ADAGAMMA: STATE-DEPENDENT DISCOUNTING FOR TEMPORAL ADAPTATION IN REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

The discount factor in reinforcement learning controls both the effective planning horizon and the strength of bootstrapping, yet most deep RL methods use a single fixed value across all states. State-dependent discounting is a natural way to adapt temporal credit assignment to local state structure, but naively learning a discount function inside deep actor–critic methods creates a self-referential target: the learned discount scales the same bootstrap term against which it is trained. Consequently, TD-error-based objectives can admit a degenerate shortcut, reducing target dependence on downstream value estimates instead of learning meaningful temporal structure. We propose **AdaGamma**, a practical framework for state-dependent discounting in deep actor–critic reinforcement learning. AdaGamma learns a bounded discount function $\gamma_\phi(s)$ and trains it using a return-consistency objective that matches a one-step adaptive bootstrap to a multi-step return computed with a lagged, stop-gradient reference discount. This removes the direct target manipulation path and encourages the discount to encode state-dependent value propagation. We analyze the Bellman operator induced by a fixed state-dependent discount in finite tabular MDPs, proving contraction, exact soft policy iteration, and Lipschitz sensitivity under standard boundedness assumptions. Empirically, AdaGamma integrates into SAC and PPO, avoids TD-collapse, improves continuous-control benchmarks, and achieves statistically significant gains in a four-week online A/B test on the JD Logistics platform. The code can be accessed at `https://anonymous.4open.science/r/AdaGamma_ICLR2027-5825`.

1 INTRODUCTION

The discount factor is a central design choice in reinforcement learning: it sets the effective planning horizon and the strength of bootstrapping in value estimation (Sutton & Barto, 1998; 2018; Puterman, 1994). Most deep RL methods, including TD3 (Fujimoto et al., 2018), TRPO (Schulman et al., 2017a), DDPG (Lillicrap et al., 2019), PPO (Schulman et al., 2017b), and SAC (Haarnoja et al., 2018), use one fixed value for all states (Mnih et al., 2015; Schulman et al., 2017b; Haarnoja et al., 2018).

For a discounted objective, γ is part of the problem specification. But when deep RL is used to optimize an undiscounted or long-horizon return, $\gamma < 1$ is better viewed as an estimation-time bias–variance choice rather than a property of the task. Discounting shortens the horizon over which errors compound, trading asymptotic fidelity for lower variance, and thus acts as a complexity control on the value class (Amit et al., 2020; Rathnam et al., 2024). This trade-off is inherently local: target variance, transition reliability, and reachable remaining horizon all vary across states (François-Lavet et al., 2016; Pitis, 2019b; Hu et al., 2022). A single global γ therefore imposes one compromise on states that may require different ones, motivating state-conditioned discounting.

Learning a discount module $\gamma_\phi(s)$ is the natural response, but it creates a structural problem. Unlike most auxiliary modules in actor–critic systems, γ_ϕ directly multiplies the bootstrap target it is trained against. Hence any objective that scores γ_ϕ by the resulting TD residual admits a degenerate shortcut: reducing the bootstrap coefficient weakens the target's dependence on downstream value estimates, allowing the loss to decrease without learning meaningful state-dependent temporal

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.

structure. The module becomes a lever for manipulating the target rather than a description of value propagation. Empirically, this is what occurs. On `SafetyPointGoal1-v0` under SAC, with the paper's γ-regularizers held fixed, squared-TD training of γ_ϕ degenerates to a near-constant discount ($\mathrm{Std}_s[\gamma_\phi(s)] \approx 0.0014$), while return-consistency retains $\mathrm{Std}_s[\gamma_\phi(s)] \approx 0.022$. A cross-validated TD-error objective also collapses despite its held-out split, giving mean $\gamma = 0.9000$ on all three reported environments (Table 5).

AdaGamma removes this self-reference instead of penalizing its symptoms. We supervise γ_ϕ against an n-step return computed with a lagged scalar reference discount $\bar{\gamma}$, and the supervisory target is stop-gradient with respect to the current $\gamma_\phi(s_t)$. Shrinking the horizon can no longer trivially reduce the loss: the one-step estimate $r + \gamma_\phi(s)V(s')$ matches the multi-step return only when $\gamma_\phi(s)$ captures how much longer-horizon value the immediate transition carries. The learned discount thus performs adaptive bootstrapping, propagating farther where future information is reliable and relying less on bootstrapped targets where it is not.

On the theory side, we characterize the Bellman operator induced by a *fixed* state-dependent discount function in a finite tabular MDP. Under bounded rewards and $\gamma(\cdot) \le \gamma_{\max} < 1$, the operator is a contraction, exact soft policy iteration converges, and Q^π_γ is Lipschitz in $\gamma(\cdot)$. We state the scope explicitly: the analysis covers this tabular operator with a frozen $\gamma(\cdot)$, not function approximation and not the joint dynamics of γ_ϕ and the critic, and it does not claim that state-dependent discounting dominates constant discounting for every MDP. The results align most directly with Bellman-style updates and soft policy iteration; our PPO variant is an empirical adaptation rather than an instance of the theory.

We instantiate AdaGamma in SAC and adapt it to PPO through corresponding return-estimation changes. Both improve continuous-control benchmarks, with larger gains in SAC. Finally, we deploy the SAC variant in a live recommender system on the JD Logistics platform, where it outperforms a SAC baseline in a four-week online A/B test.

Contributions.

- **Failure mode and method.** We identify a failure mode of TD-trained learned discounting: because $\gamma_\phi(s)$ appears in its own target, it can converge to a state-independent fixed point. AdaGamma avoids this by fitting γ_ϕ to multi-step returns computed with an external reference discount.
- **Theory.** We characterize the tabular Bellman operator induced by a fixed state-dependent discount, establishing contraction, exact soft policy iteration, and Lipschitz sensitivity under bounded rewards and $\gamma(\cdot) \le \gamma_{\max} < 1$, and we state what the analysis does and does not cover.
- **Integration and validation.** AdaGamma integrates into SAC and PPO; we report continuous-control results, a TD-versus-RC collapse ablation, and a four-week online A/B test on JD Logistics.

Related work is discussed in Appendix A.

2 Preliminaries

Notation We consider an infinite-horizon MDP $(\mathcal{S}, \mathcal{A}, p, r)$ with continuous state space $\mathcal{S}$ and action space $\mathcal{A}$. The transition density is denoted by $p(s'|s,a)$, equivalently $p : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, \infty)$, and the reward function $r : \mathcal{S} \times \mathcal{A} \to [r_{\min}, r_{\max}]$ is bounded. We use $\rho_\pi(s_t)$ and $\rho_\pi(s_t, a_t)$ to denote the state and state-action marginals of the trajectory distribution induced by policy $\pi(a_t|s_t)$.

The Discount Factor's Dual Role The discount factor γ serves two conceptually distinct functions: it defines the *objective* (the γ-discounted return) and it acts as a *damping coefficient* in the Bellman backup that controls how much bootstrapped future estimates influence current value updates. A low γ prioritizes short-term rewards and reduces variance at the cost of bias; a high γ encourages long-horizon planning but amplifies errors in value estimates. In standard algorithms, γ is fixed, typically to 0.99 or 0.995.

Soft Actor-Critic (SAC) SAC (Haarnoja et al., 2018) maximizes the entropy-augmented objective:

$$J(\pi) = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t \left(r(s_t, a_t) + \alpha \mathcal{H}(\pi(\cdot|s_t)) \right) \right]. \tag{1}$$

The soft Q-function satisfies the Bellman equation:

$$Q^\pi(s_t, a_t) = r(s_t, a_t) + \gamma \, \mathbb{E}_{s_{t+1} \sim p} \left[V^\pi(s_{t+1}) \right], \tag{2}$$

where $V^\pi(s_t) = \mathbb{E}_{a_t \sim \pi}[Q^\pi(s_t, a_t) - \alpha \log \pi(a_t|s_t)]$.

PPO and Generalized Advantage Estimation PPO (Schulman et al., 2017b) uses GAE to compute advantage estimates. The standard GAE recursion with fixed γ and trace parameter λ is:

$$\delta_t = r_t + \gamma V(s_{t+1}) - V(s_t), \tag{3}$$

$$\hat{A}_t = \delta_t + \gamma\lambda \, \hat{A}_{t+1}, \quad \hat{A}_{T-1} = \delta_{T-1}. \tag{4}$$

Expanding the recursion yields $\hat{A}_t = \sum_{l=0}^{T-1-t} (\gamma\lambda)^l \delta_{t+l}$, a geometrically-weighted sum of TD residuals.

3 AdaGamma: Method and Theory

We present AdaGamma as a unified framework with three components: (i) the gamma network architecture (Section 3.1), (ii) algorithm-specific adapters that integrate $\gamma_\phi(s)$ into bootstrapped value estimation (Sections 3.2.1 and 3.2.2), and (iii) the training objective for the gamma network (Section 3.3). We then provide supporting theory for the state-dependent discount operator underlying the method. Our theoretical results characterize basic operator-level properties such as well-posedness, policy-evaluation behavior, and comparison bounds under suitable assumptions; they are not intended as a full end-to-end convergence guarantee for the complete deep RL algorithm with function approximation. Algorithm 1 in Appendix G.1 presents the full SAC-AdaGamma procedure, and Algorithm 2 in Appendix G.2 presents the PPO-AdaGamma variant.

3.1 Gamma Network Architecture

The gamma network is a small MLP $g_\phi : \mathcal{S} \to \mathbb{R}$ whose output is passed through a sigmoid and rescaled to $[\gamma_{\min}, \gamma_{\max}]$:

$$\gamma_\phi(s_t) = \gamma_{\min} + (\gamma_{\max} - \gamma_{\min}) \cdot \sigma(g_\phi(s_t)), \tag{5}$$

where $\sigma(\cdot)$ is the sigmoid function and $\gamma_{\min}, \gamma_{\max}$ are hyperparameters (we use $\gamma_{\min} = 0.900$, $\gamma_{\max} = 0.999$ in experiments). This bounded parameterization ensures $\gamma_\phi(s) \in [\gamma_{\min}, \gamma_{\max}] \subset [0, 1)$ for all states and keeps the bootstrap coefficient strictly below one. Together with bounded rewards and standard boundedness or integrability conditions on the entropy term, this keeps the induced discounted backup well behaved.

The gamma network is deliberately kept lightweight—a two-layer MLP with hidden dimension 256—to avoid overfitting and to ensure negligible computational overhead. It takes the same state representation as the policy and value networks but uses no shared parameters, preventing gradient interference.

3.2 The Common Interface: Bootstrapped Value Targets

Despite their differences, both SAC and PPO share one operation where γ appears: the bootstrapped target used in value estimation. AdaGamma interfaces with each algorithm at precisely this point, replacing the scalar γ with $\gamma_\phi(s_t)$.

3.2.1 Off-Policy Adapter (SAC)

For SAC, the soft Q-function with adaptive discount becomes:

$$Q^\pi(s_t, a_t) = r(s_t, a_t) + \mathbb{E}_{s_{t+1} \sim p} \left[\gamma_\phi(s_t) \, \mathbb{E}_{a_{t+1} \sim \pi} \left[Q^\pi(s_{t+1}, a_{t+1}) - \alpha \log \pi(a_{t+1}|s_{t+1}) \right] \right]. \tag{6}$$

The practical target computation becomes:

$$\hat{Q}(s_t, a_t) = r_t + \gamma_\phi(s_t)(1 - d_t)\left[\min_{i=1,2} Q_{\bar{\theta}_i}(s_{t+1}, a'_{t+1}) - \alpha \log \pi_\psi(a'_{t+1}|s_{t+1})\right], \tag{7}$$

where $a'_{t+1} \sim \pi_\psi(\cdot|s_{t+1})$ and $d_t \in \{0, 1\}$ is the terminal flag. The policy update and entropy tuning are unchanged from standard SAC.

3.2.2 On-Policy Adapter (PPO): Modified GAE

PPO is an empirical adaptation of the same interface; we do not claim a parallel theory for clipping or GAE. The state-dependent discount enters the TD residual and the GAE recursion:

$$\delta_t = r_t + \gamma_\phi(s_t)(1 - d_t)\, V(s_{t+1}) - V(s_t), \tag{8}$$

and the modified GAE backward recursion becomes:

$$\hat{A}_{T-1} = \delta_{T-1}, \qquad \hat{A}_t = \delta_t + \gamma_\phi(s_t)\lambda(1 - d_t)\hat{A}_{t+1}. \tag{9}$$

Unrolling (9) yields a product-of-discounts weighting of future residuals (Proposition 1 in Appendix B.1).

A low-γ nonterminal state along a trajectory suppresses subsequent residuals, inducing a learned form of temporal segmentation; terminal states truncate the recursion in the usual way. Advantage normalization and freezing γ_ϕ within a PPO epoch are described in Appendix G.2.

3.3 Training the Gamma Network

3.3.1 The Collapse Problem

A natural first attempt is to train γ_ϕ by minimizing the squared TD error δ_t^2, since $\gamma_\phi(s_t)$ appears in δ_t. However, this creates a degenerate shortcut: γ_ϕ can collapse to a state-independent fixed point rather than learn meaningful state-dependent horizons. We verify this TD collapse empirically in Section 4.5, and show that AdaGamma avoids it.

3.3.2 Return-Consistency Objective

We instead train the gamma network via a *return-consistency* objective. The idea is to use multi-step returns as a supervision signal that the gamma network cannot trivially game.

For each transition (s_t, a_t, r_t, s_{t+1}), compute two value estimates:

1. The one-step bootstrap under the learned γ: $\hat{V}_1(s_t) = r_t + \gamma_\phi(s_t)\, V(s_{t+1})$.
2. An n-step Monte Carlo return under a reference discount $\bar{\gamma}$, truncated at terminals:

$$G_t^{(n)} = \sum_{k=0}^{n-1} \left(\prod_{j=0}^{k-1} \bar{\gamma}(1 - d_{t+j})\right) r_{t+k} + \left(\prod_{j=0}^{n-1} \bar{\gamma}(1 - d_{t+j})\right) V(s_{t+n}), \tag{10}$$

   computed with stop-gradient on V.

