Curriculum-based Deep Reinforcement Learning for Autonomous Driving in Simulated Highway Environments

Raahim A. Samad Poonawala
Department of Computer Science
Lahore University of Management Sciences
Lahore, 54792
27100419@lums.edu.pk

Saim Bilal
Department of Computer Science
Lahore University of Management Sciences
Lahore, 54792
27100208@lums.edu.pk

Muhammad Bin Tariq
Department of Computer Science
Lahore University of Management Sciences
Lahore, 54792
27100017@lums.edu.pk

Abstract

Inspired by CARLA [Dosovitskiy et al., 2017] and CuRLA [Uppuluri et al., 2025], we investigate curriculum-based deep reinforcement learning for autonomous driving in simulated highway environments. Our goal is to understand whether structured progression from simple to complex driving scenarios improves sample efficiency and generalization compared to training directly on complex tasks. We formulate autonomous driving as a Markov Decision Process in the OpenAI Highway Gymnasium environment and train agents using two deep RL algorithms, namely: PPO [Schulman et al., 2017] and DQN [Mnih et al., 2015] from the Stable-Baselines3 library, further self-implementing DQN as SimpleDQN. Our initial experiments focus on non-curriculum baselines across four tasks: multi-lane highway driving, highway merging, signalized intersections, and roundabouts, with standardized kinematic observations and discrete meta-actions. We then design a four-stage curriculum that starts with low-traffic highway driving and gradually increases complexity through merging, intersections, and dense roundabouts under a fixed interaction budget. This paper presents the problem formulation, experimental setup for both non-curriculum and curriculum regimes in MetaDrive and HighwayEnv, training and evaluation results on held-out scenarios, and an in-depth discussion on those results. **All our code and simulations are available here.**

1 Introduction

The domain of autonomous driving presents a complex safety-critical dynamic environment, where intelligent agents must be robust; able to adapt and generalize to a wide range of scenarios. Deep reinforcement learning (RL) has shown promise in training agents to make sequential decisions through trial-and-error interactions with the environment. However, training an agent to behave safely and efficiently in complex traffic scenarios can be extremely sample-inefficient and unstable.

Curriculum learning, first proposed by Bengio [Bengio et al., 2009], involves structuring the training process from simpler to more complex tasks. It proposes to address this challenge by exposing the learner to a sequence of increasingly difficult tasks rather than training directly on the hardest

Preprint. Under review.

configuration. This approach has been proposed as a way to improve learning efficiency and generalization in RL. In the context of RL, a curriculum can be expressed as an ordered sequence of Markov Decision Processes (MDPs) that gradually increase complexity, enabling more efficient knowledge transfer across tasks [Narvekar et al., 2020].

In this work, we investigate curriculum-based deep RL for autonomous driving in simulated highway environments, such as OpenAI's Highway Gymnasium [Leurent, 2018] and MetaDrive [Li et al., 2022]. We aim to understand whether a structured progression through driving scenarios can lead to better performance compared to training directly on complex tasks.

2 Related Work

2.1 Curriculum Reinmforcement Learning

In reinforcement learning, curriculum learning has been formalized at the level of tasks or Markov Decision Processes (MDPs). Narvekar et al. [2020] propose a general framework in which a curriculum is defined as an ordered sequence of source tasks, each represented as an MDP, and survey methods for selecting and sequencing such tasks. Their framework highlights that curricula can be used to improve sample efficiency, overcome local optima, and bridge the gap between simple training tasks and a target task that may be too difficult to learn from scratch. They also emphasize that curriculum design is largely orthogonal to the choice of underlying RL algorithm: value-based and actor–critic methods can, in principle, be combined with appropriately chosen task sequences.

Our work follows this view of curricula as ordered sequences of MDPs, instantiated in the context of autonomous driving tasks. We define a hand-crafted curriculum over four HighwayEnv scenarios that gradually increases complexity along two axes: road topology (highway, merge, intersection, roundabout) and traffic density. Unlike most prior curriculum RL work, which focuses on a single algorithm, we explicitly evaluate the interaction between curriculum design and two algorithm families: value-based off-policy control (DQN) and on-policy policy-gradient actor–critic (PPO).

2.2 Reinforcement Learning for Autonomous Driving and Driving Simulators

A variety of driving simulators have been developed to support RL research. HighwayEnv [Leurent, 2018] is a lightweight collection of Gym-compatible environments for tactical decision-making in autonomous driving, including lane-keeping, lane-changing, merging, and intersection navigation. It provides configurable kinematic observations, discrete meta-actions, and customizable traffic conditions, making it well-suited for fast experimentation with deep RL algorithms. MetaDrive [Li et al., 2022] is a more recent, realistic and diverse simulator that emphasizes compositional scenario generation and generalization to unseen maps, hence being more compute-intensive; it supports both single-agent and multi-agent RL tasks and has been used to study generalization, safe exploration, and multi-agent interactions.

2.3 Curriculum Learning for Autonomous Driving

Several researchers have proposed curriculum-based approaches tailored to autonomous driving. A common pattern is to fix a particular deep RL algorithm (often PPO or a related actor–critic method) and vary the environment complexity or reward shaping over training stages. For example, Uppuluri et al. [2025] introduce CuRLA, a curriculum learning framework for CARLA in which a PPO agent is combined with a variational autoencoder (VAE) to operate on compressed visual observations. They design a two-fold curriculum that gradually increases scenario difficulty and tightens safety constraints via collision penalties, reporting improvements in both training performance and evaluation robustness relative to non-curriculum baselines.

