RAD: Training an End-to-End Driving Policy via Large-Scale 3DGS-based Reinforcement Learning

Hao Gao[1,◇] **Shaoyu Chen**[1,2,†] **Bo Jiang**[1] **Bencheng Liao**[1] **Yiang Shi**[1]
Xiaoyang Guo[2] **Yuechuan Pu**[2] **Haoran Yin**[2] **Xiangyu Li**[2] **Xinbang Zhang**[2]
Ying Zhang[2] **Wenyu Liu**[1] **Qian Zhang**[2] **Xinggang Wang**[1,✉]
[1] Huazhong University of Science & Technology [2] Horizon Robotics

Abstract

Existing end-to-end autonomous driving (AD) algorithms typically follow the Imitation Learning (IL) paradigm, which faces challenges such as causal confusion and an open-loop gap. In this work, we propose **RAD**, a 3DGS-based closed-loop **R**einforcement Learning (RL) framework for end-to-end **A**utonomous **D**riving. By leveraging 3DGS techniques, we construct a photorealistic digital replica of the real physical world, enabling the AD policy to extensively explore the state space and learn to handle out-of-distribution scenarios through large-scale trial and error. To enhance safety, we design specialized rewards to guide the policy in effectively responding to safety-critical events and understanding real-world causal relationships. To better align with human driving behavior, we incorporate IL into RL training as a regularization term. We introduce a closed-loop evaluation benchmark consisting of diverse, previously unseen 3DGS environments. Compared to IL-based methods, RAD achieves stronger performance in most closed-loop metrics, particularly exhibiting a $3\times$ lower collision rate. Abundant closed-loop results are presented in the supplementary material. Code is available at `https://github.com/hustvl/RAD` for facilitating future research.

1 Introduction

End-to-end autonomous driving (AD) is currently a trending topic in both academia and industry. It replaces a modularized pipeline with a holistic one by directly mapping sensory inputs to driving actions, offering advantages in system simplicity and generalization ability. Most existing end-to-end AD algorithms [1, 2, 3, 4, 5, 6, 7, 8] follow the Imitation Learning (IL) paradigm, which trains a neural network to mimic human driving behavior. However, despite their simplicity, IL-based methods face significant challenges in real-world deployment.

One key issue is **causal confusion**. IL trains a network to replicate human driving policies by learning from demonstrations. However, this paradigm primarily captures correlations rather than causal relationships between observations (states) and actions. As a result, IL-trained policies may struggle to identify the true causal factors behind planning decisions, leading to shortcut learning [9], *e.g.*, merely extrapolating future trajectories from historical ones [10, 11]. Furthermore, since IL training data predominantly consists of common driving behaviors and does not adequately cover long-tailed distributions, IL-trained policies tend to converge to trivial solutions, lacking sufficient sensitivity to safety-critical events such as collisions.

Another major challenge is the **gap between open-loop training and closed-loop deployment**. IL policies are trained in an open-loop manner using well-distributed driving demonstrations. However,

◇ The work was done when Hao Gao (`g_hao@hust.edu.cn`) was an intern of Horizon Robotics.
† Project lead (`shaoyu.chen@horizon.auto`). ✉ Corresponding author (`xgwang@hust.edu.cn`).

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

 https://doi.org/10.52202/085713-1092

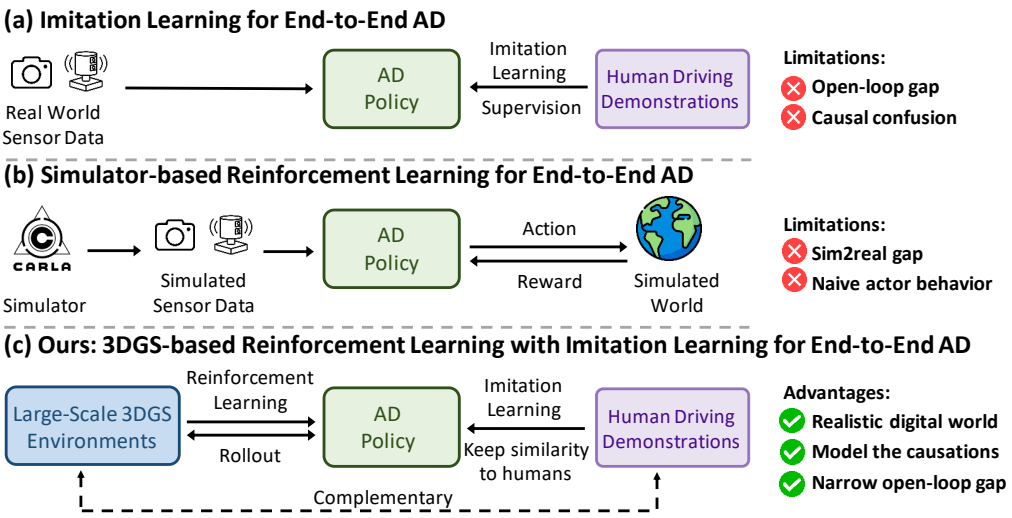


Figure 1: Different training paradigms of end-to-end autonomous driving (AD).