The gamma network is trained to minimize:

$$L_\gamma^{\mathrm{RC}}(\phi) = \mathbb{E}_{(s_t, a_t)\sim\mathcal{D}}\left[\left(r_t + \gamma_\phi(s_t)\, \mathrm{sg}[V(s_{t+1})] - \mathrm{sg}[G_t^{(n)}]\right)^2\right], \tag{11}$$

where $\mathrm{sg}[\cdot]$ denotes the stop-gradient operator. The gamma network learns to set $\gamma_\phi(s_t)$ so that the local one-step bootstrap better matches a longer-horizon target, thereby adapting the degree of temporal propagation to the local state. In states where one-step bootstrapping provides a reliable summary of longer-horizon value, the learned discount tends to be larger. In states where immediate transitions are less predictive or more weakly aligned with longer-horizon outcomes, the learned discount tends to be smaller. Critically, pushing $\gamma_\phi(s_t)$ toward the lower end of its admissible range

does *not* in general minimize this loss, because the corresponding one-step estimate need not equal the multi-step return $G_t^{(n)}$.

In its simplest form, $\bar{\gamma}$ can be fixed to a constant such as 0.99. In our SAC safety implementation, however, we use a slowly adaptive reference discount to better match the current training distribution. After warmup, every M episodes we sample a replay batch $\mathcal{B}$, compute the replay-mean predicted discount $\mathbb{E}_{s\sim\mathcal{B}}[\gamma_\phi(s)]$, and update

$$\bar{\gamma} \leftarrow (1-\tau_{\text{ref}})\bar{\gamma} + \tau_{\text{ref}}\,\mathbb{E}_{s\sim\mathcal{B}}[\gamma_\phi(s)]. \tag{12}$$

The supervisory target is computed using a lagged scalar reference discount and is stop-gradient with respect to the current $\gamma_\phi(s_t)$. Thus the gamma network cannot reduce the loss by directly changing the same bootstrap coefficient that defines its target. The EMA update keeps the n-step supervisory target smoother than using a fully state-dependent product inside $G_t^{(n)}$, while allowing the effective reference horizon to track the replay distribution over the course of training.

For off-policy algorithms, the n-step return is computed from sequences of transitions in the replay buffer and is truncated at episode boundaries. For on-policy algorithms, the rollout data naturally provides multi-step returns, again with terminal masks applied. For SAC, $V(s_{t+1})$ denotes the sampled soft value $\min_i Q_{\bar{\theta}_i}(s_{t+1}, a') - \alpha \log \pi(a'|s_{t+1})$, with $a' \sim \pi(\cdot|s_{t+1})$. For PPO, V is the learned state-value network.

A cross-validated TD baseline, used only as a comparator, is defined in Appendix E.

3.3.3 Full Training Objective

The complete gamma network loss combines the return-consistency objective with regularization:

$$J_\gamma(\phi) = L_\gamma^{\text{RC}}(\phi) + \lambda_{\text{dev}}\,\mathbb{E}_{s_t\sim\mathcal{D}}\left[(\gamma_\phi(s_t) - \gamma_{\text{target}})^2\right] + \lambda_{\text{var}}\,\text{Var}_{s_t\sim\mathcal{D}}[\gamma_\phi(s_t)] + \lambda_{\text{bound}}\,L_{\text{boundary}}, \tag{13}$$

where the deviation penalty anchors γ_ϕ near γ_{target}, the variance penalty encourages smoothness, and L_{boundary} discourages values near the boundaries:

$$L_{\text{boundary}} = \mathbb{E}_{s_t\sim\mathcal{D}}\left[\text{ReLU}(\gamma_{\min} + \epsilon_b - \gamma_\phi(s_t)) + \text{ReLU}(\gamma_\phi(s_t) - \gamma_{\max} + \epsilon_b)\right]. \tag{14}$$

3.4 Operator-Level Analysis for State-Dependent Discounting

We now study the Bellman operators induced by state-dependent discounting and establish supporting guarantees for the resulting policy-evaluation and policy-improvement dynamics. Our goal is not to prove end-to-end convergence of the full AdaGamma algorithm, which additionally involves neural function approximation, stochastic optimization, target networks, an adaptive reference discount, and alternating actor–critic–gamma updates. Instead, we isolate the operator-level consequences of replacing a scalar discount factor by a state-dependent discount function.

For analytical clarity, we specialize the general setting of Section 3.3 to a finite tabular MDP. Unless stated otherwise, the following standing assumption is imposed throughout this section.

Assumption 1. *The state space $\mathcal{S}$ and the action space $\mathcal{A}$ are finite with $m := |\mathcal{A}|$, the reward is bounded as $|r(s,a)| \le R$, $\forall (s,a) \in \mathcal{S}\times\mathcal{A}$, and the entropy temperature $\alpha > 0$ is fixed. All expectations are evaluated exactly. The state-dependent discount function satisfies $\gamma : \mathcal{S} \to [0,\beta]$, where $\beta < 1$.*

For policy-iteration results, we additionally consider the unrestricted class of stationary tabular policies and assume exact policy evaluation and exact statewise soft-greedy improvement.

3.4.1 Soft Policy Evaluation

We first consider policy evaluation under a fixed state-dependent discount function. For a stationary policy π, define

$$V_Q^\pi(s) = \sum_{a\in\mathcal{A}} \pi(a \mid s)\big(Q(s,a) - \alpha \log \pi(a \mid s)\big), \tag{15}$$

with the convention $0 \log 0 = 0$. The corresponding state-dependent soft Bellman operator is

$$(\mathcal{T}_\gamma^\pi Q)(s,a) = r(s,a) + \gamma(s)\,\mathbb{E}_{s'\sim P(\cdot|s,a)}\left[V_Q^\pi(s')\right]. \tag{16}$$

Lemma 1. *(Fixed-policy contraction under state-dependent discounting) Under Assumption 1, for any stationary policy π, the operator $\mathcal{T}_\gamma^\pi$ is a β-contraction in the sup norm:*

$$\|\mathcal{T}_\gamma^\pi Q - \mathcal{T}_\gamma^\pi Q'\|_\infty \le \beta \|Q - Q'\|_\infty. \tag{17}$$

Consequently, $\mathcal{T}_\gamma^\pi$ admits a unique fixed point Q_γ^π, and the exact iteration $Q^{k+1} = \mathcal{T}_\gamma^\pi Q^k$ converges to Q_γ^π from any initial Q^0.

Proof. See Appendix B.2. □

The n-fold operator $(\mathcal{T}_\gamma^\pi)^n$ is a β^n-contraction and shares the same unique fixed point; along a trajectory the difference of backups is weighted by the product of encountered discounts (Proposition 2 in Appendix B.3).

3.4.2 Exact Soft Policy Iteration under a Fixed State-Dependent Discount

We next study exact soft policy improvement and policy iteration for a *fixed* state-dependent discount function. Throughout this subsection, the same discount function $\gamma(\cdot)$ is used in both policy evaluation and policy improvement.

For $\alpha > 0$, define the soft optimality operator

$$(\mathcal{T}_\gamma Q)(s,a) = r(s,a) + \gamma(s)\, \mathbb{E}_{s' \sim P(\cdot|s,a)} \left[\alpha \log \sum_{b \in \mathcal{A}} \exp\left(\frac{Q(s',b)}{\alpha} \right) \right]. \tag{18}$$

For any Q, the corresponding statewise soft-greedy policy is

$$\pi_Q(a \mid s) = \frac{\exp(Q(s,a)/\alpha)}{\sum_{b \in \mathcal{A}} \exp(Q(s,b)/\alpha)}. \tag{19}$$

Exact soft-greedy improvement is monotone (Lemma 2 in Appendix B.4): replacing π by $\pi_{Q_\gamma^\pi}$ cannot decrease the soft Q-function.

Theorem 1. *(Exact soft policy iteration) Under Assumption 1, let π_0 be any stationary tabular policy and define, for $k \ge 0$, $Q_k := Q_\gamma^{\pi_k}$ and $\pi_{k+1} = \pi_{Q_k}$, where $Q_\gamma^{\pi_k}$ denotes the exact soft Q-function of π_k under the fixed state-dependent discount $\gamma(\cdot)$. Then $\{Q_k\}_k$ converges in the sup norm to the unique fixed point $Q_\gamma^\star$ of the soft optimality operator $\mathcal{T}_\gamma$. Moreover, $\pi_k \to \pi^\star := \pi_{Q_\gamma^\star}$, where $\pi^\star$ is the unique statewise soft-greedy policy associated with $Q_\gamma^\star$. In addition,*

$$Q_\gamma^\star(s,a) \ge Q_\gamma^\pi(s,a), \qquad \forall\, \pi, \quad \forall\, (s,a) \in \mathcal{S} \times \mathcal{A}, \tag{20}$$

where π ranges over all stationary tabular policies.

Proof. See Appendix B.5. □

The map $\gamma(\cdot) \mapsto Q_\gamma^\pi$ is Lipschitz in the sup norm, with constant scaling as $(R + \alpha \log m)/((1-\beta_1)(1-\beta_2))$ (Theorem 2 in Appendix B.6).

If an exogenous sequence γ_k varies sufficiently slowly, exact fixed-policy backups track the moving fixed points (Corollary 1 in Appendix B.7); $\{\gamma_k\}$ is not the endogenous return-consistency dynamics.

4 Numerical Experiments

4.1 Experimental Setup

Algorithms and baselines. We instantiate AdaGamma in SAC (Haarnoja et al., 2018) and PPO (Schulman et al., 2017b), using the same loss form (13) with algorithm-specific adapters (off-policy for SAC, on-policy for PPO). The gamma network is a two-layer MLP (hidden size 256) trained via the return-consistency objective of Section 3.3.2. Under a matched training budget, we compare against: fixed-γ SAC/PPO (γ=0.99), an uncertainty-rule adaptive-γ baseline (inspired by Kim et al.

Table 1: Test performance of fixed, adaptive, and AdaGamma variants on `SafetyPointGoal1-v0`, `Humanoid-v4`, and `Ant-v4`.

Method	Reward	Cost	Method	Reward	Cost
`SafetyPointGoal1-v0`					
SAC-Fixed-γ	27.82 ± 0.91	52.53 ± 40.75	PPO-Fixed-γ	22.58 ± 1.34	51.26 ± 34.51
SAC-CrossValidate	27.34 ± 1.05	45.84 ± 31.14	PPO-CrossValidate	20.62 ± 5.81	54.13 ± 27.09
SAC-Uncertainty	27.47 ± 1.24	37.46 ± 32.93	PPO-Uncertainty	22.59 ± 1.35	51.25 ± 34.51
SAC-AdaGamma	$\mathbf{28.25 \pm 1.05}$	$\mathbf{29.37 \pm 18.59}$	PPO-AdaGamma	$\mathbf{26.31 \pm 0.90}$	$\mathbf{41.82 \pm 19.42}$
`Humanoid-v4`					
SAC-Fixed-γ	5370.80 ± 11.24	-	PPO-Fixed-γ	411.61 ± 39.99	-
SAC-CrossValidate	616.25 ± 21.88	-	PPO-CrossValidate	291.42 ± 14.48	-
SAC-Uncertainty	6606.08 ± 59.69	-	PPO-Uncertainty	434.32 ± 98.70	-
SAC-AdaGamma	$\mathbf{6907.99 \pm 21.76}$	-	PPO-AdaGamma	$\mathbf{476.51 \pm 51.29}$	-
`Ant-v4`					
SAC-Fixed-γ	3101.48 ± 748.16	-	PPO-Fixed-γ	989.57 ± 113.71	-
SAC-CrossValidate	1399.44 ± 107.24	-	PPO-CrossValidate	1036.33 ± 110.85	-
SAC-Uncertainty	3620.87 ± 187.90	-	PPO-Uncertainty	1018.51 ± 222.44	-
SAC-AdaGamma	$\mathbf{4129.57 \pm 1155.08}$	-	PPO-AdaGamma	$\mathbf{1172.47 \pm 189.84}$	-

(2022); details in Appendix C), and a cross-validated adaptive-γ network. All methods use 5 seeds and are evaluated every 10^4 steps. For AdaGamma, we use 5-step returns, initialize γ and $\bar{\gamma}$ at 0.98, warm up for 10^5 steps, and update $\bar{\gamma}$ every 5 episodes (EMA coefficient 0.1).

Environments. We evaluate AdaGamma in four settings: `SafetyPointGoal1-v0` (Ji et al., 2023) for reward–cost tradeoffs and seed-shift generalization; `Humanoid-v4` and `Ant-v4` for high-dimensional continuous control; classic Gymnasium tasks (Towers et al., 2024) for smaller-scale SAC/PPO comparisons; and a JD Logistics target-marketing deployment for real population-specific decision making. Unless otherwise stated, we report mean $\pm$ standard deviation over random seeds.

Hyperparameter details are given in Appendix K; learning curves for SAC-AdaGamma and PPO-AdaGamma on `SafetyPointGoal1-v0`, `Humanoid-v4`, and `Ant-v4` are shown in Appendix G.3.

4.2 Continuous-control benchmarks

Table 1 reports SAC and PPO on `SafetyPointGoal1-v0`, `Humanoid-v4`, and `Ant-v4`. AdaGamma attains the best reward on all three tasks for both algorithms, and the lowest cost on the safety task. The uncertainty-rule baseline already improves on fixed γ, while the cross-validated TD baseline is weak on `Humanoid-v4`. The largest gain is SAC on `Humanoid-v4`. Snapshots are in Appendix J.

4.3 Online deployment on JD Logistics platform

We further evaluate AdaGamma in a live recommender system on the JD Logistics platform through a controlled four-week online A/B test. We randomly assign approximately one million users ($N \approx 10^6$) to the experiment, split evenly between AdaGamma-enhanced SAC and a standard SAC baseline ($\approx 5 \times 10^5$ users per arm). Users are stratified and split evenly between the two arms so that the distributions of pre-experiment features are balanced, including user order volume, order cycle, coupon usage frequency, coupon claiming statistics, and other basic user attributes. All these pre-experiment balance checks pass standard p-value tests. During the experiment the two groups are treated identically except for the deployed recommendation policy, so that the observed difference can be attributed to the RL algorithm.

Task and reward. The policy selects coupon recommendations under a limited low-tier quota. The reward combines re-scaled absolute order growth, relative order growth, and a penalty for discounts deeper than a medium-tier reference, so that order gains must outweigh subsidy cost (Appendix D).

