# Self-Improving Autonomous Vehicles via Real-World Reinforcement Learning

**Daehyeok Kwon**[1], **Seung-Woo Seo**[1] and **Sang-Hyun Lee**[2*]
[1]Seoul National University
[2]Ajou University
{rnzchy03, sseo}@snu.ac.kr, sanghyunlee@ajou.ac.kr

## Abstract

End-to-end autonomous driving systems have demonstrated advantages over traditional modular systems. Despite this progress, these end-to-end systems still struggle to be deployed in real-world driving environments, as they inevitably encounter undertrained scenarios in which autonomous vehicles may take unsafe actions. Reinforcement Learning (RL) provides a theoretical framework for addressing this challenge by enabling autonomous vehicles to self-improve: continuously collecting additional scenarios and learning from them. However, training autonomous vehicles with RL is not straightforward in the real world. Collecting real-world driving data involves costly interactions with the environment, and significant human intervention is required both to prevent autonomous vehicles from entering unsafe states and to reset them for subsequent episodes. In this paper, we introduce a novel real-world RL algorithm that allows autonomous vehicles to collect informative scenarios and learn from them with minimal human intervention. Our algorithm considers the learning progress of autonomous vehicles to identify informative scenarios and abort episodes before they enter unsafe states. To evaluate our algorithm, we introduce challenging urban driving tasks that require autonomous vehicles to reset themselves to initial states. The experimental results show that our real-world RL algorithm outperforms baselines with much less human intervention.

## 1 Introduction

Autonomous driving has been actively researched for decades. Conventional autonomous driving systems typically adopt modular architectures where each module has an independent objective. The modular systems exhibit high interpretability and maintainability. However, they often lead to suboptimal driving performance, as modules are optimized locally and information bottlenecks occur between them [Shao *et al.*, 2023; Jia *et al.*, 2023; Chen *et al.*, 2024]. Recently, end-to-end autonomous driving systems have emerged as a promising approach to overcome the drawbacks of modular systems. In contrast to modular systems, end-to-end systems are implemented as a single differentiable and trainable model, allowing them to mitigate information bottlenecks and globally optimize all components with a unified objective [Chen *et al.*, 2024]. Many recent works have shown that end-to-end systems significantly outperform modular systems in diverse driving scenarios [Bojarski *et al.*, 2016; Hu *et al.*, 2023; Jiang *et al.*, 2023; Tong *et al.*, 2023; Weng *et al.*, 2024; Liao *et al.*, 2025; Zhou *et al.*, 2025].

Despite these advantages, end-to-end systems face difficulties in being deployed in real-world driving environments, as these systems unavoidably encounter undertrained scenarios where they may make unsafe decisions. Reinforcement learning (RL) is a promising approach to address this challenge, as it provides a theoretical framework that enables autonomous vehicles to self-improve by continuously collecting additional scenarios and learning from them. Several recent works have demonstrated that leveraging RL to collect additional scenarios and learn from them in simulated driving environments helps reduce undertrained scenarios [Gao *et al.*, 2025; Huang *et al.*, 2025]. However, deploying autonomous vehicles trained in simulation to the real world is still not straightforward due to the fidelity gap between simulated and real-world driving environments.

Directly training real-world autonomous vehicles with RL is a simple approach to overcome the fidelity gap [Kendall *et al.*, 2019]. However, this approach has several critical issues. First, collecting real-world driving data involves a large number of costly interactions with the environment [Gao *et al.*, 2025]. Since naively or randomly collecting data leads to unnecessary costs, it is crucial to identify and selectively acquire informative data. Second, repetitive human intervention is required during the real-world training process [Yahya *et al.*, 2017; Chebotar *et al.*, 2017; Kendall *et al.*, 2019]. Specifically, humans must prevent autonomous vehicles from entering unsafe states and reset them for subsequent episodes. While this issue is often neglected, such intervention severely hinders learning efficiency. Consequently, it is essential to develop a method that enables autonomous vehicles to gather informative transitions and learn from them with minimal human intervention.

*Corresponding author.

In this paper, we propose a new real-world RL algorithm for self-improving autonomous vehicles. Our algorithm identifies informative scenarios that are neither too challenging nor too familiar for autonomous vehicles being trained. The key idea is to estimate the expected amount of obtainable information from under-explored states for each scenario. Our algorithm also determines when to abort episodes before autonomous vehicles enter unsafe states by taking into account their learning progress. It is based on the observation that autonomous vehicles may not know how to avoid unsafe states in under-explored states. Finally, our algorithm utilizes rule-based autonomous driving behaviors to safely return autonomous vehicles to initial states for the subsequent episodes. It is inspired by prior experimental results, demonstrating that while the rule-based behaviors cannot achieve human-like driving performance, they can perform safe and rule-abiding behaviors in diverse driving scenarios [Nothdurft *et al.*, 2011; Geiger *et al.*, 2012; Ziegler *et al.*, 2014; Broggi *et al.*, 2015].

The main contribution of our work is a real-world RL algorithm for self-improving autonomous vehicles that can continuously collect informative scenarios and learn from them. To evaluate our algorithm, we introduce challenging urban driving tasks that require autonomous vehicles to reset to initial states on their own. The experimental results demonstrate that our real-world RL algorithm identifies informative scenarios and prevents autonomous vehicles from entering unsafe states, outperforming baselines with much less human intervention across diverse driving tasks.

## 2 Related Works

Many previous works have demonstrated that RL enables an autonomous vehicle to continuously collect diverse driving scenarios and learn from them on its own [Isele *et al.*, 2018; Toromanoff *et al.*, 2020; Lee and Seo, 2024]. Isele *et al.* [2018] showed that autonomous vehicles trained with RL successfully understand and handle a variety of occluded intersection scenarios. Toromanoff *et al.* [2020] proposed an end-to-end RL algorithm that allows autonomous vehicles to obey traffic lights and avoid surrounding objects. While these previous works achieved appealing performance in diverse driving scenarios, they assumed that autonomous vehicles are trained in simulated environments. Deploying autonomous vehicles trained in simulations to the real world is still infeasible due to the fidelity gap between simulated and real-world environments. We introduce a real-world RL algorithm that enables autonomous vehicles to continuously gather diverse driving scenarios and learn by themselves in the real world.

