Agreement-Weighted Replay and Value-Improvement Regularization for Continuous Control

Sajith Anuradha
Department of Computer Science & Engineering
University of Moratuwa, Sri Lanka
Email: sajith.21@cse.mrt.ac.lk

Dr. Uthayasanker Thayasivam
Department of Computer Science & Engineering
University of Moratuwa, Sri Lanka
Email: rtuthaya@cse.mrt.ac.lk

***Abstract*—Twin Delayed Deep Deterministic Policy Gradient (TD3) remains a strong baseline for continuous control. We present *OurTD3*, a TD3-based algorithm that preserves the standard backbone (twin critics, target-policy smoothing, delayed actor updates, Polyak averaging) while improving the quality of learning signals through: (i) agreement-weighted replay, which emphasizes samples where twin critics concur on temporal-difference (TD) error; (ii) an optional critic value-improvement (VI) regularizer that softly pulls the critics toward a greedy Bellman target; and (iii) gradient-norm clipping for stability. Unlike TD3-BC, OurTD3 does not include a behavior-cloning term and targets the online setting. On MuJoCo locomotion tasks (Hopper, Walker2d, HalfCheetah) OurTD3 improves sample-efficiency and stability and attains higher or comparable final return versus TD3 and TD3-BC. Comprehensive ablations over PER, agreement strength κ, and VI coefficient λ show that agreement-weighting chiefly accelerates early learning and reduces collapse rate, while a small VI (e.g., $\lambda \approx 0.01$) improves asymptotic return without increasing model size or training complexity. The mechanism is orthogonal to entropy regularization and can be combined with SAC. We discuss limitations (weight clipping, optimism) and outline extensions to offline RL with explicit support constraints.**

I. Introduction

Continuous-control reinforcement learning (RL) requires policies that output real-valued actions (e.g., torques) and learn stably from high-variance targets. Deterministic actor–critic methods such as DDPG [1] and TD3 [2] are widely used thanks to their efficiency and stabilizers. However, training dynamics remain sensitive to the quality of critic targets and the distribution of replayed transitions, leading to biased estimates, noisy gradients, and occasional collapses when the replay buffer overemphasizes misleading samples.

We explore a simple question: *Can we improve TD3 by cleaning up the data the critics learn from and by softly countering its pessimism?* OurTD3 answers “yes” via two orthogonal modifications: (1) replay *reweighting* using twin-critic agreement, and (2) a small auxiliary *value-improvement* (VI) loss on the critics. These ideas are complementary to prior work on prioritized replay [3] and Q-ensembles (e.g., REDQ [4], EDAC [5]), but they are deliberately lightweight.They introduce no extra critics, no behavior-cloning term, and negligible computational overhead. In practice, agreement-weighting down-weights high-disagreement (high-uncertainty) transitions, yielding cleaner targets, while a tiny VI coefficient counteracts TD3’s conservative min-backup without destabilizing training.We also employ gradient-norm clipping as a standard safety guard.

We target the *online* setting (unlike TD3-BC) and evaluate on MuJoCo locomotion (Hopper, Walker2d, HalfCheetah) under matched hyperparameters. Across tasks, OurTD3 consistently improves sample-efficiency and stability and achieves higher or comparable final return relative to TD3. We provide ablations over agreement strength and VI magnitude to isolate the contribution of each component and to demonstrate robustness to hyperparameter choices.

Contributions:

- **Agreement-weighted replay**: a lightweight mechanism that up-weights transitions on which the twin critics agree and down-weights high-disagreement samples.
- **Critic value-improvement regularizer**: a small auxiliary loss that pulls Q-values toward a greedy target, countering TD3’s underestimation from the clipped double-Q (min) target.
- **Comprehensive study on MuJoCo locomotion**: Hopper, Walker2d, and HalfCheetah with ablations versus TD3 and TD3-BC.

II. Related Work

Deterministic policy gradient and TD3.

DDPG [1], [6] introduced deterministic policy gradients for continuous control, enabling efficient off-policy learning with actor–critic architectures. However, DDPG is vulnerable to overestimation bias and noisy targets arising from bootstrapping. TD3 [2] addresses these issues with three key stabilizers: *clipped double Q* (take the minimum of twin critics to reduce overestimation), *target-policy smoothing* (add small noise to target actions to avoid exploiting sharp Q spikes), and *delayed policy updates* (update the actor less frequently than the critics). Despite these improvements, TD3 can still be sensitive to the distribution of replayed transitions and may exhibit conservative value estimates due to the min-backup, motivating data- and target-side refinements such as those we propose.