Primary metric and subsidy guardrail. The primary evaluation metric is the relative uplift in average recommendation-induced orders per user. Each week we measure the incremental improvement over the previous seven days relative to the state at group assignment. As a guardrail we track the realized overall discount rate (coupon subsidy as a fraction of GMV). The two arms finished

the experiment with essentially matched overall discount intensity (relative gap below 0.1%), so the uplift in average recommendation-induced orders per user cannot be attributed to a more aggressive subsidy. As shown in Figure 1, AdaGamma consistently outperforms standard SAC over four weeks; standard significance tests indicate reliable gains. These results suggest that adaptive discounting reallocates a matched discount budget toward user states that benefit from a longer planning horizon, rather than simply spending more.

4.4 Analysis of Learned $\gamma_\phi(s)$

Statewise structure. On `Ant-v4`, SAC-AdaGamma learns a non-constant $\gamma_\phi(s)$ that varies with torso height and planar velocity (Figure 5 in Appendix F.3). Together with the matched fixed-γ ablation in Table 7, this suggests that the gains are not explained solely by the learned mean discount.

Cross-algorithm consistency. Mean learned discounts of SAC-AdaGamma and PPO-AdaGamma are closely aligned on most tasks (Table 4 in Appendix F.4).

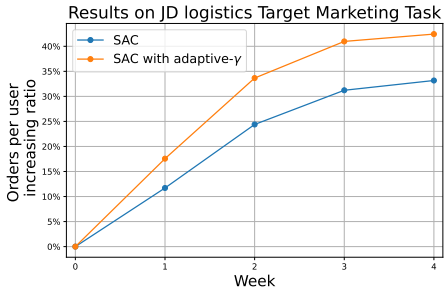


Figure 1: Four-week online A/B test on the JD Logistics target-marketing task. The plotted metric is the relative uplift in average recommendation-induced orders per user for standard SAC and SAC with AdaGamma. Overall discount intensity was matched across arms.

4.5 Empirical Verification of TD-target Collapse

Section 3.3.1 argues that scoring γ_ϕ by the squared TD residual admits a degenerate shortcut: reducing the bootstrap coefficient can weaken the target's dependence on downstream value estimates and lower the TD loss without learning meaningful state-dependent temporal structure. We test this claim on `SafetyPointGoal1-v0` under SAC, comparing return-consistency (AdaGamma-RC) against a squared-TD γ objective (AdaGamma-TD) with the same backbone, γ-range $[0.9, 0.999]$, 100k-step γ-warmup, and 5 seeds. Both arms disable adaptive $\bar{\gamma}$, so the reference-discount EMA cannot serve as an anti-collapse anchor. Collapse is measured by the distribution of $\gamma_\phi(s)$ on a probe set of $2{,}048$ states frozen at the end of warmup.

Under the squared-TD objective, γ_ϕ degenerates to a near-constant discount (mean $\approx$0.911, $\mathrm{Std}_s[\gamma_\phi(s)]\approx$0.0014), whereas return-consistency retains a non-degenerate state-dependent distribution (mean $\approx$0.971, $\mathrm{Std}_s[\gamma_\phi(s)]\approx$0.022—roughly $15\times$ larger cross-state dispersion). Evaluation return drops modestly under TD ($\approx$26.4 vs $\approx$27.2), consistent with a discount-module degeneracy rather than total unlearnability on this task. Figure 2 and Table 2 in Appendix F.1 summarize the distributions. The same qualitative failure appears for the cross-validated TD baseline in Table 5 (mean γ=0.9000 on all three reported environments).

4.6 Ablation Studies

Leave-one-out on the γ objective. To isolate which term in (13) prevents the TD-target failure of Section 4.5, we expand that collapse ablation into a leave-one-out on the three auxiliary penalties, keeping the return-consistency objective fixed. Concretely, we train RC with the full objective; RC with each of λ_{dev}, λ_{var}, and λ_{bound} set to zero in turn; and RC with all three disabled. The two squared-TD arms of Section 4.5 remain as collapse controls, so the comparison holds the backbone, γ-range, warmup, and seeds fixed and varies only the γ objective and which auxiliaries are present. Default coefficients are λ_{dev}=0.005, λ_{var}=0.012, λ_{bound}=0.05; an ablated arm sets the corresponding λ to 0. All variants use seeds 42–46, disable adaptive $\bar{\gamma}$, and follow the frozen-probe protocol of Section 4.5 (`SafetyPointGoal1-v0`, SAC, 300k steps). Shared hyperparameters are listed in Table 15. Every return-consistency variant remains non-degenerate: dropping λ_{dev}, λ_{var}, or λ_{bound} individually, or disabling all three, yields $\mathrm{Std}_s[\gamma_\phi(s)] \in [0.0215, 0.0238]$, matching the full objective (0.0216 ± 0.0013). RC-only is slightly *more* dispersed than the full loss, as expected

because λ_{var} penalizes cross-state variance rather than creating it. By contrast, replacing L_γ^{RC} with δ^2 under the *same* regularizers still collapses to a near-constant discount ($\text{Std}_s[\gamma_\phi(s)] \approx 0.0014$); without regularizers it pins to a point mass at $\gamma_{\min}$. The auxiliaries therefore modulate the distribution—λ_{var} slightly compresses it, λ_{bound} reduces mass near the floor—but they are not what prevents collapse. The failure mode is the δ^2 objective, and return-consistency is the load-bearing term in (13). Table 3 and Figure 3 in Appendix F.2 summarize the comparison; Figure 4 reports mean-γ and floor-mass diagnostics.

Training objective comparison. Section 4.2 shows that AdaGamma consistently outperforms the SAC and PPO baselines, and Section 4.4 analyzes the learned $\gamma_\phi(s)$. Here we focus on the SAC variants. Table 5 in Appendix F.5 reports the mean learned $\gamma_\phi(s)$. The fixed baseline stays at $\gamma = 0.99$, whereas the cross-validated baseline collapses to the minimum candidate in all environments, matching its weak performance. The uncertainty-based baseline chooses less extreme but still global discounts. In contrast, AdaGamma learns task-dependent average discounts, indicating that its gains come from adapting temporal weighting rather than uniformly increasing or decreasing γ.

Comparison to fixed-γ. Table 4 compares the mean learned $\gamma_\phi(s)$ under SAC and PPO. To test whether the gains simply reflect a better fixed discount, we set fixed γ to the corresponding learned mean and re-evaluate SAC and PPO on `SafetyPointGoal1-v0`, `Humanoid-v4`, and `Ant-v4`; results are given in Table 7 of Appendix F.8. In all cases, this matched fixed discount still underperforms AdaGamma. We also expand the fixed-γ grid on these tasks in Table 8 of Appendix F.8, where AdaGamma remains superior. Thus, the improvement stems from state-dependent discounting rather than a better global γ.

Network architecture. We vary the hidden dimension of the gamma network over $\{8, 16, 32, 64, 128, 256, 512\}$ on `Ant-v4`. Performance improves with capacity and saturates around 256 hidden units (Figure 6 in Appendix F.6). Beyond this point, larger networks add computation with little gain, indicating that AdaGamma has low practical overhead.

Return horizon n. We vary the multi-step return length $n \in \{5, 10, 20\}$ in the return-consistency objective on `SafetyPointGoal1-v0` for both SAC and PPO. As shown in Table 6 of Appendix F.7, $n = 5$ gives the best overall reward–cost tradeoff, while longer horizons reduce stability, particularly for PPO. This suggests that the return-consistency target should balance temporal coverage against estimation noise; in this task, $n = 5$ provides the best tradeoff.

Additional experiments. Results on classic control tasks are reported in Appendix I. We also extend AdaGamma to DDPG (Lillicrap et al., 2019) and TRPO (Schulman et al., 2017a), with results on `SafetyPointGoal1-v0`, `Humanoid-v4`, and `Ant-v4` reported in Appendix H.

5 Conclusion

AdaGamma provides a simple and unified way to learn state-dependent discount factors in deep reinforcement learning. By interfacing only through the bootstrapped value target, it integrates naturally with both SAC and PPO, while the return-consistency objective prevents the degenerate behavior that arises from naive discount learning. On the theory side, we establish operator-level convergence properties for soft policy iteration with a fixed state-dependent discount function under tabular assumptions, and characterize the value discrepancy incurred. Empirically, AdaGamma improves reward–cost tradeoffs in safety-constrained settings, scales to high-dimensional continuous control, remains competitive on classic control benchmarks, and shows stable discount patterns across algorithms. Its positive results in a four-week JD Logistics A/B test further suggest that state-adaptive temporal weighting is not only algorithmically useful, but also practical in real-world sequential decision making.

AI Use Disclosure

In this work, we used generative AI tools for assisting with the verification of mathematical proofs. We have not used generative AI tools for any other tasks with required disclosure, and the remaining required disclosure tasks are not applicable to this work. Additionally, we used generative AI tools for editing the paper to improve readability and for editing software code. We have reviewed all

AI-assisted work. In particular, the AI-assisted code modifications were verified and tested for correctness by 2 authors. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REFERENCES

Ron Amit, Ron Meir, and Kamil Ciosek. Discount factor as a regularizer in reinforcement learning. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, 2020.

Pierre-Luc Bacon, Jean Harb, and Doina Precup. The option-critic architecture. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, pp. 1726–1734. AAAI Press, 2017.

Dylan J. Foster, Akshay Krishnamurthy, David Simchi-Levi, and Yunzong Xu. Offline reinforcement learning: Fundamental barriers for value function approximation, 2022. URL https://arxiv.org/abs/2111.10919.

Vincent François-Lavet, Raphael Fonteneau, and Damien Ernst. How to discount deep reinforcement learning: Towards new dynamic strategies, 2016. URL https://arxiv.org/abs/1512.02011.

Scott Fujimoto, Herke van Hoof, and David Meger. Addressing function approximation error in actor-critic methods, 2018. URL https://arxiv.org/abs/1802.09477.

Yang Gu, Yuhu Cheng, C. L. Philip Chen, and Xuesong Wang. Proximal policy optimization with policy feedback. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 52(7):4600–4610, 2022. doi: 10.1109/TSMC.2021.3098451.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor, 2018. URL https://arxiv.org/abs/1801.01290.

Anna Harutyunyan, Will Dabney, Diana Borsa, Nicolas Heess, Remi Munos, and Doina Precup. The termination critic. In Kamalika Chaudhuri and Masashi Sugiyama (eds.), *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, volume 89 of *Proceedings of Machine Learning Research*, pp. 2231–2240. PMLR, 16–18 Apr 2019. URL https://proceedings.mlr.press/v89/harutyunyan19a.html.

Linjian Hou, Zhengming Wang, and Han Long. An improvement for value-based reinforcement learning method through increasing discount factor substitution. In *2021 IEEE 24th International Conference on Computational Science and Engineering (CSE)*, pp. 94–100, 2021. doi: 10.1109/CSE53436.2021.00023.

Hao Hu, Yiqin Yang, Qianchuan Zhao, and Chongjie Zhang. On the role of discount factor in offline reinforcement learning, 2022. URL https://arxiv.org/abs/2206.03383.

Jiaming Ji, Borong Zhang, Jiayi Zhou, Xuehai Pan, Weidong Huang, Ruiyang Sun, Yiran Geng, Yifan Zhong, Josef Dai, and Yaodong Yang. Safety gymnasium: A unified safe reinforcement learning benchmark. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023. URL https://openreview.net/forum?id=WZmlxIuIGR.

Zeyu Jia, Alexander Rakhlin, Ayush Sekhari, and Chen-Yu Wei. Offline reinforcement learning: Role of state aggregation and trajectory data. *CoRR*, abs/2403.17091, 2024. doi: 10.48550/ARXIV.2403.17091. URL https://doi.org/10.48550/arXiv.2403.17091.

MyeongSeop Kim, Jung-Su Kim, Myoung-Su Choi, and Jae-Han Park. Adaptive discount factor for deep reinforcement learning in continuing tasks with uncertainty. *Sensors*, 22(19), 2022. ISSN 1424-8220. doi: 10.3390/s22197266. URL https://www.mdpi.com/1424-8220/22/19/7266.

Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning, 2019. URL https://arxiv.org/abs/1509.02971.

Alberto Maria Metelli, Mirco Mutti, and Marcello Restelli. A tale of sampling and estimation in discounted reinforcement learning, 2023. URL https://arxiv.org/abs/2304.05073.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei Rusu, Joel Veness, Marc Bellemare, Alex Graves, Martin Riedmiller, Andreas Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518:529–33, 02 2015. doi: 10.1038/nature14236.

Abhishek Naik, Roshan Shariff, Niko Yasui, Hengshuai Yao, and Richard S. Sutton. Discounted reinforcement learning is not an optimization problem, 2019. URL https://arxiv.org/abs/1910.02140.

Silviu Pitis. Rethinking the discount factor in reinforcement learning: a decision theoretic approach. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'19/IAAI'19/EAAI'19. AAAI Press, 2019a. ISBN 978-1-57735-809-1. doi: 10.1609/aaai.v33i01.33017949. URL https://doi.org/10.1609/aaai.v33i01.33017949.

Silviu Pitis. Rethinking the discount factor in reinforcement learning: A decision theoretic approach, 2019b. URL https://arxiv.org/abs/1902.02893.

Doina Precup, Richard S. Sutton, and Satinder P. Singh. Eligibility traces for off-policy policy evaluation. In *Proceedings of the Seventeenth International Conference on Machine Learning*, ICML '00, pp. 759–766, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc. ISBN 1558607072.

Martin L. Puterman. Markov decision processes: Discrete stochastic dynamic programming. In *Wiley Series in Probability and Statistics*, 1994. URL https://api.semanticscholar.org/CorpusID:260799408.

Sarah Rathnam, Sonali Parbhoo, Siddharth Swaroop, Weiwei Pan, Susan A. Murphy, and Finale Doshi-Velez. Rethinking discount regularization: New interpretations, unintended consequences, and solutions for regularization in reinforcement learning. *Journal of Machine Learning Research*, 25(255):1–48, 2024. URL http://jmlr.org/papers/v25/24-0087.html.

