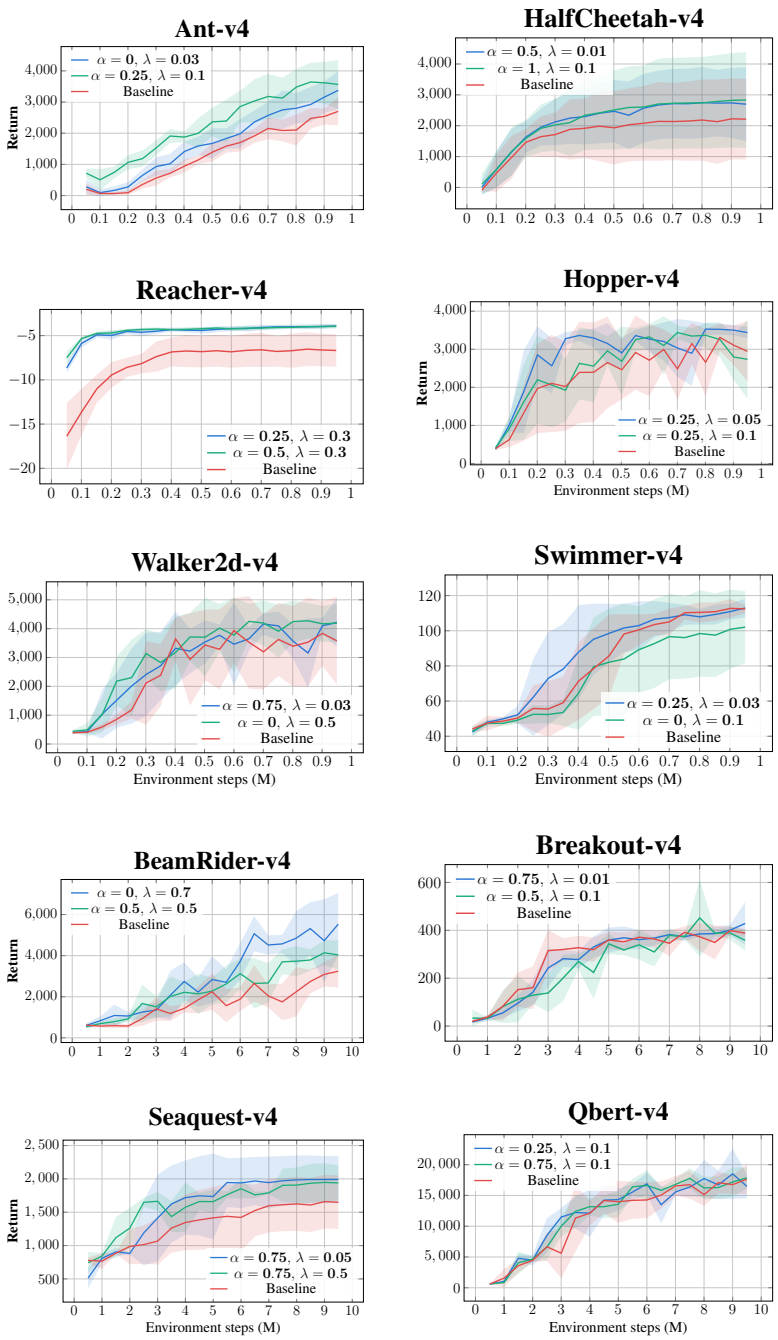


Figure 3: Periodic evaluation performance across the six MuJoCo and four Atari environments. Solid lines show the mean episodic return across five training seeds, and shaded regions indicate the standard deviation. The blue line shows the first configuration (α, λ_g) in Table 3, while the second is in green. Red represents the baseline, vanilla PPO. The x-axis represents environment steps on a million-step scale and the y-axis shows the mean episodic return averaged over five training seeds..

Table 3: Final evaluation performance comparison between the RL PPO baseline and two selected configurations for each MuJoCo and Atari environment. Returns are raw episodic returns over 100 evaluation seeds with one evaluation episode per seed. IQM denotes the interquartile mean; 95% confidence interval is across five seeds.

Environment	Configuration	Mean	Median	Std	IQM	95% CI
Ant	Baseline	2675.6	2660.7	360.9	2561.1	[2388.4, 3132.3]
	$\alpha = 0.00,\ \lambda_g = 0.03$	3361.6	3511.5	**524.0**	3394.0	[2728.7, 3938.1]
	$\alpha = 0.25,\ \lambda_g = 0.10$	**3638.7**	**3862.0**	582.8	**3597.9**	**[2960.0, 4298.9]**
HalfCheetah	Baseline	2196.2	1651.3	1162.5	1648.4	[1539.2, 3580.7]
	$\alpha = 0.50,\ \lambda_g = 0.01$	2797.7	**2746.1**	**1034.4**	**2696.1**	**[1682.7, 4064.1]**
	$\alpha = 1.00,\ \lambda_g = 0.10$	**2813.2**	1768.2	1377.6	2581.4	[1642.0, 4563.9]
Hopper	Baseline	2915.8	3343.4	580.7	2904.5	[2231.7, 3539.5]
	$\alpha = 0.25,\ \lambda_g = 0.05$	3463.6	**3526.8**	**101.8**	**3494.3**	**[3331.1, 3550.3]**
	$\alpha = 0.25,\ \lambda_g = 0.10$	**3476.6**	3500.1	142.6	3493.6	[3297.7, 3630.0]
Reacher	Baseline	-7.09	-6.19	1.07	-6.95	$[-8.45, -5.97]$
	$\alpha = 0.25,\ \lambda_g = 0.30$	-4.24	-4.17	**0.11**	-4.25	$[-4.34, -4.10]$
	$\alpha = 0.50,\ \lambda_g = 0.30$	$\mathbf{-4.13}$	$\mathbf{-4.01}$	0.13	$\mathbf{-4.12}$	$[\mathbf{-4.28}, \mathbf{-4.02}]$
Swimmer	Baseline	**113.4**	**113.6**	4.2	**113.9**	[108.2, 117.9]
	$\alpha = 0.25,\ \lambda_g = 0.03$	111.8	113.0	**4.0**	112.4	**[106.8, 115.4]**
	$\alpha = 0.00,\ \lambda_g = 0.10$	100.6	108.6	18.0	108.2	[78.4, 112.2]
Walker2d	Baseline	3580.3	3923.6	1336.7	3892.9	[1895.6, 4904.9]
	$\alpha = 0.75,\ \lambda_g = 0.03$	3880.7	3926.2	700.5	4021.9	[3050.2, 4609.9]
	$\alpha = 0.00,\ \lambda_g = 0.50$	**4099.9**	**4512.4**	**618.9**	**4177.7**	**[3329.0, 4654.7]**
BeamRider	Baseline	2919.4	2608.0	568.8	3144.6	[2219.4, 3335.2]
	$\alpha = 0.00,\ \lambda_g = 0.70$	**5565.0**	**5268.0**	**464.7**	**5605.8**	**[5037.2, 6097.6]**
	$\alpha = 0.50,\ \lambda_g = 0.50$	4956.3	5053.0	487.6	4974.4	[4380.4, 5522.3]
Breakout	Baseline	410.9	**406.0**	35.4	**400.8**	[378.2, 455.5]
	$\alpha = 0.75,\ \lambda_g = 0.01$	**413.7**	393.0	45.5	398.2	[375.8, 471.3]
	$\alpha = 0.50,\ \lambda_g = 0.10$	391.5	385.5	**13.1**	391.5	**[377.0, 407.0]**
Seaquest	Baseline	1638.5	1820.0	359.9	1792.1	[1200.6, 1875.4]
	$\alpha = 0.75,\ \lambda_g = 0.05$	**2009.1**	**1840.0**	338.2	**1841.5**	**[1836.9, 2405.1]**
	$\alpha = 0.75,\ \lambda_g = 0.50$	1967.7	1830.0	**303.2**	1821.3	[1805.1, 2325.6]
Qbert	Baseline	17822.8	**19150.0**	1257.7	18085.2	[16192.3, 19009.8]
	$\alpha = 0.25,\ \lambda_g = 0.10$	**19294.2**	19050.0	1358.5	**19561.3**	**[17513.7, 20518.9]**
	$\alpha = 0.75,\ \lambda_g = 0.10$	18119.7	18975.0	**882.2**	18180.0	[17123.5, 19063.8]