In contrast, we focus on HighwayEnv as a lightweight yet diverse testbed that includes multi-lane highways, merges, intersections, and roundabouts. We design a four-stage curriculum that starts with low-traffic highway driving and progresses through increasingly complex tasks and traffic conditions, under a fixed interaction budget. We instantiate this curriculum with two different deep RL algorithms (DQN and PPO) and compare them against a non-curriculum baseline that trains on a fixed mixture of tasks.

2.4 Choice of Reinforcement Learning Algorithms

The specific choice of RL algorithms in our study is motivated by their prominence in the deep RL and autonomous driving literature and by their complementary design philosophies. Deep Q-Networks (DQN) extend tabular Q-learning to high-dimensional settings by approximating the action-value function with a neural network and stabilizing training via a target network and experience replay. As an off-policy, value-based method that operates naturally over discrete actions, DQN provides a canonical baseline for decision-making with meta-actions such as lane changes and acceleration commands.

Proximal Policy Optimization (PPO) is an on-policy actor–critic method that directly parameterizes a stochastic policy and uses a clipped surrogate objective to prevent overly large policy updates. PPO has become a de facto standard in continuous-control benchmarks and is frequently used as the base learner in curriculum-based driving work, including CARLA-based studies.

By evaluating DQN and PPO on the same kinematic observation and discrete meta-action space in HighwayEnv, we situate our experiments within the broader deep RL and autonomous driving literature while enabling a controlled comparison of algorithm families. In particular, we can analyze how curriculum learning interacts with (i) value-based vs. policy-gradient updates, (ii) on-policy vs. off-policy data usage, and (iii) entropy-regularized vs. standard objectives. To the best of our knowledge, prior curriculum learning work for autonomous driving has not systematically compared curricula across such diverse algorithmic paradigms within a single experimental framework.

3 Mathematical Formulation

We formulate the autonomous driving problem as a Markov Decision Process (MDP) defined by the tuple $(\mathcal{S}, \mathcal{A}, P, R, \gamma)$, where:

- $\mathcal{S}$ is the state space, representing the kinematic observations of the ego vehicle and surrounding traffic. Each state $s \in \mathcal{S}$ includes features such as position, velocity, acceleration, lane information, and distances to nearby vehicles.
- $\mathcal{A}$ is the action space, consisting of discrete meta-actions that the agent can take. These actions include lane changes (left, right, maintain lane) and speed adjustments (accelerate, decelerate, maintain speed).
- $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ is the state transition probability function, defining the dynamics of the environment. It specifies the probability of transitioning to state s' from state s after taking action a. Note that for our purposes, we do not require explicit knowledge of P, as we employ model-free reinforcement learning algorithms. Thus, the simulator implicitly defines these transition dynamics through its interactions.
- $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function, which provides feedback to the agent based on its actions. The reward structure is designed to encourage safe and efficient driving behaviors, such as maintaining a desired speed, avoiding collisions, and adhering to traffic rules.
- $\gamma \in [0, 1)$ is the discount factor, which determines the importance of future rewards in the agent's decision-making process.

The objective of the agent is to learn a policy $\pi : \mathcal{S} \to \mathcal{A}$ that maximizes the expected cumulative discounted reward:

$$J(\pi) = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) \right]$$

where the expectation is taken over the trajectories generated by following policy π.

As Narvekar et al. [2020] described, there are three main classes of methods to learn an optimal policy π^*:

- Value-based methods: These methods focus on estimating the optimal action-value function $Q^*(s, a)$, which represents the expected cumulative reward of taking action a in state s and following the optimal policy thereafter. The optimal policy can then be derived by selecting the action that maximizes $Q^*(s, a)$ for each state s.

- Policy-based methods: These methods directly parameterize the policy $\pi_\theta(a|s)$ and optimize the policy parameters θ to maximize the expected cumulative reward $J(\pi_\theta)$ using gradient ascent techniques.
- Actor-critic methods: These methods combine value-based and policy-based approaches by maintaining both a policy (the actor) and a value function (the critic). The critic estimates the value function, which is used to update the policy.

For the purposes of this study, we chose two representative algorithms from these classes: Deep Q-Networks (DQN) as a value-based method, and Proximal Policy Optimization (PPO) as a policy-based method. Soft Actor-Critic (SAC) as an actor-critic method was not evaluated beyond preliminary setups due to computational constraints and training instability enforced through the combined value-based and policy-based approaches working in tandem. Each of these algorithms were implemented using the Stable-Baselines3 library [Raffin et al., 2021] and trained on the HighwayEnv [Leurent, 2018] simulator. For the MetaDrive [Li et al., 2022] simulator, we only used the PPO and DQN algorithms due to computational constraints.

4 Experimental Setup

In this section we describe our experimental setups in the OpenAI Highway Gymnasium environment (HighwayEnv) and in the MetaDrive simulator. HighwayEnv serves as our primary testbed for comparing curriculum and non-curriculum training under a fixed interaction budget, while MetaDrive provides a more complex, procedurally generated environment used to design a complementary curriculum and per-stage non-curriculum baselines.

4.1 HighwayEnv Experiments

4.1.1 Tasks and Environment Configuration

We use the `highway-env` suite of Gymnasium-compatible tactical driving tasks as our main benchmark. Specifically, we consider four scenarios:

- `highway-v0`: multi-lane highway driving.
- `merge-v0`: highway with an on-ramp merge.
- `intersection-v0`: signalized intersection.
- `roundabout-v0`: circular roundabout with entering and exiting traffic.

All tasks share a common base configuration. Episodes have a fixed duration of 40 s, with simulation and control frequencies set to 15 Hz and 5 Hz, respectively. Traffic density is controlled by a scalar parameter $\rho \in [0, 1]$, and the number of other vehicles in the scene is set to $\text{vehicles\_count} = \lfloor 50\rho \rfloor$. We adjust ρ per task to create a range of difficulties, and slightly perturb it between training and test for held-out evaluation.