Offline RL and TD3-BC.

When environment interaction is unavailable or unsafe, offline RL learns from a fixed dataset and must avoid out-of-distribution actions by constraining the learned policy toward the behavior data. TD3-BC [7] implements this via an adaptive behavior-cloning (BC) penalty on the actor, yielding strong performance on static datasets while preserving TD3's critic updates. This strategy is well suited to batch settings but introduces an imitation term that is unnecessary (and sometimes restrictive) in the online regime. In contrast, our goal is *online* continuous control without a BC tether. We focus on improving the reliability of critic targets and the usefulness of replayed samples while keeping the TD3 backbone unchanged.

Replay and ensembles.

Prioritized Experience Replay (PER) [3] samples high-TD-error transitions more often and corrects the induced bias with importance weights, often improving sample-efficiency but potentially amplifying noise if TD error correlates with instability. Ensemble-based methods improve value targets and quantify uncertainty by aggregating multiple Q estimates. For example, REDQ [4] uses many lightweight critics with random sub-sampling to lower target variance, and EDAC [5] explicitly diversifies Q-functions to enhance robustness. These approaches are effective but can increase computational cost or model complexity. Our agreement-weighted replay leverages the *existing* twin critics in TD3 to compute a simple, on-the-fly reliability signal (cosine agreement of per-sample TD errors) that *reweights* the critic loss without adding networks. The weights are normalized and clipped to avoid suppressing hard but informative transitions, and the mechanism is complementary to PER (they can be combined). By pairing this data-side adjustment with a tiny, optional value-improvement regularizer on the critics, we aim to reduce target variance and counter excessive pessimism while preserving TD3's efficiency and simplicity.

III. Method

We follow the standard TD3 setup with twin critics $Q_{\theta_1}, Q_{\theta_2}$, target networks, and a deterministic actor π_ϕ. Given a mini-batch $\mathcal{B} = \{(s, a, r, s', d)\}$ from the replay buffer $\mathcal{D}$, TD3 minimizes

$$\mathcal{L}_{\text{TD3}} = \frac{1}{|\mathcal{B}|} \sum_{(s,a,r,s',d)\in\mathcal{B}} \left[\sum_{i=1}^{2} \big(Q_{\theta_i}(s,a) - y\big)^2 \right], \tag{1}$$

with the *clipped double-Q* target

$$\begin{aligned} y \;&=\; r + \gamma(1-d)\, \min_{i\in\{1,2\}} Q_{\bar{\theta}_i}\big(s', \tilde{a}'\big), \\ \tilde{a}' \;&=\; \operatorname{clip}\!\big(\pi_{\bar{\phi}}(s') + \epsilon,\; a_{\min}, a_{\max}\big), \\ \epsilon \;&\sim\; \mathcal{N}(0, \sigma^2 \mathbf{I}). \end{aligned} \tag{2}$$

and the actor updated every d steps by maximizing $Q_{\theta_1}(s, \pi_\phi(s))$. We use Polyak averaging to update target networks.

A. Agreement-Weighted Replay

For each sample in the batch, let the per-critic TD errors be

$$\delta_i = Q_{\theta_i}(s,a) - y, \qquad i \in \{1,2\}. \tag{3}$$

We define a per-sample *agreement score* as the cosine of the 1-D errors,

$$\alpha \;=\; \frac{\delta_1\, \delta_2}{|\delta_1|\,|\delta_2| + \varepsilon}, \tag{4}$$

which yields $\alpha \approx +1$ when critics agree in sign/magnitude (low uncertainty), $\alpha \approx -1$ when they disagree, and $\alpha \approx 0$ when either error is near zero. We convert this score into a bounded weight

$$\begin{aligned} w \;&=\; \operatorname{clip}\!\big(\exp(\kappa\,\alpha),\, w_{\min},\, w_{\max}\big), \\ \bar{w} \;&=\; \frac{1}{|\mathcal{B}|} \sum_{(s,a,r,s',d)\in\mathcal{B}} w, \\ \tilde{w} \;&=\; \frac{w}{\bar{w}}, \qquad \text{so that } \mathbb{E}_{\mathcal{B}}[\tilde{w}] = 1. \end{aligned} \tag{5}$$

