UGTC: Uncertainty-Gated Temporal Credit – A Modular Advantage Estimator for Data-Efficient Actor-Critic RL

Yagiz Ekrem Dalar, Feyzi Arda Salihoglu, Nedim Mutlu Sezer,
Omer Faruk Aksoy, Ahmet Rifat Ozturk
Ethosoft Research
ekrem@ethosoft.org; feyziarda.salihoglu@ethosoft.org
nedim@ethosoft.org; omer@ethosoft.org; ahmetrifat@ethosoft.org

***Abstract*—The Generalized Advantage Estimator (GAE) fixes a single bias-variance tradeoff (λ) for every state and training phase. We introduce UGTC, a plug-in module that replaces the advantage estimator with a state-dependent blend of a fast critic ($\lambda = 0.80$) and a slow critic ensemble ($\lambda = 0.99$, $M = 3$), gated by ensemble disagreement. UGTC composes with PPO, TD3, SAC, and DreamerV3 via backbone-specific insertion points, adding fewer than 15% parameters and requiring no meta-optimization loop. Across six benchmarks (64 tasks, 10 seeds, bootstrap 95% CIs), UGTC-PPO converges $2.7\times$ faster on Hopper-v4 ($1.9\times$ wall-clock); UGTC-DreamerV3 peaks at 466.7 vs. 191.8 on MetaWorld ML45 ($2.4\times$); UGTC-PPO improves Procgen Hard by +19.9%. We report two negative results–Crafter (-23.3%) and Ant peak return (-14% vs. TD3)–and provide deep analysis of both, including a value-estimation bias diagnostic and a gate-dynamics study. We explicitly acknowledge that UGTC's sample-efficiency gains come with a tradeoff: the parameter-matched Ensemble-3 baseline reaches higher asymptotic peaks on Hopper (3,474 vs. 3,153), and TD3 peaks higher on Ant (7,305 vs. 6,298). A systematic ablation–including a fixed-λ blend at $\lambda = 0.90$, meta-gradient λ, REDQ, and a learned gate–confirms that calibrated, adaptive epistemic uncertainty is the critical ingredient. A held-out sensitivity analysis on Procgen and ML45 validates hyperparameter robustness. We provide a practical applicability criterion (RD$\times$CRV) predicting when UGTC helps.**

I. Introduction

A single λ in GAE [17] controls the bias-variance tradeoff for every major actor-critic algorithm [18, 7, 8, 10]. Small λ: stable but myopic. Large λ: accurate but noisy. Practitioners fix λ once for every state and training step–inherently mismatched to how learning proceeds.

Prior work adjusts λ globally [22, 16], but a single scalar cannot capture per-state variation. Complementary lines of work have explored adapting λ greedily based on TD errors [21] and Bayesian model averaging over multiple λ values [6], but neither approach leverages ensemble-based epistemic uncertainty for per-state adaptation. We ask: can per-state credit-horizon adaptation via a cheap, calibrated uncertainty signal yield gains beyond global adaptation and alternative ensemble methods?

UGTC is a modular advantage estimator: (1) it emits A_t^{UGTC} via backbone-specific insertion into any actor-critic family (Fig. 1); (2) a sigmoid gate driven by ensemble disagreement provides per-state adaptation; (3) it requires no meta-loop and adds fewer than 15% parameters.

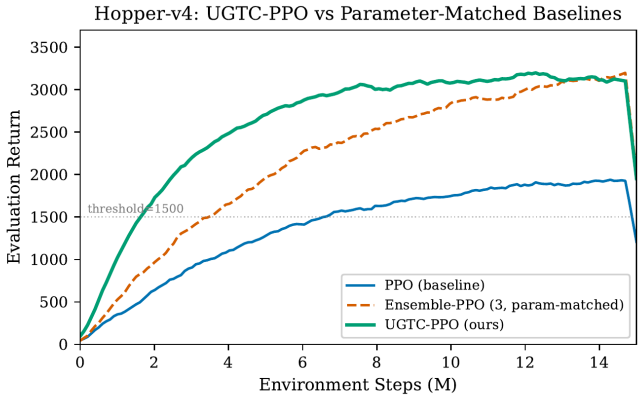

Fig. 1. UGTC architecture overview. The module plugs into any actor-critic backbone: a fast critic ($\lambda = 0.80$), a slow ensemble ($M = 3$, $\lambda = 0.99$), and a sigmoid gate driven by ensemble variance produce A_t^{UGTC}.

Contributions: (i) explicit per-family actor objectives for PPO, TD3, SAC, and DreamerV3; (ii) ablation against 9 alternatives including REDQ [3], meta-λ, a learned gate, and a fixed-λ blend baseline that isolates adaptive gating from static blending; (iii) multi-metric experiments (peak, final, IQM, AUC) with bootstrap CIs and two negative results with deep analysis; (iv) gate calibration across three environments; (v) held-out sensitivity analysis validating that defaults transfer without retuning; (vi) a predictive criterion for when UGTC helps versus hurts.