John Schulman, Sergey Levine, Philipp Moritz, Michael I. Jordan, and Pieter Abbeel. Trust region policy optimization, 2017a. URL https://arxiv.org/abs/1502.05477.

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

Matthias Schultheis, Constantin A. Rothkopf, and Heinz Koeppl. Reinforcement learning with non-exponential discounting, 2022. URL https://arxiv.org/abs/2209.13413.

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

R.S. Sutton and A.G. Barto. Reinforcement learning: An introduction. *IEEE Transactions on Neural Networks*, 9(5):1054–1054, 1998. doi: 10.1109/TNN.1998.712192.

Mark Towers, Ariel Kwiatkowski, Jordan Terry, John U Balis, Gianluca De Cola, Tristan Deleu, Manuel Goulão, Andreas Kallinteris, Markus Krimmel, Arjun KG, et al. Gymnasium: A standard interface for reinforcement learning environments. *arXiv preprint arXiv:2407.17032*, 2024.

Qingda Wei and Xianping Guo. Markov decision processes with state-dependent discount factors and unbounded rewards/costs. *Operations Research Letters*, 39(5):369–374, 2011. ISSN 0167-6377. doi: https://doi.org/10.1016/j.orl.2011.06.014. URL https://www.sciencedirect.com/science/article/pii/S016763771100068X.

Naoto Yoshida, Eiji Uchibe, and Kenji Doya. Reinforcement learning with state-dependent discount factor. In *2013 IEEE Third Joint International Conference on Development and Learning and Epigenetic Robotics (ICDL)*, pp. 1–6, 2013. doi: 10.1109/DevLrn.2013.6652533.

Wenhao Zhan, Baihe Huang, Audrey Huang, Nan Jiang, and Jason D. Lee. Offline reinforcement learning with realizability and single-policy concentrability, 2022. URL https://arxiv.org/abs/2202.04634.

A Related Work

Fixed discounting and discount as regularization. Most deep RL methods use a fixed discount factor (Sutton & Barto, 1998; Mnih et al., 2015; Haarnoja et al., 2018; Schulman et al., 2017b), despite the fact that γ strongly influences the effective planning horizon, variance, and optimization behavior (François-Lavet et al., 2016; Amit et al., 2020; Naik et al., 2019; Hu et al., 2022). A growing line of work interprets discounting not only as a preference parameter, but also as a form of regularization (Amit et al., 2020; Rathnam et al., 2024). In particular, Amit et al. (2020) show that, for several TD-learning methods, reducing γ can be viewed as introducing an explicit regularization effect, while Rathnam et al. (2024) show that globally shrinking γ may induce an undesirable non-uniform prior over transition dynamics. Related analyses in offline RL further show that γ jointly controls regularization and pessimism, affecting the trade-off between policy quality and data efficiency (Hu et al., 2022; Foster et al., 2022; Zhan et al., 2022). These results provide part of the motivation for adaptive discounting, although they do not directly address how to learn a state-dependent discount function in deep actor–critic training.

Flexible discounting: schedules and non-exponential variants. Another line of work studies discounting schemes that vary over training, over time, or beyond the standard exponential form. François-Lavet et al. (2016) discuss dynamic discount strategies, and Hou et al. (2021) propose increasing discount factors over time steps to alleviate the underweighting of distant rewards. More generally, Schultheis et al. (2022) study RL with non-exponential discount functions, and related work broadens the space of generalized return criteria (Pitis, 2019a; Naik et al., 2019). These approaches enlarge the class of temporal preference models, but the discount rule is typically global, time-indexed, or fixed by design. Our setting is different: we focus on learned *state-conditioned* discounting within a conventional bootstrapped actor–critic pipeline, where the main challenge is not defining a richer return criterion, but ensuring that the learned discount does not destabilize TD-based training.

State-dependent discounting and generalized RL objectives. A more expressive formulation allows the discount factor to depend on the state or state-action pair. Wei & Guo (2011) study MDPs with state-dependent discount factors and establish general existence and characterization results. Yoshida et al. (2013) introduce state-dependent discounting in model-free RL and derive an ExQ-learning algorithm with convergence guarantees. From a broader decision-theoretic perspective, Pitis (2019b) argue that fixed discounting is only one special case in a larger class of sequential preference models, while Naik et al. (2019) emphasize that discounting is a substantive modeling choice rather than a mere numerical convenience. These works establish that variable discounting is mathematically meaningful and behaviorally expressive, but they do not directly provide a practical implementation recipe for modern deep actor–critic methods with function approximation.

Adaptive discounting in deep RL. Several recent works study adaptive discounting more directly in deep RL. Kim et al. (2022) propose an uncertainty- or advantage-driven adjustment rule for PPO and SAC, while Gu et al. (2022) introduce a PPO variant with policy-feedback-dependent discounting. Related approaches also adapt horizon-related quantities through dynamic schedules, uncertainty signals, or return-design heuristics (François-Lavet et al., 2016; Kim et al., 2022; Gu et al., 2022). Compared with these methods, our approach learns an explicit state-conditioned neural module $\gamma_\phi(s)$ and emphasizes a different algorithmic issue: the collapse behavior that can arise when such a module is trained naively through bootstrapped TD objectives. Our return-consistency objective is designed specifically to regularize this failure mode.

Long horizons, statistical fragility, and temporal abstraction. Large discount factors require reliable long-horizon estimation, which becomes difficult when data are noisy, sparse, or poorly mixed (François-Lavet et al., 2016; Naik et al., 2019; Metelli et al., 2023). More generally, sample-complexity and offline-RL analyses show that discounted problems become statistically harder as the effective horizon grows or coverage deteriorates (Foster et al., 2022; Hu et al., 2022; Jia et al., 2024; Zhan et al., 2022). In particular, Metelli et al. (2023) study the γ-discounted mean estimation problem and derive lower bounds depending on both the discount factor and the mixing properties of the underlying Markov chain. This helps explain why a globally large γ can be statistically fragile, and why adapting bootstrap strength across states may be beneficial.

State-dependent discounting is also related to temporal abstraction and termination. In the options framework, termination determines the duration of temporally extended behavior and thus implicitly controls an effective planning horizon (Precup et al., 2000; Bacon et al., 2017). More broadly,

temporally extended actions modulate the scale of credit assignment and planning, and recent work shows benefits from decoupling behavior and target termination conditions (Harutyunyan et al., 2019). Although our method does not introduce explicit options or termination policies, the product-of-gammas structure in our modified GAE recursion plays a related conceptual role: low-γ regions shorten the effective credit-assignment horizon, while high-γ regions preserve longer-range dependencies. In this sense, our approach is complementary to temporal-abstraction methods rather than a replacement for them.

B PROOFS

B.1 PROOF OF PROPOSITION 1

Proposition 1. *(Product-of-Gammas Weighting) Under state-dependent discounting, for a nonterminal trajectory segment, the GAE advantage estimate expands as*

$$\hat{A}_t = \delta_t + \sum_{l=1}^{T-1-t} \left(\prod_{k=0}^{l-1} \gamma_\phi(s_{t+k}) \right) \lambda^l \delta_{t+l}. \tag{21}$$

With terminal masks, each factor $\gamma_\phi(s_{t+k})$ is replaced by $\gamma_\phi(s_{t+k})(1 - d_{t+k})$.

Proof. We prove the result by induction on $T - 1 - t$. The displayed expansion is for a nonterminal trajectory segment; terminal masks simply truncate the recursion because any factor $1 - d_{t+k} = 0$ eliminates all subsequent terms.

Base case. For $t = T - 1$, we have $\hat{A}_{T-1} = \delta_{T-1}$. The expansion also gives $\hat{A}_{T-1} = \delta_{T-1}$, since the summation is empty. Thus the claim holds.

Inductive step. Assume that the expansion holds for $\hat{A}_{t+1}$, i.e.,

$$\hat{A}_{t+1} = \delta_{t+1} + \sum_{l=1}^{T-2-t} \left(\prod_{k=0}^{l-1} \gamma_\phi(s_{t+1+k}) \right) \lambda^l \delta_{t+1+l}. \tag{22}$$

Then, using the recursion $\hat{A}_t = \delta_t + \gamma_\phi(s_t)\lambda \hat{A}_{t+1}$, we obtain

$$\begin{aligned}
\hat{A}_t &= \delta_t + \gamma_\phi(s_t)\lambda\, \hat{A}_{t+1} && (23)\\
&= \delta_t + \gamma_\phi(s_t)\lambda \left[\delta_{t+1} + \sum_{l=1}^{T-2-t} \left(\prod_{k=0}^{l-1} \gamma_\phi(s_{t+1+k}) \right) \lambda^l \delta_{t+1+l} \right] && (24)\\
&= \delta_t + \gamma_\phi(s_t)\lambda\, \delta_{t+1} + \sum_{l=1}^{T-2-t} \gamma_\phi(s_t) \left(\prod_{k=0}^{l-1} \gamma_\phi(s_{t+1+k}) \right) \lambda^{l+1} \delta_{t+1+l} && (25)\\
&= \delta_t + \left(\prod_{k=0}^{0} \gamma_\phi(s_{t+k}) \right) \lambda \delta_{t+1} + \sum_{l=1}^{T-2-t} \left(\prod_{k=0}^{l} \gamma_\phi(s_{t+k}) \right) \lambda^{l+1} \delta_{t+1+l}. && (26)
\end{aligned}$$

Now let $m = l + 1$ in the last summation. Then m ranges from 2 to $T - 1 - t$, and we have

$$\begin{aligned}
\hat{A}_t &= \delta_t + \left(\prod_{k=0}^{0} \gamma_\phi(s_{t+k}) \right) \lambda \delta_{t+1} + \sum_{m=2}^{T-1-t} \left(\prod_{k=0}^{m-1} \gamma_\phi(s_{t+k}) \right) \lambda^m \delta_{t+m} && (27)\\
&= \delta_t + \sum_{m=1}^{T-1-t} \left(\prod_{k=0}^{m-1} \gamma_\phi(s_{t+k}) \right) \lambda^m \delta_{t+m}. && (28)
\end{aligned}$$

This completes the induction. □

B.2 PROOF OF LEMMA 1

Lemma 1. *(Fixed-policy contraction under state-dependent discounting) Under Assumption 1, for any stationary policy π, the operator $\mathcal{T}_\gamma^\pi$ is a β-contraction in the sup norm:*

$$\|\mathcal{T}_\gamma^\pi Q - \mathcal{T}_\gamma^\pi Q'\|_\infty \le \beta \|Q - Q'\|_\infty. \tag{17}$$

Consequently, $\mathcal{T}_\gamma^\pi$ admits a unique fixed point Q_γ^π, and the exact iteration $Q^{k+1} = \mathcal{T}_\gamma^\pi Q^k$ converges to Q_γ^π from any initial Q^0.

Proof. For any Q, Q' and (s, a), the reward and entropy terms cancel, so

$$\begin{aligned}\left|(\mathcal{T}_\gamma^\pi Q)(s,a) - (\mathcal{T}_\gamma^\pi Q')(s,a)\right| &\le \gamma(s)\,\mathbb{E}_{s'\sim P(\cdot|s,a)}\left[\left|V_Q^\pi(s') - V_{Q'}^\pi(s')\right|\right] \\ &\le \gamma(s)\,\mathbb{E}_{s'\sim P(\cdot|s,a)}\left[\sum_{a'}\pi(a'|s')|Q(s',a') - Q'(s',a')|\right] \\ &\le \beta\|Q - Q'\|_\infty.\end{aligned}$$

Taking the supremum over (s, a) proves (17). The existence and uniqueness of Q_γ^π, together with convergence from any initialization, then follow from the Banach fixed-point theorem. □

B.3 Proof of Proposition 2

Proposition 2. *(Multi-step contraction and trajectory weighting) Under Assumption 1, for every integer $n \ge 1$, the n-fold Bellman operator $(\mathcal{T}_\gamma^\pi)^n$ is a β^n-contraction:*

$$\|(\mathcal{T}_\gamma^\pi)^n Q - (\mathcal{T}_\gamma^\pi)^n Q'\|_\infty \le \beta^n \|Q - Q'\|_\infty. \tag{29}$$

Moreover, $\mathcal{T}_\gamma^\pi$ and $(\mathcal{T}_\gamma^\pi)^n$ have the same unique fixed point Q_γ^π. Along a trajectory generated by P and π,

$$\left((\mathcal{T}_\gamma^\pi)^n Q - (\mathcal{T}_\gamma^\pi)^n Q'\right)(s_0,a_0) = \mathbb{E}_\pi\left[\left(\prod_{j=0}^{n-1}\gamma(s_j)\right)(Q - Q')(s_n,a_n)\,\middle|\, s_0,a_0\right]. \tag{30}$$

Proof. Applying Lemma 1 recursively gives

$$\|(\mathcal{T}_\gamma^\pi)^n Q - (\mathcal{T}_\gamma^\pi)^n Q'\|_\infty \le \beta\|(\mathcal{T}_\gamma^\pi)^{n-1} Q - (\mathcal{T}_\gamma^\pi)^{n-1} Q'\|_\infty \le \cdots \le \beta^n\|Q - Q'\|_\infty,$$

which proves (29).

Since Q_γ^π is the fixed point of $\mathcal{T}_\gamma^\pi$, we obtain $(\mathcal{T}_\gamma^\pi)^n Q_\gamma^\pi = Q_\gamma^\pi$. Because $(\mathcal{T}_\gamma^\pi)^n$ is itself a contraction, its fixed point is unique; hence the one-step and n-step operators share the same fixed point Q_γ^π.