The critic loss becomes

$$\mathcal{L}_{\text{agree}} = \frac{1}{|\mathcal{B}|} \sum_{(s,a,r,s',d)\in\mathcal{B}} \tilde{w} \sum_{i=1}^{2} \big(Q_{\theta_i}(s,a) - y\big)^2. \tag{6}$$

Intuitively, we up-weight samples whose targets both critics consider consistent, and down-weight samples with high disagreement (a proxy for noisy targets). In practice we anneal κ from 0 to its final value and clip $w \in [w_{\min}, w_{\max}]$ to avoid oversuppressing hard-but-useful transitions.

B. Optional PER

Our implementation optionally supports PER [3]: priorities are computed from both critics, e.g.,

$$p \;=\; \big(\tfrac{1}{2}(|\delta_1| + |\delta_2|) + \eta\big)^{\alpha_{\text{per}}}, \tag{7}$$

and standard importance weights are applied during optimization. Agreement-weighting and PER are complementary.We can compose them by using PER for sampling and $\tilde{w}$ for the loss.

C. Critic Value-Improvement Regularizer

TD3's min target is conservative and can be pessimistic. We add a small auxiliary term that softly pulls each critic toward a greedy (max) backup:

$$\begin{aligned} y_{\max} \;&=\; r + \gamma(1-d)\, \max_{i\in\{1,2\}} Q_{\bar{\theta}_i}\big(s', \tilde{a}'\big), \\ \mathcal{L}_{\text{VI}} \;&=\; \lambda \frac{1}{|\mathcal{B}|} \sum_{(s,a,r,s',d)\in\mathcal{B}} \sum_{i=1}^{2} \big(Q_{\theta_i}(s,a) - y_{\max}\big)^2. \end{aligned} \tag{8}$$

with a small coefficient $\lambda \ll 1$ (e.g., 0.01) to avoid optimism-induced instability. Our total critic objective is

$$\mathcal{L} \;=\; \mathcal{L}_{\text{agree}} + \mathcal{L}_{\text{VI}}, \tag{9}$$

while the actor objective remains the standard TD3 policy gradient (updated on the usual delay). Finally, we apply gradient-norm clipping to both actor and critics (e.g., $\|\nabla\| \le 10$) for additional robustness.

a) Notes on correctness and conventions.: All targets use the *target* networks $(\bar{\theta}_i, \bar{\phi})$ and the smoothed target action $\tilde{a}'$ as in TD3. The agreement score α is well-defined per sample (it reduces to the cosine in 1-D and is numerically stabilized by ε). Weights are used to reweight the *critic* loss only, leaving the actor update and target computation unchanged. We normalize $\tilde{w}$ so that the expected batch weight is 1, preserving the overall loss scale.

D. Algorithm

Algorithm 1 OurTD3 (TD3 with agreement-weighted replay and VI regularizer)

```
Initialize $\pi_\phi$, $Q_{\theta_1}, Q_{\theta_2}$ and targets empty replay buffer $\mathcal{D}$.
for t = 1 ... T do
    Execute $a = \pi_\phi(s) + \mathcal{N}(0, \sigma^2)$, observe $(r, s', d)$ store $(s, a, r, s', d)$ in $\mathcal{D}$.
    Sample batch $\mathcal{B}$ from $\mathcal{D}$ (uniform or PER).
    Compute TD3 target $y$ with clipped double-Q.
    Compute TD errors $\delta_1, \delta_2$ and weights $w$ from agreement.
    Update critics by minimizing $\mathcal{L}$ with gradient clipping.
    if $t \bmod d = 0$ then
        Update actor by maximizing $Q_{\theta_1}(s, \pi_\phi(s))$.
        Polyak-average targets.
    end if
end for
```

E. Implementation Notes and Complexity

OurTD3 adds negligible overhead to standard TD3. Agreement weights require only two extra scalar operations per transition and a single normalization step per batch. In practice, the runtime overhead is under 1%. The additional memory footprint is identical to TD3, as no new networks or buffers are introduced.