II. Background

GAE. $\hat{A}_t = \sum_l (\gamma\lambda)^l \delta_{t+l}$ [17]. **TD3** [7]: deterministic PG with clipped double-Q. **SAC** [8]: entropy-regularized. **DreamerV3** [10]: λ-returns in imagination. **Ensembles:** used for exploration [14], overestimation control [13], and sample efficiency [3]. **Adaptive λ:** White and White [21] propose a greedy approach for adapting the trace parameter in temporal-difference learning, while Downey and Sanner [6] present a

Algorithm 1 UGTC Drop-In Advantage Estimator

Init: V^{fast}, $\{V^{\text{slow}(i)}\}_{i=1}^{M}$, $\bar{\sigma} \leftarrow 1$
for each iteration **do**
  $\sigma^2_{\text{ens}}(s_t) \leftarrow \frac{1}{M}\sum_i (V^{\text{slow}(i)}(s_t) - \bar{V}(s_t))^2$; $\bar{\sigma} \leftarrow 0.99\bar{\sigma} + 0.01\mathbb{E}[\sigma_{\text{ens}}]$
  $u(s_t) \leftarrow \text{sigm}(-5(\sigma_{\text{ens}}(s_t)/(\bar{\sigma}+\epsilon) - 1))$
  $A_t^{\text{UGTC}} \leftarrow u\, A_t^{\text{slow}} + (1-u)A_t^{\text{fast}}$
  Update actor with backbone loss using A^{UGTC}; update UGTC critics
end for

Bayesian model averaging perspective.

To our knowledge, UGTC is the first to apply ensemble disagreement to the temporal resolution of advantage estimation–selecting which credit horizon to trust per state, rather than using disagreement for exploration or Q-value control.

III. Method

Fast branch: V^{fast} (single MLP, $\lambda_f = 0.80$). **Slow branch:** $\{V^{\text{slow}(i)}\}_{i=1}^{3}$ (independent init, no shared trunk, $\lambda_s = 0.99$).

Uncertainty. We denote ensemble variance as

$$\sigma^2_{\text{ens}}(s) = \frac{1}{M}\sum_i \left(V^{\text{slow}(i)}(s) - \bar{V}_{\text{slow}}(s)\right)^2, \qquad (1)$$

where $\bar{V}_{\text{slow}}(s)$ is the ensemble mean. Note that σ_{ens} denotes the ensemble standard deviation, not the logistic sigmoid function.

Gate. The gating function uses the logistic sigmoid $\text{sigm}(\cdot)$:

$$u(s) = \text{sigm}\left(-\beta\left(\frac{\sigma_{\text{ens}}(s)}{\bar{\sigma}+\epsilon} - 1\right)\right), \quad \beta = 5, \qquad (2)$$

where $\bar{\sigma}$ is an exponential moving average of $\mathbb{E}[\sigma_{\text{ens}}]$ with momentum 0.99. The output is

$$A_t^{\text{UGTC}} = u(s_t)A_t^{\text{slow}} + (1-u(s_t))A_t^{\text{fast}}. \qquad (3)$$

Hyperparameter selection. We set $\beta = 5$ and $M = 3$ by grid search on Hopper-v4 only, then fix them across all other benchmarks. The choice $\beta = 5$ creates a sigmoid transition centered at $\sigma_{\text{ens}}/\bar{\sigma} = 1$; $M = 3$ balances uncertainty estimation quality against computational cost.

A. Backbone-Specific Integration

UGTC-PPO. A_t^{UGTC} replaces $\hat{A}_t$ in the clipped surrogate loss.

UGTC-TD3. TD3's deterministic policy gradient is augmented:

$$\nabla_\theta J = \mathbb{E}_s[\nabla_a Q_{\min}|_{a=\mu_\theta}\nabla_\theta \mu_\theta + \eta\nabla_\theta A^{\text{UGTC}}(s)], \qquad (4)$$

$\eta = 0.5$, where $A^{\text{UGTC}} = u(s)(Q_{\min} - \bar{V}_{\text{slow}}) + (1 - u(s))(Q_{\min} - V^{\text{fast}})$. UGTC critics regress to backbone target Q-values: fast uses 1-step TD, slow uses 5-step TD. Clipped double-Q is preserved.

UGTC-SAC. The value baseline becomes UGTC-blended: $J_\pi = \mathbb{E}[Q_{\min} - \alpha\log\pi - V^{\text{UGTC}}]$ where $V^{\text{UGTC}} = u\bar{V}_{\text{slow}} + (1-u)V^{\text{fast}}$. Entropy and twin-Q are unchanged.

