# Sample-Efficient Off-Policy Reinforcement Learning for High-Frequency Control: A Comparative Study in TrackMania

**Allyne Zhang**
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA 15213
allynez@cs.cmu.edu

**Aspen Chen**
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA 15213
sipengch@cs.cmu.edu

**Tyler Ho**
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA 15213
tylerho@cs.cmu.edu

## Abstract

High-frequency control systems require algorithms that maximize learning from expensive, non-parallelizable data collection. We evaluate three off-policy RL algorithms (SAC, TQC, and REDQ) in TrackMania, a racing simulator with real-time constraints mirroring physical robotics. Our results reveal that algorithm performance is highly environment-dependent: TQC achieves 20% improvement over SAC on fullspeed tracks through conservative value estimation, while SAC dominates on technical tracks where TQC's truncation proves overly conservative. REDQ fails entirely in this visual control domain, highlighting the gap between theoretical sample efficiency and practical performance. These findings underscore the importance of empirical evaluation across diverse environments and suggest that no single algorithm universally dominates in sample-constrained, high-frequency control.

## 1 Introduction

### 1.1 Problem and Motivation

Learning control policies for autonomous systems faces a fundamental bottleneck: data collection is expensive while training computation is abundant. High-fidelity physics simulators cannot always be parallelized, and real-world robotics data must be collected sequentially in real-time. This motivates off-policy reinforcement learning with high UTD ratios that perform many gradient updates per collected transition. However, aggressive experience reuse introduces challenges: value overestimation, distribution shift from stale replay data, and training instability.

We study these challenges in TrackMania, a racing game that cannot be parallelized on one computer due to the demanding graphics of the game. This constraint mirrors real robotics: collecting 10 hours of data requires 10 hours of wall-clock time. TrackMania also provides realistic dynamics (traction limits, surface friction variation) and precision requirements analogous to autonomous driving. We ask which of distributional RL (TQC), ensemble methods (REDQ), or standard actor-critic (SAC) enable stable, sample-efficient learning under these constraints.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

### 1.2 Contributions

- Systematic comparison of TQC, REDQ, and SAC in high-frequency vision-based control with real-time data constraints
- Extensions to the open-source tmrl framework:
  - Implementation of TQC with distributional critics and quantile truncation
  - Extension of REDQ to support image-based observations with CNN encoders
  - Integration of Impoola-CNN [12], a ResNet-inspired encoder with global average pooling
  - Integration of frozen DINOv3 [11] vision transformer backbones with precomputed feature caching for efficient replay buffer storage

## 2 Related Work

### 2.1 Off-Policy RL for Continuous Control

Off-policy reinforcement learning methods learn from replay buffers populated by past interactions, enabling far more updates per sample than on-policy algorithms [10]. DDPG first demonstrated that deterministic actor–critic methods can solve a wide range of continuous-control tasks, including end-to-end learning from pixels [9]. TD3 reduces overestimation by using twin critics and delayed policy updates [4], and SAC further improves robustness by maximizing entropy while remaining off-policy [5]. These methods, however, are typically tuned for modest update-to-data (UTD) ratios and low-dimensional MuJoCo-style benchmarks, leaving open how they behave under high UTD in high-frequency, vision-based domains like TrackMania.

### 2.2 Distributional and Ensemble Methods for Sample Efficiency

Recent work has focused on improving sample efficiency and stability by reshaping the critic. TQC extends SAC with distributional critics and truncation of the highest quantiles to finely control overestimation bias, achieving strong results on continuous-control benchmarks [7]. REDQ instead uses a large ensemble of critics with randomized subsampling in the target to support very high UTD ratios while controlling bias and variance [1]. Both methods were primarily evaluated on state-based MuJoCo tasks. Our project aims to examine how these distributional and ensemble strategies transfer to a high-frequency, visual control setting and reveals environment-dependent behavior and outright failure modes.

### 2.3 Offline RL and Conservative Value Estimation

Offline RL studies learning policies from fixed datasets without further environment interaction, motivated by domains where data collection is expensive or unsafe. Benchmarks such as D4RL provide standardized offline datasets across locomotion, navigation, and driving [3], and conservative methods like CQL modify the Bellman objective to penalize Q-values on out-of-distribution actions and reduce extrapolation error [6]. Our setting is *semi-offline*: data is gathered sequentially under strict real-time constraints and then heavily reused via high-UTD off-policy updates. Rather than introducing a new conservative objective, we ask how existing off-policy methods (SAC, TQC, REDQ) behave when pushed toward offline-style experience reuse in a high-frequency visual control task.