4.1.2 Observation and Action Spaces

To standardize the input across tasks, we use the built-in `Kinematics` observation:

$$s_t = \text{concat}\left[(\text{presence}_i, x_i, y_i, v_{x,i}, v_{y,i})_{i=0}^{N-1}\right],$$

where $N = 5$ is the number of vehicles (ego plus nearest neighbors), and the features are normalized and expressed in relative coordinates. This yields a fixed-dimensional, low-level kinematic state vector.

The action space is discrete and defined via `DiscreteMetaAction`, which maps a small set of high-level actions (e.g., keep lane, change lane left/right, accelerate, brake) to continuous control commands for the underlying vehicle dynamics. This choice allows us to apply both value-based (DQN) and policy-gradient (PPO) methods without changing the environment.

4.1.3 Algorithms and Hyperparameters (MetaDrive and HighwayEnv)

We instantiate three reinforcement learning algorithms: PPO and DQN from stable_baselines3 and our own implementation of DQN by the name of SimpleDQN, each using a feed-forward MLP policy over the input observations in HighwayEnv. MetaDrive contains the implementation of PPO and DQN from stable_baselines3 only.

PPO. We use an `MlpPolicy` with two hidden layers of 256 units each, employing the `Tanh` activation function for better performance on normalized observations. The learning rate is set to 5×10^{-4}, with a batch size of 64 and a rollout length of 2048 steps. We run 10 optimization epochs per update, and the clipping parameter for PPO is set to 0.2. The discount factor is $\gamma = 0.99$, and we use Generalized Advantage Estimation (GAE) with a λ of 0.95. The entropy coefficient is set to 0.01 to encourage exploration, and the value function coefficient is set to 0.5. Gradient clipping is applied with a maximum gradient norm of 0.5. State-dependent exploration is disabled by setting `use_sde` to False, and we normalize the advantages during training.

DQN. We use an `MlpPolicy` with two hidden layers of 256 units each, utilizing the `ReLU` activation function, which is effective for Q-learning tasks. The learning rate is set to 10^{-4}, with a larger replay buffer size of 100,000 and a batch size of 32. The target network is updated every 1000 steps, and we use a hard update for the target network with $\tau = 1.0$. The exploration fraction is set to 0.1, with an initial ϵ of 1.0 and a final ϵ of 0.05, using an ϵ-greedy exploration schedule **only in HighwayEnv**. The discount factor is $\gamma = 0.99$, and the training frequency is every 4 environment steps. Gradient steps are performed once per update, and we disable memory optimization for compatibility.

SimpleDQN. The agent utilizes a `SimpleReplayBuffer` to store experience tuples, where each tuple includes `state`, `action`, `reward`, `next_state`, and `done`. The buffer has a capacity of 50,000, and the batch size for each training update is 64. The `SimpleQNetwork` defines the Q-function with a simple MLP architecture, where the number of hidden layers is 2, each with 256 units. The Q-network is trained using the Adam optimizer with a learning rate of 5×10^{-4}. The model is updated with gradient clipping set to `max_grad_norm` = 10.0.

The exploration strategy decays epsilon from an initial value of 1.0 to a final value of 0.05 over 50,000 steps, using an ϵ-greedy exploration schedule. Training begins after 500 steps as defined by `learning_starts`. The discount factor is $\gamma = 0.99$, and the training frequency is every 1 environment step. Gradient steps are performed 1 time per update, and memory optimization is again disabled for compatibility. The agent also tracks training metrics if a `metrics_tracker` is provided, logging metrics such as loss, episode rewards, and episode lengths.

4.1.4 Non-Curriculum Baseline

In the non-curriculum baseline, the agent trains directly trains on a fixed mixture of four road topologies: `highway-v0`, `merge-v0`, `intersection-v0`, and `roundabout-v0`, with varying traffic densities corresponding to the hardest (ie. last) curriculum stage (described in Section 4.1.5). Conceptually, this corresponds to sampling an MDP $\mathcal{M} \sim p(\mathcal{M})$ from a distribution over these tasks and running standard RL. This regime serves as a hard baseline where the agent must learn to navigate diverse traffic patterns simultaneously without structured guidance. We impose a strict budget of 100,000 training steps.

To assess zero-shot generalization, we employ a rigorous evaluation suite distinct from the training distribution. The primary testbed is a five-block `HELDOUT_MULTIBLOCK_SCENARIO` featuring significantly higher traffic densities: `merge-v0` ($\rho = 0.45$), `intersection-v0` ($\rho = 0.45$), `highway-v0` ($\rho = 0.50$), `roundabout-v0` ($\rho = 0.48$), and a second `intersection-v0` ($\rho = 0.50$). Additionally, we subject the agents to task-specific "stress tests" to isolate performance on critical maneuvers under pressure: `Intersection_Stress` ($\rho = 0.45$), `Merge_Stress` ($\rho = 0.50$), and `Roundabout_Stress` ($\rho = 0.40$).

4.1.5 Curriculum Regimen

In the curriculum setting, four-stage curriculum that progresses through the tasks in increasing order of complexity (defined as an MDP sequence). We allocate the same total budget (100,000 steps) across these stages, capped at 25,000 steps each. The curriculum progresses as follows:

- **Stage 1 (Highway Low):** Single block `highway-v0` at traffic density $\rho = 0.20$. Focuses on introductory cruising and lane discipline (Success threshold: 0.80).
- **Stage 2 (Highway Merge):** Two blocks (`highway-v0` at $\rho = 0.25$, `merge-v0` at $\rho = 0.30$) to introduce merging maneuvers (Success threshold: 0.82).
- **Stage 3 (Highway Merge Intersection):** Three blocks (`highway-v0` at $\rho = 0.30$, `merge-v0` at $\rho = 0.35$, `intersection-v0` at $\rho = 0.35$) to combine dense highway driving, merging, and intersection negotiation (Success threshold: 0.85).
- **Stage 4 (All Blocks):** Four blocks (`highway-v0` at $\rho = 0.35$, `merge-v0` at $\rho = 0.40$, `intersection-v0` at $\rho = 0.40$, `roundabout-v0` at $\rho = 0.45$) with block shuffling enabled to encourage order-invariant generalization (Success threshold: 0.90).