Table 4: Full evaluation results for **Ant-v4** across all tested α–λ_g configurations. IQM = interquartile mean; 95% confidence interval is across five seeds. **Bold** = highest value per metric across all rows. $\lambda_g = 0$ rows omitted (equivalent to baseline).

α	λ_g	Mean	Median	Std	IQM	95% CI
Baseline	—	2675.6	2660.7	360.9	2561.1	[2388.4, 3132.3]
$\alpha = 0.00$	$\lambda_g = 0.01$	2796.5	2822.6	280.7	2746.3	[2513.2, 3153.8]
	$\lambda_g = 0.03$	3361.6	3511.5	524.0	3394.0	[2728.7, 3938.1]
	$\lambda_g = 0.05$	3116.6	3249.2	68.1	3100.0	[3051.5, 3204.7]
	$\lambda_g = 0.10$	3060.1	2959.5	567.6	2930.3	[2520.0, 3706.6]
	$\lambda_g = 0.30$	3314.3	3090.2	520.3	3163.9	[2891.4, 4006.7]
	$\lambda_g = 0.50$	3466.7	3668.9	774.3	3440.5	[2561.8, 4381.8]
	$\lambda_g = 0.70$	3200.2	3335.2	390.5	3168.1	[2793.2, 3650.0]
$\alpha = 0.25$	$\lambda_g = 0.01$	3241.8	3178.5	768.2	3193.2	[2513.2, 4190.3]
	$\lambda_g = 0.03$	2753.6	2778.8	248.2	2722.7	[2507.7, 3071.1]
	$\lambda_g = 0.05$	2801.0	3087.1	286.1	2743.5	[2504.5, 3143.2]
	$\lambda_g = 0.10$	3638.7	3862.0	582.8	3597.9	[2960.0, 4298.9]
	$\lambda_g = 0.30$	3474.7	3139.9	680.5	3331.5	[2846.1, 4366.9]
	$\lambda_g = 0.50$	3132.7	3083.3	362.8	3013.9	[2826.7, 3581.7]
	$\lambda_g = 0.70$	2733.7	2707.6	168.3	2680.4	[2585.7, 2947.7]
$\alpha = 0.50$	$\lambda_g = 0.01$	2742.2	2811.0	441.1	2630.7	[2361.5, 3326.2]
	$\lambda_g = 0.03$	3271.8	3477.5	863.0	3279.0	[2231.9, 4250.5]
	$\lambda_g = 0.05$	3525.3	3710.9	529.8	3647.9	[2874.7, 3999.9]
	$\lambda_g = 0.10$	3606.0	**4180.2**	802.5	3725.1	[2583.5, 4372.6]
	$\lambda_g = 0.30$	2796.1	2910.7	214.6	2777.4	[2565.2, 3057.1]
	$\lambda_g = 0.50$	2563.6	2752.1	264.7	2617.6	[2221.2, 2784.4]
	$\lambda_g = 0.70$	2175.1	2342.5	267.1	2213.1	[1837.8, 2414.9]
$\alpha = 0.75$	$\lambda_g = 0.01$	3310.8	3636.0	507.8	3386.8	[2670.3, 3822.6]
	$\lambda_g = 0.03$	3034.7	3150.1	332.1	3047.5	[2645.4, 3421.6]
	$\lambda_g = 0.05$	3293.2	3248.4	487.8	3232.4	[2768.4, 3887.8]
	$\lambda_g = 0.10$	3061.9	3142.9	317.0	3068.3	[2685.9, 3394.4]
	$\lambda_g = 0.30$	3110.4	3207.0	254.3	3109.2	[2815.0, 3405.4]
	$\lambda_g = 0.50$	2146.6	2076.7	252.5	2069.7	[1934.8, 2458.7]
	$\lambda_g = 0.70$	1529.3	1511.7	**57.6**	1509.3	[1482.0, 1603.6]
$\alpha = 1.00$	$\lambda_g = 0.01$	2522.4	2645.1	285.9	2476.0	[2240.0, 2886.0]
	$\lambda_g = 0.03$	**3804.2**	4008.1	617.7	**3955.4**	[3002.5, 4388.6]
	$\lambda_g = 0.05$	3136.1	3236.6	446.6	3162.7	[2628.1, 3619.2]
	$\lambda_g = 0.10$	3184.1	3077.4	272.4	3062.8	[3015.7, 3515.9]
	$\lambda_g = 0.30$	2917.3	2876.9	197.1	2879.1	[2727.7, 3157.4]
	$\lambda_g = 0.50$	2018.5	1932.8	157.6	2001.6	[1864.6, 2217.3]
	$\lambda_g = 0.70$	1257.0	1264.9	129.5	1247.3	[1113.6, 1416.3]

Table 5: Full evaluation results for **HalfCheetah-v4** across all tested α–λ_g configurations. IQM = interquartile mean; 95% confidence interval is across five seeds. **Bold** = highest value per metric across all rows. $\lambda_g = 0$ rows omitted (equivalent to baseline).