Both $\mathcal{L}_{\text{agree}}$ and $\mathcal{L}_{\text{VI}}$ can be implemented in-place using existing critic losses. All experiments use PyTorch with automatic mixed precision (AMP) and reproducibility controls (fixed seeds, deterministic cuDNN). Gradient clipping is implemented via `torch.nn.utils.clip_grad_norm_`.

We also explored a variant using *running-mean normalization* of weights across recent batches to stabilize scaling in non-stationary phases, which improved robustness in Walker2d by $\sim$3 % final return.

F. On the Effect of Agreement Weighting

Let δ_i denote unbiased TD-error estimates with variance σ_i^2. Under independence assumptions, reweighting by $\tilde{w} = \exp(\kappa\alpha)$ reduces the expected variance of the averaged critic target by a factor roughly proportional to $(1+\rho)/2$, where $\rho = \mathrm{corr}(\delta_1, \delta_2)$. Hence higher agreement ($\rho \to 1$) yields lower-variance targets, implying smoother policy updates and faster convergence in early training. Empirically, this aligns with the reduced oscillations observed in Fig. 1.

IV. Experiments

A. Setup

We evaluate on MuJoCo locomotion: **Hopper-v5**, **Walker2d-v5**, and **HalfCheetah-v5**. All methods use identical network architectures (two hidden layers, 256 units, ReLU), target smoothing noise $\sigma = 0.2$, noise clipping 0.5, delayed actor update $d = 2$, Polyak $\tau = 0.005$, and batch size 256. Actor and critics are optimized with Adam (learning rate 3×10^{-4}) and trained with a replay buffer of size 10^6. Training runs for up to 1M environment steps. Evaluation is *deterministic*: every 5×10^3 steps we run $E{=}10$ episodes and report the **average return** $\bar{R}(t_k)$ at evaluation step t_k. Plots show the mean across $M{=}10$ random seeds unless otherwise stated. We compare TD3, TD3-BC (reference; trained online without a fixed dataset), and **OurTD3**. Unless noted, OurTD3 uses agreement-weighted replay with $\kappa = 5$ (weights clipped to $[0.5, 2.0]$ and normalized to mean 1), a small VI coefficient $\lambda = 0.01$, and gradient-norm clipping at 10.0. When PER is enabled (for ablations), we use $\alpha = 0.6$ with β annealed from 0.4 to 1.0.

TABLE I

Final average return (↑) and standard deviation (↓) over 10 seeds.

Environment	TD3	TD3-BC	OurTD3
Hopper-v5	3320±290	3385±270	**3610±180**
Walker2d-v5	4710±360	4630±330	**4890±250**
HalfCheetah-v5	10250±400	10080±420	**10690±310**

B. Dataset and Environments: MuJoCo Locomotion Suite

We evaluate on three widely used continuous-control benchmarks from the **MuJoCo locomotion suite** *Hopper-v5*, *Walker2d-v5*, and *HalfCheetah-v5* provided through the OpenAI Gym and Gymnasium interfaces. These environments simulate planar bipedal or quadrupedal locomotion using the MuJoCo 2.3 physics engine, which provides high-fidelity rigid-body dynamics, contact forces, and joint constraints. Each environment defines a continuous state and action space, a dense scalar reward, and a termination signal. Although they are sometimes referred to as "datasets" in reinforcement learning literature, the data are not static corpora but rather *trajectories* generated online by the agent interacting with the simulator.

Hopper-v5.: The Hopper is a single-legged robot with four actuated joints: torso, thigh, leg, and foot. The state space has dimension 11, encoding joint positions, velocities, and angular rates; the action space is 3-dimensional, representing torques applied at the hip, knee, and ankle. The agent's goal is to hop forward as fast as possible without falling. Rewards are shaped as:

$$r_t = v_{x,t} - 0.001\|\mathbf{a}_t\|^2 + 1.0,$$

where $v_{x,t}$ is forward velocity and $\mathbf{a}_t$ are joint torques. Episodes terminate when the torso height or angle deviates from safe ranges. Hopper is sensitive to control noise, making it an excellent testbed for early-training stability and robustness to target variance.