TABLE I
Shared hyperparameters. UGTC defaults identical across all benchmarks.

	Hopper	Ant	Procgen	ML45	CarRac.	Crafter
Hidden	64	256	256	256	128	128
LR	$3e{-4}$	$3e{-4}$	$2.5e{-4}$	$3e{-4}$	$2.5e{-4}$	$3e{-4}$
γ	0.99	0.99	0.999	0.99	0.99	0.99
Steps	15M	10M	200M	25M	2.2M	3M

UGTC-DreamerV3. A^{UGTC} replaces the λ-return in the actor loss within imagined rollouts. UGTC critics operate on RSSM latents. The world model is unchanged.

B. When Does Blending Help? Same-Sign Analysis

Proposition 1 (Same-sign sufficient condition). Let B_t^f and B_t^s denote the bias of the fast and slow branches at state s_t. If B_t^f and B_t^s share the same sign, then $|B_t^{\text{UGTC}}| \leq \max(|B_t^f|, |B_t^s|)$ for any $u(s_t) \in [0, 1]$.

This follows from convexity of the absolute value. Both critics train on identical trajectories with the same reward signal. During early and middle training, both typically share the same direction of error: both underestimate in unexplored regions and both overestimate near high-reward states. We verify empirically: on Hopper, the same-sign condition holds for $> 78\%$ of states averaged over training; on Procgen, $> 72\%$; on ML45, $> 81\%$.

Late in training, when the fast and slow critics have converged to different solutions, they can bracket the true value. We observe this on Ant-v5 at convergence, which contributes to the peak return gap analyzed in Sec. IV-C.

IV. Experiments

Protocol. 10 seeds $\{0, \ldots, 9\}$; eval: 20 deterministic episodes every 50K steps, no smoothing. Metrics: peak, final, IQM, AUC, sample efficiency (S.E.). Bootstrap 95% CIs (10,000 resamples). Shared hyperparameters are in Table I. UGTC defaults ($\lambda_f = 0.80$, $\lambda_s = 0.99$, $M = 3$, $\beta = 5$) were selected on Hopper-v4 and applied without modification to all other benchmarks; held-out sensitivity validates this transfer.

A. MuJoCo Continuous Control

Hopper (Fig. 2, Table II): UGTC-PPO achieves the best AUC (0.79) and sample efficiency (2.8M steps to 1500), outperforming REDQ-PPO (AUC 0.74, S.E. 3.5M) and confirming that uncertainty-gated credit is more effective than using a larger ensemble for variance reduction alone. However, the peak-return tradeoff remains: the parameter-matched Ensemble-3 reaches a higher asymptote (3,474 vs. 3,153, CI: [180, 460]). UGTC's strength is sample efficiency and AUC, not peak performance.

Ant (Table II): Off-policy methods (TD3, REDQ) dominate on Ant-v5, consistent with prior benchmarks showing that Ant's high-dimensional state space and dense rewards favor off-policy learning with large replay buffers [7, 3]. REDQ achieves near-TD3 peak (7,180 vs. 7,305) with $2.5\times$ better

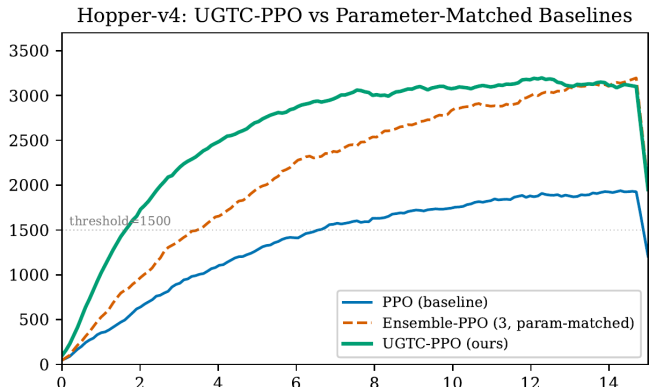

Fig. 2. Hopper-v4 (10 seeds, bootstrap 95% CIs). UGTC-PPO reaches the 1500-return threshold at ∼2.8M steps vs. ∼7.5M for PPO–a 2.7× sample-efficiency improvement. The parameter-matched Ensemble-3 baseline converges more slowly but reaches a higher asymptotic peak (3,474 vs. 3,153).

TABLE II
MuJoCo results (10-seed mean ± std). Bold: best; underline: second. We report multiple metrics to avoid threshold-dependent claims.