α	λ_g	**Mean**	**Median**	**Std**	**IQM**	**95% CI**
Baseline	—	2196.2	1651.3	1162.5	1648.4	[1539.2, 3580.7]
$\alpha = 0.00$	$\lambda_g = 0.01$	1544.4	1562.7	79.4	1532.1	[1465.3, 1645.0]
	$\lambda_g = 0.03$	1685.8	1692.8	68.9	1700.5	[1598.8, 1755.3]
	$\lambda_g = 0.05$	2261.1	1720.3	1113.7	1731.1	[1645.9, 3562.5]
	$\lambda_g = 0.10$	2684.8	1796.5	1270.2	2492.5	[1581.3, 4284.5]
	$\lambda_g = 0.30$	1844.8	1766.6	220.3	1769.8	[1692.8, 2113.5]
	$\lambda_g = 0.50$	1958.5	1637.6	991.1	1578.0	[1264.4, 3153.4]
	$\lambda_g = 0.70$	1961.2	1949.9	657.2	1772.9	[1367.3, 2814.9]
$\alpha = 0.25$	$\lambda_g = 0.01$	2083.2	1676.3	795.5	1714.2	[1619.2, 3055.8]
	$\lambda_g = 0.03$	2436.7	1779.0	1021.7	2296.2	[1518.5, 3713.7]
	$\lambda_g = 0.05$	2483.4	1706.3	1050.9	2288.4	[1586.0, 3836.3]
	$\lambda_g = 0.10$	2216.1	1720.4	1103.5	1703.5	[1573.9, 3520.0]
	$\lambda_g = 0.30$	2130.1	1692.7	1014.4	1687.5	[1505.0, 3332.0]
	$\lambda_g = 0.50$	1923.8	2005.1	466.5	1882.0	[1422.2, 2497.7]
	$\lambda_g = 0.70$	1934.3	2116.5	296.7	1957.7	[1571.0, 2221.0]
$\alpha = 0.50$	$\lambda_g = 0.01$	2797.7	**2746.1**	1034.4	**2696.1**	[1682.7, 4064.1]
	$\lambda_g = 0.03$	2153.4	1768.4	916.0	1734.1	[1595.9, 3254.6]
	$\lambda_g = 0.05$	1655.1	1645.2	73.2	1653.8	[1572.0, 1742.6]
	$\lambda_g = 0.10$	2646.6	1775.8	1210.9	2458.7	[1599.9, 4170.1]
	$\lambda_g = 0.30$	2041.0	2084.2	588.8	1948.0	[1409.4, 2765.7]
	$\lambda_g = 0.50$	2328.1	2255.3	343.4	2327.7	[1946.4, 2742.9]
	$\lambda_g = 0.70$	856.7	902.2	107.6	851.7	[730.4, 974.2]
$\alpha = 0.75$	$\lambda_g = 0.01$	2173.1	1724.6	1002.9	1717.4	[1599.9, 3359.1]
	$\lambda_g = 0.03$	1714.9	1709.9	72.0	1715.7	[1632.9, 1796.0]
	$\lambda_g = 0.05$	2086.4	1628.6	949.2	1626.5	[1578.2, 3203.3]
	$\lambda_g = 0.10$	2188.9	1623.5	1168.0	1643.4	[1529.8, 3587.2]
	$\lambda_g = 0.30$	2382.5	2519.6	632.2	2456.8	[1621.7, 3028.0]
	$\lambda_g = 0.50$	1350.4	1344.1	114.8	1333.7	[1228.6, 1488.4]
	$\lambda_g = 0.70$	402.8	376.4	88.7	396.6	[308.3, 513.5]
$\alpha = 1.00$	$\lambda_g = 0.01$	1640.1	1659.2	90.3	1643.0	[1528.9, 1738.8]
	$\lambda_g = 0.03$	1626.5	1610.2	89.0	1600.5	[1551.3, 1743.2]
	$\lambda_g = 0.05$	1645.4	1629.3	**67.1**	1629.6	[1584.4, 1731.7]
	$\lambda_g = 0.10$	**2813.2**	1768.2	1377.6	2581.4	[1642.0, 4563.9]
	$\lambda_g = 0.30$	1994.4	2062.6	466.5	2012.5	[1445.0, 2503.7]
	$\lambda_g = 0.50$	983.7	901.4	221.4	953.4	[758.8, 1268.0]
	$\lambda_g = 0.70$	287.4	271.9	80.9	269.4	[212.9, 385.3]

Table 6: Full evaluation results for **Hopper-v4** across all tested α–λ_g configurations. IQM = interquartile mean; 95% confidence interval is across five seeds. **Bold** = highest value per metric across all rows. $\lambda_g = 0$ rows omitted (equivalent to baseline).

α	λ_g	**Mean**	**Median**	**Std**	**IQM**	**95% CI**
Baseline	—	2915.8	3343.4	580.7	2904.5	[2231.7, 3539.5]
$\alpha = 0.00$	$\lambda_g = 0.01$	3418.8	3426.5	65.7	3404.6	[3352.5, 3498.1]
	$\lambda_g = 0.03$	2535.6	2902.3	762.4	2647.5	[1561.2, 3237.7]
	$\lambda_g = 0.05$	2638.2	3289.2	785.7	2706.8	[1659.6, 3421.8]
	$\lambda_g = 0.10$	2638.7	3280.1	846.1	2776.4	[1552.3, 3462.3]
	$\lambda_g = 0.30$	2774.0	2927.5	554.3	2746.6	[2142.6, 3317.9]
	$\lambda_g = 0.50$	3093.6	3445.9	580.3	3299.6	[2331.3, 3538.8]
	$\lambda_g = 0.70$	2153.4	1946.0	642.0	2360.7	[1334.4, 2670.0]
$\alpha = 0.25$	$\lambda_g = 0.01$	3201.3	3443.8	445.1	3358.4	[2653.8, 3546.8]
	$\lambda_g = 0.03$	3090.0	3350.3	500.0	3162.1	[2454.8, 3613.2]
	$\lambda_g = 0.05$	3463.6	3526.8	101.8	**3494.3**	[3331.1, 3550.3]
	$\lambda_g = 0.10$	**3476.6**	3500.1	142.6	3493.9	[3297.7, 3630.0]
	$\lambda_g = 0.30$	2942.0	3269.7	610.3	3147.2	[2172.1, 3413.6]
	$\lambda_g = 0.50$	2631.7	2710.7	687.1	2604.1	[1929.3, 3459.1]
	$\lambda_g = 0.70$	2507.2	3120.9	935.7	2746.8	[1270.9, 3333.9]
$\alpha = 0.50$	$\lambda_g = 0.01$	2512.0	2542.5	723.2	2461.6	[1729.9, 3386.2]
	$\lambda_g = 0.03$	2790.5	**3565.0**	940.3	2870.0	[1634.9, 3628.0]
	$\lambda_g = 0.05$	3246.4	3251.5	231.8	3244.2	[2990.1, 3524.0]
	$\lambda_g = 0.10$	3089.5	3345.8	393.6	3189.5	[2571.4, 3437.4]
	$\lambda_g = 0.30$	2813.9	3352.8	900.2	3112.6	[1622.6, 3541.0]
	$\lambda_g = 0.50$	1449.0	1077.3	807.5	1357.2	[664.5, 2451.4]
	$\lambda_g = 0.70$	710.0	688.7	246.1	719.1	[420.1, 997.7]
$\alpha = 0.75$	$\lambda_g = 0.01$	3212.0	3422.6	257.7	3285.7	[2899.4, 3431.0]
	$\lambda_g = 0.03$	2976.4	3345.0	484.1	3013.0	[2371.4, 3497.5]
	$\lambda_g = 0.05$	2329.0	1786.8	974.1	2304.9	[1352.7, 3517.2]
	$\lambda_g = 0.10$	3134.4	3268.6	308.4	3128.4	[2767.3, 3480.5]
	$\lambda_g = 0.30$	1586.3	886.8	1067.8	1361.3	[649.7, 2976.3]
	$\lambda_g = 0.50$	438.4	413.1	**56.9**	413.5	[403.7, 506.2]
	$\lambda_g = 0.70$	650.7	332.3	718.5	328.1	[210.6, 1519.5]
$\alpha = 1.00$	$\lambda_g = 0.01$	2617.1	3266.0	852.0	2844.7	[1487.8, 3325.1]
	$\lambda_g = 0.03$	2827.0	3476.6	846.9	3031.8	[1708.5, 3551.2]
	$\lambda_g = 0.05$	2608.4	3113.7	540.7	2712.4	[1910.8, 3146.6]
	$\lambda_g = 0.10$	2892.2	3436.3	941.8	3324.5	[1754.6, 3460.8]
	$\lambda_g = 0.30$	1913.9	1317.1	885.8	1834.6	[1052.0, 3025.1]
	$\lambda_g = 0.50$	916.0	704.4	376.3	794.1	[618.2, 1407.5]
	$\lambda_g = 0.70$	374.9	341.5	139.6	333.4	[260.4, 542.9]