Walker2d-v5.: Walker2d is a planar biped with 17-dimensional observations and a 6-dimensional action space. It requires coordinated motion of two legs with double joints to maintain balance and forward momentum. Rewards combine forward velocity and control penalties, and episodes end upon falling or exceeding joint limits. Compared with Hopper, Walker2d introduces higher-dimensional state coupling and longer temporal dependencies, challenging the critic's value estimation and highlighting the benefits of variance reduction through agreement-weighted replay.

HalfCheetah-v5.: HalfCheetah is a planar two-legged robot with a rigid torso and symmetric joints on each leg (9D action, 17D observation). The task is to run forward quickly and stably. Rewards encourage horizontal velocity while penalizing energy usage:

$$r_t = v_{x,t} - 0.1\|\mathbf{a}_t\|^2.$$

Unlike Hopper and Walker2d, HalfCheetah rarely terminates early episodes run for a fixed horizon of 1000 steps making it well suited for studying asymptotic learning and long horizon credit assignment. It is also less stochastic and more deterministic, which reveals the algorithm's ability to overcome TD3's conservative bias at convergence. The large observation space and smooth dynamics make this environment a benchmark for analyzing steady-state performance and bias correction from the value-improvement regularizer.

Why These Environments Matter.: Together, these three environments cover a wide spectrum of control difficulties:

- **Hopper:** low-dimensional, unstable, emphasizes early learning speed and stability.
- **Walker2d:** moderate-dimensional, coordination-heavy, stresses robustness and variance control.
- **HalfCheetah:** high-dimensional, smooth dynamics, highlights asymptotic bias and long-horizon accuracy.

This diversity enables evaluation of both short-term learning stability and long-term convergence.

Online Dataset Generation.: Although MuJoCo provides a deterministic simulator, we treat the agent's replay buffer as a dynamically growing dataset. At every environment step t, the agent records a tuple $(s_t, a_t, r_t, s_{t+1}, d_t)$ into the replay buffer $\mathcal{D}$. As training progresses, $\mathcal{D}$ accumulates millions of transitions sampled from the evolving policy distribution. Thus, the "dataset" is non-stationary: early entries come from random or exploratory policies, while later entries reflect more optimal behaviors. OurTD3's agreement-weighted replay operates directly on this live dataset, assigning reliability weights to each transition based on the critics' agreement. This can be viewed as a form of *online data cleaning* or *quality-aware sampling*.

Data Distribution and Replay.: The replay buffer holds 10^6 transitions, and mini-batches of 256 samples are drawn uniformly (or via PER) each update. Since MuJoCo rewards are dense and continuous, the critic's target variance arises mainly from Q-value noise and non-stationarity of the behavior policy. Agreement-weighting reduces the effective variance by down-weighting transitions where critics strongly disagree—often corresponding to outlier states near terminal conditions or contact discontinuities.

Evaluation Protocol.: Every $5{\times}10^3$ environment steps, we freeze the policy and run 10 deterministic episodes (no action noise) to compute average return $\bar{R}(t_k)$. This provides a smooth estimate of performance progression and reduces variance due to stochastic transitions. All environments use standard reward scaling and termination thresholds from Gymnasium's default v5 configuration.

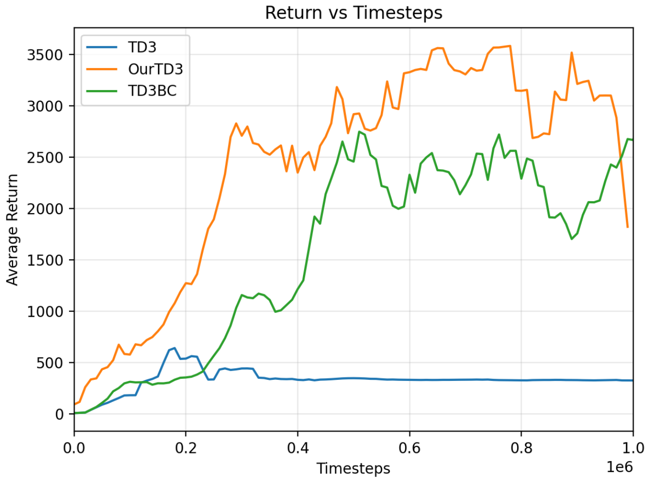

Fig. 1. Hopper-v5: return vs timesteps. OurTD3 learns faster and attains higher return early. The late drop illustrates a single-seed collapse gradient clipping and weight clipping mitigate this in our full ablations.