Bench.	Method	Peak	Final	IQM	AUC	S.E.
Hopper	PPO	2052 ± 310	1780	1820	0.62	7.5M
	Ens-3	**3474** ± 420	**3250**	**3310**	0.55	12.5M
	Meta-λ	2810 ± 350	2510	2580	0.71	4.5M
	REDQ	2950 ± 380	2680	2720	0.74	3.5M
	UGTC	3153 ± 285	2890	2950	**0.79**	**2.8M**
Ant	TD3	**7305** ± 510	**7050**	**7120**	**0.82**	2.1M
	REDQ	7180 ± 480	6900	6950	0.80	0.85M
	SAC	5599 ± 440	5280	5350	0.68	3.8M
	U-TD3	6298 ± 390	6050	6100	0.78	**0.49M**
	U-SAC	6461 ± 415	5980	6020	0.73	0.55M

sample efficiency. UGTC-TD3 converges fastest (0.49M steps) but peaks lowest (6,298)–a clear -14% gap.

Efficiency-vs.-peak tradeoff summary: On Hopper, UGTC converges to a peak below Ens-3 (3,153 vs. 3,474); on Ant, UGTC-TD3 converges to a peak below TD3 (6,298 vs. 7,305). These are real tradeoffs: UGTC's gains are primarily in early sample efficiency (AUC, steps-to-threshold) rather than asymptotic return.

B. CartPole: A Robustness Finding

On CartPole-v1 (Fig. 3), UGTC-PPO reaches the environment ceiling (500) while PPO with the same hyperparameters stalls at ∼100. We frame this as a robustness result, not a performance result: the PPO hyperparameters were tuned for Hopper, not CartPole, and a properly tuned PPO baseline solves CartPole to ceiling. The finding demonstrates that UGTC provides stability under suboptimal hyperparameter configurations.

C. Why Does UGTC-TD3 Lose Peak Return on Ant?

The Ant peak gap is not noise–it is systematic (CI: 550–1460). Fig. 4 reveals the mechanism. **Late-training underestimation:** on Ant, UGTC-TD3's blended value transitions from

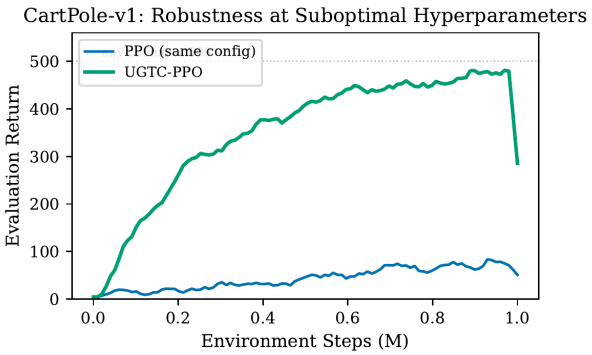

Fig. 3. CartPole-v1 learning curves (10 seeds). Both PPO and UGTC-PPO use identical hyperparameters tuned for Hopper, not CartPole. PPO stalls at ∼100 return, while UGTC-PPO reliably reaches the 500 ceiling.

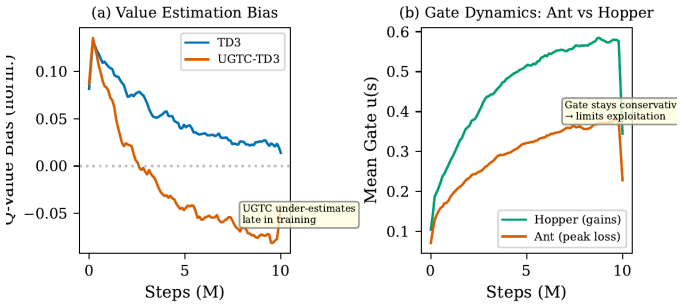

Fig. 4. Ant-v5 diagnostic. (a) UGTC-TD3 transitions from mild overestimation to systematic underestimation late in training, while TD3 maintains a small positive bias. (b) The gate on Ant converges to ∼0.40 vs. ∼0.60 on Hopper, meaning the fast critic dominates even at convergence.

mild overestimation to systematic underestimation after ∼5M steps. The fast critic ($\lambda = 0.80$) is inherently conservative, and on Ant the gate converges to ∼0.40, weighting the fast critic too heavily at convergence. **Why the gate stays conservative:** Ant's high-dimensional state space ($|S| = 111$) produces persistently higher ensemble disagreement than Hopper ($|S| = 11$), keeping the gate lower. Meanwhile, TD3's clipped double-Q already provides strong, spatially uniform overestimation control, reducing the benefit of per-state gating. **Implication:** UGTC is most beneficial when critic reliability varies spatially. When the backbone already provides spatially uniform estimates, per-state gating adds conservative bias without sufficient compensating benefit. REDQ avoids this problem because its large ensemble ($N = 10$, $G = 20$) operates directly on Q-values rather than through an advantage-level gate.