Table 7: Full evaluation results for **Reacher-v4** across all tested α–λ_g configurations. IQM = interquartile mean; 95% confidence interval is across five seeds. **Bold** = highest value per metric across all rows. $\lambda_g = 0$ rows omitted (equivalent to baseline).

α	λ_g	**Mean**	**Median**	**Std**	**IQM**	**95% CI**
Baseline	—	−7.09	−6.19	1.07	−6.95	[−8.45, −5.97]
$\alpha = 0.00$	$\lambda_g = 0.01$	−5.26	−4.60	0.36	−5.20	[−5.73, −4.89]
	$\lambda_g = 0.03$	−5.65	−5.16	1.49	−5.04	[−7.49, −4.63]
	$\lambda_g = 0.05$	−5.59	−5.07	1.15	−5.32	[−7.05, −4.55]
	$\lambda_g = 0.10$	−4.73	−4.37	0.42	−4.65	[−5.24, −4.28]
	$\lambda_g = 0.30$	−4.29	−4.14	0.22	−4.24	[−4.56, −4.05]
	$\lambda_g = 0.50$	−4.47	−4.33	0.06	−4.46	[−4.54, −4.40]
	$\lambda_g = 0.70$	−5.02	−4.93	0.14	−5.03	[−5.16, −4.85]
$\alpha = 0.25$	$\lambda_g = 0.01$	−7.69	−7.00	1.76	−7.40	[−9.81, −5.81]
	$\lambda_g = 0.03$	−5.48	−5.00	1.01	−5.15	[−6.72, −4.65]
	$\lambda_g = 0.05$	−5.52	−4.87	0.75	−5.39	[−6.43, −4.76]
	$\lambda_g = 0.10$	−4.62	−4.29	**0.11**	−4.61	[−4.75, −4.51]
	$\lambda_g = 0.30$	−4.24	−4.17	**0.11**	−4.25	[−4.34, −4.10]
	$\lambda_g = 0.50$	−4.34	−4.23	**0.11**	−4.34	[−4.47, −4.22]
	$\lambda_g = 0.70$	−5.02	−4.89	0.25	−4.96	[−5.33, −4.79]
$\alpha = 0.50$	$\lambda_g = 0.01$	−6.09	−5.18	0.95	−5.89	[−7.28, −5.21]
	$\lambda_g = 0.03$	−5.35	−4.80	0.94	−5.03	[−6.57, −4.60]
	$\lambda_g = 0.05$	−4.90	−4.54	0.29	−4.89	[−5.23, −4.56]
	$\lambda_g = 0.10$	−4.45	−4.27	0.24	−4.43	[−4.74, −4.22]
	$\lambda_g = 0.30$	**−4.13**	**−4.01**	0.13	**−4.12**	[−4.28, −4.02]
	$\lambda_g = 0.50$	−4.43	−4.34	0.17	−4.44	[−4.62, −4.22]
	$\lambda_g = 0.70$	−5.23	−5.03	0.27	−5.18	[−5.57, −4.96]
$\alpha = 0.75$	$\lambda_g = 0.01$	−7.39	−6.67	1.18	−7.15	[−8.91, −6.26]
	$\lambda_g = 0.03$	−5.27	−5.08	0.31	−5.25	[−5.65, −4.93]
	$\lambda_g = 0.05$	−5.35	−5.02	0.50	−5.20	[−5.97, −4.93]
	$\lambda_g = 0.10$	−4.46	−4.29	0.32	−4.42	[−4.84, −4.11]
	$\lambda_g = 0.30$	−4.25	−4.10	0.17	−4.20	[−4.48, −4.10]
	$\lambda_g = 0.50$	−4.46	−4.34	0.13	−4.42	[−4.62, −4.34]
	$\lambda_g = 0.70$	−5.26	−5.09	0.22	−5.24	[−5.47, −5.03]
$\alpha = 1.00$	$\lambda_g = 0.01$	−6.61	−5.90	2.00	−6.34	[−9.11, −4.85]
	$\lambda_g = 0.03$	−4.93	−4.58	0.40	−4.79	[−5.46, −4.62]
	$\lambda_g = 0.05$	−4.84	−4.66	0.39	−4.81	[−5.31, −4.42]
	$\lambda_g = 0.10$	−4.41	−4.25	0.33	−4.34	[−4.80, −4.10]
	$\lambda_g = 0.30$	−4.29	−4.15	0.18	−4.29	[−4.51, −4.09]
	$\lambda_g = 0.50$	−4.37	−4.26	0.14	−4.40	[−4.51, −4.19]
	$\lambda_g = 0.70$	−5.30	−5.03	0.32	−5.28	[−5.69, −4.94]

Table 8: Full evaluation results for **Swimmer-v4** across all tested α–λ_g configurations. IQM = interquartile mean; 95% confidence interval is across five seeds. **Bold** = highest value per metric across all rows. $\lambda_g = 0$ rows omitted (equivalent to baseline).