Finally, repeatedly expanding the difference between the two Bellman backups yields (30), which makes explicit the product-of-discounts structure of multi-step propagation. □

B.4 Proof of Lemma 2

Lemma 2. *(Exact soft policy improvement) Let π be any stationary tabular policy, and let Q_γ^π denote its soft Q-function under the fixed discount function $\gamma(\cdot)$. Define $\pi_{\text{new}} = \pi_{Q_\gamma^\pi}$. Then*

$$Q_\gamma^{\pi_{\text{new}}}(s,a) \ge Q_\gamma^\pi(s,a), \qquad \forall (s,a) \in \mathcal{S}\times\mathcal{A}. \tag{31}$$

Proof. For any vector $q \in \mathbb{R}^{|\mathcal{A}|}$, the Gibbs variational identity gives

$$\alpha \log \sum_{a\in\mathcal{A}} \exp\Big(\frac{q_a}{\alpha}\Big) = \max_{\mu\in\Delta(\mathcal{A})} \sum_{a\in\mathcal{A}} \mu(a)\big(q_a - \alpha\log\mu(a)\big), \tag{32}$$

whose unique maximizer is

$$\mu^\star(a) = \frac{\exp(q_a/\alpha)}{\sum_{b\in\mathcal{A}}\exp(q_b/\alpha)}.$$

Applying (32) statewise with $q_a = Q_\gamma^\pi(s,a)$ and using $\pi_{\text{new}} = \pi_{Q_\gamma^\pi}$ yields

$$V_{Q_\gamma^\pi}^{\pi_{\text{new}}}(s) \ge V_{Q_\gamma^\pi}^\pi(s), \qquad \forall s \in \mathcal{S}. \tag{33}$$

Because $\gamma(s) \ge 0$, equation 33 implies

$$\mathcal{T}_\gamma^{\pi_{\text{new}}} Q_\gamma^\pi \ge \mathcal{T}_\gamma^\pi Q_\gamma^\pi = Q_\gamma^\pi, \tag{34}$$

where the equality follows from the fact that Q_γ^π is the unique fixed point of $\mathcal{T}_\gamma^\pi$.

Moreover, $\mathcal{T}_\gamma^{\pi_{\text{new}}}$ is monotone: for any $Q_1 \le Q_2$ componentwise,

$$\mathcal{T}_\gamma^{\pi_{\text{new}}} Q_1 \le \mathcal{T}_\gamma^{\pi_{\text{new}}} Q_2.$$

Combining this monotonicity with (34) gives

$$Q_\gamma^\pi \le \mathcal{T}_\gamma^{\pi_{\text{new}}} Q_\gamma^\pi \le \left(\mathcal{T}_\gamma^{\pi_{\text{new}}}\right)^2 Q_\gamma^\pi \le \cdots .$$

By Lemma 1, $\mathcal{T}_\gamma^{\pi_{\text{new}}}$ is a β-contraction. Hence its iterates converge to its unique fixed point $Q_\gamma^{\pi_{\text{new}}}$. Taking the limit in the above monotone sequence yields

$$Q_\gamma^\pi \le \lim_{\ell\to\infty} \left(\mathcal{T}_\gamma^{\pi_{\text{new}}}\right)^\ell Q_\gamma^\pi = Q_\gamma^{\pi_{\text{new}}},$$

which proves (31). □

B.5 Proof of Theorem 1

Theorem 1. *(Exact soft policy iteration) Under Assumption 1, let π_0 be any stationary tabular policy and define, for $k \ge 0$, $Q_k := Q_\gamma^{\pi_k}$ and $\pi_{k+1} = \pi_{Q_k}$, where $Q_\gamma^{\pi_k}$ denotes the exact soft Q-function of π_k under the fixed state-dependent discount $\gamma(\cdot)$. Then $\{Q_k\}_k$ converges in the sup norm to the unique fixed point $Q_\gamma^\star$ of the soft optimality operator $\mathcal{T}_\gamma$. Moreover, $\pi_k \to \pi^\star := \pi_{Q_\gamma^\star}$, where $\pi^\star$ is the unique statewise soft-greedy policy associated with $Q_\gamma^\star$. In addition,*

$$Q_\gamma^\star(s,a) \ge Q_\gamma^\pi(s,a), \qquad \forall\, \pi, \quad \forall\, (s,a) \in \mathcal{S}\times\mathcal{A}, \tag{20}$$

where π ranges over all stationary tabular policies.

Proof. The log-sum-exp mapping is 1-Lipschitz with respect to the sup norm. Together with $\gamma(s) \le \beta < 1$, this implies that $\mathcal{T}_\gamma$ is a β-contraction. Hence, by the Banach fixed-point theorem, $\mathcal{T}_\gamma$ admits a unique fixed point, denoted by $Q_\gamma^\star$.

We first establish monotonicity of the policy-iteration sequence. By the Gibbs variational identity in (32) and the definition $\pi_{k+1} = \pi_{Q_k}$,

$$\mathcal{T}_\gamma Q_k = \mathcal{T}_\gamma^{\pi_{k+1}} Q_k \ge \mathcal{T}_\gamma^{\pi_k} Q_k = Q_k, \tag{35}$$

where the last equality follows from $Q_k = Q_\gamma^{\pi_k}$.

Since $\mathcal{T}_\gamma^{\pi_{k+1}}$ is monotone, repeated application of (35) gives

$$Q_k \le \mathcal{T}_\gamma^{\pi_{k+1}} Q_k \le \left(\mathcal{T}_\gamma^{\pi_{k+1}}\right)^2 Q_k \le \cdots .$$

By Lemma 1, $\mathcal{T}_\gamma^{\pi_{k+1}}$ is a β-contraction, and its iterates converge to its unique fixed point $Q_\gamma^{\pi_{k+1}} = Q_{k+1}$. Therefore,

$$Q_k \le \mathcal{T}_\gamma Q_k \le Q_{k+1}. \tag{36}$$

Thus, $\{Q_k\}$ is componentwise nondecreasing.

Next, we establish a uniform upper bound. For any stationary policy π, the soft optimality operator dominates the corresponding policy-evaluation operator:

$$\mathcal{T}_\gamma Q \ge \mathcal{T}_\gamma^\pi Q, \qquad \forall\, Q. \tag{37}$$

In particular, setting $Q = Q_\gamma^\pi$ yields

$$\mathcal{T}_\gamma Q_\gamma^\pi \ge \mathcal{T}_\gamma^\pi Q_\gamma^\pi = Q_\gamma^\pi.$$

Since $\mathcal{T}_\gamma$ is both monotone and contractive, iterating it from Q_γ^π gives a nondecreasing sequence converging to $Q_\gamma^\star$. Hence

$$Q_\gamma^\pi \le Q_\gamma^\star$$

for every stationary tabular policy π. In particular,

$$Q_k \le Q_\gamma^\star, \qquad \forall k.$$

Because $\mathcal{S} \times \mathcal{A}$ is finite, the componentwise nondecreasing and bounded sequence $\{Q_k\}$ converges in the sup norm to some $\overline{Q}$. Taking limits in (36) and using the continuity of $\mathcal{T}_\gamma$ yields

$$\overline{Q} \le \mathcal{T}_\gamma \overline{Q} \le \overline{Q}.$$

Consequently,

$$\mathcal{T}_\gamma \overline{Q} = \overline{Q}.$$

By uniqueness of the fixed point of $\mathcal{T}_\gamma$,

$$\overline{Q} = Q^\star_\gamma.$$

Therefore,

$$Q_k \to Q^\star_\gamma \qquad \text{in the sup norm.}$$

Finally, since $\alpha > 0$, the statewise softmax mapping $Q \mapsto \pi_Q$ is continuous. Hence

$$\pi_{k+1} = \pi_{Q_k} \to \pi_{Q^\star_\gamma} = \pi^\star.$$

The dominance argument above also establishes (20). □

B.6 Proof of Theorem 2

Theorem 2. *(Lipschitz sensitivity to the discount function) Under Assumption 1, fix a stationary policy π. For $i \in \{1, 2\}$, let $\gamma_i : \mathcal{S} \to [0, \beta_i]$, where $\beta_i < 1$. Then*

$$\|Q^\pi_{\gamma_1} - Q^\pi_{\gamma_2}\|_\infty \le \frac{C\,\|\gamma_1 - \gamma_2\|_\infty}{(1-\beta_1)(1-\beta_2)}, \tag{38}$$

where $C := R + \alpha \log m$. In particular, if $\gamma_2(s) \equiv \bar{\gamma}$ for some $\bar{\gamma} \in [0, 1)$,

$$\|Q^\pi_{\gamma_1} - Q^\pi_{\bar{\gamma}}\|_\infty \le \frac{(R + \alpha \log m)\max_{s\in\mathcal{S}} |\gamma_1(s) - \bar{\gamma}|}{(1-\beta_1)(1-\bar{\gamma})}. \tag{39}$$

Proof. Let

$$H_\pi(s) := -\sum_{a\in\mathcal{A}} \pi(a \mid s) \log \pi(a \mid s)$$

denote the entropy of $\pi(\cdot \mid s)$. Since $\mathcal{A}$ is finite,

$$0 \le H_\pi(s) \le \log m, \qquad \forall s \in \mathcal{S}. \tag{40}$$

We first bound the soft value function associated with $Q^\pi_{\gamma_2}$. Since $Q^\pi_{\gamma_2}$ is the fixed point of $T^\pi_{\gamma_2}$, the Bellman equation and $\gamma_2(s) \le \beta_2$ imply

$$\|Q^\pi_{\gamma_2}\|_\infty \le R + \beta_2 \|V^\pi_{Q^\pi_{\gamma_2}}\|_\infty \le R + \beta_2 \left(\|Q^\pi_{\gamma_2}\|_\infty + \alpha \log m\right). \tag{41}$$

Hence

$$\|Q^\pi_{\gamma_2}\|_\infty \le \frac{R + \alpha\beta_2 \log m}{1-\beta_2}. \tag{42}$$

Consequently,

$$\|V^\pi_{Q^\pi_{\gamma_2}}\|_\infty \le \|Q^\pi_{\gamma_2}\|_\infty + \alpha \log m \le \frac{R + \alpha \log m}{1-\beta_2} = \frac{C}{1-\beta_2}. \tag{43}$$

We now compare the two fixed-point equations. For every (s, a), adding and subtracting the term $\gamma_1(s)\mathbb{E}[V^\pi_{Q^\pi_{\gamma_2}}(s')]$ yields

$$Q^\pi_{\gamma_1}(s,a) - Q^\pi_{\gamma_2}(s,a) = \gamma_1(s)\mathbb{E}_{s'\sim P(\cdot|s,a)}\left[V^\pi_{Q^\pi_{\gamma_1}}(s') - V^\pi_{Q^\pi_{\gamma_2}}(s')\right] + \big(\gamma_1(s) - \gamma_2(s)\big)\mathbb{E}_{s'\sim P(\cdot|s,a)}\left[V^\pi_{Q^\pi_{\gamma_2}}(s')\right]. \tag{44}$$

Because the policy π is fixed, the entropy terms cancel in $V^\pi_{Q^\pi_{\gamma_1}} - V^\pi_{Q^\pi_{\gamma_2}}$. Therefore,

$$\|V^\pi_{Q^\pi_{\gamma_1}} - V^\pi_{Q^\pi_{\gamma_2}}\|_\infty \le \|Q^\pi_{\gamma_1} - Q^\pi_{\gamma_2}\|_\infty. \tag{45}$$

Taking sup norms in (44) and using (43), equation 45, we obtain

$$\|Q^\pi_{\gamma_1} - Q^\pi_{\gamma_2}\|_\infty \le \beta_1 \|Q^\pi_{\gamma_1} - Q^\pi_{\gamma_2}\|_\infty + \|\gamma_1 - \gamma_2\|_\infty \frac{C}{1-\beta_2}. \tag{46}$$

Rearranging gives

$$\|Q^\pi_{\gamma_1} - Q^\pi_{\gamma_2}\|_\infty \le \frac{C\,\|\gamma_1 - \gamma_2\|_\infty}{(1-\beta_1)(1-\beta_2)},$$

which proves (38).

Finally, when $\gamma_2(s) \equiv \bar{\gamma}$, we may take $\beta_2 = \bar{\gamma}$, yielding (39). □

B.7 Proof of Corollary 1

Corollary 1. *(Conditional tracking under a varying discount) Under Assumption 1, fix a stationary policy π and let $\gamma_k : \mathcal{S} \to [0, \beta]$ with $\beta < 1$. Define $Q^\star_k := Q^\pi_{\gamma_k}$, $Q_{k+1} = (\mathcal{T}^\pi_{\gamma_k})^n Q_k$ and $e_k := \|Q_k - Q^\star_k\|_\infty$. Then*

$$e_{k+1} \le \beta^n e_k + \frac{R + \alpha \log m}{(1-\beta)^2} \|\gamma_{k+1} - \gamma_k\|_\infty. \tag{47}$$

Hence if $\|\gamma_{k+1} - \gamma_k\|_\infty \to 0$, then $e_k \to 0$. If $\sup_k \|\gamma_{k+1} - \gamma_k\|_\infty \le \eta$, then

$$\limsup_{k\to\infty} e_k \le \frac{(R + \alpha \log m)\eta}{(1-\beta)^2(1-\beta^n)}. \tag{48}$$

Proof. Since $Q^\star_k = Q^\pi_{\gamma_k}$ is the unique fixed point of $\mathcal{T}^\pi_{\gamma_k}$, it is also a fixed point of its n-fold iterate $(\mathcal{T}^\pi_{\gamma_k})^n Q^\star_k = Q^\star_k$. Therefore,

$$\begin{aligned} e_{k+1} &= \|Q_{k+1} - Q^\star_{k+1}\|_\infty \\ &= \|(\mathcal{T}^\pi_{\gamma_k})^n Q_k - Q^\star_{k+1}\|_\infty \\ &\le \|(\mathcal{T}^\pi_{\gamma_k})^n Q_k - Q^\star_k\|_\infty + \|Q^\star_k - Q^\star_{k+1}\|_\infty \\ &= \|(\mathcal{T}^\pi_{\gamma_k})^n Q_k - (\mathcal{T}^\pi_{\gamma_k})^n Q^\star_k\|_\infty + \|Q^\star_k - Q^\star_{k+1}\|_\infty. \end{aligned} \tag{49}$$

By Proposition 2, $(\mathcal{T}^\pi_{\gamma_k})^n$ is a β^n-contraction. Hence

$$e_{k+1} \le \beta^n e_k + \|Q^\star_{k+1} - Q^\star_k\|_\infty. \tag{50}$$

Next, by Theorem 2, we have

$$\|Q^\star_{k+1} - Q^\star_k\|_\infty \le \frac{R + \alpha \log m}{(1-\beta)^2} \|\gamma_{k+1} - \gamma_k\|_\infty.$$

Substituting this bound into (50) yields (47).