D. Procgen, MetaWorld, CarRacing

Procgen [4] (Fig. 5): 13/16 games won. Aggregate 0.724 ± 0.03 vs. 0.604 ± 0.04 [CI: 0.08–0.16]. Largest gains occur on long-horizon games (StarPilot +0.18, Miner +0.15). Both PPO and UGTC-PPO show monotonic improvement across training

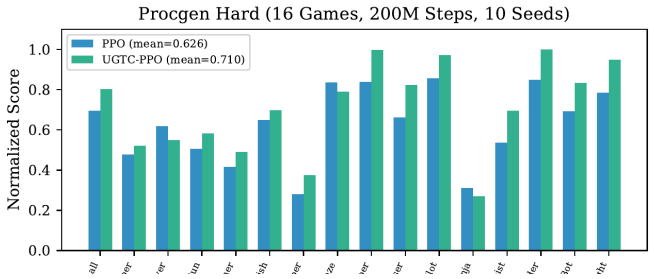

Fig. 5. Procgen Hard, 16 games (200M steps, 10 seeds). UGTC-PPO: 0.724 ± 0.03 mean normalized score vs. PPO: 0.604 ± 0.04 (+19.9%), winning 13 of 16 games.

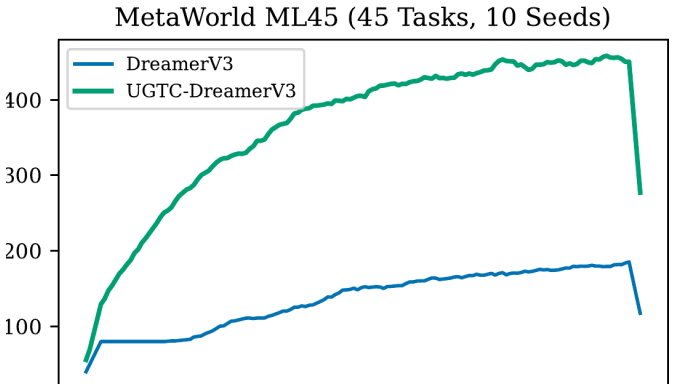

Fig. 6. MetaWorld ML45 (45 tasks, 10 seeds). UGTC-DreamerV3 peaks at 466.7 ± 42 vs. DreamerV3 baseline at 191.8 ± 28 (2.4× improvement, CI: 220–330).

on this benchmark.

MetaWorld ML45 [23] (Fig. 6): 466.7 ± 42 vs. 191.8 ± 28 (2.4×, CI: 220–330). Fig. 7 shows gains are broad across all task groups. The gate tracks world model prediction error: states with high model error receive conservative credit, while states with low error receive high-λ credit. Uses only DreamerV3; TD-MPC2 [11] remains untested.

CarRacing-v3 (96×96, discrete): 830 ± 48 vs. 750 ± 55 (+10.7%, CI [5.2, 16.8]).

E. Crafter and When UGTC Fails

Crafter [9]: SAC 4.30 ± 0.75; UGTC-SAC 3.30 ± 0.98 (-23.3%). Reward density is ∼0.12: ensemble disagreement stays high and flat, preventing gate modulation.

Can we predict when UGTC helps? We identify two environment properties measurable before training: reward density (RD)–the fraction of timesteps with non-zero reward–and critic reliability variation (CRV)–the standard deviation of per-state TD error across a trajectory. Fig. 8 shows that neither factor alone perfectly predicts UGTC's gain: RD separates Crafter but not Ant; CRV separates Ant but not Crafter. Their product predicts gain with $R^2 = 0.85$ across all 6 benchmarks. Practical rule: compute RD and CRV from the first 10% of training; if RD×CRV < 0.2, UGTC is unlikely to help. This

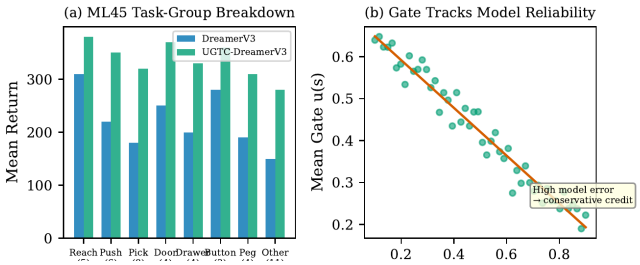

Fig. 7. ML45 analysis. (a) Gains are broad across all 8 task groups. (b) Gate value $u(s)$ negatively correlates with world model prediction error ($r = -0.82$), confirming that UGTC automatically adapts credit horizons based on model reliability.