α	λ_g	**Mean**	**Median**	**Std**	**IQM**	**95% CI**
Baseline	—	**113.4**	**113.6**	4.2	**113.9**	[108.2, 117.9]
$\alpha = 0.00$	$\lambda_g = 0.01$	101.4	106.6	13.8	106.1	[84.4, 111.7]
	$\lambda_g = 0.03$	97.5	103.0	22.6	102.7	[70.7, 117.2]
	$\lambda_g = 0.05$	99.3	104.1	17.0	104.8	[78.0, 113.7]
	$\lambda_g = 0.10$	100.6	108.6	18.0	108.2	[78.4, 112.2]
	$\lambda_g = 0.30$	73.0	52.8	28.3	70.0	[48.1, 107.8]
	$\lambda_g = 0.50$	49.4	49.6	**0.8**	49.4	[48.4, 50.3]
	$\lambda_g = 0.70$	45.3	46.2	1.8	45.7	[43.1, 46.9]
$\alpha = 0.25$	$\lambda_g = 0.01$	97.4	110.5	25.8	104.5	[63.4, 118.2]
	$\lambda_g = 0.03$	111.8	113.0	4.0	112.4	[106.8, 115.4]
	$\lambda_g = 0.05$	95.8	103.5	17.4	101.9	[73.4, 109.3]
	$\lambda_g = 0.10$	77.0	61.2	28.5	74.7	[49.9, 111.8]
	$\lambda_g = 0.30$	77.2	53.2	33.2	71.3	[48.9, 119.7]
	$\lambda_g = 0.50$	48.3	48.6	2.4	48.2	[45.7, 51.0]
	$\lambda_g = 0.70$	46.0	45.5	3.8	45.9	[41.9, 50.5]
$\alpha = 0.50$	$\lambda_g = 0.01$	87.8	104.0	26.6	92.5	[53.6, 112.1]
	$\lambda_g = 0.03$	80.2	96.9	23.5	83.0	[51.3, 100.1]
	$\lambda_g = 0.05$	82.6	91.7	26.5	79.9	[52.5, 112.0]
	$\lambda_g = 0.10$	97.0	105.8	23.3	105.5	[69.1, 113.6]
	$\lambda_g = 0.30$	58.1	50.7	17.9	50.0	[46.9, 79.8]
	$\lambda_g = 0.50$	47.2	47.6	1.7	47.9	[45.1, 48.4]
	$\lambda_g = 0.70$	42.8	42.8	1.2	42.6	[41.7, 44.3]
$\alpha = 0.75$	$\lambda_g = 0.01$	89.1	95.8	23.3	94.6	[59.1, 111.6]
	$\lambda_g = 0.03$	91.6	95.0	28.1	94.7	[56.8, 122.2]
	$\lambda_g = 0.05$	101.3	107.8	17.9	107.7	[78.6, 115.4]
	$\lambda_g = 0.10$	96.1	103.5	22.3	105.0	[68.9, 112.2]
	$\lambda_g = 0.30$	52.4	52.4	2.4	51.9	[50.3, 55.5]
	$\lambda_g = 0.50$	47.8	47.5	1.6	48.1	[45.8, 49.4]
	$\lambda_g = 0.70$	42.4	42.6	1.9	42.3	[40.2, 44.5]
$\alpha = 1.00$	$\lambda_g = 0.01$	82.3	80.7	21.8	81.2	[57.7, 108.9]
	$\lambda_g = 0.03$	100.0	109.0	25.4	108.9	[69.5, 118.5]
	$\lambda_g = 0.05$	90.2	94.9	20.2	91.0	[66.1, 111.9]
	$\lambda_g = 0.10$	73.4	59.2	25.2	71.2	[49.5, 104.0]
	$\lambda_g = 0.30$	49.8	50.4	2.1	49.3	[47.9, 52.4]
	$\lambda_g = 0.50$	46.3	46.5	2.1	46.7	[43.7, 48.4]
	$\lambda_g = 0.70$	42.9	43.3	1.9	42.8	[40.7, 45.2]

Table 9: Full evaluation results for **Walker2d-v4** across all tested α–λ_g configurations. IQM = interquartile mean; 95% confidence interval is across five seeds. **Bold** = highest value per metric across all rows. $\lambda_g = 0$ rows omitted (equivalent to baseline).

α	λ_g	**Mean**	**Median**	**Std**	**IQM**	**95% CI**
Baseline	—	3580.3	3923.6	1336.7	3893.0	[1895.6, 4905.0]
$\alpha = 0.00$	$\lambda_g = 0.01$	3571.7	4464.9	1214.3	3657.6	[2081.2, 4719.1]
	$\lambda_g = 0.03$	3852.9	4144.7	925.9	4030.0	[2662.9, 4774.7]
	$\lambda_g = 0.05$	3323.3	3776.7	547.6	3262.3	[2713.5, 3985.8]
	$\lambda_g = 0.10$	3515.0	3862.8	1071.4	3720.7	[2199.1, 4633.3]
	$\lambda_g = 0.30$	2965.2	3166.6	923.4	2780.6	[2034.1, 4086.4]
	$\lambda_g = 0.50$	4099.9	**4512.4**	618.9	4177.7	[3329.0, 4654.7]
	$\lambda_g = 0.70$	1531.1	1231.6	536.9	1322.4	[1129.7, 2136.7]
$\alpha = 0.25$	$\lambda_g = 0.01$	3890.8	4111.1	482.0	3932.6	[3314.8, 4410.9]
	$\lambda_g = 0.03$	3676.5	3975.5	839.6	3823.5	[2627.8, 4494.2]
	$\lambda_g = 0.05$	3106.9	3803.9	762.8	3194.9	[2227.4, 3962.0]
	$\lambda_g = 0.10$	3509.6	4066.9	1060.2	3956.3	[2207.9, 4239.7]
	$\lambda_g = 0.30$	3119.3	3551.3	797.5	3368.6	[2107.5, 3773.0]
	$\lambda_g = 0.50$	2531.8	2109.5	1221.6	2270.7	[1395.4, 4122.5]
	$\lambda_g = 0.70$	2387.3	2238.1	654.3	2403.1	[1602.5, 3067.6]
$\alpha = 0.50$	$\lambda_g = 0.01$	3372.3	3843.3	992.3	3248.4	[2270.2, 4549.9]
	$\lambda_g = 0.03$	3487.4	3919.9	865.0	3539.1	[2415.9, 4375.8]
	$\lambda_g = 0.05$	3405.2	3606.1	725.6	3557.9	[2516.0, 4082.3]
	$\lambda_g = 0.10$	**4211.5**	4507.7	867.8	**4483.5**	[3125.9, 4925.5]
	$\lambda_g = 0.30$	2786.0	2291.4	1160.6	2444.9	[1723.2, 4253.2]
	$\lambda_g = 0.50$	2498.8	1704.5	1054.7	2383.9	[1561.2, 3794.0]
	$\lambda_g = 0.70$	2211.0	1932.4	616.3	2009.7	[1686.1, 2982.0]
$\alpha = 0.75$	$\lambda_g = 0.01$	3053.8	3782.2	1028.0	2960.3	[1995.6, 4337.4]
	$\lambda_g = 0.03$	3880.7	3926.2	700.5	4021.9	[3050.2, 4609.9]
	$\lambda_g = 0.05$	3950.1	3827.5	803.6	3729.7	[3248.2, 4922.3]
	$\lambda_g = 0.10$	3406.3	3441.1	659.2	3393.0	[2636.6, 4189.9]
	$\lambda_g = 0.30$	3137.2	3391.9	407.0	3176.4	[2626.6, 3559.1]
	$\lambda_g = 0.50$	2576.6	2728.6	1135.9	2700.2	[1194.6, 3805.0]
	$\lambda_g = 0.70$	1621.0	1376.2	702.8	1481.7	[946.0, 2506.3]
$\alpha = 1.00$	$\lambda_g = 0.01$	3433.9	4011.2	907.3	3772.5	[2430.6, 4115.5]
	$\lambda_g = 0.03$	4056.7	4196.4	558.3	4069.2	[3393.6, 4659.5]
	$\lambda_g = 0.05$	3691.6	3858.2	**369.4**	3831.4	[3247.4, 3967.2]
	$\lambda_g = 0.10$	2856.8	3632.0	1177.9	3305.0	[1324.1, 3701.5]
	$\lambda_g = 0.30$	3061.0	3237.1	1137.5	3102.4	[1706.5, 4314.9]
	$\lambda_g = 0.50$	2583.9	2909.6	731.2	2577.2	[1754.7, 3412.9]
	$\lambda_g = 0.70$	1262.0	748.8	755.0	1057.5	[619.2, 2264.0]