Let $\rho := \beta^n < 1$ and $\delta_k := \frac{R+\alpha \log m}{(1-\beta)^2} \|\gamma_{k+1} - \gamma_k\|_\infty$, then the tracking recursion can be written as

$$e_{k+1} \le \rho e_k + \delta_k.$$

Repeated substitution gives, for every $K \ge 1$,

$$e_K \le \rho^K e_0 + \sum_{j=0}^{K-1} \rho^{K-1-j} \delta_j. \tag{51}$$

If $\|\gamma_{k+1} - \gamma_k\|_\infty \to 0$, then $\delta_k \to 0$. Since $\rho < 1$, the convolution of the vanishing sequence $\{\delta_k\}$ with the geometrically decaying kernel in equation 51 also vanishes, and therefore $e_k \to 0$.

Finally, suppose that $\sup_k \|\gamma_{k+1} - \gamma_k\|_\infty \le \eta$, then

$$\delta_k \le D := \frac{(R + \alpha \log m)\eta}{(1-\beta)^2}.$$

Using (51),

$$e_K \le \rho^K e_0 + D \sum_{j=0}^{K-1} \rho^{K-1-j} = \rho^K e_0 + D \frac{1-\rho^K}{1-\rho}.$$

Taking the limit superior and recalling that $\rho = \beta^n$ gives

$$\limsup_{K\to\infty} e_K \le \frac{D}{1-\rho} = \frac{(R + \alpha \log m)\eta}{(1-\beta)^2(1-\beta^n)},$$

which proves (48). □

C Details of Uncertainty-rule adaptive-γ baseline

Inspired by Kim et al. (2022), who adapt the discount factor based on value-estimation uncertainty measured via ensemble disagreement, we implement an uncertainty-driven baseline within both the SAC and PPO frameworks. In SAC, we treat the absolute difference between the two Q-heads, $|Q_{\theta_1}(s,a) - Q_{\theta_2}(s,a)|$, as a lightweight uncertainty proxy. In PPO, which lacks a twin-critic architecture by default, we introduce an auxiliary value network V_{ϕ_2} alongside the primary value network V_{ϕ_1} and use their disagreement $|V_{\phi_1}(s) - V_{\phi_2}(s)|$ as the analogous uncertainty signal. Crucially, the auxiliary V_{ϕ_2} is used *solely* for computing the per-state discount factor; it does not participate in PPO's standard operations such as advantage estimation (GAE) or the policy gradient objective, which continue to rely exclusively on the primary V_{ϕ_1}. In both cases, a per-state discount is computed via

$$\gamma(s) = \gamma_{\min} + (\gamma_{\max} - \gamma_{\min}) \exp\big(-\eta\, \beta\, d(s)\big), \tag{52}$$

where $d(s) \ge 0$ denotes the critic disagreement, $\beta > 0$ is a learnable scaling parameter, and $\eta > 0$ is the uncertainty scale. The intuition follows Kim et al. (2022): when the two critics disagree strongly (high uncertainty), the discount is pushed toward $\gamma_{\min}$ to shorten the effective planning horizon and reduce reliance on unreliable bootstrapped targets; when they agree (low uncertainty), the discount approaches $\gamma_{\max}$ to enable longer-horizon credit assignment. Compared with the original formulation in Kim et al. (2022), which uses a larger ensemble and a different scaling rule, our variant is adapted to each algorithm's native architecture—reusing SAC's existing twin Q-networks and adding a minimal auxiliary value network for PPO—and grants the baseline additional flexibility through the learnable β. This provides a competitive reference point that already incorporates state-dependent temporal adaptation, against which we isolate the additional benefit of AdaGamma's return-consistency training objective.

D Reward Calculation in the JD Logistics Deployment

This section gives the detailed calculation of the reward components used in the JD Logistics deployment. Let order_t denote the per-user order volume in the current cycle, and let order_{t-1} denote the corresponding quantity in the previous cycle. Let $\mathrm{discount}_t$ be the realized discount under the selected coupon action, and let $\mathrm{discount}^{\mathrm{med}}$ denote the medium-tier reference discount.

The raw absolute order-growth reward is

$$r_t^{\mathrm{abs}} = a\,(\mathrm{order}_t - \mathrm{order}_{t-1})\,, \tag{53}$$

where a is a scaling coefficient. This term directly rewards an increase in per-user order volume relative to the previous cycle.

The raw relative order-growth reward is

$$r_t^{\mathrm{rel}} = b\,\frac{\mathrm{order}_t - \mathrm{order}_{t-1}}{\mathrm{order}_{t-1} + \epsilon}, \tag{54}$$

where b is a scaling coefficient and $\epsilon > 0$ is a small constant used for numerical stability. This term captures the proportional growth in order volume and is complementary to the absolute-growth term.

The raw discount-cost penalty is

$$r_t^{\mathrm{pen}} = -c \max\Big\{\mathrm{discount}_t - \mathrm{discount}^{\mathrm{med}},\, 0\Big\}\,, \tag{55}$$

where c is a scaling coefficient. This term penalizes the additional discount cost relative to the medium-tier reference. The penalty is applied only when the selected action corresponds to a discount deeper than the medium-tier reference.

Because the three raw quantities have different units and numerical ranges, we independently re-scale them before computing the final reward:

$$\tilde{r}_t^k = \mathrm{Rescale}_k\big(r_t^k\big), \qquad k \in \{\mathrm{abs}, \mathrm{rel}, \mathrm{pen}\}. \tag{56}$$

The re-scaled components are then combined as

$$r_t = \alpha_1 \tilde{r}_t^{\mathrm{abs}} + \alpha_2 \tilde{r}_t^{\mathrm{rel}} + \alpha_3 \tilde{r}_t^{\mathrm{pen}}. \tag{57}$$

Here $\alpha_1, \alpha_2, \alpha_3$ determine the relative importance of absolute growth, relative growth, and discount cost after all three components have been mapped to a shared scale.

E Cross-validated TD baseline

As a comparator, we split a batch into halves A and B, update the critic on A, and train γ_ϕ on the squared residual of B:

$$\mathcal{L}_\gamma^{\mathrm{CV}}(\phi) = \mathbb{E}_{(s_t,a_t)\in B}\big[(\delta_t^{(A)})^2\big], \tag{58}$$

where $\delta_t^{(A)}$ is the SAC or PPO one-step residual after the A-split update. Because that critic was not optimized on B, shrinking γ does not automatically reduce the loss; empirically this baseline still collapses (Table 5).

F Experiment Results

F.1 TD-target collapse: numerical summary

Both collapse arms disable adaptive $\bar{\gamma}$. Statistics are means over the final 10% of training, then mean $\pm$ standard deviation across 5 seeds, on a frozen 2048-state probe set. A leave-one-out on the three auxiliary terms of (13) is reported in Section 4.6, with numerical summaries in Appendix F.2.

Table 2: TD-target collapse ablation on `SafetyPointGoal1-v0` (SAC, 300k steps, 5 seeds).

Method	mean γ	**Std** γ	$\mathrm{Frac}_{\mathrm{floor}}$	Return
AdaGamma-RC	0.971 ± 0.003	0.0216 ± 0.0013	0.000	27.17 ± 0.12
AdaGamma-TD	0.911 ± 0.000	0.0014 ± 0.0003	0.000	26.35 ± 0.20

F.2 Leave-one-out on the full γ objective

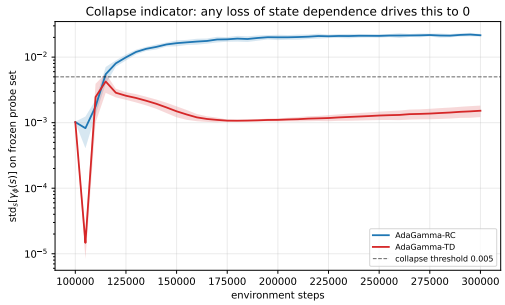
Numerical results for the leave-one-out of Section 4.6 appear below. Statistics are means over the final 10% of training, then mean $\pm$ standard deviation across 5 seeds, on a frozen 2048-state probe set. Shared hyperparameters are listed in Table 15.

F.3 Statewise structure of $\gamma_\phi(s)$

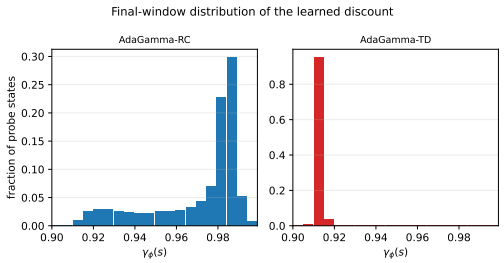
Figure 5 visualizes SAC-AdaGamma's learned discount on `Ant-v4` from rollout states of a fixed trained policy.

F.4 Cross-algorithm consistency

Table 4 reports the mean learned discount for SAC-AdaGamma and PPO-AdaGamma across representative tasks.



(a) Cross-state standard deviation of γ_ϕ on the frozen probe set (log scale).



(b) Final-window histogram of $\gamma_\phi(s)$.

Figure 2: TD-target collapse under SAC on `SafetyPointGoal1-v0`. Shaded bands are ± 1 standard deviation across 5 seeds. RC remains non-degenerate; squared-TD collapses to a near-constant discount.

Table 3: Leave-one-out on (13) for the γ module on `SafetyPointGoal1-v0` (SAC, 300k steps, 5 seeds). Statistics are means over the final 10% of training, then mean $\pm$ standard deviation across seeds, on a frozen 2048-state probe set. All runs disable adaptive $\bar{\gamma}$. **Std** γ is the primary collapse indicator; $\text{Frac}_{\text{floor}}$ is the fraction of probe states in the bottom 5% of the γ range.

Obj.	Auxiliaries	mean γ	**Std** γ	$\text{Frac}_{\text{floor}}$	Return
RC	full	0.971 ± 0.003	0.0216 ± 0.0013	0.000	27.17 ± 0.12
RC	w/o λ_{dev}	0.971 ± 0.002	0.0215 ± 0.0013	0.000	26.88 ± 0.10
RC	w/o λ_{var}	0.971 ± 0.003	0.0217 ± 0.0015	0.000	27.46 ± 0.31
RC	w/o λ_{bound}	0.971 ± 0.003	0.0234 ± 0.0014	0.000	27.29 ± 0.27
RC	none	0.970 ± 0.003	0.0238 ± 0.0017	0.000	27.20 ± 0.19
TD δ^2	paper defaults	0.911 ± 0.000	0.0014 ± 0.0003	0.000	26.35 ± 0.20
TD δ^2	none	0.900 ± 0.000	0.0000 ± 0.0000	1.000	26.09 ± 0.37

F.5 Training Objective Comparison

Table 5 summarizes the mean learned discounts of SAC variants across the three main benchmarks.

F.6 Network architecture

Figure 6 reports the performance of SAC-AdaGamma on `Ant-v4` as the hidden dimension of the gamma network varies over $\{8, 16, 32, 64, 128, 256, 512\}$.

F.7 SAC/PPO with AdaGamma under different return horizons

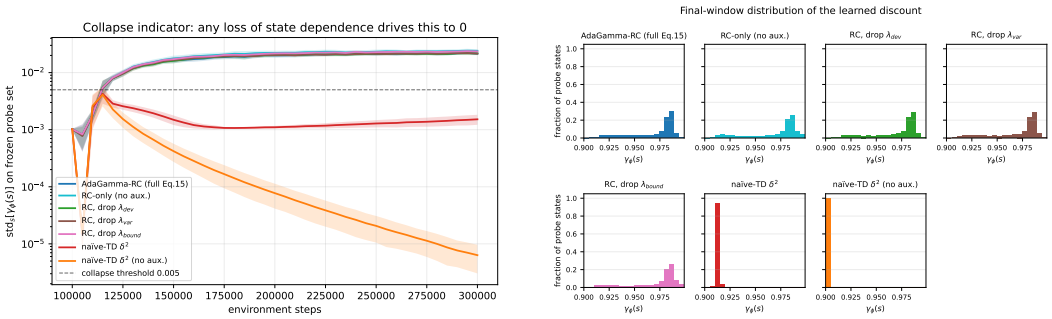
Table 6 reports the rewards and costs of SAC-AdaGamma and PPO-AdaGamma under different return horizons n.

F.8 Comparison to fixed-γ

Table 7 compares AdaGamma with fixed discounts chosen around the learned mean discount. Table 8 reports a broader fixed-γ grid search for SAC and PPO.

G Algorithmic Framework under AdaGamma

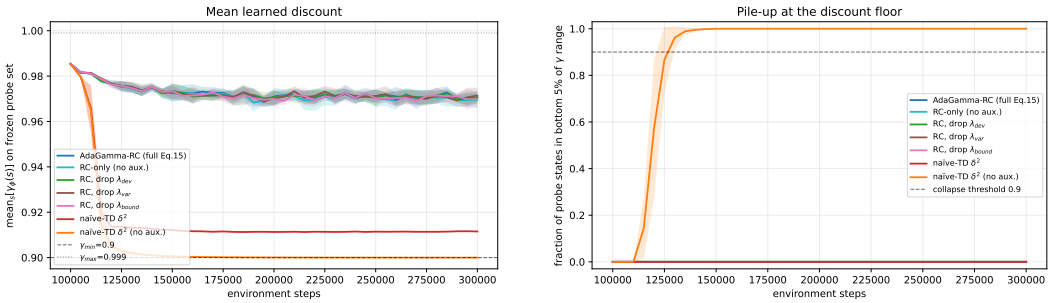
This section summarizes the SAC and PPO implementations of AdaGamma.



(a) Cross-state standard deviation of γ_ϕ on the frozen probe set (log scale).

(b) Final-window histogram of $\gamma_\phi(s)$.

Figure 3: Leave-one-out on (13) under SAC on `SafetyPointGoal1-v0`. Shaded bands are ± 1 standard deviation across 5 seeds. Every return-consistency variant remains non-degenerate; squared-TD collapses to a near-constant discount with the paper's regularizers, and to a point mass at $\gamma_{\min}$ without them.



(a) Mean probe γ_ϕ over training.

(b) Fraction of probe states in the bottom 5% of the γ range.

Figure 4: Additional diagnostics for (13) leave-one-out. Mean γ alone cannot establish collapse. Only TD without auxiliaries piles up at $\gamma_{\min}$; TD with regularizers settles interior to the range as a near-constant, so floor mass stays near zero while standard deviation collapses.