Directly training agents with RL in the real world has received significant attention as it simply avoids the fidelity gap problem [Yahya *et al.*, 2017; Chebotar *et al.*, 2017; Kendall *et al.*, 2019]. Yahya *et al.* [2017] and Chebotar *et al.* [2017] showed that their RL algorithms enable robotic arms to successfully open doors or hit a puck into a goal using a hockey stick. Kendall *et al.* [2019] demonstrated that RL algorithms enable real-world autonomous vehicles to learn driving strategies for lane following. These works highlight the feasibility of addressing real-world problems with RL, but require repeated human intervention during their reset processes. For example, in [Yahya *et al.*, 2017] and [Chebotar *et al.*, 2017], human operators close doors or return the puck to the front of the robotic arms between every episode. In [Kendall *et al.*, 2019], safety drivers take over control to return autonomous vehicles to the center of the lane whenever they deviate from the target lane. These human interventions significantly hinder training efficiency in the real world.

Several recent works automate their reset processes to reduce required human interventions [Levine *et al.*, 2018; Zhu *et al.*, 2019; Nagabandi *et al.*, 2020; Zeng *et al.*, 2020]. Levine *et al.* [2018] collected object grasping data by using metal bins with sloped sides to prevent objects from wedging into corners. They collect approximately 800,000 grasping trials, requiring human intervention only to replace objects in the bins. Nagabandi *et al.* [2020] trained a real-world robotic hand to rotate two Baoding balls around the palm. They automate the reset process by using a 7-DoF Franka-Emika arm to pick up dropped balls and return them to the robotic hand. These works demonstrated that automating the reset process effectively reduces human intervention, leading to better training efficiency in the real world. To the best of our knowledge, no previous work has discussed how to automate the reset process for training autonomous vehicles in the real world. Our algorithm leverages rule-based autonomous driving algorithms to safely reset autonomous vehicles. Furthermore, it identifies informative scenarios and prevents them from entering unsafe states.

## 3 Proposed Method

The goal of our work is to empower autonomous vehicles with the ability to self-improve. To do so, our real-world RL algorithm enables autonomous vehicles themselves to continuously collect informative scenarios and learn from them. Specifically, our algorithm learns both where to return autonomous vehicles to collect informative transitions in the following episodes and when to abort episodes to prevent them from entering unsafe states. Figure 1 provides an overview of the training procedure of our algorithm.

### 3.1 Problem Formulation

We use a Markov decision process (MDP) to model an environment. MDP is defined as the tuple $(S, A, P, R, \rho_0, \gamma)$, where $S$ denotes the set of states, $A$ denotes the set of actions, and $P : S \times A \times S \to \mathbb{R}^+$ denotes the state transition model. The function $R : S \times A \times S \to \mathbb{R}$ denotes the reward function, which outputs a scalar feedback called a reward, $r$. $\rho_0 : S \to \mathbb{R}^+$ represents the initial state distribution, and $\gamma$ represents the discount factor. Key components that represent behaviors of an agent are the forward policy and the forward state-action value function: The forward policy $\pi_f(a|s)$ maps a state to a probability distribution over actions and the forward state-action value function $Q^{\pi_f}(s, a)$ represents the expected return obtained when the agent takes the action $a$ in the state $s$ and follows the policy $\pi_f$.

RL aims to find the optimal policy $\pi_f^*$ that maximizes the expected cumulative rewards when the state transition model is unknown. While RL algorithms have achieved remarkable