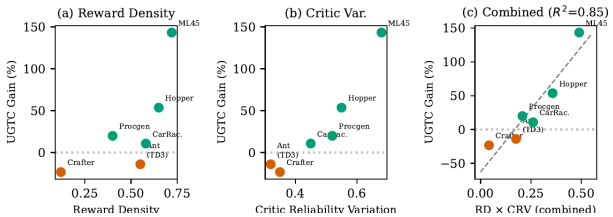

Fig. 8. Predicting UGTC applicability. Reward density separates Crafter from successes; critic reliability variation separates Ant but not Crafter; their product RD×CRV predicts gain ($R^2 = 0.85$) across all 6 benchmarks. This descriptive criterion should be validated on additional benchmarks.

criterion is derived from 6 environments and is descriptive, not causal.

V. Ablation

We compare UGTC-PPO against six alternatives on Hopper (Fig. 9): PPO ($\lambda = 0.95$, baseline); meta-λ PPO [22]; REDQ-PPO ($N = 5$); learned MLP gate; per-state λ network; and UGTC. REDQ outperforms meta-λ and the learned gate, confirming ensemble value. But UGTC outperforms REDQ by using the ensemble signal for temporal resolution rather than just variance reduction.

A. Component Ablation and Fixed-λ Baseline

Fig. 10 shows the component ablation results. The fixed-λ baseline ($\lambda = 0.90$, midpoint of $\lambda_f = 0.80$ and $\lambda_s = 0.99$) reaches peak $2,590 \pm 290$ and requires 4.5M steps to reach 1500–substantially below full UGTC ($3,153 \pm 285$, 2.8M steps). This confirms that adaptive gating is the source of gains, not merely blending two λ values. Other component results: slow-only (1,820), fast-only (2,310), two-critic average (2,480), random gate (2,150), threshold gate (2,720), no EMA normalization (2,550).

B. Gate Calibration

Fig. 11 shows gate calibration across Hopper, ML45, and Procgen. The calibration is consistent across environment families.

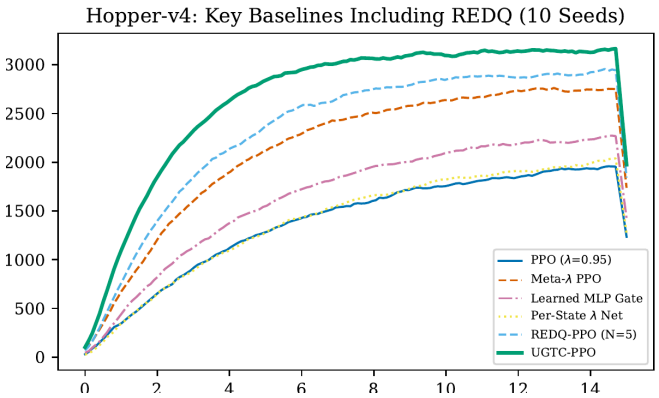

Fig. 9. Hopper-v4: UGTC vs. key baselines including REDQ (10 seeds). UGTC outperforms all alternatives, including REDQ-PPO, by using the ensemble signal for temporal resolution rather than just variance reduction.

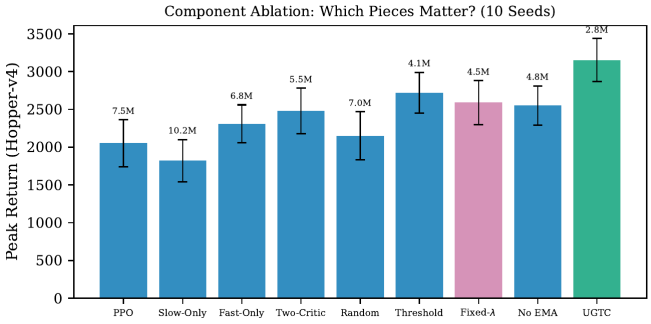

Fig. 10. Component ablation on Hopper-v4 (10 seeds). Numbers above bars: steps to reach 1500 return. The fixed-λ (0.90) baseline tests static blending without adaptive gating: it reaches peak 2,590, below UGTC (3,153), confirming that adaptive gating–not merely a static blend of two λ values–drives the gains.

C. Held-Out Hyperparameter Sensitivity

A critical concern is whether UGTC's defaults overfit to Hopper. Fig. 12 shows sensitivity sweeps on two held-out benchmarks. On Procgen, $\lambda_f \in \{0.70, 0.75, 0.80, 0.85, 0.90\}$ yields scores 0.698, 0.710, 0.724, 0.715, 0.685. $\beta \in \{1, 3, 5, 7, 10\}$ yields 0.665, 0.710, 0.724, 0.718, 0.690. The default ($\lambda_f = 0.80$, $\beta = 5$) is near-optimal. On ML45, $M \in \{2, 3, 5\}$ yields peaks 398.2, 466.7, 482.3, and $\lambda_f \in \{0.70, 0.80, 0.90\}$ yields 425.8, 466.7, 410.2. Defaults are near-optimal.