real-world driving is a closed-loop process where minor trajectory errors at each step accumulate over time, leading to compounding errors and out-of-distribution scenarios. IL-trained policies often struggle in these unseen situations, raising concerns about their robustness.

A straightforward solution to these problems is to perform closed-loop Reinforcement Learning (RL) training, which requires a driving environment that can interact with the AD policy. However, using real-world driving environments for closed-loop training poses prohibitive safety risks and operational costs. Simulated driving environments with sensor data simulation capabilities [12, 13] (which are required for end-to-end AD) are typically built on game engines [14, 15] but fail to provide realistic sensor simulation results.

In this work, we establish a 3DGS-based [16] closed-loop RL training paradigm. Leveraging 3DGS techniques, we construct a photorealistic digital replica of the real world, where the AD policy can extensively explore the state space and learn to handle out-of-distribution situations through large-scale trial and error. To ensure effective responses to safety-critical events and a better understanding of real-world causations, we design specialized safety-related rewards. However, RL training presents several critical challenges, which this paper addresses.

One significant challenge is the **Human Alignment Problem**. The exploration process in RL can lead to policies that deviate from human-like behavior, disrupting the smoothness of the action sequence. To address this, we incorporate imitation learning as a regularization term during RL training, helping to maintain similarity to human driving behavior. As illustrated in Fig. 1, RL and IL work together to optimize the AD policy: RL enhances IL by addressing causation and the open-loop gap, while IL improves RL by ensuring better human alignment.

Another major challenge is the **Sparse Reward Problem**. RL often suffers from sparse rewards and slow convergence. To alleviate this issue, we introduce dense auxiliary objectives related to collisions and deviations, which help constrain the full action distribution. Additionally, we streamline and decouple the action space to reduce the exploration cost associated with RL.

To validate the effectiveness of our approach, we construct a closed-loop evaluation benchmark comprising diverse, unseen 3DGS environments. Our method, RAD, outperforms IL-based approaches across most closed-loop metrics, notably achieving a collision rate that is $3\times$ lower.

The contributions of this work are summarized as follows:

- We propose the first 3DGS-based RL framework for training end-to-end AD policy. The reward, action space, optimization objective, and interaction mechanism are specially designed to enhance training efficiency and effectiveness.
- We propose to combine RL and IL to synergistically optimize the end-to-end AD policy. RL complements IL by modeling the causations and narrowing the open-loop gap, while IL complements RL in terms of human alignment.

https://doi.org/10.52202/085713-1092

- We validate the effectiveness of RAD on a closed-loop evaluation benchmark consisting of diverse, unseen 3DGS environments. RAD achieves stronger performance in closed-loop evaluation, particularly a $3\times$ lower collision rate, compared to IL-based methods.

2 Related Work

Dynamic Scene Reconstruction. Implicit neural representations have been widely used in novel view synthesis and dynamic scene reconstruction, as in UniSim [17], MARS [18], and NeuRAD [19], which leverage neural scene graphs for structured decomposition. However, their slow rendering speeds hinder real-time applications. Several recent works [20, 21, 22] have demonstrated the effectiveness of 3D Gaussian Splatting (3DGS) [16] for dynamic urban scene reconstruction. While prior works [17, 18, 22] primarily utilize reconstructed scenes for closed-loop evaluation, we go a step further by incorporating scenes reconstructed via 3DGS into the RL training loop to enhance policy learning.

End-to-End Imitation Learning for Autonomous Driving. Recent advances in learning-based planning have shown strong potential, driven by large-scale data. UniAD [1] integrates multiple perception tasks to boost planning, while VAD [2] improves efficiency with compact vectorized representations. Follow-up works [23, 4, 3, 10, 24, 25, 26, 27, 7] enhance the single-trajectory paradigm, whereas VADv2 [6] and Hydra-MDP [5] shift towards multi-modal planning with improved scoring. DiffusionDrive [8] proposes a truncated diffusion policy that denoises an anchored Gaussian distribution to a multi-mode driving action distribution. While existing methods largely follow an IL paradigm, we enhance it by incorporating closed-loop RL to further optimize the policy.