Table 10: Full evaluation results for **BeamRider** across all tested α–λ_g configurations. IQM = interquartile mean; 95% confidence interval is across five seeds. **Bold** = highest value per metric across all rows. $\lambda_g = 0$ rows omitted (equivalent to baseline).

α	λ_g	**Mean**	**Median**	**Std**	**IQM**	**95% CI**
Baseline	—	2919.4	2608.0	568.8	3144.6	[2219.4, 3335.2]
$\alpha = 0.00$	$\lambda_g = 0.01$	2756.5	2608.0	691.5	2855.6	[1868.1, 3458.1]
	$\lambda_g = 0.03$	3026.7	2776.0	497.5	3032.0	[2446.2, 3615.1]
	$\lambda_g = 0.05$	3227.4	3300.0	729.7	3232.4	[2434.5, 4094.9]
	$\lambda_g = 0.10$	3386.0	3480.0	503.0	3357.1	[2833.5, 4004.5]
	$\lambda_g = 0.30$	4373.3	4320.0	427.3	4366.2	[3906.2, 4881.9]
	$\lambda_g = 0.50$	4641.4	4732.0	459.5	4767.4	[4089.1, 5039.8]
	$\lambda_g = 0.70$	**5565.0**	5268.0	464.7	**5605.8**	[5037.2, 6097.6]
$\alpha = 0.25$	$\lambda_g = 0.01$	3807.0	3900.0	831.1	3755.2	[2955.8, 4739.3]
	$\lambda_g = 0.03$	2975.0	2720.0	429.8	2809.4	[2667.7, 3510.8]
	$\lambda_g = 0.05$	3139.9	3240.0	591.3	3281.1	[2429.0, 3654.0]
	$\lambda_g = 0.10$	3039.6	2160.0	1126.7	2899.1	[1783.8, 4373.0]
	$\lambda_g = 0.30$	4324.2	4860.0	636.0	4229.2	[3660.3, 5094.1]
	$\lambda_g = 0.50$	4131.6	4527.0	1335.2	4640.7	[2478.1, 5142.9]
	$\lambda_g = 0.70$	5239.3	**5544.0**	770.4	5237.7	[4351.7, 6162.8]
$\alpha = 0.50$	$\lambda_g = 0.01$	2322.8	2160.0	651.1	2323.7	[1537.1, 3054.5]
	$\lambda_g = 0.03$	2688.8	2160.0	567.7	2857.9	[1995.4, 3199.2]
	$\lambda_g = 0.05$	2650.1	2160.0	462.8	2858.9	[2092.3, 2942.4]
	$\lambda_g = 0.10$	3493.6	3390.0	739.3	3490.7	[2613.5, 4340.8]
	$\lambda_g = 0.30$	4162.3	3932.0	845.4	4085.9	[3235.7, 5201.4]
	$\lambda_g = 0.50$	4956.3	5053.0	487.6	4974.4	[4380.4, 5522.3]
	$\lambda_g = 0.70$	4863.6	4928.0	545.8	4781.8	[4305.0, 5441.3]
$\alpha = 0.75$	$\lambda_g = 0.01$	2699.0	2412.0	755.0	2643.6	[1934.7, 3609.7]
	$\lambda_g = 0.03$	3224.9	3720.0	930.5	3567.9	[2013.7, 3917.8]
	$\lambda_g = 0.05$	2804.8	2888.0	1063.8	2851.0	[1583.7, 3992.5]
	$\lambda_g = 0.10$	3809.5	3900.0	1039.3	3619.2	[2863.4, 5010.7]
	$\lambda_g = 0.30$	4023.3	3900.0	556.9	3885.3	[3523.5, 4751.7]
	$\lambda_g = 0.50$	4371.7	4146.0	750.2	4456.2	[3475.3, 5154.9]
	$\lambda_g = 0.70$	4298.4	4412.0	920.0	4335.2	[3244.4, 5307.7]
$\alpha = 1.00$	$\lambda_g = 0.01$	2941.5	3074.0	782.6	2752.0	[2212.0, 3960.9]
	$\lambda_g = 0.03$	2912.6	3000.0	987.5	2832.7	[1815.4, 4097.6]
	$\lambda_g = 0.05$	2824.6	2902.0	653.7	2937.9	[1996.0, 3487.0]
	$\lambda_g = 0.10$	3821.0	3975.0	746.4	4060.4	[2880.5, 4424.6]
	$\lambda_g = 0.30$	3814.8	3420.0	**326.6**	3812.4	[3435.8, 4192.6]
	$\lambda_g = 0.50$	4424.9	4572.0	365.1	4444.8	[3982.0, 4811.0]
	$\lambda_g = 0.70$	4742.9	4832.0	812.8	4799.7	[3807.1, 5558.8]

Table 11: Full evaluation results for **Breakout** across all tested α–λ_g configurations. IQM = interquartile mean; 95% confidence interval is across five seeds. **Bold** = highest value per metric across all rows. $\lambda_g = 0$ rows omitted (equivalent to baseline).