We initialize the policy parameters at Stage 1 and train for T_{stage} steps using the stage-specific environment. At the end of each stage we (i) evaluate the current policy on that stage's environment and record mean and standard deviation of returns, and (ii) save a stage-specific checkpoint. The model parameters are then carried forward unchanged to the next stage. After completing all four stages, we evaluate the final policy on the same held-out test configurations $\mathcal{M}_{\text{test}}$ used in the non-curriculum setting (Section 4.1.4).

This protocol allows us to compare curriculum and non-curriculum training under identical sample budgets and observation/action spaces, while varying only the order and structure of the encountered tasks.

4.1.6 Evaluation Metrics

Our primary evaluation metric is the average undiscounted episode return on held-out configurations:

$$\bar{R} = \frac{1}{N} \sum_{i=1}^{N} R^{(i)},$$

where $R^{(i)}$ is the total reward in evaluation episode i and N is the number of evaluation episodes. When available from the environment, we also log auxiliary metrics such as collision rate, success/completion rate, and average episode length.

4.2 MetaDrive Experiments

MetaDrive is a more complex, procedurally generated driving simulator with diverse road layouts and traffic patterns. In this project, we use it to (i) design a complementary curriculum and (ii) train per-stage non-curriculum baselines that we evaluate on challenging held-out scenarios. Due to computational constraints, we restrict MetaDrive experiments to PPO and DQN for non-curriculum and PPO for curriculum.

4.2.1 Curriculum Stage Design

We define four curriculum stages $\{C0, C1, C2, C3\}$ with increasing geometric and traffic complexity. Each stage has its own interaction budget and reward-shaping parameters:

- **C0 (Straight, no traffic).** Single straight road segment (map `"S"`) with no other vehicles; budget $100{,}000$ steps. The reward mainly encourages staying on the road and maintaining moderate speed.
- **C1 (Roundabout, no traffic).** Single roundabout (map `"O"`) with no traffic; budget $150{,}000$ steps. Topology is more complex than C0, but the ego vehicle still drives alone.
- **C2 (Light-traffic PG map).** A procedurally generated 10-block map with light traffic (e.g., density $\rho \approx 0.05$); budget $200{,}000$ steps. The reward increases emphasis on speed and safety, penalizing collisions and traffic violations more strongly.
- **C3 (Dense-traffic PG map).** A 20-block map with dense traffic (e.g., $\rho \approx 0.30$); budget $200{,}000$ steps. This is the most challenging stage, with a greater success bonus and higher penalties (compared to other stages).

The total MetaDrive curriculum budget is the sum of the four stage budgets. In addition, we also add a progressively larger step penalty for each stage which encourages the ego vehicle to reach its destination faster.

4.2.2 Environment Configuration and Reward Shaping

For each stage, we instantiate a MetaDrive environment with the specified map and traffic level and wrap it in a custom `CurriculumRewardWrapper`. This wrapper combines:

- a base term proportional to the original environment reward,
- a speed term that rewards driving up to a stage-specific speed limit,
- penalties for collisions, off-road events, and traffic violations,
- a terminal success bonus for safe arrival at the destination,
- and a small per-step penalty to discourage unnecessarily long episodes.

For DQN, we apply a discrete-action wrapper that maps a small set of steering–throttle pairs to discrete actions; PPO operates directly in the continuous action space. All Algorithm hyperparameters remain similar for MetaDrive as HighwayEnv, briefed in Section 4.1.3.

4.2.3 Non-Curriculum Baselines

At this checkpoint, we use the stages above to define per-stage non-curriculum baselines. For each stage C_k and each algorithm (PPO, DQN), we:

1. train a separate model from scratch on that stage only, using the stage's budget as the total number of environment steps, and
2. evaluate the trained policy on the same stage.

For each run, we log average return, and episode length. These stage-wise baselines will serve as reference points for future curriculum experiments in MetaDrive.

4.2.4 Held-out Test Scenarios

To assess generalization, we additionally evaluate each trained policy on an unseen held-out scenario. A fixed six-block map (denoted `"SCrRXO"`) consisting of straight, circular, ramp, intersection, and roundabout segments, with medium traffic. This tests transfer to a more complex but structured layout.

For the held-out scenario, we run a fixed number of evaluation episodes and log the same metrics as above. Overall, HighwayEnv and MetaDrive together provide a progression from lightweight, fast-to-train tactical tasks to richer, procedurally generated environments.

5 Results

In this section, we present the quantitative results of our experiments. We first analyze the performance on the MetaDrive simulator, comparing Non-Curriculum baselines against our Curriculum learning approach. We then present the comparative results for the HighwayEnv benchmark.

5.1 MetaDrive Results

For MetaDrive, we evaluated the performance of PPO and DQN agents for Non-Curriculum and only PPO for Curriculum due to computational constraints. A more holistic model comparison in context of both frameworks is presented for HighwayEnv. We focus on four key metrics: the mean episode reward, collision rate, and success rate during training stages and the zero-shot generalization performance on the held-out `SCrRXO` map.

5.1.1 Non-Curriculum Baselines

We trained independent non-curriculum agents for DQN and PPO directly on the target configurations. Table 1 summarizes the final training performance across all four complexity stages.

