Advanced Machine Learning -Progress Report

Conductor: Dr. Uthayasanker Thayasivam

Assessment Type: Individual Research Paper

Academic Year: 2024/2025

Student Id:210043V

Project Id-RL003

1. Literature Review (TD3 & TD3-based extensions)

Problem & motivation

Continuous-control reinforcement learning (RL) agents operate in real-valued action spaces (e.g., torques for locomotion), where small value-estimation errors can compound quickly because training relies on bootstrapped targets. Early actor–critic methods like DDPG learn a deterministic policy and a critic in tandem, but they suffer from a systematic overestimation bias.When the critic is used both to choose and evaluate actions, approximation noise gets amplified, inflating Q-values and producing unstable policy updates. Straightforward fixes from discrete control such as Double DQN do not directly carry over to deterministic actor–critic, because the online and target critics are highly correlated and the policy changes during learning, so the “decoupled max” idea does not sufficiently break the bias/variance loop. In short, the field needed a method that directly reduces overestimation and target variance in continuous action settings without sacrificing the sample-efficiency and simplicity that made DDPG popular.

TD3: core idea and contributions

Twin Delayed Deep Deterministic Policy Gradient (TD3) answers that need with three mutually supportive mechanisms that target the above failure modes while keeping the backbone simple:

1. Clipped Double-Q targets. Maintain two independent critics and compute the backup using the minimum of their target estimates. This prefers mild underestimation over harmful overestimation, reducing maximization bias without changing the policy class.

2. Delayed policy updates. Update the actor (and target networks) less frequently than the critics. This two-timescale schedule lets the critics settle before the policy chases moving targets, lowering variance in the policy gradient.

3. Target-policy smoothing. Add small clipped Gaussian noise to the target action when computing the bootstrap. This acts like an Expected-SARSA-style regularizer for continuous actions, discouraging the critic from developing brittle peaks that the actor can exploit.

TD3-based variants relevant to this project

Value-Improved Actor-Critic on TD3 (VI-TD3)

A recent line of work adds a lightweight value-improvement (VI) operator to the critic update while leaving the TD3 actor update unchanged. Intuitively, TD3 already discourages overestimation with a conservative backup. The VI term then tightens the critic further by encouraging a greedier value-improvement step inside the critic loss. Because this change is loss-level and does not alter the policy objective, network shapes, or stabilizers, it slots neatly into TD3 and typically yields better sample-efficiency (learning more from the same data) with negligible extra compute.

Replay upgrades for TD3 (vMFER or PER)

Orthogonal to loss-level changes are data-level upgrades that improve which transitions TD3 learns from.

- vMFER replay computes a per-transition weight based on critic-gradient agreement. transitions where twin critics agree on improvement direction are up-weighted, while highly discordant ones are down-weighted (or delayed). This reduces noisy updates

and often accelerates early learning without touching the architecture or targets.

- Prioritized Experience Replay (PER) samples transitions with probability increasing in their TD-error, then uses importance weights to correct bias. PER is easy to bolt onto TD3 and tends to improve sample-efficiency, especially on locomotion tasks where a small subset of transitions is particularly informative. A small mix of short n-step targets can further help credit assignment while remaining a minor change.

Both upgrades are non-intrusive.They leave the TD3 actor, twin critics, min-backup, and smoothing intact only the sampling/weighting of replay batches changes.

Gap & planned contribution

TD3 already addresses the core bias/variance pathologies of deterministic actor–critic, but it does not adapt its stabilizers during training, nor does it explicitly manage which data is most trustworthy at each step. Recent work shows that a loss-level refinement (VI-TD3) and data-level selection (vMFER or PER) each help separately. My project proposes to

(1) reproduce a faithful TD3 baseline under the standard protocol.

(2) evaluate VI-TD3 as a minimal critic-side enhancement (Hyperparameter tuning)

(3) evaluate one replay upgrade (vMFER or PER) as a minimal data-side enhancement.

(4) test the combined method (VI-TD3 + Replay) to examine additivity.

The hypothesis is that sharper value learning (via VI) and more informative batches (via replay weighting/prioritization) act on different levers, yielding higher sample-efficiency and equal or better final returns at similar compute. All comparisons will follow the TD3 evaluation discipline (fixed budget, periodic noise-free evals, multiple seeds) and will report AUC, final return, variance, and throughput, so improvements are quantified, fair, and reproducible.

2. Methodology Outline

M1:TD3 Baseline + Lightweight Tuning

Motivation

I first build a plain, reference TD3 to (a) learn the end-to-end pipeline hands-on, (b) create a fair yardstick for all comparisons, and (c) squeeze safe, reproducible gains using basics. This gives experience with robust training practice and ensures any improvements are due to our choices, not hidden setup differences.

Hyperparameter Optimization

On top of this faithful baseline, I will pursue a lightweight hyperparameter-optimization pass aimed at safe, course-friendly improvements rather than wholesale algorithm changes. Concretely, I vary only the knobs that affect optimization and stability while leaving the architecture intact. This includes selecting the optimizer family, introducing schedules for the actor and critic learning rates, adding global gradient-norm clipping, adjusting batch size, scheduling the target-policy smoothing parameters, and using a simple schedule for the actor-update delay. I will also sweep the Polyak averaging coefficient for target networks, test a conservative update-to-data ratio for critic updates per environment step, and taper exploration noise over the course of training. These changes are intentionally modest and orthogonal, designed to squeeze performance from the same model without creating a new variant.