Reinforcement Learning. Reinforcement Learning is a promising technique that has not been fully explored. AlphaGo [28] and AlphaGo Zero [29] have demonstrated the power of Reinforcement Learning in the game of Go. Recently, OpenAI O1 [30] and Deepseek-R1 [31] have leveraged Reinforcement Learning to develop reasoning abilities. RL has also been applied in autonomous driving [32, 33, 34, 35, 36], though often using non-photorealistic simulators (e.g., CARLA [12]) or requiring perfect perception inputs. To the best of our knowledge, RAD is the first to train an end-to-end autonomous driving agent using RL in a photorealistic 3DGS environment.

Combining Imitation and Reinforcement Learning. To combine the sample efficiency of IL with the robustness of RL, CADRE [37] and CIRL [38] adopt a two-stage pipeline, first imitating expert demonstrations and then fine-tuning policies with RL to improve success rates in basic and complex CARLA scenarios. Lu *et al.* [35] and Huang *et al.* [39] instead perform joint IL and RL optimization, where IL constrains RL exploration to maintain human-like behaviors while improving robustness and rare-case success. Although these approaches demonstrate the benefits of combining IL and RL, they are still limited to non-photorealistic simulators or rely on structured BEV representations and virtual sensor data. In contrast, RAD is the first to jointly integrate IL and RL within a 3DGS-based photorealistic digital-twin environment, enabling fully end-to-end policy learning from real-world sensor inputs.

3 RAD

3.1 End-to-End Driving Policy

The overall framework of RAD is depicted in Fig. 2. RAD takes multi-view image sequences as input, transforms the sensor data into scene token embeddings, outputs the probabilistic distribution of actions, and samples an action to control the vehicle.

BEV Encoder. We first employ a BEV encoder [40] to transform multi-view image features from the perspective view to the Bird's Eye View (BEV), obtaining a feature map in the BEV space. This feature map is then used to learn instance-level map features and agent features.

Map Head. Then we utilize a group of map tokens [41, 42, 43] to learn the vectorized map elements of the driving scene from the BEV feature map, including lane centerlines, lane dividers, road boundaries, arrows, traffic signals, *etc*.

Agent Head. Besides, a group of agent tokens [44] is adopted to predict the motion information of other traffic participants, including location, orientation, size, speed, and multi-mode future trajectories.

 https://doi.org/10.52202/085713-1092

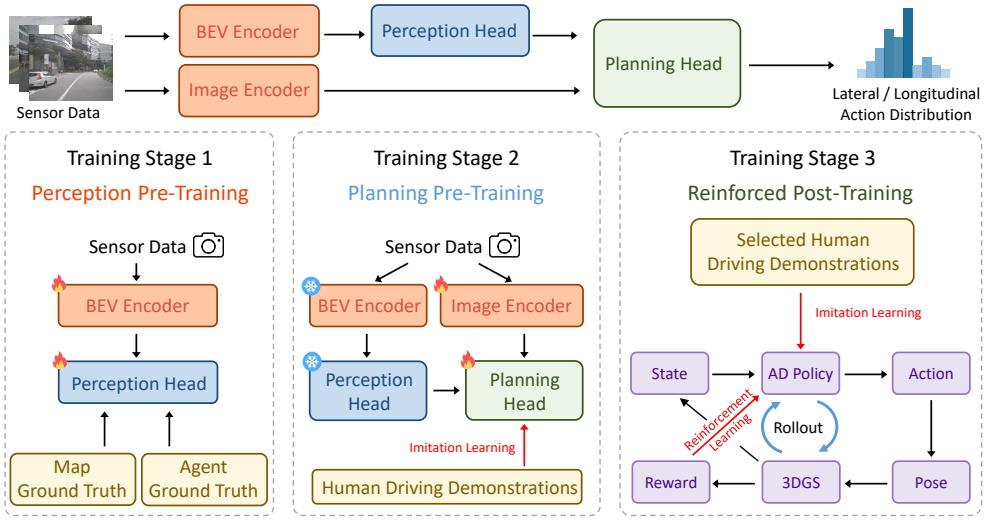


Figure 2: Overall framework of RAD. RAD takes a three-stage training paradigm. In the perception pre-training, ground-truths of map and agent are used to guide instance-level tokens to encode corresponding information. In the planning pre-training stage, large-scale driving demonstrations are used to initialize the action distribution. In the reinforced post-training stage, RL and IL synergistically fine-tune the AD policy.