### 2.4 Vision-Based Racing and TrackMania

Vision-based control has been studied extensively in simulated environments. DQN showed that deep networks can learn control directly from pixels on Atari [10], and CARLA established a high-fidelity urban driving simulator for imitation learning and RL in autonomous driving [2]. TrackMania, exposed through the open-source TMRL framework, offers a complementary testbed with high-speed racing, realistic physics, and limited simulation parallelism [8]. Prior TMRL work mainly focused on online SAC. Our project aims to extend this framework to support high-UTD off-policy training and to systematically compare SAC, TQC, and REDQ under vision-based, high-frequency control and real-time data-collection constraints.

Table 1: Hyperparameter settings for different algorithms.

| Algorithm | Critics | Batch Size | Temperature | Learning Rate | |
|---|---|---|---|---|---|
| | | | | Actor | Critic |
| REDQ | 10 | 256 | 0.2 | $3 \times 10^{-4}$ | $5 \times 10^{-5}$ |
| SAC | 2 | 256 | 0.2 | $3 \times 10^{-4}$ | $3 \times 10^{-4}$ |
| TQC | 5 | 256 | 0.2 | $3 \times 10^{-4}$ | $3 \times 10^{-4}$ |

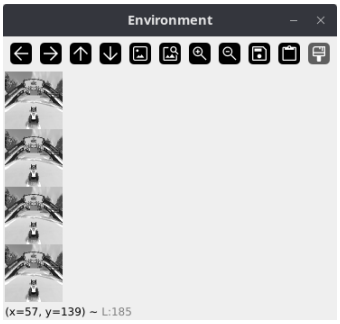
# 3 Methodology

## 3.1 Problem Formulation

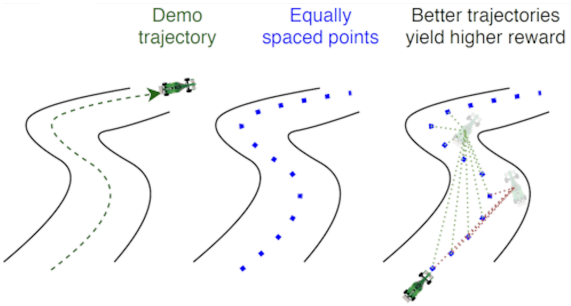
We formulate racing as a Partially Observable MDP (POMDP) $(\mathcal{S}, \mathcal{O}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \Omega, \gamma)$ where:

- **State** $s \in \mathcal{S}$: The complete physical state including vehicle pose (position, orientation, velocity, angular velocity), all opponents' states, track surface properties, and tire conditions
- **Observation** $o \in \mathcal{O}$: 4 consecutive $64 \times 64$ grayscale frames concatenated with telemetry vector (speed, gear, RPM) and 2 previous actions. The observation function $\Omega(s)$ maps states to observations
- **Action** $a \in \mathcal{A}$: Continuous 3D vector [gas, brake, steer] $\in [-1, 1] \times [-1, 1] \times [-1, 1]$ at 20Hz. Both gas and brake are clipped to $[0, 1]$ when sent to the game client.
- **Reward** $r_t = \mathcal{R}(s_t, a_t)$: Number of waypoints traversed per timestep (waypoints equally spaced along demo trajectory)
- **Dynamics** $s_{t+1} \sim \mathcal{P}(\cdot|s_t, a_t)$: Deterministic physics simulation with surface-dependent friction coefficients (provided by Trackmania)

The objective is maximizing expected discounted return $J(\pi) = \mathbb{E}_{\tau\sim\pi}[\sum_{t=0}^{\infty} \gamma^t r_t]$, where the policy $\pi(a|o)$ conditions on observations rather than full states. We use asynchronous data collection: a parallel worker continuously collects experience tuples $(o_t, a_t, r_t, o_{t+1})$ from the current policy into a replay buffer (capacity 1M), while the training process samples mini-batches independently.




(a) Observation space for the agent consists of four 64×64 grayscale images.




(b) Reward for the agent is calculated by number of waypoints traversed per timestep.

Figure 1: Environment observation space and reward structure.