Algorithm	C0 (Straight)	C1 (Roundabout)	C2 (Light)	C3 (Dense)
DQN	136.2 ± 5.1	47.3 ± 0.0	38.9 ± 0.0	32.5 ± 0.0
PPO	128.5 ± 4.2	52.1 ± 8.6	43.5 ± 6.8	41.2 ± 7.5

Table 1: Non-Curriculum Performance: Mean Training Reward $\pm$ Std (Final 10k Steps)

Figure 1 illustrates the training progression for DQN. On the simplest stage (C0, red line), the agent rapidly converges to a high mean reward of ≈ 136.2. However, as complexity increases, a distinct pattern emerges. On stages C1 (blue), C2 (green), and C3 (orange), the reward curves exhibit a sharp initial rise followed by a complete plateau. Specifically, for C3, the reward stabilizes at exactly 32.5 with zero variance (Standard Deviation ≈ 0.0). Corresponding metrics logged during these epochs indicate a collision rate of 1.0 and a success rate of 0.0.

Meanwhile, Figure 2 displays the PPO training dynamics. In contrast to DQN, the PPO curves do not flatline. On the Dense Traffic stage (C3, blue line), the reward fluctuates between 0 and 60, averaging ≈ 41.2 (Table 1). The Light Traffic stage (C2, green line) shows extreme volatility, oscillating between 0 and 250. Unlike DQN, PPO maintains a non-zero standard deviation in rewards throughout the training budget, indicating varied episode termination conditions rather than deterministic failure in DQN.

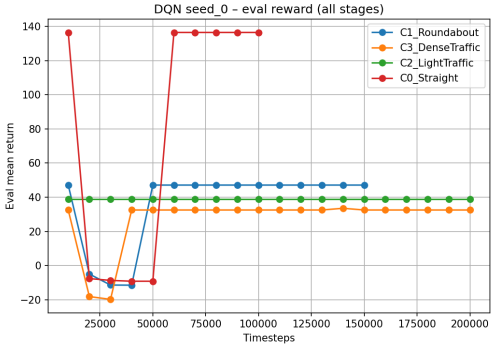

Figure 1: DQN Training Reward (Non-Curriculum). Note the red line (C0) converging high, while C1, C2, and C3 lines flatline completely, indicating deterministic episode outcomes.

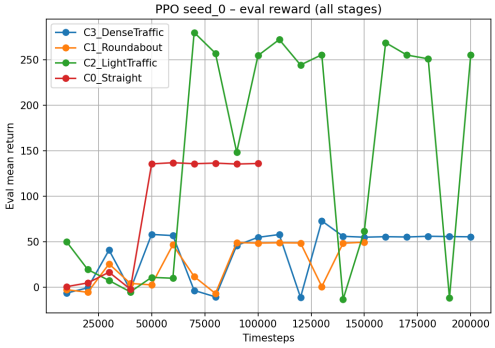

Figure 2: PPO Training Reward (Non-Curriculum). The C2 (green) and C3 (blue) curves show continued variance, contrasting with the flatlines observed in DQN.

5.1.2 Curriculum Learning (PPO)

We trained a PPO agent using the four-stage curriculum ($C0 \rightarrow C1 \rightarrow C2 \rightarrow C3$) described in Section 4.3.1. Table 2 summarizes the agent's performance at the end of each curriculum stage.

Stage	Training Mean Reward	Held-out (SCrRXO) Reward	Status
C0 (Straight)	135.6 ± 5.1	32.1	Solved
C1 (Roundabout)	76.8 ± 8.2	64.3	Solved
C2 (Light Traffic)	265.3 ± 12.4	-49.9	Overfitting
C3 (Dense Traffic)	43.3 ± 6.5	45.5	Recovered

Table 2: PPO Curriculum Progression: Training and Held-out Performance

Notably, we observed a "catastrophic forgetting" event after Stage C2. As seen in Table 2, while the training reward on C2 was high (265.3), the held-out performance dropped to -49.9. However, training on the final Stage C3 allowed the agent to recover, achieving a final held-out score of 45.5.

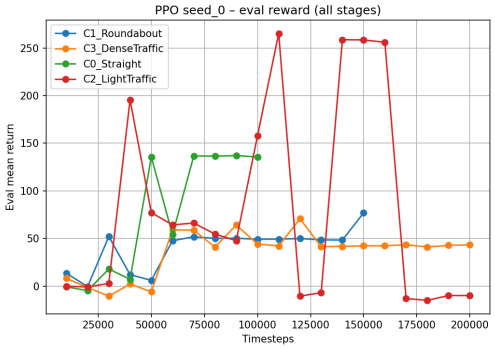

Figure 3: Curriculum PPO Training Progression. The agent sequentially masters Straight (Red), Roundabout (Blue), and Light Traffic (Green) before tackling Dense Traffic (Orange).

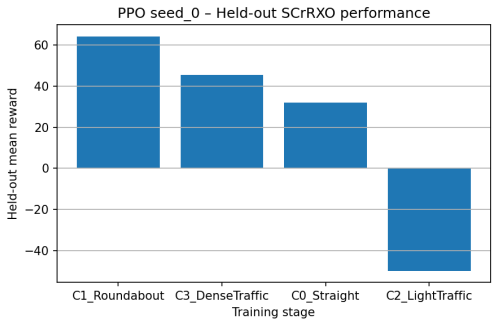

Figure 4: Held-out performance on `SCrRXO` for PPO Curriculum. The drop after Stage C2 indicates overfitting to sparse traffic, while the recovery after C3 demonstrates the curriculum's final robustness.