Image Encoder. Apart from the above instance-level map and agent tokens, we use an individual image encoder [45, 46] to transform the original images into image tokens. These image tokens provide dense and rich scene information for planning, complementary to the instance-level tokens.

Action Space. We propose a decoupled discrete action representation by separating lateral and longitudinal actions. Each action is defined over a short 0.5-second horizon, assuming constant linear and angular velocities. This modeling assumption allows direct computation of control signals from the current action predictions. The resulting formulation reduces the dimensionality of the action space, facilitating more efficient policy optimization.

Planning Head. We use E_{scene} to denote the scene representation, which consists of map tokens, agent tokens, and image tokens. We initialize a planning embedding denoted as E_{plan}. A cascaded Transformer decoder ϕ takes the planning embedding E_{plan} as the query and the scene representation E_{scene} as both key and value.

The output of the decoder ϕ is then combined with navigation information E_{navi} and ego state E_{state} to output the probabilistic distributions of the lateral action a^x and the longitudinal action a^y:

$$\begin{aligned} \pi(a^x \mid s) &= \text{softmax}(\text{MLP}(\phi(E_{\text{plan}}, E_{\text{scene}}) \quad + E_{\text{navi}} + E_{\text{state}})), \\ \pi(a^y \mid s) &= \text{softmax}(\text{MLP}(\phi(E_{\text{plan}}, E_{\text{scene}}) \quad + E_{\text{navi}} + E_{\text{state}})), \end{aligned} \tag{1}$$

where E_{plan}, E_{navi}, E_{state}, and the output of MLP are all of the same dimension $(1 \times D)$.

The planning head also outputs the value functions $V_x(s)$ and $V_y(s)$, which estimate the expected cumulative rewards for the lateral and longitudinal actions, respectively:

$$\begin{aligned} V_x(s) &= \text{MLP}(\phi(E_{\text{plan}}, E_{\text{scene}}) + E_{\text{navi}} + E_{\text{state}}), \\ V_y(s) &= \text{MLP}(\phi(E_{\text{plan}}, E_{\text{scene}}) + E_{\text{navi}} + E_{\text{state}}). \end{aligned} \tag{2}$$

The value functions are used in RL training (Sec. 3.5).

3.2 Training Paradigm

We adopt a three-stage training paradigm: perception pre-training, planning pre-training, and reinforced post-training, as shown in Fig. 2.

Perception Pre-Training. Information in the image is sparse and low-level. In the first stage, the map head and the agent head explicitly output map elements and agent motion information, which are supervised with ground-truth labels. Consequently, map tokens and agent tokens implicitly encode

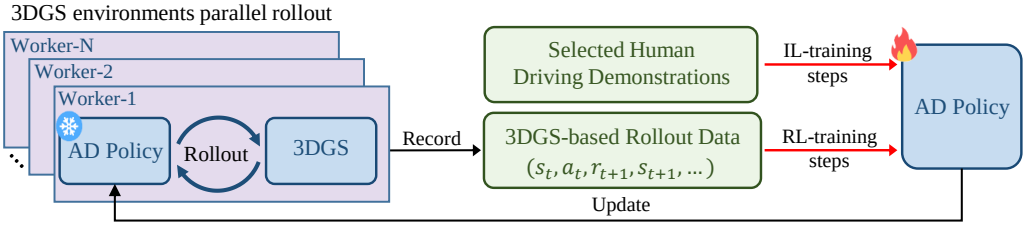


Figure 3: Post-training. N workers parallelly run. The generated rollout data $(s_t, a_t, r_{t+1}, s_{t+1}, ...)$ are recorded in a rollout buffer. Rollout data and human driving demonstrations are used in RL- and IL-training steps to fine-tune the AD policy synergistically.

the corresponding high-level information. In this stage, we only update the parameters of the BEV encoder, the map head, and the agent head.

Planning Pre-Training. In the second stage, to prevent the unstable cold start of RL training, IL is first performed to initialize the probabilistic distribution of actions based on large-scale real-world driving demonstrations from expert drivers. In this stage, we only update the parameters of the image encoder and the planning head, while the parameters of the BEV encoder, map head, and agent head are frozen. The optimization objectives of perception tasks and planning tasks may conflict with each other. However, with the training stage and parameters decoupled, such conflicts are mostly avoided.