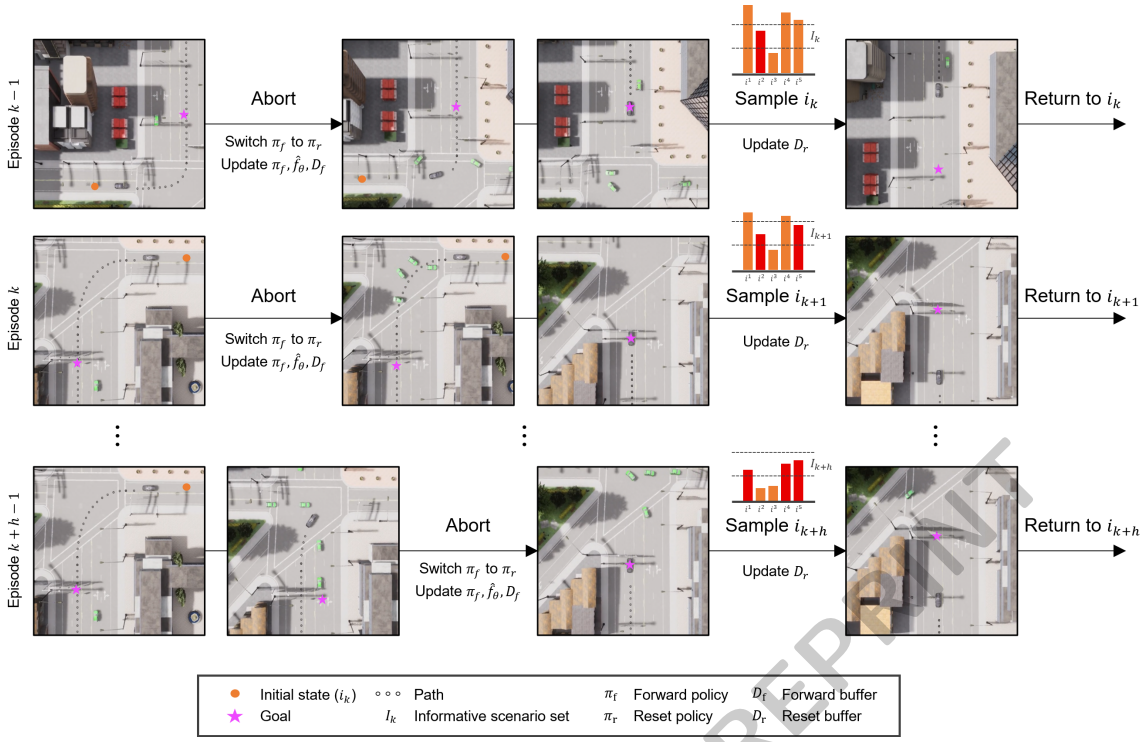


Figure 1: Overview of our real-world RL algorithm. The episode is aborted when the estimated state novelty is too high. After that, an autonomous vehicle is controlled with the reset policy to return to the initial state of the next scenario without human intervention. The next scenario is sampled from informative scenarios. The timing of the switch to the reset policy is pushed further back as the training progresses.

results in various domains, they typically require countless interactions with environments and assume that resetting an agent is managed outside the training procedure [Mnih *et al.*, 2015; Vinyals *et al.*, 2019]. This assumption makes it difficult to apply such algorithms to train autonomous vehicles in the real world, as interacting with real-world driving environments is costly and resetting autonomous vehicles demands significant human intervention.

Our work seeks to address three main challenges that are critical to train autonomous vehicles with RL in the real world. First, we must identify which scenarios can provide autonomous vehicles with informative transitions. Scenarios that are too easy or too difficult can preclude autonomous vehicles from obtaining informative transitions or can easily lead them into risky states, such as collisions with surrounding objects. Second, we must decide when to abort an episode to prevent autonomous vehicles from entering unsafe states. Entering unsafe states involves significant cost to recover and may even make further training impossible. Finally, after aborting an episode, we must safely return autonomous vehicles to an initial state for a subsequent episode while complying with traffic rules. In the remainder of this section, we discuss how our real-world RL algorithm addresses each of these challenges in detail.

## 3.2 Identifying Informative Scenarios

We hypothesize that informative scenarios are neither too challenging nor too familiar for an autonomous vehicle being trained. If scenarios are too challenging, an autonomous vehicle easily enters unsafe states and fails to gather sufficient transitions. Conversely, if scenarios are too familiar, an autonomous vehicle may rarely collect informative transitions even when it completes episodes without aborts. Identifying informative scenarios is therefore critical for enabling an autonomous vehicle to collect sufficient useful transitions.

To identify informative scenarios based on the learning progress of an autonomous vehicle, our algorithm estimates the expected amount of obtainable information from under-explored states within a scenario, $e_i$, where $i$ is an index of scenarios, and then determines the scenario for the $k$th episode, $i_k$, as follows:

$$i_k \sim \mathrm{Unif}(I_k), \quad \text{where } I_k \triangleq \{i \in I \mid \lambda_1 \leq e_i \leq \lambda_2\}. \quad (1)$$

$I_k$ is the set of informative scenarios for the $k$th episode, $i$ is a scenario included in the set of all scenarios $I$, and $\lambda_1$ and $\lambda_2$ are its lower and upper bounds, respectively. The scenario for the subsequent episode is uniformly sampled from the set of informative scenarios identified at the current episode. This can prevent an autonomous vehicle from entering scenarios that involve too many under-explored or familiar states. We would like to emphasize that the set of informative scenarios can adapt to the learning progress of an autonomous vehicle, suggesting that our real-world RL algorithm implicitly generates a curriculum for scenarios. Experimental results described in Section 4.4 indicate that identifying informative scenarios improves sample efficiency.

Estimating the expected amount of obtainable information from under-explored states within a scenario has two

critical challenges. First, there are no dominant or widely accepted approaches for estimating the amount of obtainable information. To address this challenge, our algorithm leverages the concept of state novelty to approximate the obtainable information. This is motivated by the intuition that novel states tend to provide higher information gain. Fortunately, several recent works have introduced feasible state-novelty estimation approaches [Bellemare *et al.*, 2016; Pathak *et al.*, 2017; Burda *et al.*, 2018]. We use random network distillation (RND) [Burda *et al.*, 2018] to estimate the novelty of states, as it is simple to implement and works well in high-dimensional observations. RND defines a target network $f(s) : S \to \mathbb{R}^k$ and a predictor network $\hat{f}_\theta(s) : S \to \mathbb{R}^k$. The target network is randomly initialized and then fixed, and the predictor network is trained to minimize the expected prediction error $\|\hat{f}_\theta(s) - f(s)\|^2$ with collected transitions. RND empirically demonstrated that the prediction error is higher in unseen states than in frequently explored states. Based on this interesting experimental result, we regard the prediction error as the estimated novelty of a state.

The other challenge, which may seem paradoxical, is to access under-explored states within a scenario. Our key idea to address this challenge is to collect transitions when an autonomous vehicle is returned to the next initial state after the current episode is aborted. These transitions collected during the reset process involve under-explored states, as an autonomous vehicle is trained with transitions collected before episodes are aborted. Our algorithm contrasts with most previous works in that they neither collect transitions during the reset phase nor leverage them. Note that an autonomous vehicle is expected to pass the goal of the current episode during the reset process, and our algorithm does not collect transitions after an autonomous vehicle reaches the goal state, as such transitions are unrelated to driving scenarios.

As a result, the expected amount of obtainable information from under-explored states within a scenario $i$ can then be estimated as follows:

$$e_i = \mathbb{E}_{s \sim D_r^i} \left[ \|\hat{f}_\theta(s) - f(s)\| \right], \tag{2}$$

where $D_r^i$ is the reset buffer that contains transitions collected during the reset process for a scenario $i$. Note that the transitions sampled from the reset buffer are used only to estimate the obtainable information of under-explored states, and are not used to train any other models, including the predictor network or the policy.