## 3.2 Algorithms

In this section we introduce the three algorithms that we tried on TrackMania. We include the details of hyperparameters that we used in Table 1.

### 3.2.1 Soft Actor-Critic

Soft Actor-Critic (SAC) [5] is an entropy-regularized actor-critic with twin critics for overestimation mitigation. In the maximum-entropy framework, the optimal policy maximizes

$$\pi^\star = \arg\max_\pi \sum_{t=0}^{\infty} \mathbb{E}_{(s_t,a_t)\sim\rho_\pi}\big[r(s_t,a_t) + \alpha\mathcal{H}(\pi(\cdot \mid s_t))\big], \tag{1}$$

where $\mathcal{H}(\pi(\cdot \mid s)) = \mathbb{E}_{a\sim\pi(\cdot|s)}[-\log\pi(a \mid s)]$ is the Shannon entropy and $\alpha > 0$ is the temperature. The soft state-value and action-value functions satisfy

$$V^\pi(s) = \mathbb{E}_{a\sim\pi(\cdot|s)}\big[Q^\pi(s,a) - \alpha\log\pi(a \mid s)\big], \tag{2}$$

$$Q^\pi(s,a) = \mathbb{E}\Big[r(s,a) + \gamma\,\mathbb{E}_{s'\sim p(\cdot|s,a)}\big[V^\pi(s')\big]\Big]. \tag{3}$$

In deep SAC, we approximate $Q^\pi$ with twin critics $Q_{\theta_1}, Q_{\theta_2}$ and use a target network with parameters $\bar\theta_1, \bar\theta_2$. Given a replay buffer $\mathcal{D}$ and a next-action sample $a' \sim \pi_\phi(\cdot \mid s')$, the soft Bellman target is

$$y(r,s') = r + \gamma\Big(\min_{i\in\{1,2\}} Q_{\bar\theta_i}(s',a') - \alpha\log\pi_\phi(a' \mid s')\Big), \tag{4}$$

and each critic minimizes

$$J_Q(\theta_i) = \mathbb{E}_{(s,a,r,s')\sim\mathcal{D}}\Big[\tfrac{1}{2}\big(Q_{\theta_i}(s,a) - y(r,s')\big)^2\Big], \quad i = 1,2. \tag{5}$$

The actor is updated by minimizing the KL-regularized objective

$$J_\pi(\phi) = \mathbb{E}_{s\sim\mathcal{D},\, a\sim\pi_\phi(\cdot|s)}\big[\alpha\log\pi_\phi(a \mid s) - \min_i Q_{\theta_i}(s,a)\big], \tag{6}$$

which moves $\pi_\phi$ towards actions with high soft Q-values while maintaining entropy. When using an adaptive temperature, $\alpha$ is optimized to match a target entropy $\mathcal{H}_{\text{targ}}$ via

$$J(\alpha) = \mathbb{E}_{s\sim\mathcal{D},\, a\sim\pi_\phi(\cdot|s)}\big[-\alpha\big(\log\pi_\phi(a \mid s) + \mathcal{H}_{\text{targ}}\big)\big], \tag{7}$$

so that the gradient update decreases $\alpha$ when the policy is too stochastic and increases it otherwise.

### 3.2.2 Truncated Quantile Critics

Truncated Quantile Critics (TQC) [7] extends SAC with distributional critics outputting $K$ quantiles per Q-network. Each critic $i \in \{1,\dots,N\}$ represents a return distribution $Z_{\theta_i}(s,a)$ via atoms

$$Z_{\theta_i}(s,a) = \frac{1}{K}\sum_{k=1}^{K} \delta_{\theta_{i,k}(s,a)}, \tag{8}$$

where $\theta_{i,k}(s,a)$ approximates the $\tau_k$-quantile of the return at fixed quantile levels $\tau_k \in (0,1)$. The distributional Bellman target uses a sample $a' \sim \pi_\phi(\cdot \mid s')$ and soft value

$$y_j(r,s') = r + \gamma\big(\tilde\theta_j(s',a') - \alpha\log\pi_\phi(a' \mid s')\big), \tag{9}$$

where $\{\tilde\theta_j(s',a')\}_{j=1}^K$ are target quantiles from a target critic. Training minimizes the quantile Huber loss