5.1.3 Curriculum vs. Non-Curriculum

We compare the final Curriculum PPO agent against the Non-Curriculum PPO baseline specifically on the dense traffic task (C3) to demonstrate the curricula's ability to perform on complex real-world scenarios. Table 3 highlights the performance gains and efficiency improvements.

Metric	Non-Curriculum	Curriculum	Change
Mean Training Reward (C3)	41.2	**43.3**	$+5.1\%$
Held-out SCrRXO Reward	43.5	**45.5**	$+4.6\%$
Total Training Time	$\approx$ 6 hrs	$\approx$ **4.5** hrs	$\mathbf{-25\%}$

Table 3: Efficiency and Performance Comparison (MetaDrive PPO)

As shown in Table 3, the Curriculum approach yielded marginal but consistent gains in both training reward ($+5.1\%$) and held-out generalization ($+4.6\%$). More significantly, the curriculum agent achieved these results with a **25% reduction in total wall-clock training time**. By progressively exposing the agent to harder tasks, the curriculum avoided the extended "warm-up" periods of low-quality exploration seen in the non-curriculum runs on complex maps.

5.2 HighwayEnv Results

While our MetaDrive experiments were limited to PPO for the curriculum regime due to computational constraints, the HighwayEnv benchmark allows for a more holistic comparison of both PPO and DQN

across both frameworks. We focus on four key metrics: mean episode reward across testing scenarios, mean training times, collision and success rates during training stages, and zero-shot generalization performance on the held-out multi-block stress tests.

5.2.1 Curriculum vs. Non-Curriculum

Algorithm	Regime	Mean reward	Mean success rate	Mean completion rate	Train time (hr)	Timesteps
DQN	non-curriculum	59.9745	0.5000	0.6933	0.6588	100000
DQN	curriculum	66.7224	0.5000	0.8038	0.5759	95000
SimpleDQN	non-curriculum	64.8184	0.5667	0.6683	0.7578	100000
SimpleDQN	curriculum	68.5214	0.4500	0.8325	0.8559	100000
PPO	non-curriculum	68.9275	0.7000	0.6875	1.2550	100000
PPO	curriculum	99.9139	0.8500	0.8388	0.8921	80000

Table 4: Summary of evaluation metrics for each algorithm under non-curriculum and curriculum training regimes.

Algorithm	Δ Mean reward	Δ Success rate	Δ Completion rate	Δ Train time (hr)
DQN	$+11.25\%$	0.00%	$+15.92\%$	-12.58%
SimpleDQN	$+5.71\%$	-20.59%	$+24.57\%$	$+12.95\%$
PPO	$+44.96\%$	$+21.43\%$	$+22.00\%$	-28.92%

Table 5: Curriculum effect percentage (curriculum minus non-curriculum) for each algorithm.

Table 4 reports the primary evaluation metrics: mean episode reward, mean success rate, mean completion rate, wall-clock training time in hours, and total environment timesteps. Table 5 shows the curriculum effect (curriculum minus non-curriculum) to make relative improvements easier to read.

Across the three algorithms we observe three main behavioural patterns:

1. **PPO exhibits the strongest gains under curriculum learning.** PPO shows a substantial +44.96% increase in mean reward, accompanied by sizable improvements in success rate (+21.43%) and completion rate (+22.00%). Training time decreases by nearly 29%. These large percentage gains reinforce that PPO benefits disproportionately from staged skill acquisition, likely due to its on-policy updates, stable advantage estimates, and smooth policy changes.

2. **DQN demonstrates moderate but consistent improvements.** Mean reward improves by +11.25% and completion rate by +15.92%, with no change in success rate. Training time decreases by 12.58%. This profile suggests that curriculum helps DQN learn more reliable episode trajectories, but the ϵ-greedy exploration and replay-driven updates limit gains in strict binary success.

3. **SimpleDQN shows mixed effects.** While mean reward and completion rate increase by +5.71% and +24.57% respectively, success rate drops sharply by 20.59%, and training time increases by 12.95%. This combination indicates that the handcrafted SimpleDQN may be more sensitive to non-stationarity introduced across curriculum stages, potentially overfitting to reward structure while struggling with stricter success criteria.

The percentage-based results above highlight the overall quantitative impact of curriculum learning across PPO, DQN, and SimpleDQN. However, these metrics alone cannot fully explain *why* certain algorithms benefit more than others, nor why some show mixed effects. To understand the underlying behavioural dynamics — such as how agents navigate merges, intersections, or dense traffic, where they fail, and which behaviours transfer across stages — we now turn to a deeper qualitative and quantitative analysis.

6 Discussion

6.1 MetaDrive Analysis

The results presented in Section 5.1 highlight a critical difference in how value-based (DQN) and policy-gradient (PPO) methods handle the complex trade-off between speed and safety in autonomous driving. A comprehensive analysis is done via simulation runs in HighwayEnv as MetaDrive's simulations crash after an episode, usually in a small number of frames (3-4 seconds).

6.1.1 Non-Curriculum Dynamics

The "flatline" behavior observed in Figure 1 and the zero standard deviation shown in Table 1 for stages C1-C3 are indicative of reward hacking, similar to the findings of Mnih et al. [2015], where agents in similar settings exploited simplistic strategies, leading to reward-maximization that ignored long-term consequences. The agent converged to a deterministic strategy: maximizing acceleration to accumulate distance-based rewards before inevitably colliding. Since the accumulated velocity reward outweighed the fixed collision penalty, the agent effectively learned that "crashing fast" is optimal. The discrete action space of DQN likely exacerbated this, preventing the fine-grained control necessary for high-speed evasion in dense traffic. This is also evidenced by a collision rate of 1.0 and a success rate of 0.0 across the final evaluation epochs.