### 3.3 Aborting Episodes to Prevent Unsafe States

An autonomous vehicle is likely to encounter unsafe states even if episodes are identified as informative scenarios. Episodes must be aborted before an autonomous vehicle enters unsafe states. We observed that the estimated state novelty is high when episodes must be aborted. This is intuitive because an autonomous vehicle may not know how to avoid unsafe states on under-explored states, where the estimated novelty is typically high. Therefore, we use the estimated state novelty as a metric to determine when to abort episodes. Specifically, our algorithm estimates the novelty of the current state at each time step and then aborts an episode if the estimated novelty is too high. Note that, similar to Section 3.2, we use RND to estimate the novelty of the current state.

**Algorithm 1** Overall Training Procedure

**Given:** Informative scenario set $I_1$
Initialize forward policy and buffer $\pi_f(a|s), D_f$
Initialize reset buffers $\{D_r^j\}_{j=1,\dots,n}$
Initialize target and predictor networks $f(s), \hat{f}_\theta(s)$
Sample scenario $i_1 \sim \text{Unif}(I_1)$
**for** $k \leftarrow 1 \dots K$ **do**
  **for** $t \leftarrow 1 \dots T_{\text{forward}}$ **do**
    **if** $\lambda_0 \leq \|\hat{f}_\theta(s_t) - f(s_t)\|$ **then**
      Abort and switch to reset policy $\pi_r(a_t|s_t)$
    **end if**
    Select forward action $a_t \sim \pi_f(a_t|s_t)$
    Obtain and add transition to forward buffer $D_f$
    Update forward policy $\pi_f(a|s)$ and predictor $\hat{f}_\theta(s)$
  **end for**
  **for** $t \leftarrow 1 \dots T_{\text{reset}}$ **do**
    Select reset action $a_t \sim \pi_r(a_t|s_t)$
    Obtain and add transition to reset buffer $D_r^{i_k}$
  **end for**
  Estimate informative scenario set $I_{k+1}$
  Sample next scenario $i_{k+1} \sim \text{Unif}(I_{k+1})$
  Return to initial state of scenario $i_{k+1}$ with reset policy $\pi_r(a|s)$
**end for**

### 3.4 Returning with Safety-aware Reset Behaviors

After aborting an episode, an autonomous vehicle must return to an initial state for the subsequent episode. The reset process requires the autonomous vehicle to perform safety-aware behaviors and adhere to traffic rules. Furthermore, the autonomous vehicle must handle a sequence of diverse driving tasks during the reset process. Imagine training an autonomous vehicle to address a detour task. The route from the aborted state to the initial state for the subsequent episode is likely to include other driving tasks, such as lane-change and intersection tasks. These challenges make it difficult for autonomous vehicles to learn both how to reach a goal state and how to return to an initial state simultaneously.

Our algorithm leverages rule-based autonomous driving algorithms to return autonomous vehicles with safety-aware and rule-abiding behaviors. This is based on prior experimental results as follows: rule-based algorithms cannot perform flexible behaviors like expert drivers, but they can perform safety-aware and rule-abiding behaviors across diverse driving scenarios. Most RL algorithms overlook the benefits of rule-based algorithms. In contrast, our work sheds new light on these benefits for training autonomous vehicles in the real world. Note that while our algorithm relies on rule-based driving behaviors to return autonomous vehicles, it learns when to abort episodes and where to return them.

### 3.5 Training Procedure Details

Algorithm 1 describes the overall training procedure of our real-world RL algorithm. Before aborting an episode, an au-

tonomous vehicle is trained and controlled with the RL forward policy $\pi_f(a|s)$. After the episode is aborted, the vehicle is controlled with the rule-based reset policy $\pi_r(a|s)$ to reach a goal state. Once the vehicle reaches the goal state, the informative scenario set for the subsequent episode is identified, and the reset policy returns the vehicle to the initial state of a scenario sampled from the identified set. To estimate the informative scenario set, we define independent reset buffers for each scenario, which makes it efficient to sample corresponding transitions. We would like to note that the forward policy trained with an RL algorithm can outperform the rule-based reset policy.

# 4 Experiments

Our experiments aim to answer the following questions: 1) Can our real-world RL algorithm enable autonomous vehicles to achieve competitive driving performance? 2) Can our real-world RL algorithm identify informative scenarios? 3) How does identifying informative scenarios affect the performance of our real-world RL algorithm? and 4) Can our real-world RL algorithm prevent autonomous vehicles from entering risky states? To answer these questions, we introduce five urban driving tasks and use them to evaluate our real-world RL algorithm against baselines.

## 4.1 Baselines

The baselines used in our experiments are as follows: 1) an agent that randomly selects actions with access to external resets (Random), 2) an RL agent that has access to external resets (Oracle), 3) an autonomous algorithm that learns both how to solve tasks and how to reset simultaneously (LNT) [Eysenbach *et al.*, 2018], and 4) a variant of our real-world RL algorithm that leverages rule-based autonomous driving algorithms without episode aborts and informative scenario identification (Ours w/o Curr). While LNT demonstrated that learning reset behaviors at the same time is efficient to reduce human intervention in simple robotic tasks, urban driving tasks have multiple complex scenarios along the reset route. We expect that LNT fails to learn reset driving behaviors and triggers significant human intervention. Unlike LNT, our algorithm focuses on identifying informative scenarios and avoiding unsafe states, while leveraging rule-based reset driving behaviors.