$$J_Z(\theta_i) = \mathbb{E}_{(s,a,r,s')\sim\mathcal{D}}\left[\frac{1}{K}\sum_{k=1}^{K}\sum_{j=1}^{K}\rho_{\tau_k}^{\kappa}\big(y_j(r,s') - \theta_{i,k}(s,a)\big)\right], \tag{10}$$

where $\rho_\tau^\kappa$ is the quantile Huber loss used in QR-DQN. To control overestimation, TQC aggregates the $NK$ target quantiles from all critics, sorts them, and drops the largest $d$ atoms. Denote the sorted targets as $z_{(1)} \le \cdots \le z_{(NK)}$; the truncated mixture value is

$$Q_{\text{TQC}}(s',a') = \frac{1}{NK-d}\sum_{j=1}^{NK-d} z_{(j)}(s',a'). \tag{11}$$

This truncated value is then used both in the critic targets and in the actor loss. The policy objective becomes a soft-actor update similar to SAC,

$$J_\pi(\phi) = \mathbb{E}_{s\sim\mathcal{D},\, a\sim\pi_\phi(\cdot|s)}\big[\alpha\log\pi_\phi(a \mid s) - Q_{\text{TQC}}(s,a)\big], \tag{12}$$

where $Q_{\text{TQC}}(s,a)$ is computed by averaging the non-truncated atoms of the current critics. By discarding the top quantiles, TQC introduces an implicit pessimism that systematically reduces overestimation bias.

### 3.2.3 Randomized Ensembled Double Q-Learning

Randomized Ensembled Double Q-Learning (REDQ) [1] augments SAC with an ensemble of $N$ critics and a high update-to-data ratio $G \gg 1$, while using random subsets for target computation to control bias and variance. Let $\{Q_{\theta_i}\}_{i=1}^{N}$ be the critic ensemble and $\{Q_{\bar{\theta}_i}\}_{i=1}^{N}$ their target networks. At each critic update, we sample a random subset $\mathcal{M} \subset \{1, \ldots, N\}$ of size $M$ (without replacement) and draw $a' \sim \pi_\phi(\cdot \mid s')$. The REDQ target is

$$y_{\mathrm{REDQ}}(r, s') = r + \gamma\Big( \min_{j \in \mathcal{M}} Q_{\bar{\theta}_j}(s', a') - \alpha \log \pi_\phi(a' \mid s') \Big), \tag{13}$$

so that each critic $Q_{\theta_i}$ minimizes

$$J_Q(\theta_i) = \mathbb{E}_{(s,a,r,s') \sim \mathcal{D}} \Big[ \tfrac{1}{2} \big( Q_{\theta_i}(s, a) - y_{\mathrm{REDQ}}(r, s') \big)^2 \Big], \quad i = 1, \ldots, N. \tag{14}$$

The policy is updated using the ensemble-averaged Q-value,

$$J_\pi(\phi) = \mathbb{E}_{s \sim \mathcal{D},\, a \sim \pi_\phi(\cdot|s)} \Big[ \alpha \log \pi_\phi(a \mid s) - \frac{1}{N} \sum_{i=1}^{N} Q_{\theta_i}(s, a) \Big], \tag{15}$$

which reduces variance compared to using a single critic. REDQ further sets a large update-to-data ratio $G$, i.e., performs $G$ gradient updates per environment step, leveraging the ensemble and in-target minimization to keep the induced overestimation bias small while achieving high sample efficiency.

## 3.3 Experimental Setup

### 3.3.1 Tracks

**Fall 2020-01 (Fullspeed)** "Fall 2020-01" is the opening fullspeed map of the official Fall 2020 Trackmania campaign by Nadeo, clocking in at roughly 19–20 seconds for a clean run. It is tagged as a mixed-surface fullspeed race, combining approximately half its length on asphalt with the rest split between dirt and grass, which gives the track a distinctly flowing but slightly slippery character. The layout offers a wide racing line that rewards players who commit to smooth, high-speed trajectories rather than aggressive braking or micro-adjustments; the key challenge lies in reading the transitions between grip levels as you move from asphalt onto dirt and grass and back again while trying to stay full throttle. Because of the short duration and generous line width, the map becomes a precision speedrun where small optimizations—early steering inputs, careful air control on subtle bumps, and efficient exits off each surface change—can easily decide whether the agent could hit author time or even approach world-record pace.