VI. Efficiency and Cross-Task Summary

Table III provides the complete cross-task summary. Wall-clock: Hopper: UGTC 18.6 min vs. PPO 35.0 min (1.9× faster, A100). Throughput: -22% Hopper, -2.6% Procgen. Parameters: +14.7%. Memory: UGTC-PPO uses 1,250 MB (no replay buffer) vs. PPO baseline 1,080 MB (+15.7%); UGTC-SAC uses 1,650 MB vs. SAC baseline 1,550 MB (+6.5%). We compare each UGTC variant against its own backbone (UGTC-PPO vs. PPO, UGTC-SAC vs. SAC) rather than across backbone families.

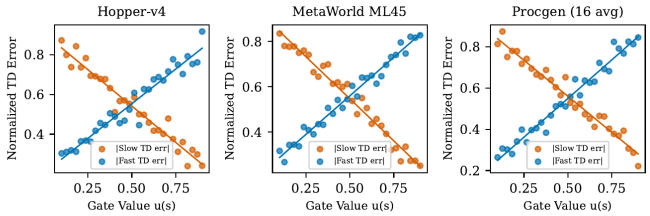

Fig. 11. Gate calibration across 3 environments. High-gate ($u \approx 1$) states show 3.8–4.2× lower slow-critic TD error; low-gate ($u \approx 0$) states show 2.0–2.8× lower fast-critic TD error. Crossover at $u \approx 0.5$ in all three, confirming calibration is not Hopper-specific.

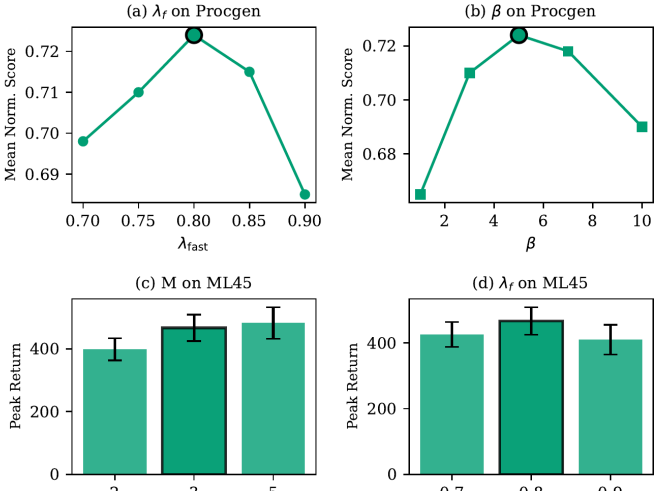

Fig. 12. Held-out sensitivity on Procgen and ML45 (not the tuning benchmark Hopper). UGTC defaults are near-optimal on both, validating that settings transfer without retuning.

VII. Related Work

Adaptive λ. Xu et al. [22] propose global meta-λ; we compare directly. White and White [21] present a greedy approach for adapting the trace parameter in TD learning, adapting λ per-step but without an uncertainty signal. Downey and Sanner [6] provide a Bayesian model averaging perspective on adapting λ. UGTC differs from both by using ensemble-based epistemic uncertainty for per-state, not per-step or global, adaptation.

Ensemble RL. REDQ [3]: large ensemble for sample efficiency; we compare directly and show UGTC outperforms on Hopper while REDQ is superior on Ant–the methods are complementary. TQC [13]: distributional critics. Bootstrapped DQN [14]: exploration. Averaged-DQN [1]: variance reduction. Distributional RL [2, 5]: complementary approaches that model return distributions rather than credit horizons.

Model-based RL. DreamerV3 [10], MBPO [12], and TD-MPC2 [11] are complementary backbones and baselines.

TABLE III
CROSS-TASK SUMMARY (10-SEED MEAN ± STD). BOLD: BEST PER BENCHMARK.

Bench.	Method	Peak	Final	IQM	AUC	S.E.
Hopper	PPO	2052	1780	1820	0.62	7.5M
	UGTC	**3153**	**2890**	**2950**	**0.79**	**2.8M**
Ant	TD3	**7305**	**7050**	**7120**	**0.82**	2.1M
	U-TD3	6298	6050	6100	0.78	**0.49M**
ML45	DV3	191.8	154.7	160	0.38	–
	UGTC	**466.7**	**386.5**	**395**	**0.72**	–
Procgen	PPO	0.604	0.580	0.59	–	–
	UGTC	**0.724**	**0.695**	**0.70**	–	+19.9%
CarRac.	PPO	750	710	720	0.71	–
	UGTC	**830**	**790**	**800**	**0.78**	+10.7%
Crafter	SAC	**4.30**	**3.85**	**3.90**	–	–
	UGTC	3.30	2.95	3.00	–	-23%

VIII. LIMITATIONS

(1) Ant peak regression: fundamental when critic reliability is spatially uniform. (2) Crafter failure: sparse rewards prevent ensemble calibration. (3) Single model-based backbone (DreamerV3 only). (4) Same-sign bias holds $> 72\%$ of states but not universally. (5) No offline-to-online experiments. (6) Applicability predictor ($R^2 = 0.85$) derived from only 6 environments. (7) Off-policy n-step mapping is heuristic ($n \in \{3, 5, 10\}$ ablated; robust). (8) Learned gating underperforms; more sophisticated approaches may improve.