M2: VI-TD3 + Replay Upgrade (vMFER or PER)- Combo

Overview & motivation

I will keep a faithful TD3 backbone (twin critics with min backup, delayed actor updates, target-policy smoothing, Polyak targets) and stack two orthogonal enhancements .As I found there are two variant,

1. VI-TD3 (loss-level)

   add a lightweight value-improvement (VI) term inside the critic update to sharpen value estimation without touching the actor or architecture.

2. Replay upgrade (data-level)

improve which data TD3 learns from using either vMFER (agreement-weighted sampling/weighting) or PER (TD-error-based prioritization with bias correction).

Because one acts on the update rule and the other on data selection/weighting, they combine cleanly and are expected to yield complementary gains in sample-efficiency and stability.

Components

- VI-TD3 (critic-side enhancement).

  Augment the TD3 critic loss with a small VI term that nudges Q-estimates toward a greedier improvement operator. Actor objective, delay, targets, and networks remain unchanged.

- vMFER replay (reliability-weighted data).

  Estimate a per-transition weight from critic-gradient agreement and use it to reweight critic (and optionally actor) updates; refresh weights periodically and cap extremes.
  or

- PER replay (informativeness-weighted data).

  Sample transitions with probability increasing in TD-error and use importance weights to correct bias; optionally pair with short n-step targets for better credit assignment.

How I integrate both into the TD3 loop

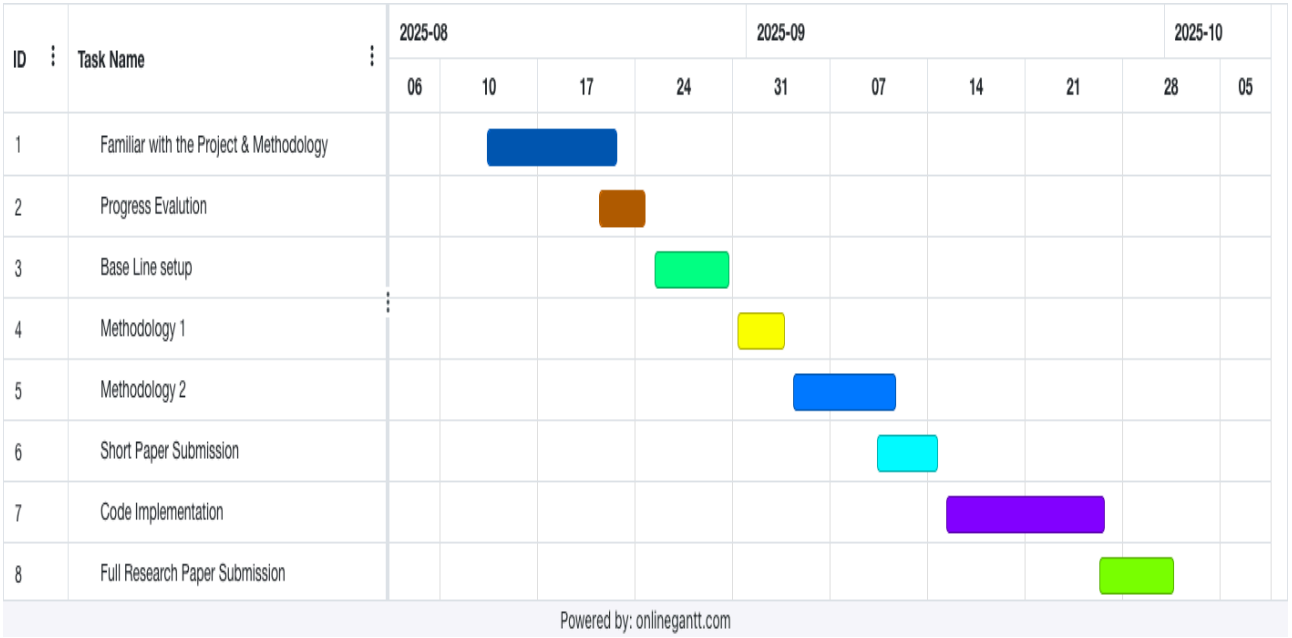
1. Batching: Draw a minibatch from replay.
   - PER: sample by priority and carry importance weights.
   - vMFER: compute normalized per-sample weights from critic-agreement; refresh on a schedule.
2. Targets: Compute the standard TD3 target (twin-critic min + target-policy smoothing).
3. Critic update: Apply the TD3 critic loss plus the VI term; multiply by replay weights (PER or vMFER).

4. Actor update (delayed): Update the actor as in TD3. Optionally apply replay weights for vMFER (keeping it normalized and clipped).

5. Target networks: Maintain Polyak averaging and keep TD3's actor-delay and smoothing intact.

6. Logging: Track TD-error statistics, twin-critic disagreement, gradient norms, and policy-change magnitude to explain effects.

3.Timeline



References

1. Fujimoto, S., van Hoof, H., & Meger, D. (2018). Addressing Function Approximation Error in Actor-Critic Methods. arXiv:1802.09477.

2. Oren, G., et al. (2024). Value-Improved Actor-Critic (includes a VI-TD3 instantiation). arXiv preprint.
3. Fujimoto, S., & Meger, D. (2021). A Minimalist Approach to Offline Reinforcement Learning (TD3+BC). NeurIPS 2021.
4. Zhu, D., et al. (2024). vMFER: Von Mises–Fisher Experience Resampling for RL. IJCAI 2024.
5. Sheikh, S., Phielipp, M., & Boloni, L. (2022). Maximizing Ensemble Diversity in Deep RL (MED-RL). ICLR 2022.