**Eaux (Technical)** Eaux is a compact 20-second technical Trackmania map, featured in the "High Quality Shorts" mappack on Trackmania Exchange as a mini tech race by Whiskey.wp. The route is laid out on 100% asphalt, so the challenge comes not from mixed surfaces but from precise car control through a sequence of tight hairpins and linked corners that quickly punish any over-steering or late braking. Subtle elevation changes force players to manage weight transfer carefully, adjusting their lines as the car crests and dips to maintain grip and exit speed. As a result, Eaux plays like a concentrated mechanics test: success depends on clean setups into each hairpin, smooth throttle modulation, and the ability to carry momentum while staying within the narrow margin for error imposed by the short overall runtime.

### 3.3.2 Training & Metrics

We train each agent until track completion becomes relatively consistent, operationalized as achieving a completion rate above 50% over recent evaluation episodes, with a hard cap of 30 wall-clock hours to ensure comparable computational budgets across algorithms.

For evaluation, we measure performance using the episode return, defined as the cumulative reward obtained within an episode, and report the mean episode return, as well as the standard deviations, over all episodes in a given epoch. This metric directly reflects the agent's ability to both progress along the track and avoid catastrophic failures (e.g., crashes or timeouts), and provides a consistent basis for comparing different algorithms and hyperparameter settings.

### 3.4 Ablation Studies

To isolate the contributions of individual design choices, we conducted systematic ablations across two primary dimensions: visual encoder architecture and temporal observation representation. All ablations were performed on the Power Source track using SAC as the base algorithm to control for algorithmic confounds.

**Visual Encoder Architecture.** We evaluated three encoder architectures with varying capacity and inductive biases:

- *Vanilla CNN*: A lightweight 4-layer convolutional network (64-64-128-128 channels) with stride-2 downsampling, designed for real-time inference. Processes stacked grayscale frames directly.
- *Impoola-CNN* [12]: A 15-layer ResNet-inspired encoder that extends Impala-CNN by replacing the flattening operation with global average pooling (GAP). This architectural modification reduces translation sensitivity and creates a more balanced parameter distribution, with only 4% of parameters in the final linear layer compared to 73% in standard Impala-CNN. The GAP layer aggregates spatial information while maintaining input size independence.
- *DINOv3* [11]: A frozen vision transformer backbone (ConvNeXt-Tiny variant) pretrained via self-supervised learning on large-scale image data. This required architectural modifications to the tmrl pipeline: rather than storing raw images in the replay buffer, we precompute and cache DINOv3 feature embeddings (768-dimensional vectors) to amortize inference cost during training.

**Temporal Observation Representation.** We investigated the trade-off between temporal context and computational efficiency:

- *4-frame grayscale history*: Four consecutive $64 \times 64$ grayscale frames stacked channel-wise, enabling implicit velocity estimation through frame differencing.
- *Single RGB frame*: One $64 \times 64$ RGB frame providing richer color information (surface type discrimination) but no explicit temporal context.

## 4 Results

### 4.1 Performance Comparison

Table 2 presents final performance metrics across the last 10% of epochs for all algorithms on both track types. On the Fall 2020-01 (Fullspeed) track, TQC achieves the highest final return of $154.4 \pm 3.7$, representing a 20.4% improvement over the SAC baseline ($128.3 \pm 0.3$). REDQ failed to converge on both tracks, achieving only $3.8 \pm 0.4$ on Fall 2020-01 and $3.2 \pm 0.0$ on Eaux, suggesting that ensemble-based uncertainty estimation alone is insufficient for this domain.

Table 2: Final performance after training (mean $\pm$ SE)

| **Algorithm** | **Eaux** | **Fall 2020-01** |
|---|---|---|
| REDQ | $3.2 \pm 0.0$ | $3.8 \pm 0.4$ |
| SAC | $\mathbf{150.9 \pm 1.7}$ | $128.3 \pm 0.3$ |
| TQC | $5.5 \pm 0.0$ | $\mathbf{154.4 \pm 3.7}$ |

Notably, performance patterns differ substantially between track types. While TQC dominates on the Fullspeed track, SAC achieves the best performance on the Eaux (Technical) track with a return of $150.9 \pm 1.7$. This suggests that distributional value estimation with truncation may be overly conservative for technical tracks requiring precise maneuvers. The improvement metrics are summarized in Table 3.