In contrast, PPO converged to a safe but slow local optimum. The persistent variance observed in Figure 2 (blue and green lines) confirms that the agent was actively attempting to survive, resulting in varied episode lengths rather than the deterministic crashes of DQN. While this approach is safer—evidenced by the non-zero survival rates—the lower peak rewards in Table 1 (e.g., 41.2 vs potential higher scores) suggest the policy became overly conservative, often failing to reach the destination within the time limit due to excessive caution, emphasizing the need for a balance of safety and exploration Fahim et al. [2018]. This safety-first approach resulted in lower variance in episode returns and a higher survival rate in dense traffic scenarios compared to DQN. While PPO's absolute reward accumulation was sometimes lower due to its reluctance to drive at maximum speed, its behavior was more aligned with the desired characteristics of an autonomous agent.

6.1.2 Curriculum Dynamics

The curriculum successfully stabilized early training but introduced a challenge at Stage C2. The sharp drop in held-out performance (-49.9) shown in Table 2 suggests that training on Light Traffic (C2) caused the agent to overfit to empty roads, unlearning the collision-avoidance behaviors developed in Stage C1. However, the curriculum was ultimately validated by the recovery in Stage C3, where the agent successfully adapted its high-speed policy from C2 to the dense traffic of C3.

This adaptive process also explains the training time efficiency noted in Table 3. By entering the final dense traffic stage with pre-learned primitives for steering (from C1) and speed control (from C2), the curriculum agent required fewer samples to stabilize than the non-curriculum agent, which had to learn these basic controls simultaneously with complex collision avoidance. This demonstrates that while the final performance gains were modest, the curriculum structure provided a more compute-efficient path to convergence.

PPO's failure to learn aggressive maneuvers that might be necessary for efficiency were helped mitigated by the curriculum approach through forcing the agent to master vehicle control in simpler settings before facing dense traffic. However, the fundamental conservative bias of the policy remained. This suggests that for complex simulators like MetaDrive, reward shaping needs to be dynamically adjusted—perhaps penalizing passivity more in later stages—to encourage more active driving behaviors without compromising safety.

6.2 HighwayEnv Analysis

The results presented in Section 5.2, together with qualitative behavioral observations from simulation runs, reveal clear distinctions in how PPO, DQN, and SimpleDQN respond to the HighwayEnv driving task under curriculum and non-curriculum training. Although all agents acquired basic path-following abilities, their stability, risk preferences, and generalization capacity varied significantly.

6.2.1 PPO Behavioral Dynamics

The PPO agent exhibits a distinctive "throttle-as-knob" control strategy across both curriculum and non-curriculum regimes. Behavioral observations from simulation runs reveal that PPO strongly prefers lane-keeping and safe following distances, even in relatively sparse traffic conditions. When any vehicle appears in the observation space, the agent immediately reduces speed, resuming acceleration only after the vehicle exits its field of view. This risk-averse behavior explains PPO's higher baseline success rate (0.70 in non-curriculum) compared to DQN variants (0.50-0.57), but also accounts for its initially lower mean reward (68.93 vs. potential maximum) due to conservative speed profiles.

Curriculum learning amplifies PPO's strengths while introducing adaptive lane-changing behavior. The staged progression through highway-v0 $\rightarrow$ merge-v0 $\rightarrow$ intersection-v0 $\rightarrow$ roundabout-v0 forces the agent to develop active maneuvering skills that complement its inherent caution. The curriculum PPO agent demonstrates more frequent lane changes when beneficial, particularly in merge and roundabout scenarios, while maintaining the safe following behavior learned in early stages. This combination produces the dramatic +44.96% reward improvement and +21.43% success rate gain observed in Table 5.

However, we observed evidence of catastrophic forgetting during curriculum transitions, mirroring the Stage C2 phenomenon in MetaDrive. When evaluated on basic highway-v0 scenarios after training on the complex configuration, the curriculum PPO agent occasionally exhibited performance degradation compared to its starting checkpoint (initial curriculum). The agent would sometimes over-apply complex intersection behaviors (e.g., unnecessary deceleration, overly cautious gap acceptance) to simple highway cruising tasks. This suggests that while curriculum learning improves final-stage performance, it can erode mastery of earlier, simpler skills; a trade-off that may be acceptable if the target deployment is the complex environment.

6.2.2 DQN Behavioral Dynamics

DQN demonstrates the same fundamental pathology observed in MetaDrive, though less severely due to HighwayEnv's simpler dynamics and discrete action space. Behavioral analysis reveals that the non-curriculum DQN agent frequently adopts high-speed, collision-prone policies, particularly in dense traffic scenarios. The agent exploits the distance-based reward component by maximizing velocity before inevitable crashes, similar to the "crashing fast" strategy identified in MetaDrive. The epsilon-greedy exploration schedule decays too rapidly, causing the agent to commit to suboptimal Q-value estimates before adequately exploring safe maneuvering strategies.

Curriculum learning provides partial mitigation. The +11.25% reward improvement and +15.92% completion rate increase (Table 4) indicate that staged exposure helps DQN learn more robust Q-values in early stages, which transfer forward. By mastering lane discipline in Stage1_Highway_Low (density 0.20) before facing merges and intersections, the curriculum DQN agent develops better credit assignment for safe behaviors. However, the zero change in success rate reveals that DQN's discrete action space and off-policy learning fundamentally limit its ability to execute the fine-grained control needed for consistent success in dense, multi-task scenarios.

The 12.58% training time reduction for curriculum DQN likely stems from faster convergence in early stages, where the simplified environments allow efficient replay buffer population with meaningful transitions. In contrast, non-curriculum DQN wastes early training samples on the complex Stage4_All_Blocks composite map, where random exploration yields predominantly low-reward, collision-terminating episodes.

