Project Final Report

Team 34

Daniel Geyfman, geyfmand, 17782117
Bailey Deng, baileyd1, 46475318
Richard Zhang, richarzz, 64283611

1 Project Summary

Our goal was to train an autonomous racing policy that drives the virtual **AWS DeepRacer** car around time-trial tracks *as fast as possible while remaining on-track*.

DeepRacer resembles a 1⁄18-scale vehicle equipped with a forward-facing camera; in simulation, each step delivers a 160 x 120 px grayscale image from which AWS extracts 15 engineered state variables (speed, distance-from-center, heading, etc.). The agent must output a steering angle and throttle at 15 Hz. Solving this problem is **non-trivial** because:

- the continuous action space is high-variance and delayed-reward; naïve exploration quickly crashes
- rewards are sparse—reaching the finish line may take thousands of steps at the start
- the car must balance two competing objectives: hugging the racing line in corners and maximizing speed on straights. Classical control or scripted heuristics struggle here; instead, we rely on deep **reinforcement learning (RL)** to learn nuanced, track-specific behaviours

By Week 10, our best model completes *Forever Raceway* in **16.397 s**, ranking **38 / 678 (top 6 %)** in the 2025 U.S. Open Practice leaderboard. We also generalised to three unseen tracks, demonstrating transfer ability (see section 3).

---

2 Approaches

2.1 Baseline (Status-Report model)

- **Algorithm:** AWS default **PPO-Clip** ($\gamma = 0.999$, $\lambda = 0.95$, $\varepsilon = 0.2$).
- **Action space:** 9 discrete (steering, speed) pairs, max 4 $m\,s^{-1}$.
- **Reward:** distance-from-center bonus; +10 on finish; −100 if off-track.
- **Training:** 4 Robomaker workers × 30 min = 2 hrs per iteration.
- **Result:** Avg 17.776 s, best 16.731 s on Forever Raceway.

2.2 Proposed Improvements

a. Racing-Line Reward

We replaced center-line shaping with an *ideal-line* lookup produced by hand-driving once and smoothing with a Catmull–Rom spline. Reward: where is lateral distance to the spline and current speed. This directly incentivises higher speed on straights (second term) while preserving line adherence.

b. Turn-Aware Acceleration

We introduced a *curvature gate*: allowable throttle scales with upcoming track curvature estimated three waypoints ahead. Implemented as an observation-level feature so the network learns *when* to lift.

c. Long-Horizon Training Budget

Instead of many short jobs, we ran **one 8-hour job** (still within the free tier) which covers ≈190k steps starting from random waypoints—mitigating over-fitting to a single racing line.

d. Hyper-parameter Sweep

A Bayesian search (Ax) over {γ, learning-rate, entropy-β, clip-ε} yielded a three percent faster convergence curve; final settings are listed in Appendix A.

e. Failed Variant: Soft Actor–Critic

We ported the reward to AWS SAC but—even with a doubled replay buffer—SAC lagged PPO by roughly nine percent after six hours and was abandoned.

f. Pseudocode

```
for episode in range(N):
    s ← env.reset(random_waypoint=True)
    done ← False
    while not done:
        a ← π_θ(s)
        s′, r, done ← env.step(a)
        buffer.add(s, a, r)
        s ← s′
    # PPO-clip update
    for minibatch in buffer:
        r(θ) = π_θ(a|s) / π_{θ_old}(a|s)
        L = mean(min(rÂ, clip(r,1−ε,1+ε)Â))
        θ ← θ + η ∇_θ L
  θ_old ← θ
```

---

3 Evaluation

3.1 Experimental Setup

- **Simulator:** AWS DeepRacer v2.5 (fast sim)
- **Hardware:** 4 × c5.large workers + 1 × ml.c5.xlarge learner (AWS free‑tier credits)
- **Metrics:** *Lap time* (lower = better); *%‑off‑track*; *lap completion rate*.
- **Test Protocol:** Best checkpoint chosen via validation reward, then frozen. For each track we run **ten laps** with random start offsets, report *mean*, *std*, and *best*.

3.2 Quantitative Results

Track	Baseline Mean (s)	Final Mean (s)	Final Best (s)	Δ % (mean)
Forever Raceway	17.78 ± 0.42	**16.62 ± 0.31**	**16.397**	**−6.5 %**
Ace Speedway	31.08 ± 0.55	**30.12 ± 0.48**	29.924	−3.1 %
Rogue Circuit	30.41 ± 0.60	**29.56 ± 0.37**	29.269	−2.8 %
BreadCentric Speedway	46.27 ± 0.71	**45.38 ± 0.64**	45.064	−1.9 %

The biggest gains appear on Forever Raceway—the optimisation target—while generalisation still provides *1–3 %* speed‑ups on unseen tracks.

3.3 Ablation Study

Removing each component from §2.2 yields the following slowdown on Forever Raceway (mean of 5 laps):

- − Racing‑line reward: +0.88 s
- − Curvature gate: +0.46 s
- − Long‑horizon training: +0.33 s This confirms the reward redesign is the dominant contributor.

3.4 Qualitative Analysis

Fig. 1 (reward‑per‑episode curve) shows stable monotonic improvement; Fig. 2 overlays trajectories—our agent hugs the inside of turn 3 and carries ≈3.9 m s^{-1} onto the back straight compared with 3.1 m s^{-1} baseline. In video playback, the car exhibits earlier braking points but smoother apex clipping, eliminating the small fishtail evident in the status-report model.

---

4 References & Resources Used

1. Schulman et al., *Proximal Policy Optimization Algorithms*, arXiv:1707.06347 (2017).
2. Haarnoja et al., *Soft Actor-Critic: Off-Policy Maximum Entropy Deep RL*, ICML 2018.
3. Mark Ross, "DeepRacer Model Optimisation Tips," AWS YouTube Channel, 2024.
4. Ray G., "Using Log Analysis to Drive Experiments and Win the AWS DeepRacer F1 ProAm Race," AWS ML Blog, 2023.
5. AWS DeepRacer Developer Guide, Release 2.5, 2025.
6. Ax Developers, *Adaptive Experimentation Platform v0.3*, GitHub 2025.

14)