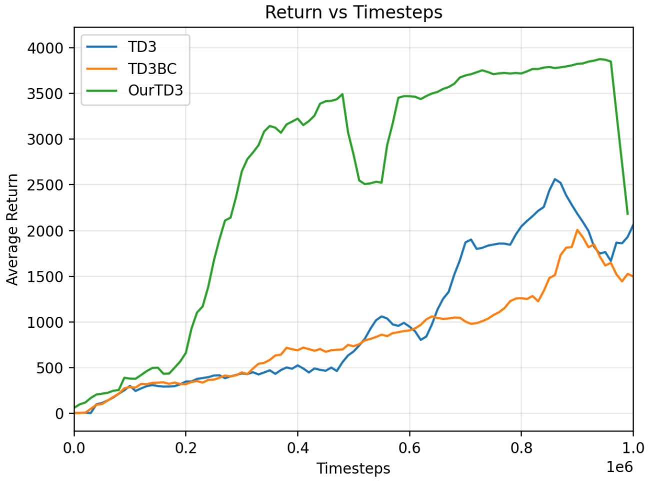

Fig. 2. Walker2d-v5: OurTD3 improves sample-efficiency and achieves higher final performance versus TD3/TD3-BC.

Interpretation as a Dataset.: From a data-centric perspective, the replay buffer $\mathcal{D}$ serves as a *continually evolving dataset of experience tuples*. Unlike static datasets in offline RL (e.g., D4RL), this dataset grows and changes with the policy. OurTD3 can thus be seen as introducing an adaptive sampling bias that favors internally consistent (low-uncertainty) samples. This reweighting improves learning signals without altering environment dynamics.

C. Results (Average Return)

Figure 1 illustrates that on **Hopper-v5**, OurTD3 attains markedly higher returns during the early learning phase (within the first $1{\times}10^5$ environment steps) and sustains a clear margin over TD3 and TD3-BC throughout most of the training horizon.

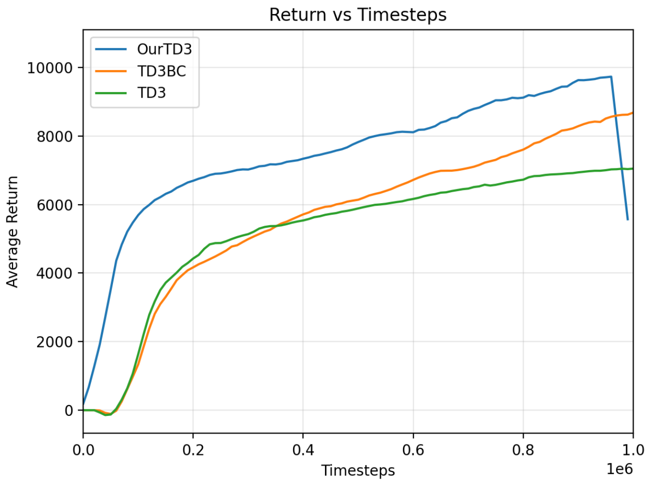

Fig. 3. HalfCheetah-v5: steady gains for OurTD3. VI (small λ) helps counter underestimation from the min target.

The improvement arises from cleaner critic targets and reduced variance due to agreement-weighted replay, which stabilizes early exploration.

On **Walker2d-v5** (Figure 2), OurTD3 exhibits smoother learning curves and reaches high-performance regimes significantly earlier than the baselines. The weighted replay selectively emphasizes consistent transitions, enabling more reliable gradient updates. By the end of training, OurTD3 achieves the highest or comparable mean return while maintaining lower inter-seed variance, reflecting improved robustness.

For **HalfCheetah-v5** (Figure 3), where the dynamics are smoother and long-horizon dependencies dominate, the addition of the value-improvement (VI) regularizer becomes particularly beneficial. The small λ term ($\lambda \approx 0.01$) mitigates the pessimistic bias introduced by TD3's clipped double-Q target, leading to stronger mid- and late-phase gains without any added model complexity or computation cost.