## 4.2 Implementation Details

Our real-world RL algorithm includes two main learnable models: the forward policy and the RND predictor, each represented by neural networks. The forward policy has two hidden layers of 512 units with ReLU activations and an additional softmax layer to output a categorical distribution over high-level actions, such as go, crawl, and stop. The input of the forward policy consists of the predicted trajectories of surrounding vehicles and the planned trajectories of an autonomous vehicle. We used Soft Actor-Critic (SAC) [Haarnoja *et al.*, 2018], a state-of-the-art off-policy RL algorithm, and the Adam optimizer to update the forward policy. To provide a fair comparison, the forward policies of our algorithm and baselines were designed with the same structure and were trained with the same RL algorithm and optimizer.

| HYPERPARAMETER | VALUE |
|---|---|
| Batch Size | 256 |
| Buffer Size (Forward) | 50000 |
| Buffer Size (Reset) | 1000 |
| Learning Rate | $1 \times 10^{-3}$ |
| Discount Factor | 0.99 |
| Temperature | 0.4 |
| Gradient Step | 1 |
| Target Update Interval | 1 |
| Target Smoothing Coefficient | 0.005 |
| $\lambda_{0,1,2}$ | 1.4/1.0/1.7 |

Table 1: Hyperparameters

The RND predictor has three convolutional layers and a linear output layer as follows: 32 filters of size 8×8 and stride 4, 64 filters of size 4×4 and stride 4, and 32 filters of size 3×3 and stride 1. The output of the last convolutional layer is fed into a linear layer that has 512 hidden units and outputs 256-dimensional feature vectors. The input of the RND predictor is a bird's-eye view segmentation mask. Similar to the forward policy, we utilized Adam optimizer to update the RND predictor. The key hyperparameters described in Table 1 were tuned with the coarse grid search.

The reset policy is implemented with the autonomous driving agent provided by the open-source simulator CARLA [Dosovitskiy *et al.*, 2017]. The agent uses rule-based algorithms designed to ensure safety and follow traffic rules. We observed it performed safety-aware and rule-abiding behaviors that could handle most scenarios in our evaluation tasks.

## 4.3 Environments

Existing benchmarks in the field of autonomous driving focus on evaluating the driving performance of trained autonomous vehicles. These benchmarks do not provide metrics and settings to evaluate RL algorithms for training autonomous vehicles in the real world. Therefore, we introduce challenging urban driving tasks that require an autonomous vehicle to reset to initial states on its own, allowing us to measure how much human intervention is required when training them with RL in the real world. Figure 2 describes these tasks introduced in our work, which consist of a detour, a roundabout, and three different unsignalized intersection tasks. Similar to the reset policy discussed in Section 4.2, they were implemented within CARLA. The goal across these tasks is to reach given target locations as fast as possible without collisions, and an autonomous vehicle must understand interactions with surrounding vehicles to achieve this goal. The reward function for the detour task is defined as follows:

$$r(s_t, a_t) = \lambda_g \mathbb{1}_g(s_t, a_t) - \lambda_p \mathbb{1}_p(s_t, a_t) - \lambda_c \mathbb{1}_c(s_t, a_t) - \lambda_s \mathbb{1}_s(s_t, a_t), \quad (3)$$

where $\mathbb{1}_g$ indicates whether our agent reaches a goal, $\mathbb{1}_p$ indicates whether our agent crosses a center line, $\mathbb{1}_c$ indicates whether our agent collides or fails to keep a safe distance, $\mathbb{1}_s$ indicates whether our agent survives, and $\lambda_g$, $\lambda_p$, $\lambda_c$, and $\lambda_s$ are hyperparameters to balance each of these terms, respectively. The shared reward function for other tasks is defined as follows:

$$r(s_t, a_t) = \lambda_v \frac{v_t}{v_{\text{max}}} - \lambda_c \mathbb{1}_c(s_t, a_t) - \lambda_s \mathbb{1}_s(s_t, a_t), \quad (4)$$

| | DETOUR | | | THREE-WAY | | | FOUR-WAY | | | FIVE-WAY | | | ROUNDABOUT | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | AS ↓ | SR ↑ | MR ↓ | AS ↓ | SR ↑ | MR ↓ | AS ↓ | SR ↑ | MR ↓ | AS ↓ | SR ↑ | MR ↓ | AS ↓ | SR ↑ | MR ↓ |
| Random | 947.9 | 0.06 | 600.0 | 773.7 | 0.48 | 600.0 | 685.4 | 0.58 | 600.0 | 725.1 | 0.50 | 600.0 | 920.6 | 0.45 | 500.0 |
| Oracle | 146.5 | 1.00 | 600.0 | 414.0 | 0.86 | 600.0 | 348.8 | 0.87 | 600.0 | 288.5 | 0.94 | 600.0 | 453.2 | 0.96 | 500.0 |
| LNT | 150.0 | 1.00 | 176.0 | 465.4 | 0.72 | 398.0 | 493.3 | 0.72 | 570.0 | 471.4 | 0.74 | 538.0 | 516.4 | 0.86 | 312.0 |
| Ours (w/o Curr) | 151.5 | 1.00 | 108.7 | 408.4 | 0.81 | 129.0 | 351.9 | 0.84 | 144.0 | 290.1 | 0.93 | 75.0 | 492.0 | 0.92 | 73.0 |
| **Ours** | **151.5** | **1.00** | **70.1** | **404.3** | **0.85** | **85.0** | **347.8** | **0.87** | **104.0** | **281.6** | **0.94** | **54.0** | **476.0** | **0.95** | **47.0** |

Table 2: Quantitative Results on Urban Driving Tasks


Figure 2: Urban driving tasks introduced in our experiments. All surrounding vehicles are set to ignore traffic signals, so an autonomous vehicle being trained should consider interactions with them to solve these tasks. The black dotted line denotes the route to a given goal.

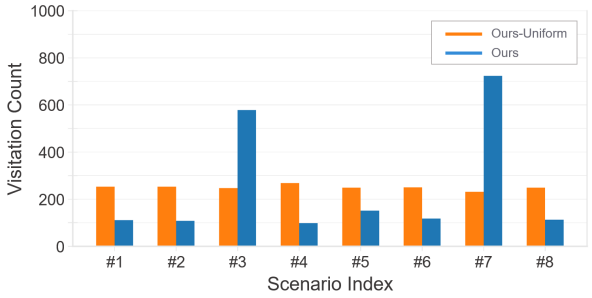


Figure 3: Effects of identifying informative scenarios on scenario visitation in modified five-way unsignalized intersection task. The scenarios numbered 3 and 7 are informative, while the others are non-informative.