Table 3: Performance relative to SAC baseline

| Map | Algorithm | Abs. Gain | Rel. Gain (%) |
|---|---|---|---|
| Fall 2020-01 | TQC | +26.1 | +20.4% |
| Fall 2020-01 | REDQ | −124.5 | −97.0% |
| Eaux | TQC | −145.4 | −96.4% |
| Eaux | REDQ | −147.7 | −97.9% |

### 4.2 Learning Dynamics

Figure 2 presents the learning curves for all algorithm-map combinations. On the Fullspeed track (left panel), TQC and SAC both achieve strong performance, though TQC demonstrates faster initial learning. SAC shows greater variance during the mid-training phase and isn't able to recover fully before hitting 30 hours. REDQ fails to make meaningful progress, remaining near zero throughout training.

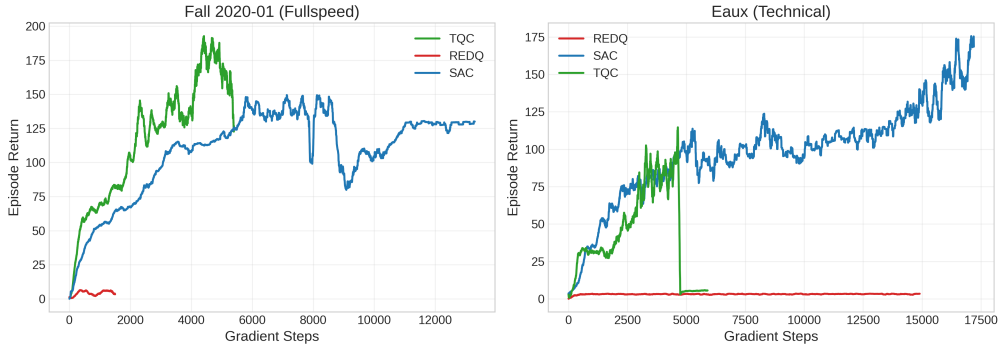



Figure 2: Learning curves across gradient steps. Left: Fall 2020-01 (Fullspeed). Right: Eaux (Technical). REDQ failed to converge on both tracks.

The Eaux track (right panel) reveals a different pattern: In early epochs, SAC and TQC maintain similar steady improvement throughout training but around epoch 5000, TQC experiences a performance collapse. This suggests TQC's quantile truncation, designed to mitigate overestimation, may be overly conservative for the precise driving style required on technical tracks. SAC's higher variance tolerance allows it to discover and maintain effective strategies.

**Sample Efficiency.** On the Fullspeed track, TQC reaches a return threshold of 125 in just 2,229 gradient steps compared to 5,394 for SAC, demonstrating more than 2× faster learning. However, this advantage does not persist on the Eaux track, where SAC's steadier learning proves more effective overall.

Table 4: Steps required to reach performance thresholds of rolling average window over 100 epochs. A dash (–) indicates the threshold was not reached during training.

| Map | Algorithm | Steps to Reach Threshold | | | | |
|---|---|---|---|---|---|---|
| | | 50 | 75 | 100 | 125 | 150 |
| Eaux | SAC | 1,342 | 2,238 | 4,800 | 13,961 | 15,882 |
| Eaux | TQC | 2,305 | 2,958 | 3,284 | – | – |
| Fall 2020-01 | SAC | 799 | 2,331 | 3,070 | 5,394 | – |
| Fall 2020-01 | TQC | 340 | 1,303 | 1,903 | 2,229 | 3,252 |

**REDQ Failure Analysis.** REDQ's complete failure on both tracks warrants explanation. Despite its design for high update-to-data (UTD) ratios, we were unable to leverage this advantage in the TrackMania environment. The computational overhead (10× more Q-network updates per environment step) without corresponding performance gains suggests several potential issues: (1) the

ensemble of 10 Q-networks may require careful hyperparameter tuning (e.g., ensemble size, target network update frequency) that is highly domain-specific, or (2) the increased computational cost may have limited the total number of environment interactions within our training budget, hindering REDQ's ability to collect sufficient diverse experience. These results suggest that ensemble methods designed for high UTD ratios do not straightforwardly transfer to vision-based continuous control tasks tuned for SAC without adaptation.

### 4.3 Statistical Significance

To validate that observed performance differences are not due to random variation, we conduct Welch's t-tests on late-phase training returns (Table 5). All pairwise comparisons achieve statistical significance ($p < 0.001$), confirming that the performance ordering is robust.