α	λ_g	Mean	Median	Std	IQM	95% CI
Baseline	—	410.9	406.0	35.4	**400.8**	[378.2, 455.5]
$\alpha = 0.00$	$\lambda_g = 0.01$	386.5	384.0	18.0	391.4	[363.8, 402.7]
	$\lambda_g = 0.03$	379.2	380.0	15.9	379.5	[361.7, 398.2]
	$\lambda_g = 0.05$	384.8	384.5	18.8	380.2	[367.5, 408.3]
	$\lambda_g = 0.10$	386.7	405.0	19.3	386.6	[365.0, 409.4]
	$\lambda_g = 0.30$	384.8	399.0	16.1	385.3	[364.9, 401.9]
	$\lambda_g = 0.50$	362.3	382.0	38.7	371.6	[311.3, 395.5]
	$\lambda_g = 0.70$	386.1	391.0	20.3	381.8	[367.1, 410.4]
$\alpha = 0.25$	$\lambda_g = 0.01$	366.4	371.0	38.7	374.3	[316.2, 404.2]
	$\lambda_g = 0.03$	398.7	**411.0**	32.1	390.9	[367.7, 437.8]
	$\lambda_g = 0.05$	386.0	393.0	29.0	388.5	[349.6, 417.1]
	$\lambda_g = 0.10$	376.4	387.0	18.5	373.8	[360.3, 399.7]
	$\lambda_g = 0.30$	381.7	385.0	7.7	384.6	[372.0, 387.8]
	$\lambda_g = 0.50$	369.6	375.0	28.8	369.2	[335.1, 402.1]
	$\lambda_g = 0.70$	359.4	371.0	35.4	374.4	[315.5, 383.2]
$\alpha = 0.50$	$\lambda_g = 0.01$	365.9	383.5	26.4	376.7	[332.1, 384.2]
	$\lambda_g = 0.03$	383.9	391.5	**7.0**	383.1	[376.2, 392.5]
	$\lambda_g = 0.05$	367.7	383.0	12.5	364.4	[357.0, 384.2]
	$\lambda_g = 0.10$	391.5	385.5	13.1	391.5	[377.0, 407.0]
	$\lambda_g = 0.30$	389.9	396.0	14.9	386.3	[375.5, 408.6]
	$\lambda_g = 0.50$	408.2	404.0	43.7	396.1	[370.0, 461.6]
	$\lambda_g = 0.70$	354.4	353.5	30.7	355.8	[316.7, 388.6]
$\alpha = 0.75$	$\lambda_g = 0.01$	**413.7**	393.0	45.5	398.2	[375.8, 471.3]
	$\lambda_g = 0.03$	385.9	396.5	16.2	386.2	[367.5, 404.2]
	$\lambda_g = 0.05$	382.4	394.0	10.3	385.1	[369.6, 392.2]
	$\lambda_g = 0.10$	382.6	404.0	23.7	391.4	[351.6, 400.3]
	$\lambda_g = 0.30$	369.0	381.5	27.3	363.1	[341.7, 403.6]
	$\lambda_g = 0.50$	370.5	391.0	17.7	367.1	[355.1, 393.6]
	$\lambda_g = 0.70$	343.7	370.0	55.6	346.8	[275.2, 404.3]
$\alpha = 1.00$	$\lambda_g = 0.01$	368.8	374.5	19.9	371.4	[343.5, 389.2]
	$\lambda_g = 0.03$	388.9	396.5	26.1	392.8	[357.3, 414.4]
	$\lambda_g = 0.05$	399.5	393.5	61.5	381.7	[344.4, 477.6]
	$\lambda_g = 0.10$	378.6	398.5	35.2	379.7	[339.6, 420.3]
	$\lambda_g = 0.30$	363.6	394.0	48.1	377.0	[300.0, 405.4]
	$\lambda_g = 0.50$	391.4	393.5	15.6	387.1	[377.8, 411.6]
	$\lambda_g = 0.70$	300.0	315.0	55.0	309.2	[231.8, 358.3]

Table 12: Full evaluation results for **Seaquest** across all tested α–λ_g configurations. IQM = interquartile mean; 95% confidence interval is across five seeds. **Bold** = highest value per metric across all rows. $\lambda_g = 0$ rows omitted (equivalent to baseline).

α	λ_g	Mean	Median	Std	IQM	95% CI
Baseline	—	1638.5	1820.0	359.9	1792.1	[1200.6, 1875.4]
$\alpha = 0.00$	$\lambda_g = 0.01$	1657.4	1820.0	363.4	1826.7	[1229.5, 1860.0]
	$\lambda_g = 0.03$	1484.0	1830.0	436.1	1546.4	[946.7, 1840.5]
	$\lambda_g = 0.05$	1474.3	1820.0	432.3	1530.5	[943.3, 1836.9]
	$\lambda_g = 0.10$	1598.4	1830.0	345.7	1720.6	[1140.9, 1860.0]
	$\lambda_g = 0.30$	1844.8	**1840.0**	16.7	1841.3	[1829.7, 1863.6]
	$\lambda_g = 0.50$	1827.5	1820.0	10.7	1827.9	[1815.6, 1840.0]
	$\lambda_g = 0.70$	1823.7	1820.0	7.2	1823.1	[1816.1, 1832.8]
$\alpha = 0.25$	$\lambda_g = 0.01$	1811.7	**1840.0**	90.3	1844.9	[1703.1, 1877.7]
	$\lambda_g = 0.03$	1491.1	**1840.0**	435.5	1548.2	[956.9, 1851.8]
	$\lambda_g = 0.05$	1914.2	**1840.0**	583.5	**1979.7**	[1236.4, 2554.7]
	$\lambda_g = 0.10$	1302.0	960.0	433.4	1246.7	[939.7, 1833.6]
	$\lambda_g = 0.30$	1666.3	**1840.0**	357.8	1843.3	[1247.2, 1850.5]
	$\lambda_g = 0.50$	1830.6	**1840.0**	**4.7**	1830.8	[1825.1, 1835.9]
	$\lambda_g = 0.70$	1808.3	1800.0	11.4	1805.9	[1798.7, 1823.1]
$\alpha = 0.50$	$\lambda_g = 0.01$	1664.3	**1840.0**	354.7	1833.3	[1246.9, 1855.9]
	$\lambda_g = 0.03$	1828.3	**1840.0**	37.5	1830.5	[1782.3, 1870.0]
	$\lambda_g = 0.05$	1664.2	**1840.0**	352.3	1833.1	[1246.4, 1854.4]
	$\lambda_g = 0.10$	1738.0	**1840.0**	420.5	1840.3	[1246.7, 2093.2]
	$\lambda_g = 0.30$	1832.3	**1840.0**	16.8	1838.5	[1811.8, 1845.7]
	$\lambda_g = 0.50$	1813.2	1820.0	23.9	1820.0	[1782.5, 1835.1]
	$\lambda_g = 0.70$	1811.4	1820.0	18.1	1808.7	[1792.1, 1834.1]
$\alpha = 0.75$	$\lambda_g = 0.01$	1496.3	**1840.0**	440.2	1545.1	[956.9, 1873.9]
	$\lambda_g = 0.03$	1828.0	**1840.0**	546.2	1841.5	[1237.7, 2397.5]
	$\lambda_g = 0.05$	**2009.1**	**1840.0**	338.2	1841.5	[1836.9, 2405.1]
	$\lambda_g = 0.10$	1757.5	**1840.0**	474.1	1794.3	[1208.6, 2244.9]
	$\lambda_g = 0.30$	1817.8	**1840.0**	37.2	1828.7	[1770.3, 1850.0]
	$\lambda_g = 0.50$	1967.7	1830.0	303.2	1821.3	[1805.1, 2325.6]
	$\lambda_g = 0.70$	1805.7	1800.0	11.5	1806.4	[1791.3, 1817.9]
$\alpha = 1.00$	$\lambda_g = 0.01$	1111.7	960.0	340.3	952.1	[919.3, 1514.3]
	$\lambda_g = 0.03$	1110.5	960.0	355.5	942.3	[916.3, 1528.5]
	$\lambda_g = 0.05$	1483.1	1820.0	433.5	1541.8	[949.7, 1843.6]
	$\lambda_g = 0.10$	1662.9	**1840.0**	351.5	1836.7	[1250.0, 1843.1]
	$\lambda_g = 0.30$	1751.9	1820.0	95.6	1781.3	[1629.6, 1836.4]
	$\lambda_g = 0.50$	1825.4	1820.0	9.2	1826.9	[1813.9, 1835.1]
	$\lambda_g = 0.70$	1810.2	1820.0	16.9	1810.3	[1791.5, 1825.4]