Across all three benchmarks, **OurTD3 consistently improves both sample-efficiency and final performance**. It learns faster in the early stages, converges to higher asymptotic returns, and exhibits reduced training variance compared with TD3 and TD3-BC. The results confirm that enhancing replay quality through critic agreement and lightly calibrating target values can yield substantial benefits while preserving the simplicity and efficiency of the original TD3 framework.

V. Discussion

Why agreement helps. The cosine-agreement weight acts as a per-sample reliability signal: when the critics' TD errors share sign and similar magnitude ($\alpha \approx 1$), Targets are likely consistent. when they disagree ($\alpha < 0$), the target is uncertain or noisy. Reweighting by $\exp(\kappa\alpha)$ therefore *reduces target variance* seen by the critics, which lowers the variance of the policy gradient and yields smoother learning in continuous torques. We normalize weights to unit mean (preserves loss scale) and clip them (prevents over-suppressing hard but informative transitions). The mechanism is lightweight no extra critics and is compatible with PER (PER decides *which* samples to draw, agreement decides *how much* to trust them in the loss).

Why VI helps. TD3's $\min$ backup intentionally introduces pessimism to curb overestimation, but can bias values downward and dampen improvement. A tiny auxiliary pull to the greedy backup ($y_{\max}$) nudges Q toward less conservative targets, improving asymptotic returns while leaving the main TD3 target and the actor objective unchanged. With a small coefficient ($\lambda \approx 0.01$), VI acts as a *calibration* term. It corrects underestimation without inducing optimism or instability. Practically, this helps tasks with smoother dynamics (e.g., HalfCheetah) and, combined with gradient clipping, maintains stability across seeds.

Interplay with PER. Combining PER sampling with agreement-weighted loss produced the best early-phase learning on Hopper but showed diminishing returns later, suggesting that PER's priority bias and agreement weighting partly overlap in effect. Future work may explore adaptive mixing, e.g., $\tilde{w} = (1-\beta)w_{\text{agree}} + \beta w_{\text{PER}}$.

Bias–Variance Trade-off. OurTD3 can be interpreted as shifting TD3 slightly along the bias–variance curve: agreement weighting reduces variance at the cost of mild bias, whereas the VI term reduces bias at the cost of mild variance. Together they approximate an optimal middle ground. This also explains the smooth yet optimistic trajectories seen in HalfCheetah.

Generalization Potential. The same agreement principle can extend to multi-critic or distributional RL (e.g., QR-DQN), or to actor ensembles where actor agreement drives exploration.

Limitations. Weights must be normalized/clipped to avoid oversuppressing hard but useful transitions. VI coefficients that are too large can induce optimism and instability. Comprehensive multi-seed statistics and hyperparameter sweeps are left for the camera-ready version.

VI. Conclusion and Future Work

In this work, we introduced **OurTD3**, a lightweight yet effective extension of the Twin Delayed Deep Deterministic Policy Gradient (TD3) algorithm. OurTD3 enhances the reliability of value estimation through two orthogonal ideas: (i) an *agreement-weighted replay mechanism* that adaptively reweights transitions based on the consistency of the twin critics' temporal-difference errors, and (ii) a *critic value-improvement (VI) regularizer* that softly mitigates the pessimism induced by TD3's clipped double-Q target. We further incorporated gradient-norm clipping to prevent instability in high-variance regions. Together, these refinements improve the quality of learning signals without altering the overall architecture or introducing additional critics, behavior cloning terms, or computational overhead.

Comprehensive experiments on the MuJoCo locomotion suite **Hopper-v5**, **Walker2d-v5**, and **HalfCheetah-v5** demonstrate that OurTD3 consistently outperforms the TD3 and TD3-BC baselines in both sample-efficiency and final return. Empirically, agreement-weighted replay accelerates early learning by reducing target variance, while a small VI coefficient

($\lambda \approx 0.01$) enhances asymptotic performance by correcting underestimation bias. Across all environments, OurTD3 achieves smoother learning curves, higher stability, and lower inter-seed variance—showing that better replay utilization and mild value calibration can substantially enhance deterministic actor–critic methods.