Reinforced Post-Training. In the reinforced post-training, RL and IL synergistically fine-tune the distribution. RL aims to guide the policy to be sensitive to critical risky events and adaptive to out-of-distribution situations. IL serves as the regularization term to keep the policy's behavior similar to that of humans.

We select a large number of risky, dense-traffic clips from collected driving demonstrations, and for each clip, train an independent 3DGS model to reconstruct it as a digital driving environment. As illustrated in Fig. 3, we deploy N parallel workers. Each worker randomly samples a 3DGS environment, performs a rollout where the AD policy controls the ego vehicle to interact with the environment, and stores the resulting trajectories $(s_t, a_t, r_{t+1}, s_{t+1}, ...)$ in a shared buffer.

For policy optimization, we alternate between RL and IL steps. In RL steps, the Proximal Policy Optimization (PPO) [47] is used to update the policy from the rollout buffer. In IL steps, expert demonstrations are used for supervised updates. After a fixed number of steps, the updated policy is synchronized across all workers to mitigate distributional shift. During training, only the image encoder and planning head are updated, while the BEV encoder, map head, and agent head are kept frozen. The detailed RL design is introduced below.

3.3 Interaction Mechanism between AD Policy and 3DGS Environment

In the 3DGS environment, the ego vehicle acts according to the AD policy. Other traffic participants can be controlled in many manners: 1) using a smart agent model; 2) using Intelligent Driver Model (IDM) for route tracking; 3) log-replay with real-world data. In order to fully recover the real-world traffic flow, we use log-replay to control other traffic participants in experiments. A conventional kinematic bicycle model is employed to iteratively update the ego vehicle's pose at every Δt seconds as follows:

$$x_{t+1}^w = x_t^w + v_t \cos(\psi_t^w)\,\Delta t,\ y_{t+1}^w = y_t^w + v_t \sin(\psi_t^w)\,\Delta t,\ \psi_{t+1}^w = \psi_t^w + \frac{v_t}{L}\tan(\delta_t)\,\Delta t, \quad (3)$$

where x_t^w and y_t^w denote the position of the ego vehicle relative to the world coordinate; ψ_t^w is the heading angle that defines the vehicle's orientation with respect to the world x-coordinate; v_t is the linear velocity of the ego vehicle; δ_t is the steering angle of the front wheels; and L is the wheelbase, i.e., the distance between the front and rear axles.

During the rollout process, the AD policy outputs actions (a_t^x, a_t^y) for a 0.5-second time horizon at time step t. We derive the linear velocity v_t and steering angle δ_t based on (a_t^x, a_t^y). Based on the kinematic model in Eq. 3, the pose of the ego vehicle in the world coordinate system is updated from $p_t = (x_t^w, y_t^w, \psi_t^w)$ to $p_{t+1} = (x_{t+1}^w, y_{t+1}^w, \psi_{t+1}^w)$.

Based on the updated p_{t+1}, the 3DGS environment computes the new ego vehicle's state s_{t+1}. The updated pose p_{t+1} and state s_{t+1} serve as the input for the next iteration of the inference process.

https://doi.org/10.52202/085713-1092

Table 1: Comparison of training strategies in Stage 3.

Stage 3	CR↓	DCR↓	SCR↓	DR↓	PDR↓	HDR↓	ADD↓	Long. Jerk↓	Lat. Jerk↓
IL	0.229	0.211	0.018	0.066	**0.039**	0.027	**0.238**	**3.928**	0.103
RL	0.143	0.128	0.015	0.080	0.065	**0.015**	0.345	4.204	0.085
RL+IL	**0.089**	**0.080**	**0.009**	**0.063**	0.042	0.021	0.257	4.495	**0.082**



(1) Dynamic Collision (2) Static Collision (3) Positional Deviation (4) Heading Deviation

Figure 4: Example diagram of four types of reward sources. (1) Collision with a dynamic obstacle ahead triggers r_{dc}. (2) Hitting a static roadside obstacle incurs r_{sc}. (3) Moving onto the curb triggers r_{pd}. (4) Drifting toward the adjacent lane triggers r_{hd}.

The 3DGS environment also generates rewards $\mathcal{R}$ (Sec. 3.4) according to multi-source information (including trajectories of other agents, map information, the expert trajectory of the ego vehicle, and the parameters of Gaussians), which are used to optimize the AD policy (Sec. 3.5).