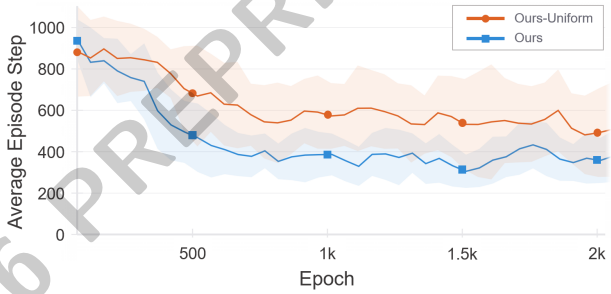


Figure 4: Effects of identifying informative scenarios on performance in modified five-way unsignalized intersection task. Our algorithm attains a lower average episode step and converges faster than uniform scenario sampling. Lines and shaded areas denote the means and standard deviations over 10 random seeds.

where $v_t$ denotes the current speed, $v_{\max}$ denotes the maximum speed, and $\lambda_v$ is the corresponding hyperparameter.

### 4.4 Experimental Results and Analysis

The evaluation metrics used in our experiments are as follows: success rate (SR), average episode step (AS), and the number of manual resets (MR). Success indicates whether an autonomous vehicle reaches a goal within a time limit without collision, and the average episode step represents how efficiently an autonomous vehicle reaches a goal. The number of manual resets indicates how much human intervention is required to train an autonomous vehicle. While both SR and AS are calculated in the evaluation procedure, MR is calculated throughout the training procedure. A manual reset is triggered when a vehicle collides, fails to reach a goal, or cannot return to an initial state for the subsequent episode.

Table 2 describes the numerical training results computed over 100 episodes for each urban driving task. While Oracle achieves the best performance across all driving tasks, it requires repetitive manual resets after every episode. This makes it impractical to use Oracle for training autonomous vehicles in real-world driving environments. LNT triggers a much larger number of manual resets than Ours (w/o Curr). We empirically observed that the reset policy of LNT failed to simultaneously learn all tasks encountered in the reset route. Our algorithm achieves competitive driving performance with significantly fewer manual resets than baselines. In particular, the gap between Ours and Ours (w/o Curr) implies that learning when to abort episodes and identifying informative scenarios are critical in reducing unnecessary manual resets.

To investigate whether our algorithm identifies informative scenarios, we modified the five-way unsignalized intersection task, where some scenarios are intentionally designed to be non-informative. When an autonomous vehicle resets to these non-informative scenarios, surrounding vehicles are not spawned, and the inputs of the RND predictor are randomly shuffled. An autonomous vehicle would struggle to collect informative transitions under these conditions. Figure 3 describes the visitation counts comparison between our real-world RL algorithm and the variant that uniformly samples the next scenario. We confirmed that our algorithm leads an autonomous vehicle to informative scenarios much more frequently than non-informative scenarios. In addition, as shown in Figure 4, we observed that our algorithm achieves a lower average episode step and converges faster than the

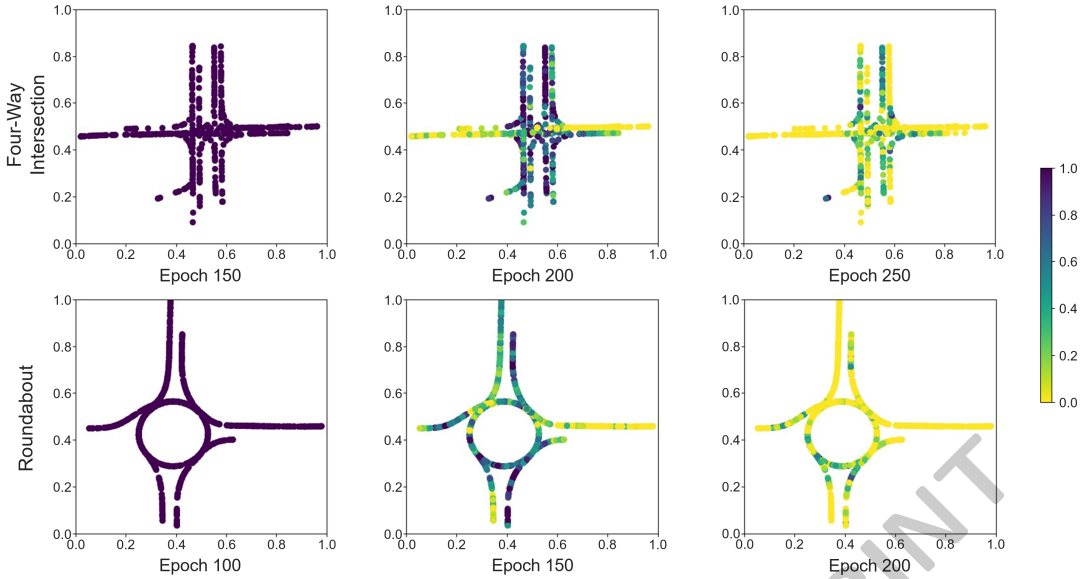


Figure 5: Estimated novelty of sampled states in four-way unsignalized intersection and roundabout tasks. Each dimension of the states is normalized to [0, 1], and their colors represent the novelty estimated by the RND predictor. These results indicate that the state space where an autonomous vehicle can be trained continually broadens as the training progresses.

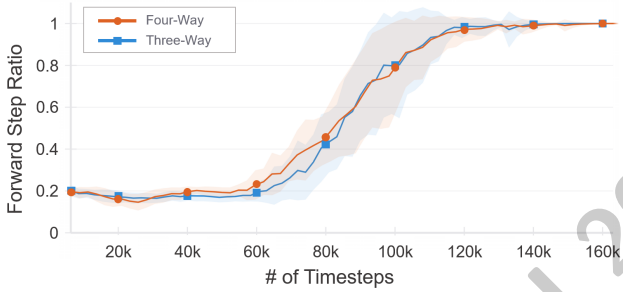


Figure 6: Forward step ratios for three-way and four-way unsignalized intersection tasks. The forward step ratio refers to the ratio between the number of forward time steps and the total number of time steps. Lines and shaded areas denote the means and standard deviations over 10 random seeds, respectively.

variant. This indicates that identifying informative scenarios contributes to better sample efficiency of our algorithm.