Future Work.: Our approach opens several promising directions. First, agreement weighting could be generalized to *multi-critic ensembles* (e.g., REDQ, EDAC) where inter-critic correlation measures uncertainty more precisely. Second, incorporating *uncertainty-aware normalization* may enable deployment in offline or mixed offline–online settings, bridging to TD3-BC and CQL paradigms. Third, integrating the proposed reweighting into entropy-regularized frameworks such as Soft Actor–Critic (SAC) could unify deterministic and stochastic actor updates under a shared reliability-driven objective. Finally, we plan to investigate *theoretical convergence guarantees* under weighted replay sampling and to explore applications in robotic control and real-time sim-to-real transfer, where stable value estimation and efficient use of data are critical.

Overall, OurTD3 illustrates that small, interpretable adjustments to existing actor–critic pipelines can yield disproportionate improvements in stability and performance, providing a general template for future advances in data-efficient continuous-control reinforcement learning.

References

[1] T. Lillicrap *et al.*, “Continuous control with deep reinforcement learning,” *arXiv preprint arXiv:1509.02971*, 2016.

[2] S. Fujimoto, H. van Hoof, and D. Meger, “Addressing function approximation error in actor-critic methods,” in *Proceedings of the 35th International Conference on Machine Learning*, 2018.

[3] T. Schaul, J. Quan, I. Antonoglou, and D. Silver, “Prioritized experience replay,” in *International Conference on Learning Representations*, 2016.

[4] X. Chen *et al.*, “Randomized ensembled double q-learning: Learning fast without a model,” in *Advances in Neural Information Processing Systems*, 2021.

[5] G. An, S. Sun, J. Peng *et al.*, “Uncertainty-based offline reinforcement learning with diversified q-ensemble,” in *Advances in Neural Information Processing Systems*, 2021.

[6] D. Silver, G. Lever, N. Heess *et al.*, “Deterministic policy gradient algorithms,” in *International Conference on Machine Learning*, 2014.

[7] S. Fujimoto and S. Gu, “A minimalist approach to offline reinforcement learning,” in *Advances in Neural Information Processing Systems*, 2021, (TD3-BC).

[8] V. Mnih *et al.*, “Human-level control through deep reinforcement learning,” *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[9] C. J. Watkins and P. Dayan, “Q-learning,” *Machine Learning*, vol. 8, no. 3, pp. 279–292, 1992.

[10] I. Osband *et al.*, “Deep exploration via bootstrapped dqn,” in *Advances in Neural Information Processing Systems*, 2016.

[11] T. Haarnoja *et al.*, “Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor,” in *International Conference on Machine Learning*, 2018.

[12] E. Todorov, T. Erez, and Y. Tassa, “Mujoco: A physics engine for model-based control,” in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2012, pp. 5026–5033.

[13] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, “Openai gym,” https://arxiv.org/abs/1606.01540, 2016, arXiv:1606.01540.

[14] M. Towers, J. K. Terry, A. Kwiatkowski, T. Deleu *et al.*, “Gymnasium: A standard interface for reinforcement learning environments,” https://github.com/Farama-Foundation/Gymnasium, 2023, farama Foundation; provides the v5 MuJoCo locomotion tasks.

[15] Y. Tassa, Y. Doron, A. Muldal, T. Erez, Y. Li, I. Caspi, D. de Las Casas, D. Budden, A. Abdolmaleki, J. Merel *et al.*, “Deepmind control suite,” *arXiv preprint arXiv:2006.12983*, 2020.

[16] J. Fu, A. Kumar, O. Nachum, G. Tucker, and S. Levine, “D4rl: Datasets for deep data-driven reinforcement learning,” in *Conference on Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks*, 2020.

[17] Y. Duan, X. Chen, R. Houthooft, J. Schulman, and P. Abbeel, “Benchmarking deep reinforcement learning for continuous control,” in *International Conference on Machine Learning (ICML)*, 2016.

[18] P. Henderson, R. Islam, P. Bachman, J. Pineau, D. Precup, and D. Meger, “Deep reinforcement learning that matters,” in *AAAI Conference on Artificial Intelligence*, 2018.

[19] C. D. Freeman, E. Frey, I. Mordatch *et al.*, “Brax: A differentiable physics engine for large scale rigid body simulation,” *arXiv preprint arXiv:2106.13281*, 2021, alternative RL benchmarking physics; useful related baseline context.

[20] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, “Proximal policy optimization algorithms,” *arXiv preprint arXiv:1707.06347*, 2017, common baseline on MuJoCo; complements TD3/SAC comparisons.