Table 13: Full evaluation results for **Qbert** across all tested α–λ_g configurations. IQM = interquartile mean; 95% confidence interval is across five seeds. **Bold** = highest value per metric across all rows. $\lambda_g = 0$ rows omitted (equivalent to baseline).

α	λ_g	Mean	Median	Std	IQM	95% CI
Baseline	—	17822.8	**19150.0**	1257.7	18085.2	[16192.3, 19009.8]
$\alpha = 0.00$	$\lambda_g = 0.01$	18420.3	16450.0	3078.6	17955.1	[15562.8, 22403.3]
	$\lambda_g = 0.03$	17679.7	16800.0	1471.5	17283.1	[16423.3, 19625.2]
	$\lambda_g = 0.05$	16987.5	15925.0	2360.3	17042.3	[14438.1, 19821.7]
	$\lambda_g = 0.10$	17115.2	16487.5	1201.6	17404.8	[15937.8, 18211.3]
	$\lambda_g = 0.30$	16337.6	15900.0	2371.4	16667.0	[13569.3, 18783.2]
	$\lambda_g = 0.50$	16023.2	16025.0	470.9	16045.3	[15481.2, 16559.0]
	$\lambda_g = 0.70$	15268.0	15300.0	1143.0	15448.6	[13936.7, 16384.9]
$\alpha = 0.25$	$\lambda_g = 0.01$	17568.5	17912.5	3281.2	18098.7	[13420.1, 20992.4]
	$\lambda_g = 0.03$	17643.5	18662.5	1781.1	17604.9	[15557.5, 19681.6]
	$\lambda_g = 0.05$	18039.5	16425.0	1191.8	18500.0	[16525.3, 18932.8]
	$\lambda_g = 0.10$	**19294.2**	19050.0	1358.5	**19561.3**	[17513.7, 20518.9]
	$\lambda_g = 0.30$	15996.9	15575.0	1198.5	15851.8	[14776.9, 17484.3]
	$\lambda_g = 0.50$	16122.5	15850.0	586.4	16185.2	[15430.8, 16784.8]
	$\lambda_g = 0.70$	13218.7	11975.0	1655.9	12864.8	[11674.5, 15350.2]
$\alpha = 0.50$	$\lambda_g = 0.01$	18712.2	17762.5	3088.2	18578.7	[15518.7, 22392.0]
	$\lambda_g = 0.03$	17060.2	16175.0	1346.6	16974.1	[15597.6, 18688.5]
	$\lambda_g = 0.05$	17827.3	17150.0	1755.8	17587.9	[16080.2, 20087.9]
	$\lambda_g = 0.10$	17983.5	18725.0	2373.0	17725.3	[15296.9, 20661.7]
	$\lambda_g = 0.30$	16118.0	15650.0	955.0	16050.6	[15159.3, 17292.8]
	$\lambda_g = 0.50$	15919.6	15337.5	2573.4	15239.2	[13657.2, 18501.4]
	$\lambda_g = 0.70$	12767.0	11650.0	2482.6	13314.9	[9494.8, 14981.7]
$\alpha = 0.75$	$\lambda_g = 0.01$	15969.8	15975.0	2080.5	16233.2	[13536.8, 18032.4]
	$\lambda_g = 0.03$	17026.7	15987.5	1794.4	16458.0	[15523.3, 19377.2]
	$\lambda_g = 0.05$	18621.5	16862.5	2929.8	18439.4	[15813.5, 22213.4]
	$\lambda_g = 0.10$	18119.7	18975.0	882.2	18180.0	[17123.5, 19063.8]
	$\lambda_g = 0.30$	15599.7	15425.0	**266.0**	15585.7	[15303.6, 15922.2]
	$\lambda_g = 0.50$	15848.6	15625.0	1514.7	15820.6	[14197.7, 17525.3]
	$\lambda_g = 0.70$	14673.8	15100.0	1737.8	15210.0	[12477.3, 16143.2]
$\alpha = 1.00$	$\lambda_g = 0.01$	18154.2	18925.0	1257.6	18280.9	[16586.5, 19299.5]
	$\lambda_g = 0.03$	15758.0	15725.0	1012.9	15838.3	[14531.6, 16903.4]
	$\lambda_g = 0.05$	16159.5	15825.0	2262.3	16403.5	[13493.0, 18546.9]
	$\lambda_g = 0.10$	17544.0	17050.0	1719.4	17614.3	[15508.2, 19440.8]
	$\lambda_g = 0.30$	15119.5	15687.5	611.4	14976.0	[14545.5, 15896.2]
	$\lambda_g = 0.50$	14429.3	15275.0	630.7	14463.8	[13718.6, 15171.2]
	$\lambda_g = 0.70$	15193.6	15775.0	599.9	15345.0	[14466.1, 15778.7]