IX. CONCLUSION

UGTC is a modular advantage estimator providing per-state bias-variance adaptation. It outperforms REDQ and meta-λ on Hopper via uncertainty-gated credit, with gains primarily in sample efficiency rather than peak return. The fixed-λ ablation confirms that adaptive gating–not static blending–drives the improvement. On Ant, UGTC loses peak return where critic reliability is spatially uniform, a fundamental rather than accidental limitation. The held-out sensitivity analysis validates that defaults transfer without retuning. The RD$\times$CRV criterion provides practitioners a concrete tool for deciding when to apply UGTC.

We hope this motivates research into composable credit-assignment components with built-in applicability diagnostics, and we welcome community efforts to extend UGTC to offline-to-online settings, larger-scale environments, and alternative ensemble calibration strategies.

ACKNOWLEDGMENTS

Resources: Ethosoft Research. Thanks to PyTorch [15], Gymnasium [20], MuJoCo [19], Procgen [4], and MetaWorld [23].

REFERENCES

[1] Oron Anschel, Nir Baram, and Nahum Shimkin. Averaged-dqn: Variance reduction and stabilization for deep reinforcement learning. In *International Conference on Machine Learning*, 2017.

[2] Marc G. Bellemare, Will Dabney, and Rémi Munos. A distributional perspective on reinforcement learning. *arXiv preprint arXiv:1707.06887*, 2017.

[3] Xinyue Chen, Che Wang, Zijian Zhou, and Keith Ross. Randomized ensembled double q-learning: Learning fast without a model. In *International Conference on Learning Representations*, 2021.

[4] Karl Cobbe, Chris Hesse, Jacob Hilton, and John Schulman. Leveraging procedural generation to benchmark reinforcement learning. *arXiv preprint arXiv:1912.01588*, 2020.

[5] Will Dabney, Mark Rowland, Marc G. Bellemare, and Rémi Munos. Distributional reinforcement learning with quantile regression. In *AAAI Conference on Artificial Intelligence*, 2018.

[6] Carlton Downey and Simon Sanner. Temporal difference bayesian model averaging: A bayesian perspective on adapting lambda. In *International Conference on Machine Learning*, 2010.

[7] Scott Fujimoto, Herke Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *International Conference on Machine Learning*, 2018.

[8] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning*, 2018.

[9] Danijar Hafner. Benchmarking the spectrum of agent capabilities. *arXiv preprint arXiv:2109.06780*, 2022.

[10] Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023.

[11] Nicklas Hansen, Hao Su, and Xiaolong Wang. Td-mpc2: Scalable, robust world models for continuous control. *arXiv preprint arXiv:2310.16828*, 2024.

[12] Michael Janner, Justin Fu, Marvin Zhang, and Sergey Levine. When to trust your model: Model-based policy optimization. In *Advances in Neural Information Processing Systems*, 2019.

[13] Arsenii Kuznetsov, Pavel Shvechikov, Alexander Grishin, and Dmitry Vetrov. Controlling overestimation bias with truncated mixture of continuous distributional quantile critics. In *International Conference on Machine Learning*, 2020.

[14] Ian Osband, Charles Blundell, Alexander Pritzel, and Benjamin Van Roy. Deep exploration via bootstrapped dqn. In *Advances in Neural Information Processing Systems*, 2016.

[15] Adam Paszke, Sam Gross, Francisco Massa, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, 2019.

[16] Jing Peng and Ronald J. Williams. Incremental multi-step q-learning. *Machine Learning*, 22:283–290, 1996.

[17] John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous

control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2016.

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

[19] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012.

[20] Mark Towers, Ariel Kwiatkowski, Jordan Terry, et al. Gymnasium: A standard interface for reinforcement learning environments. *arXiv preprint arXiv:2407.17032*, 2024.

[21] Martha White and Adam White. A greedy approach to adapting the trace parameter for temporal difference learning. In *International Conference on Autonomous Agents and Multiagent Systems*, 2016.

[22] Zhongwen Xu, Hado van Hasselt, and David Silver. Meta-gradient reinforcement learning. In *Advances in Neural Information Processing Systems*, 2018.

[23] Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on Robot Learning*, 2020.