To analyze how our real-world RL algorithm prevents autonomous vehicles from entering unsafe states, we calculate the ratio of the number of time steps before an episode is aborted to the total number of time steps. Figure 6 describes how this ratio changes over training time on the three-way and four-way unsignalized intersection tasks. We observed that episodes are aborted near initial states in the early stages of training and near goal states at the end of the training. This suggests that our real-world RL algorithm can reduce manual resets by preventing an autonomous vehicle from entering under-explored states, where it might take unsafe actions. Note that our algorithm takes into account the learning progress of an autonomous vehicle to determine when to abort episodes.

We also visualize how the states in which our algorithm allows an autonomous vehicle to explore change over time. As mentioned in Algorithm 1, the estimated novelty of these states should be lower than the abort threshold $\lambda_0$. To visualize such states, we randomly sample some states from multiple rollouts of each initial state and estimate the novelty of the sampled states. Figure 5 illustrates the visualization results in the four-way unsignalized intersection and roundabout tasks. Note that the estimated state novelty is clipped to the range $[\lambda_0, \lambda_0 + 2]$, and then normalized to [0, 1]. The visualization results show that the states having the estimated novelty below $\lambda_0$ are gradually spread out from initial states as the training progresses. Therefore, Figure 5 can then be interpreted as the qualitative evidence supporting the quantitative results shown in Figure 6.

## 5 Conclusion

We introduce a real-world RL algorithm that enables autonomous vehicles to continuously collect informative transitions and improve by themselves. The three key challenges addressed in our work are: 1) where to return autonomous vehicles to collect informative transitions, 2) when to abort episodes to prevent autonomous vehicles from entering unsafe states, and 3) how to safely return autonomous vehicles to initial states for subsequent episodes. Experimental results demonstrate that our real-world RL algorithm allows autonomous vehicles to achieve competitive driving performance with reduced human intervention compared to baselines in diverse urban driving tasks. The results also show that identifying informative scenarios facilitates sample efficiency. We will employ Vision-Language Models to assess the safety of states. Their common-sense reasoning capabilities can allow us to identify unsafe states across diverse driving tasks. We hope that our research contributes to the realization of RL-based autonomous vehicles in the real world.

## Acknowledgments

This work was supported by the Technology Innovation Program (or Industrial Strategic Technology Development Program-Technology Innovation Program) (RS-2025-25449157, Development of Commercialization Technologies for End-to-End Autonomous Driving Products) funded by the Ministry of Trade, Industry and Resources (MOTIR, Korea), the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (RS-2026-25497111), the Institute of Information & Communications Technology Planning & Evaluation (IITP) under the Artificial Intelligence Convergence Innovation Human Resources Development (IITP-2026-RS-2023-00255968) grant funded by the Korea government (MSIT), and the Ajou University research fund.

## References

[Bellemare *et al.*, 2016] Marc Bellemare, Sriram Srinivasan, Georg Ostrovski, Tom Schaul, David Saxton, and Remi Munos. Unifying count-based exploration and intrinsic motivation. *Advances in neural information processing systems*, 29, 2016.

[Bojarski *et al.*, 2016] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, et al. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*, 2016.

[Broggi *et al.*, 2015] Alberto Broggi, Pietro Cerri, Stefano Debattisti, Maria Chiara Laghi, Paolo Medici, Daniele Molinari, Matteo Panciroli, and Antonio Prioletti. Proud—public road urban driverless-car test. *IEEE Transactions on Intelligent Transportation Systems*, 16(6):3508–3519, 2015.

[Burda *et al.*, 2018] Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. Exploration by random network distillation. *arXiv preprint arXiv:1810.12894*, 2018.

[Chebotar *et al.*, 2017] Yevgen Chebotar, Karol Hausman, Marvin Zhang, Gaurav Sukhatme, Stefan Schaal, and Sergey Levine. Combining model-based and model-free updates for trajectory-centric reinforcement learning. In *International conference on machine learning*, pages 703–711. PMLR, 2017.

[Chen *et al.*, 2024] Li Chen, Penghao Wu, Kashyap Chitta, Bernhard Jaeger, Andreas Geiger, and Hongyang Li. End-to-end autonomous driving: Challenges and frontiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[Dosovitskiy *et al.*, 2017] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. In *Conference on robot learning*, pages 1–16. PMLR, 2017.

[Eysenbach *et al.*, 2018] Benjamin Eysenbach, Shixiang Gu, Julian Ibarz, and Sergey Levine. Leave no trace: Learning to reset for safe and autonomous reinforcement learning. In *International Conference on Learning Representations*, 2018.

[Gao *et al.*, 2025] Hao Gao, Shaoyu Chen, Bo Jiang, Bencheng Liao, Yiang Shi, Xiaoyang Guo, Yuechuan Pu, haoran yin, Xiangyu Li, xinbang zhang, ying zhang, Wenyu Liu, Qian Zhang, and Xinggang Wang. RAD: Training an end-to-end driving policy via large-scale 3DGS-based reinforcement learning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[Geiger *et al.*, 2012] Andreas Geiger, Martin Lauer, Frank Moosmann, Benjamin Ranft, Holger Rapp, Christoph Stiller, and Julius Ziegler. Team annieway's entry to the 2011 grand cooperative driving challenge. *IEEE Transactions on Intelligent Transportation Systems*, 13(3):1008–1017, 2012.

[Haarnoja *et al.*, 2018] 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*, pages 1861–1870. PMLR, 2018.

[Hu *et al.*, 2023] Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, et al. Planning-oriented autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 17853–17862, 2023.