Table 5: Statistical significance of performance differences (Welch's t-test)

| Map | Comparison | t-statistic | p-value |
|---|---|---|---|
| Fall 2020-01 | TQC vs SAC | 24.16 | $< 0.0001$* |
| Fall 2020-01 | REDQ vs SAC | $-278.02$ | $< 0.0001$* |
| Fall 2020-01 | TQC vs REDQ | 90.45 | $< 0.0001$* |
| Eaux | SAC vs TQC | 69.33 | $< 0.0001$* |
| Eaux | SAC vs REDQ | 160.85 | $< 0.0001$* |
| Eaux | TQC vs REDQ | 29.62 | $< 0.0001$* |

### 4.4 Ablation Results

We conducted ablation studies on the Power Source track to evaluate visual encoder architectures and temporal observation representations. Figure 3 shows the learning curves for each configuration.

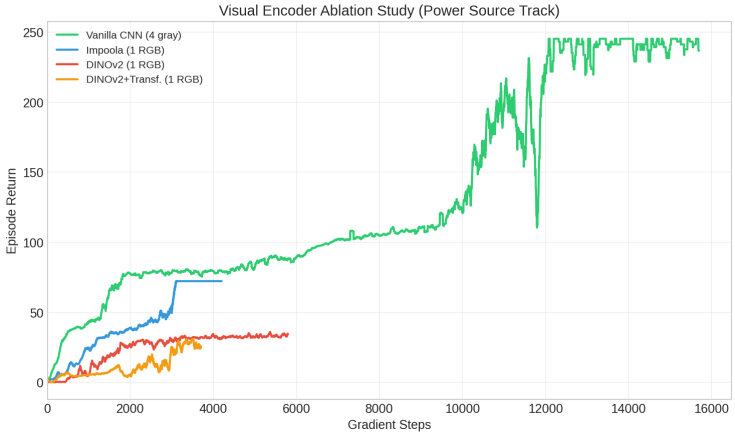



Figure 3: Visual encoder ablation learning curves. The 4-frame grayscale configuration (Vanilla CNN) dramatically outperforms all single RGB frame alternatives.

Table 6 summarizes final performance for each encoder configuration. The results reveal a striking finding: temporal context from frame stacking is far more important than encoder architecture sophistication or pretrained representations.

**Temporal Representation Dominates.** The 4-frame grayscale configuration achieves a return of $240.9 \pm 0.8$, dramatically outperforming all single RGB frame alternatives by 230–330%. This confirms that temporal context for velocity estimation is critical for vehicle control—frame stacking enables implicit optical flow computation that single frames cannot provide, regardless of encoder sophistication.

Table 6: Visual encoder ablation study results (Power Source track)

| Encoder | Input | Return | Max | vs Baseline |
|---|---|---|---|---|
| Vanilla CNN | 4 grayscale | $\mathbf{240.9 \pm 0.8}$ | 245.4 | (baseline) |
| Impoola | 1 RGB | $72.3 \pm 0.0$ | 106.6 | $-70.0\%$ |
| DINOv3 | 1 RGB | $33.4 \pm 0.1$ | 42.1 | $-86.1\%$ |
| DINOv3+Transf. | 1 RGB | $27.1 \pm 0.4$ | 44.2 | $-88.7\%$ |

# 5 Discussion

## 5.1 Significance

**Autonomous vehicles**: TQC's conservative value estimation and efficient sample utilization enable learning robust emergency controllers and collision avoidance behaviors from safe demonstrations without requiring dangerous real-world exploration. The high-frequency control insights transfer directly to autonomous driving scenarios where reaction time and data efficiency are critical safety considerations.

**High-frequency robotics**: Our findings on sample-efficient learning at high control frequencies extend naturally to quadrotor drones requiring 20-100Hz stabilization, robotic manipulation involving contact-rich tasks with complex dynamics, and industrial process control where physical data collection is expensive or constrained but offline training compute is abundant. The ability to learn effective policies from limited interaction data addresses a fundamental bottleneck in deploying RL to real robotic systems.

**Accessible research platform**: TrackMania provides a low-barrier entry point for reinforcement learning research without expensive hardware, cloud compute, or specialized equipment, effectively democratizing sample-efficiency research. The platform's realistic physics, challenging control dynamics, and active community make it an attractive testbed for developing and benchmarking algorithms before deployment on physical systems.