G.1 SAC-AdaGamma

Algorithm 1 summarizes the SAC implementation of AdaGamma.

G.2 PPO-AdaGamma

Algorithm 2 summarizes the PPO implementation of AdaGamma.

State-dependent γ makes advantage magnitudes vary across states, which interacts with PPO's clipping threshold. We normalize advantages by batch mean and standard deviation. During PPO's multiple epochs on one rollout we freeze $\gamma_\phi(s)$, computed once from the rollout states, so the gamma network cannot co-adapt with the policy inside a trust-region update. The product-of-gammas expansion also applies to n-step value targets:

$$V_{\text{target}}(s_t) = \sum_{k=0}^{n-1} \left(\prod_{j=0}^{k-1} \gamma_\phi(s_{t+j})(1 - d_{t+j}) \right) r_{t+k} + \left(\prod_{j=0}^{n-1} \gamma_\phi(s_{t+j})(1 - d_{t+j}) \right) V(s_{t+n}). \tag{59}$$

G.3 Reward Curves

This section reports the running reward curves of SAC and PPO, including AdaGamma and the baselines, on `SafetyPointGoal1-v0`, `Humanoid-v4`, and `Ant-v4`.

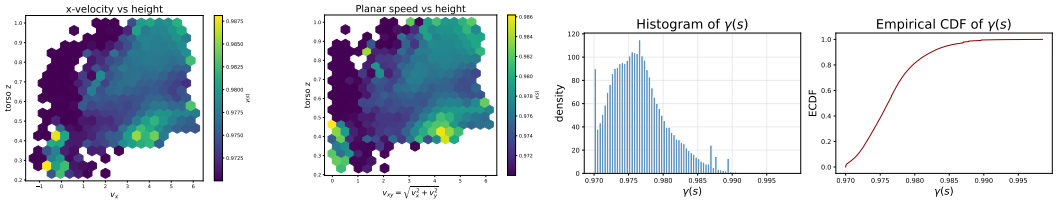


Figure 5: Statewise heatmaps and empirical distribution of $\gamma_\phi(s)$ learned by SAC-AdaGamma on `Ant-v4`.

Table 4: Mean learned $\gamma_\phi(s)$ for SAC-AdaGamma and PPO-AdaGamma.

Method Variants	SafetyPointGoal1-v0	Humanoid-v4	Ant-v4	Pendulum-v1
SAC-AdaGamma	0.9989	0.9860	0.9733	0.9908
PPO-AdaGamma	0.9988	0.9846	0.9990	0.9908

H Generalization to TRPO and DDPG

To examine whether AdaGamma transfers beyond our primary SAC/PPO implementations, we evaluate integrated DDPG (Lillicrap et al., 2019) and TRPO (Schulman et al., 2017a) agents on the same three benchmarks used in Section 4: the constrained navigation task `SafetyPointGoal1-v0` (Table 9), and the high-dimensional MuJoCo locomotion domains `Humanoid-v4` (Table 10) and `Ant-v4` (Table 11). We deliberately keep environments, seeds, horizons, evaluation protocol, and discount-related baselines aligned with the main experiments so that any differences can be attributed to the choice of base algorithm rather than task setup. `SafetyPointGoal1-v0` tests reward–cost tradeoffs under safety constraints, while `Humanoid-v4` and `Ant-v4` stress long-horizon coordination and contact-rich dynamics, respectively.

I Classic Control Results

To further assess whether state-dependent discounting is useful beyond the safety and high-dimensional locomotion settings, we evaluate AdaGamma on a set of standard Gymnasium control benchmarks with both SAC and PPO. Table 12 shows that AdaGamma improves or matches the base algorithm on most classic-control tasks. On `CartPole-v1`, both SAC–AdaGamma and PPO–AdaGamma reach the maximum score of 500, reducing the residual variance observed in the fixed-discount baselines. On `Pendulum-v1`, `MountainCarContinuous-v0`, and `Acrobot-v1`, performance remains below the task optimum and exhibits greater variability; even so, AdaGamma is often competitive with or modestly better than the fixed-discount baselines.

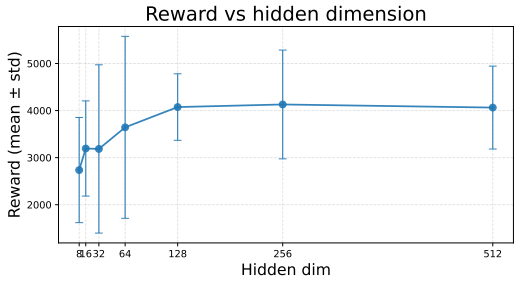


Figure 6: Performance of SAC-AdaGamma on `Ant-v4` as the gamma-network hidden dimension varies. The curve reports reward mean ± standard deviation.

Table 5: Mean value of γ learned by different SAC variants.

Method Variants	SafetyPointGoal1-v0	Humanoid-v4	Ant-v4
SAC-fixed γ	0.9900	0.9900	0.9900
SAC-CrossValidate	0.9000	0.9000	0.9000
SAC-Uncertainty	0.9495	0.9745	0.9745
SAC-AdaGamma	0.9989	0.9860	0.9733

Table 6: Mean reward and cost, with standard deviation, for SAC-AdaGamma and PPO-AdaGamma with $n \in \{5, 10, 20\}$.

Method Variants	SafetyPointGoal1-v0 Reward	SafetyPointGoal1-v0 Cost
SAC-AdaGamma($n = 5$)	**28.254 ± 1.05**	**29.37 ± 18.59**
SAC-AdaGamma($n = 10$)	27.75 ± 0.85	56.20 ± 36.93
SAC-AdaGamma($n = 20$)	27.71 ± 1.05	57.70 ± 48.42
PPO-AdaGamma($n = 5$)	**26.30 ± 0.90**	**41.82 ± 19.42**
PPO-AdaGamma($n = 10$)	25.92 ± 1.27	48.60 ± 32.44
PPO-AdaGamma($n = 20$)	16.06 ± 12.11	49.85 ± 36.39

J SNAPSHOTS OF HUMANOID-V4 AND ANT-V4

This section shows snapshots of trained SAC-AdaGamma and PPO-AdaGamma agents on `Humanoid-v4` and `Ant-v4`.

K HYPERPARAMETER CONFIGURATIONS

Table 7: Performance of SAC and PPO with fixed γ around the learned mean value.

Method	Reward	Cost	Method	Reward	Cost
SafetyPointGoal1-v0					
SAC-Fixed-$\gamma = 0.9950$	27.57 ± 0.81	51.25 ± 38.92	PPO-Fixed-$\gamma = 0.9950$	25.36 ± 1.63	53.25 ± 32.13
SAC-Fixed-$\gamma = 0.9989$	27.06 ± 4.55	53.20 ± 37.24	PPO-Fixed-$\gamma = 0.9988$	23.86 ± 1.93	46.85 ± 27.73
SAC-Fixed-$\gamma = 0.9990$	25.92 ± 1.34	50.20 ± 40.17	PPO-Fixed-$\gamma = 0.9990$	25.71 ± 1.32	48.60 ± 29.07
SAC-AdaGamma	$\mathbf{28.25 \pm 1.05}$	$\mathbf{29.37 \pm 18.59}$	PPO-AdaGamma	$\mathbf{26.31 \pm 0.90}$	$\mathbf{41.82 \pm 19.42}$
Humanoid-v4					
SAC-Fixed-$\gamma = 0.9850$	6050.05 ± 22.50	-	PPO-Fixed-$\gamma = 0.9950$	456.19 ± 27.61	-
SAC-Fixed-$\gamma = 0.9860$	5830.34 ± 6.54	-	PPO-Fixed-$\gamma = 0.9999$	442.33 ± 17.78	-
SAC-Fixed-$\gamma = 0.9990$	5576.23 ± 17.84	-	PPO-Fixed-$\gamma = 0.9990$	447.99 ± 23.06	-
SAC-AdaGamma	$\mathbf{6907.99 \pm 21.76}$	-	PPO-AdaGamma	$\mathbf{476.51 \pm 51.29}$	-
Ant-v4					
SAC-Fixed-$\gamma = 0.9700$	3601.58 ± 1812.92	-	PPO-Fixed-$\gamma = 0.9980$	1064.52 ± 193.92	-
SAC-Fixed-$\gamma = 0.9733$	3461.86 ± 1803.14	-	PPO-Fixed-$\gamma = 0.9985$	987.77 ± 234.11	-
SAC-Fixed-$\gamma = 0.9750$	3778.66 ± 644.60	-	PPO-Fixed-$\gamma = 0.9990$	1033.70 ± 192.00	-
SAC-AdaGamma	$\mathbf{4129.57 \pm 1155.08}$	-	PPO-AdaGamma	$\mathbf{1172.47 \pm 189.84}$	-

Table 8: Grid search of SAC and PPO with fixed γ.

Method	Reward	Cost	Method	Reward	Cost
SafetyPointGoal1-v0					
SAC-Fixed-$\gamma = 0.9000$	27.13 ± 1.09	53.30 ± 40.87	PPO-Fixed-$\gamma = 0.9000$	18.98 ± 2.34	51.80 ± 27.30
SAC-Fixed-$\gamma = 0.9500$	27.88 ± 1.23	49.45 ± 34.19	PPO-Fixed-$\gamma = 0.9500$	24.43 ± 1.80	49.95 ± 26.76
SAC-Fixed-$\gamma = 0.9700$	27.32 ± 1.30	51.65 ± 33.76	PPO-Fixed-$\gamma = 0.9700$	22.25 ± 1.57	46.45 ± 31.19
SAC-Fixed-$\gamma = 0.9800$	27.96 ± 1.12	46.45 ± 35.90	PPO-Fixed-$\gamma = 0.9800$	21.60 ± 4.80	50.80 ± 37.66
SAC-Fixed-$\gamma = 0.9990$	25.92 ± 1.34	50.20 ± 40.17	PPO-Fixed-$\gamma = 0.9990$	25.71 ± 1.32	48.60 ± 29.07
SAC-AdaGamma	$\mathbf{28.25 \pm 1.05}$	$\mathbf{29.37 \pm 18.59}$	PPO-AdaGamma	$\mathbf{26.31 \pm 0.90}$	$\mathbf{41.82 \pm 19.42}$
Humanoid-v4					
SAC-Fixed-$\gamma = 0.9000$	723.73 ± 72.03	-	PPO-Fixed-$\gamma = 0.9000$	305.43 ± 61.33	-
SAC-Fixed-$\gamma = 0.9500$	3280.41 ± 1826.32	-	PPO-Fixed-$\gamma = 0.9500$	349.12 ± 57.15	-
SAC-Fixed-$\gamma = 0.9700$	6185.12 ± 29.35	-	PPO-Fixed-$\gamma = 0.9700$	315.13 ± 19.90	-
SAC-Fixed-$\gamma = 0.9800$	6346.42 ± 15.40	-	PPO-Fixed-$\gamma = 0.9800$	356.19 ± 27.61	-
SAC-Fixed-$\gamma = 0.9990$	5576.23 ± 17.84	-	PPO-Fixed-$\gamma = 0.9990$	447.99 ± 23.06	-
SAC-AdaGamma	$\mathbf{6907.99 \pm 21.76}$	-	PPO-AdaGamma	$\mathbf{476.51 \pm 51.29}$	-
Ant-v4					
SAC-Fixed-$\gamma = 0.9000$	1223.375 ± 137.70	-	PPO-Fixed-$\gamma = 0.9000$	750.41 ± 70.57	-
SAC-Fixed-$\gamma = 0.9500$	3236.72 ± 1868.49	-	PPO-Fixed-$\gamma = 0.9500$	823.30 ± 120.42	-
SAC-Fixed-$\gamma = 0.9700$	3601.58 ± 1812.92	-	PPO-Fixed-$\gamma = 0.9700$	893.34 ± 141.93	-
SAC-Fixed-$\gamma = 0.9800$	3761.03 ± 1692.59	-	PPO-Fixed-$\gamma = 0.9800$	994.73 ± 187.78	-
SAC-Fixed-$\gamma = 0.9990$	918.32 ± 17.68	-	PPO-Fixed-$\gamma = 0.9990$	1033.70 ± 192.00	-
SAC-AdaGamma	$\mathbf{4129.57 \pm 1155.08}$	-	PPO-AdaGamma	$\mathbf{1172.47 \pm 189.84}$	-

Algorithm 1 AdaGamma with SAC (Off-Policy Adapter)

Input: Policy π_ψ, Q-networks $Q_{\theta_1}, Q_{\theta_2}$, gamma network g_ϕ, replay buffer $\mathcal{D}$
for each iteration **do**
  for each environment step **do**
    $a_t \sim \pi_\psi(\cdot|s_t)$, observe r_t, s_{t+1}, d_t
    $\mathcal{D} \leftarrow \mathcal{D} \cup \{(s_t, a_t, r_t, s_{t+1}, d_t)\}$
  end for
  for each gradient step **do**
    Sample mini-batch from $\mathcal{D}$
    // Compute adaptive discount:
    $\gamma_\phi(s_t) = \gamma_{\min} + (\gamma_{\max} - \gamma_{\min}) \cdot \sigma(g_\phi(s_t))$
    // Update Q-functions with adaptive target:
    $\hat{Q} = r_t + \gamma_\phi(s_t)(1 - d_t)[\min_i Q_{\bar{\theta}_i}(s_{t+1}, a') - \alpha \log \pi_\psi(a'|s_{t+1})]$
    $\theta_i \leftarrow \theta_i - \lambda_Q \nabla_{\theta_i} J_Q(\theta_i)$
    // Update policy and temperature (unchanged):
    $\psi \leftarrow \psi - \lambda_\pi \nabla_\psi J_\pi(\psi); \quad \alpha \leftarrow \alpha - \lambda_\alpha \nabla_\alpha J(\alpha)$
    // Update gamma network (return-consistency):
    Compute $G_t^{(n)}$ with reference $\bar{\gamma}$ and terminal masks (stop-gradient)
    Optionally update $\bar{\gamma}$ by EMA of replay-mean $\gamma_\phi(s)$ after warmup
    $\phi \leftarrow \phi - \lambda_\gamma \nabla_\phi J_\gamma(\phi)$ (13)
    // Soft-update target networks:
    $\bar{\theta}_i \leftarrow \tau\theta_i + (1 - \tau)\bar{\theta}_i$
  end for
end for