[Huang *et al.*, 2025] Zhiyu Huang, Xinshuo Weng, Maximilian Igl, Yuxiao Chen, Yulong Cao, Boris Ivanovic, Marco Pavone, and Chen Lv. Gen-drive: Enhancing diffusion generative driving policies with reward modeling and reinforcement learning fine-tuning. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3445–3451. IEEE, 2025.

[Isele *et al.*, 2018] David Isele, Reza Rahimi, Akansel Cosgun, Kaushik Subramanian, and Kikuo Fujimura. Navigating occluded intersections with autonomous vehicles using deep reinforcement learning. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 2034–2039. IEEE, 2018.

[Jia *et al.*, 2023] Xiaosong Jia, Penghao Wu, Li Chen, Jiangwei Xie, Conghui He, Junchi Yan, and Hongyang Li. Think twice before driving: Towards scalable decoders for end-to-end autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21983–21994, 2023.

[Jiang *et al.*, 2023] Bo Jiang, Shaoyu Chen, Qing Xu, Bencheng Liao, Jiajie Chen, Helong Zhou, Qian Zhang, Wenyu Liu, Chang Huang, and Xinggang Wang. Vad: Vectorized scene representation for efficient autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8340–8350, 2023.

[Kendall *et al.*, 2019] Alex Kendall, Jeffrey Hawke, David Janz, Przemyslaw Mazur, Daniele Reda, John-Mark Allen, Vinh-Dieu Lam, Alex Bewley, and Amar Shah. Learning to drive in a day. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8248–8254. IEEE, 2019.

[Lee and Seo, 2024] Sang-Hyun Lee and Seung-Woo Seo. Self-supervised curriculum generation for autonomous reinforcement learning without task-specific knowledge. *IEEE Robotics and Automation Letters*, 2024.

[Levine *et al.*, 2018] Sergey Levine, Peter Pastor, Alex Krizhevsky, Julian Ibarz, and Deirdre Quillen. Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. *The International journal of robotics research*, 37(4-5):421–436, 2018.

[Liao *et al.*, 2025] Bencheng Liao, Shaoyu Chen, Haoran Yin, Bo Jiang, Cheng Wang, Sixu Yan, Xinbang Zhang, Xiangyu Li, Ying Zhang, Qian Zhang, et al. Diffusiondrive: Truncated diffusion model for end-to-end autonomous driving. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 12037–12047, 2025.

[Mnih *et al.*, 2015] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

[Nagabandi *et al.*, 2020] Anusha Nagabandi, Kurt Konolige, Sergey Levine, and Vikash Kumar. Deep dynamics models for learning dexterous manipulation. In *Conference on Robot Learning*, pages 1101–1112. PMLR, 2020.

[Nothdurft *et al.*, 2011] Tobias Nothdurft, Peter Hecker, Sebastian Ohl, Falko Saust, Markus Maurer, Andreas Reschka, and Jürgen Rüdiger Böhmer. Stadtpilot: First fully autonomous test drives in urban traffic. In *2011 14th international IEEE conference on intelligent transportation systems (ITSC)*, pages 919–924. IEEE, 2011.

[Pathak *et al.*, 2017] Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *International conference on machine learning*, pages 2778–2787. PMLR, 2017.

[Shao *et al.*, 2023] Hao Shao, Letian Wang, Ruobing Chen, Steven L Waslander, Hongsheng Li, and Yu Liu. Reasonnet: End-to-end driving with temporal and global reasoning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13723–13733, 2023.

[Tong *et al.*, 2023] Wenwen Tong, Chonghao Sima, Tai Wang, Li Chen, Silei Wu, Hanming Deng, Yi Gu, Lewei Lu, Ping Luo, Dahua Lin, et al. Scene as occupancy. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8406–8415, 2023.

[Toromanoff *et al.*, 2020] Marin Toromanoff, Emilie Wirbel, and Fabien Moutarde. End-to-end model-free reinforcement learning for urban driving using implicit affordances. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7153–7162, 2020.

[Vinyals *et al.*, 2019] Oriol Vinyals, Igor Babuschkin, Wojciech M Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H Choi, Richard Powell, Timo Ewalds, Petko Georgiev, et al. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, 575(7782):350–354, 2019.

[Weng *et al.*, 2024] Xinshuo Weng, Boris Ivanovic, Yan Wang, Yue Wang, and Marco Pavone. Para-drive: Parallelized architecture for real-time autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15449–15458, 2024.

[Yahya *et al.*, 2017] Ali Yahya, Adrian Li, Mrinal Kalakrishnan, Yevgen Chebotar, and Sergey Levine. Collective robot reinforcement learning with distributed asynchronous guided policy search. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 79–86. IEEE, 2017.

[Zeng *et al.*, 2020] Andy Zeng, Shuran Song, Johnny Lee, Alberto Rodriguez, and Thomas Funkhouser. Tossingbot: Learning to throw arbitrary objects with residual physics. *IEEE Transactions on Robotics*, 36(4):1307–1319, 2020.

[Zhou *et al.*, 2025] Zewei Zhou, Tianhui Cai, Seth Z. Zhao, Yun Zhang, Zhiyu Huang, Bolei Zhou, and Jiaqi Ma. AutoVLA: A vision-language-action model for end-to-end autonomous driving with adaptive reasoning and reinforcement fine-tuning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[Zhu *et al.*, 2019] Henry Zhu, Abhishek Gupta, Aravind Rajeswaran, Sergey Levine, and Vikash Kumar. Dexterous manipulation with deep reinforcement learning: Efficient, general, and low-cost. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 3651–3657. IEEE, 2019.

[Ziegler *et al.*, 2014] Julius Ziegler, Philipp Bender, Markus Schreiber, Henning Lategahn, Tobias Strauss, Christoph Stiller, Thao Dang, Uwe Franke, Nils Appenrodt, Christoph G Keller, et al. Making bertha drive—an autonomous journey on a historic route. *IEEE Intelligent transportation systems magazine*, 6(2):8–20, 2014.