## 5.2 Limitations

**Environmental**: The real-time constraint of the simulation environment limited the scale of our dataset collection, restricting the total number of training episodes and environment interactions. The deterministic physics engine, while computationally efficient, lacks the stochasticity inherent in real-world dynamics such as tire slip variability, surface irregularities, and aerodynamic perturbations. Additionally, the idealized sensor model assumes perfect state observation which oversimplifies the perception challenges encountered in physical autonomous racing systems.

**Algorithmic**: Our study focused exclusively on actor-critic variants within the policy gradient family of reinforcement learning algorithms. We did not evaluate model-based RL approaches that could potentially improve sample efficiency through learned dynamics models, imitation learning methods that might leverage expert demonstrations, or evolutionary strategies that could explore the policy space differently.

**Evaluation**: The experimental scope was restricted to two distinct track layouts, which provides limited diversity in terms of geometric complexity, corner types, and racing line characteristics. This narrow evaluation domain may not fully capture the range of challenges present in competitive autonomous racing scenarios. Critically, our findings remain purely simulation-based without sim-to-real validation on physical vehicles, leaving questions about the transferability of learned policies unaddressed.

## 5.3 Future Work

Several promising directions emerge from this work's limitations and findings:

1. **Multi-task and transfer learning**: Developing agents capable of generalizing across diverse track configurations represents a critical next step. This includes training unified policies on varied track geometries, surface types, and mapping styles to learn transferable racing

primitives rather than track-specific behaviors. Meta-learning approaches could enable rapid adaptation to novel circuits with minimal fine-tuning.

2. **Learning from demonstration**: The availability of expert leaderboard replays presents an opportunity to incorporate imitation learning and behavior cloning. Combining demonstrations with reinforcement learning through approaches like GAIL or DAgger could accelerate training by providing strong initialization and guiding exploration toward high-performance regions of the policy space.
3. **Sim-to-real transfer**: Validating learned policies on physical platforms such as RC racing cars would provide crucial insights into real-world applicability. This would require developing robust domain randomization strategies, addressing sensor noise and actuation delays, and potentially incorporating system identification techniques to minimize the reality gap.

## 6 Conclusion

This work demonstrates that algorithm performance in high-frequency visual control is highly environment-dependent. TQC achieves superior sample efficiency on fullspeed tracks where conservative value estimation prevents overcommitment, while SAC excels on technical tracks where TQC's truncation proves overly conservative. REDQ fails entirely, highlighting that ensemble methods designed for state-based MuJoCo tasks do not straightforwardly transfer to vision-based control. These findings underscore that no single algorithm universally dominates in sample-constrained, high-frequency control, motivating careful algorithm selection based on task characteristics. By establishing TrackMania as an accessible benchmark and releasing our implementation,[1] we enable future research on sample-efficient RL for high-frequency control systems.

## Acknowledgments

We thank the TrackMania community and tmrl framework developers, and our course instructors for feedback and guidance.

## References

[1] Xue Bin Chen, Tongzheng Lu, Xinyue Xu, Zichuan Zhu, Yuxin Yang, Zuyuan Ma, Heng Xu, and Yuan Zhu. Randomized ensembled double q-learning: Learning fast without a model. In *International Conference on Learning Representations*, 2021.

[2] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, pages 1–16, 2017.

[3] Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4rl: Datasets for deep data-driven reinforcement learning. *arXiv preprint arXiv:2004.07219*, 2020.

[4] Scott Fujimoto, Herke van Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *International Conference on Machine Learning*, pages 1587–1596, 2018.

[5] 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.

[6] Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 33, pages 1179–1191, 2020.

[7] 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.

[1] `https://github.com/Tyrest/tmrl`

[8] Yannick Liégeois. Trackmania roborace league. `https://github.com/trackmania-rl/tmrl`, 2022.

[9] 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. In *International Conference on Learning Representations*, 2016.

[10] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

[11] Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Hervé Jégou, Patrick Labatut, and Piotr Bojanowski. Dinov3, 2025.

[12] Raphael Trumpp, Ansgar Schäfftlein, Mirco Theile, and Marco Caccamo. Impoola: The power of average pooling for image-based deep reinforcement learning, 2025.