Algorithm 2 AdaGamma with PPO (On-Policy Adapter)

Input: Policy π_ψ, value network V_ω, gamma network g_ϕ
for each iteration **do**
  // Collect rollout:
  for $t = 0, \ldots, T-1$ **do**
    $a_t \sim \pi_\psi(\cdot|s_t)$, observe r_t, s_{t+1}, d_t
  end for
  // Compute $\gamma_\phi(s_t)$ for all rollout states (freeze):
  $\{\gamma_t\}_{t=0}^{T-1} \leftarrow \{\gamma_\phi(s_t)\}_{t=0}^{T-1}$
  // Modified GAE with state-dependent γ:
  $\delta_{T-1} = r_{T-1} + \gamma_{T-1}(1-d_{T-1})V(s_T) - V(s_{T-1})$
  $\hat{A}_{T-1} = \delta_{T-1}$
  for $t = T-2, \ldots, 0$ **do**
    $\delta_t = r_t + \gamma_t(1-d_t)V(s_{t+1}) - V(s_t)$
    $\hat{A}_t = \delta_t + \gamma_t\lambda(1-d_t)\hat{A}_{t+1}$
  end for
  Normalize: $\hat{A}_t \leftarrow (\hat{A}_t - \text{mean})/\text{std}$
  // PPO epochs (frozen γ):
  for epoch $= 1, \ldots, K$ **do**
    Update π_ψ via clipped surrogate with $\hat{A}_t$
    Update V_ω on value targets
  end for
  // Update gamma network:
  Compute L_γ^{RC} from rollout data with terminal masks (11)
  $\phi \leftarrow \phi - \lambda_\gamma \nabla_\phi J_\gamma(\phi)$
end for

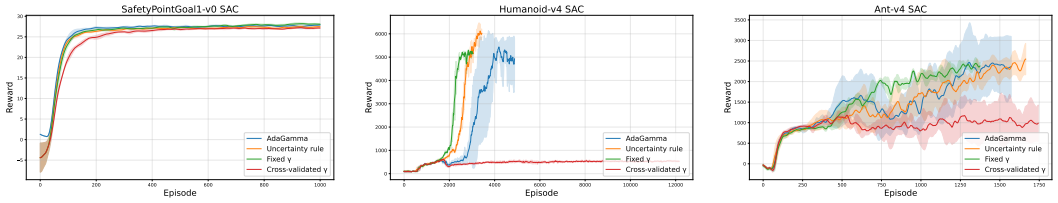


Figure 7: Learning reward curves of SAC-AdaGamma on `SafetyPointGoal1-v0`, `Humanoid-v4`, and `Ant-v4`.

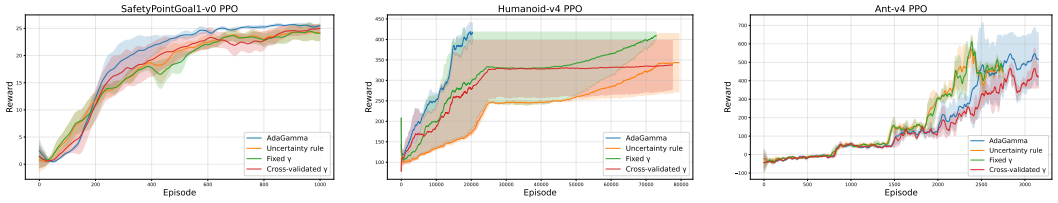


Figure 8: Learning reward curves of PPO-AdaGamma on `SafetyPointGoal1-v0`, `Humanoid-v4`, and `Ant-v4`.

Table 9: Performance in `SafetyPointGoal1-v0`.

Method Variants	Reward	Cost	Method Variants	Reward	Cost
TRPO-Fixed-γ	24.48 $\pm$ 2.01	**45.90** $\pm$ **28.90**	DDPG-Fixed-γ	27.34 $\pm$ 1.18	51.45 $\pm$ 41.13
TRPO-CrossValidate	24.38 $\pm$ 7.11	46.35 $\pm$ 28.92	DDPG-CrossValidate	27.26 $\pm$ 1.23	53.85 $\pm$ 41.92
TRPO-Uncertainty	23.50 $\pm$ 1.90	64.55 $\pm$ 35.31	DDPG-Uncertainty	27.10 $\pm$ 0.70	52.70 $\pm$ 40.64
TRPO-AdaGamma	**25.93** $\pm$ 1.46	49.40 $\pm$ 29.17	DDPG-AdaGamma	**27.45** $\pm$ **1.40**	**46.15** $\pm$ **32.72**

Table 10: Performance on `Humanoid-v4`.

Method Variants	Reward	Method Variants	Reward
TRPO-Fixed-γ	218.15 $\pm$ 11.33	DDPG-Fixed-γ	165.82 $\pm$ 5.67
TRPO-CrossValidate	221.10 $\pm$ 10.10	DDPG-CrossValidate	356.85 $\pm$ 18.47
TRPO-Uncertainty	250.65 $\pm$ 25.40	DDPG-Uncertainty	454.46$\pm$ 13.18
TRPO-AdaGamma	**284.49** $\pm$ **11.21**	DDPG-AdaGamma	**457.02** $\pm$ **45.58**

Table 11: Performance on `Ant-v4`.

Method Variants	Reward	Method Variants	Reward
TRPO-Fixed-γ	1326.83 $\pm$ 103.37	DDPG-Fixed-γ	3136.60 $\pm$ 1106.70
TRPO-CrossValidate	940.26 $\pm$ 457.39	DDPG-CrossValidate	1968.91 $\pm$ 687.19
TRPO-Uncertainty	940.26 $\pm$ 457.39	DDPG-Uncertainty	2915.64 $\pm$ 1179.86
TRPO-AdaGamma	**1451.34** $\pm$ **243.90**	DDPG-AdaGamma	**3774.19** $\pm$ **458.63**

Table 12: Test results of SAC and PPO with and without adaptive-γ.

Test Results (rewards)	SAC	SAC-AdaGamma	PPO	PPO-AdaGamma
CartPole-v1	483.20 $\pm$ 14.85	**500.00** $\pm$ **0.00**	481.90 $\pm$ 78.70	**500.00** $\pm$ **0.00**
Pendulum-v1	-64.832 $\pm$ 62.931	**-58.557** $\pm$ **56.618**	-212.27 $\pm$ 150.86	**-198.12** $\pm$ **126.97**
MountainCarContinuous-v0	94.47 $\pm$ 0.73	**94.60** $\pm$ **0.60**	84.72 $\pm$ 11.48	**86.84**$\pm$ **3.24**
Acrobot-v1	-82.91 $\pm$ 12.20	**-82.61** $\pm$ **14.64**	**-95.34** $\pm$ **21.46**	-115.186 $\pm$ 29.02

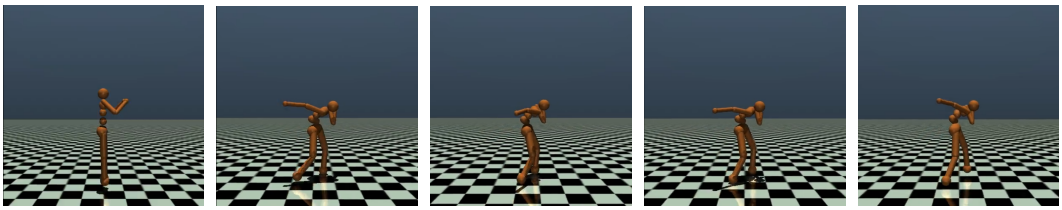


Figure 9: Snapshots of SAC-AdaGamma on `Humanoid-v4`.

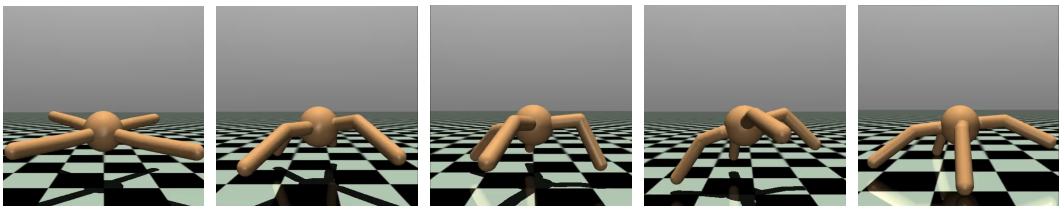


Figure 10: Snapshots of SAC-AdaGamma on `Ant-v4`.



Figure 11: Snapshots of PPO-AdaGamma on `Humanoid-v4`.

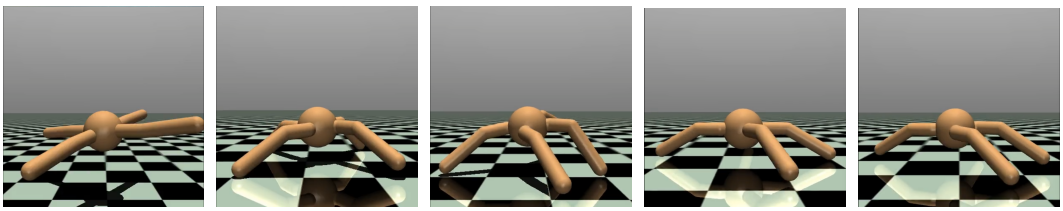


Figure 12: Snapshots of PPO-AdaGamma on `Ant-v4`.

Table 13: SAC hyperparameters by task.

	SafetyPointGoal1-v0	Humanoid-v4	Ant-v4
Task / schedule			
Max env steps	10^6	10^6	10^6
Eval interval		10^4 steps	
SAC backbone (shared `Config`)			
Actor/Critic LR		3×10^{-4}	
Target smoothing τ		5×10^{-3}	
Replay capacity		10^6	
Min buffer before train		5000	
Batch size		256	
Grad steps / env step		1	
Hidden dim (policy/Q)		256	
Entropy coef. α		0.2 (automatic tuning)	
Max grad norm		1.0	
Episode horizon cap		1000	
Safety constraint (SafetyPointGoal1 only)			
Enabled / cost limit		yes / 25	
Lagrange LR / init		5×10^{-3} / 10^{-3}	
Uncertainty-rule baseline (β)			
β_{init}; learnable; β LR		2.0; true; 10^{-3}	
Target γ (`uncertainty_target`)	—	0.99	0.99
CLI: `uncertainty-scale` $= s \Rightarrow \beta_{\text{init}} = 2s$		$s = 1$ in scripts	
Fixed / RC / cross-validated (per script)			
Warmup steps (`gamma-warmup-steps`)	10^5	10^5	10^5
γ-net LR / hidden		10^{-4} / 256	
Clip $[\gamma_{\min}, \gamma_{\max}]$	$[0.9, 0.999]$	$[0.9, 0.999]$	$[0.97, 0.999]$
n-step n	5	5	5
Fixed γ	0.99	0.99	0.99
RC: γ / $\bar{\gamma}$ init	0.98 / 0.98	0.98 / 0.98	0.98 / 0.98
RC: $\bar{\gamma}$ adaptive; after warmup		true / true	
RC: EMA τ / update every (episodes)		0.1 / 5	
RC: train every (`gamma-update-freq` steps)	20	20	20
λ_{rc}		1.0	
$\lambda_{\text{dev}}/\lambda_{\text{var}}/\lambda_{\text{bound}}$		0.005 / 0.012 / 0.05	
CV: γ; H_{cv}; loss weight	—	0.98; 20; 1.0	0.98; 20; 1.0
Uncertainty: `gamma-base`	—	0.99	0.98

Table 14: PPO hyperparameters by task.

Hyperparameter	SafetyPointGoal1-v0	Humanoid-v4	Ant-v4
Task / schedule			
Max env steps	10^6	3×10^6	10^6
Eval interval		10^4 steps	
PPO core			
Actor LR / critic LR	3×10^{-4} / 3×10^{-4}	10^{-4} / 10^{-4}	3×10^{-4} / 3×10^{-4}
Policy clip ε		0.2	
GAE λ		0.95	
PPO epochs	10	8	10
Rollout buffer size	4096	16384	4096
Mini-batch size	128	256	128
Entropy coef.	0.01	0.005	0.01
Max grad norm		0.5	
Action std (init / floor / decay Δ)		0.5 / 0.1 / 0.05	
Action-std decay period (steps)	2×10^5	10^5	2×10^5
Episode horizon cap		1000	
γ network (AdaGamma / CV / RC branches)			
Hidden dim / γ LR		256 / 3×10^{-4}	
Clip $[\gamma_{\min}, \gamma_{\max}]$		[0.9, 0.999]	
γ-net warmup (episodes)	200 *(default; script unset)*	500	200 *(default; script unset)*
Reg. $\lambda_{\text{dev}}/\lambda_{\text{var}}/\lambda_{\text{bound}}/\epsilon_{\text{bound}}$		0.01 / 0.005 / 0.05 / 0.005	
Return-consistency (RC) — where used			
Horizon n (`rc-horizon`)	$\{5, 10, 20\}$ *(separate runs)*	10	10
λ_{rc}		1.0	
$\bar{\gamma}$ init / adaptive / after warmup		0.98 / true / true	
Ref EMA τ / update every (PPO upd.)	0.1 / 1	0.05 / 5	0.1 / 1
Per-baseline settings (non-RC)			
Fixed γ	0.99	0.99	0.99
Uncertainty(`gamma-base`, $\gamma_{\min}$, scale, `num-q-ensembles`)	0.99, 0.9, 1.0, 5	0.99, 0.9, 1.0, 5	0.99, 0.9, 1.0, 5
Cross-validated (H_{cv}, λ_{cv})	10, 1.0	10, 1.0	10, 1.0

Table 15: Shared hyperparameters for the collapse ablation.

Hyperparameter	Value
Environment	SafetyPointGoal1-v0
Total env. steps	300,000
γ warmup steps	100,000
$[\gamma_{\min}, \gamma_{\max}]$	$[0.9,\ 0.999]$
γ / $\bar{\gamma}$ init	0.98 / 0.98
Adaptive $\bar{\gamma}$	off
RC n-step	5
γ network hidden dim	256
γ learning rate	1×10^{-4}
γ update frequency	every 20 critic steps
Probe size / interval	2048 / 5,000 steps
Seeds	42, 43, 44, 45, 46