6.2.3 SimpleDQN Behavioral Dynamics

SimpleDQN's contradictory results stem directly from its aggressive hyperparameter configuration. Compared to stable-baselines3's DQN, SimpleDQN employs 4x more frequent updates (train_freq=1 vs 4), a 50% smaller replay buffer (50k vs 100k), 5x higher learning rate (5e-4 vs 1e-4), and 2x more frequent target network updates (every 500 vs 1000 steps).

These settings optimize for rapid single-task convergence but create severe Q-value instability during curriculum transitions. When the agent progresses from Stage1_Highway_Low (=0.20, highway-only) to Stage2_Highway_Merge (=0.30, merge introduction), the small replay buffer cannot retain

sufficient Stage1 experiences while the high learning rate forces rapid Q-network adaptation to the new state distribution. This causes reward exploitation without skill retention: the agent learns to maximize stage-specific intermediate rewards (distance traveled, speed bonuses) while violating the stricter success criteria that require collision-free, rule-compliant completion.

The 12.95% training time increase reflects the cost of Q-value re-stabilization at each stage boundary; approximately 2,000 additional steps per transition to recover from overestimation-induced instability. Unlike PPO's smooth policy transfers or standard DQN's conservative updates, SimpleDQN's aggressive configuration makes it fundamentally unsuitable for curriculum learning without substantial hyperparameter retuning. This serves as a cautionary example: algorithms must be designed for stage-transition robustness, not just final-stage performance.

6.2.4 Impact of Curriculum Learning

Despite expectations, curriculum learning did not result in consistent or significant improvements in measured performance. While PPO occasionally benefited qualitatively, such as delaying collisions with oncoming traffic, it also exhibited increased instability and unnecessary lateral movement. For DQN, curriculum training did not mitigate unsafe behaviours and sometimes reduced behavioural flexibility.

A recurring and important pattern was **catastrophic forgetting**. When agents trained on more complex stages were later tested on earlier, simpler environments, their performance degraded noticeably. This indicates that the curriculum did not reinforce foundational behaviours and may have encouraged overspecialization to the final stage of training.

7 Conclusion

In this work, we investigated the efficacy of curriculum-based Deep Reinforcement Learning for autonomous driving, comparing it against standard non-curriculum training across both tactical (HighwayEnv) and complex procedural (MetaDrive) environments.

Our experiments revealed that the impact of curriculum learning is highly context-dependent. In simpler, tactical environments like HighwayEnv curriculum learning enriched certain qualitative behaviours, especially for PPO. But it did not translate into measurable performance gains and, in some cases, introduced instability or forgetting. The contrasting behavioural profiles of PPO and DQN highlight the need for algorithm-specific curriculum designs: value-based agents may require careful reward shaping or structured stage transitions, whereas policy-gradient agents may benefit from smoothing mechanisms to maintain behavioural stability across stages.

In the high-fidelity MetaDrive simulator, much like HighwayEnv, non-curriculum baselines either succumbed to reward hacking (DQN) or excessive conservatism (PPO). Although, our four-stage curriculum enabled the PPO agent to mildly bridge the gap between safety and task completion in an efficient amount of time, they exhibited conservative driving behaviors that may be suboptimal for real-world efficiency. Furthermore, the catastrophic forgetting observed during stage transitions indicates that simple sequential training is insufficient for robust multi-task retention, and the jump in difficulty through the stages might have been too harsh. Future work can focus on addressing the remaining safe but slow bias of our policies by integrating dynamic reward shaping and exploring hybrid architectures that combine the assertiveness of value-based methods with the stability of policy gradients.

8 Contributions

Raahim A Samad Poonawala:

- Conceptualization and implementation of MetaDrive experiments, including curriculum design.
- Conducted literature review on reinforcement learning training mechanisms and reward structures.
- Responsible for the compilation, and analysis of MetaDrive results in the report.

Muhammad Bin Tariq:

- Led the literature review for the report -> Authored the initial sections of the paper, including the abstract, introduction, and problem formulation.
- Conducted preliminary simulations and setup for HighwayEnv experiments.

Saim Bilal:

- Proposed the curriculum-learning idea.
- Executed the full training regimen for HighwayEnv experiments (curriculum and non-curriculum).
- Responsible for the compilation, and discussion of HighwayEnv results.

References

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning (ICML '09)*, pages 41–48. ACM / Machine Learning Research, 2009. doi: 10.1145/1553374.1553380.

Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator, 2017. URL `https://arxiv.org/abs/1711.03938`.

Reza Fahim, Xin Shia, and Cho-Jui Hsieh. Safety-critical deep reinforcement learning for autonomous driving. *IEEE Access*, 6:73641–73653, 2018.

Edouard Leurent. An environment for autonomous driving decision-making. `https://github.com/eleurent/highway-env`, 2018.

Quanyi Li, Zhenghao Peng, Lan Feng, Qihang Zhang, Zhenghai Xue, and Bolei Zhou. Metadrive: Composing diverse driving scenarios for generalizable reinforcement learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

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

Sanmit Narvekar, Bei Peng, Matteo Leonetti, Jivko Sinapov, Matthew E. Taylor, and Peter Stone. Curriculum learning for reinforcement learning domains: A framework and survey, 2020. URL `https://arxiv.org/abs/2003.04960`.

Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22(268):1–8, 2021. URL `http://jmlr.org/papers/v22/20-1364.html`.

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

Bhargava Uppuluri, Anjel Patel, Neil Mehta, Sridhar Kamath, and Pratyush Chakraborty. Curla: Curriculum learning based deep reinforcement learning for autonomous driving, 2025. URL `https://arxiv.org/abs/2501.04982`.