3.4 Reward Modeling

The reward is the source of the training signal, which determines the optimization direction of RL. The reward function is designed to guide the ego vehicle's behavior by penalizing unsafe actions and encouraging alignment with the expert trajectory. It is composed of four reward components: (1) collision with dynamic obstacles, (2) collision with static obstacles, (3) positional deviation from the expert trajectory, and (4) heading deviation from the expert trajectory:

$$\mathcal{R} = \{r_{\text{dc}}, r_{\text{sc}}, r_{\text{pd}}, r_{\text{hd}}\}. \tag{4}$$

As illustrated in Fig. 4, these reward components are triggered under specific conditions. In the 3DGS environment, dynamic collision is detected if the ego vehicle's bounding box overlaps with the annotated bounding boxes of dynamic obstacles, triggering a negative reward r_{dc}. Similarly, static collision is identified when the ego vehicle's bounding box overlaps with the Gaussians of static obstacles, resulting in a negative reward r_{sc}. Positional deviation is measured as the Euclidean distance between the ego vehicle's current position and the closest point on the expert trajectory. A deviation beyond a predefined threshold d_{max} incurs a negative reward r_{pd}. Heading deviation is calculated as the angular difference between the ego vehicle's current heading angle ψ_t and the expert trajectory's matched heading angle ψ_{expert}. A deviation beyond a threshold ψ_{max} results in a negative reward r_{hd}.

Any of these events, including dynamic collision, static collision, excessive positional deviation, or excessive heading deviation, triggers immediate episode termination. Because after such events occur, the 3DGS environment typically generates noisy sensor data, which is detrimental to RL training.

3.5 Policy Optimization

In the closed-loop environment, the error in each single step accumulates over time. The aforementioned rewards are not only caused by the current action but also by the actions of the preceding steps. The rewards are propagated forward with Generalized Advantage Estimation (GAE) [48] to optimize the action distribution of the preceding steps.

Specifically, for each time step t, we store the current state s_t, action a_t, reward r_t, and the estimate of the value $V(s_t)$. Based on the decoupled action space, and considering that different rewards have different correlations to lateral and longitudinal actions, the reward r_t is divided into lateral reward r_t^x and longitudinal reward r_t^y:

$$r_t^x = r_t^{\text{sc}} + r_t^{\text{pd}} + r_t^{\text{hd}}, \quad r_t^y = r_t^{\text{dc}}. \tag{5}$$

Similarly, the value function $V(s_t)$ is decoupled into two components: $V_x(s_t)$ for the lateral dimension and $V_y(s_t)$ for the longitudinal dimension. These value functions estimate the expected cumulative rewards for the lateral and longitudinal actions, respectively. The advantage estimates $\hat{A}_t^x$ and $\hat{A}_t^y$ are computed using GAE (detailed in Appendix A.3), then decomposed into components aligned with reward types:

$$\hat{A}_t^x = \hat{A}_t^{\text{sc}} + \hat{A}_t^{\text{pd}} + \hat{A}_t^{\text{hd}}, \quad \hat{A}_t^y = \hat{A}_t^{\text{dc}}. \tag{6}$$

The policy π_θ is optimized via a modified PPO objective:

$$\mathcal{L}^{\text{PPO}}(\theta) = \mathcal{L}_x^{\text{PPO}}(\theta) + \mathcal{L}_y^{\text{PPO}}(\theta), \tag{7}$$

with independent clipping mechanisms for each dimension (full equations in Appendix A.3). The clipped objective prevents excessively large updates to the policy parameters θ.

3.6 Auxiliary Objective

To address the sparse reward challenge in RL, we design directional auxiliary objectives that provide dense supervision for both longitudinal and lateral controls. These objectives adaptively adjust action probabilities based on real-time collision risk and trajectory deviation, offering more informative gradients during training.

Specifically, a dynamic collision auxiliary loss $\mathcal{L}_{\text{dc}}$ penalizes acceleration and encourages deceleration when past collisions with preceding vehicles are detected. A static collision auxiliary loss $\mathcal{L}_{\text{sc}}$ suppresses steering toward static obstacles and promotes avoidance behavior. For trajectory alignment, the positional deviation loss $\mathcal{L}_{\text{pd}}$ encourages lateral corrections toward the reference path, while the heading deviation loss $\mathcal{L}_{\text{hd}}$ promotes angular corrections to minimize orientation error. Additional implementation details are provided in Appendix A.4.

The composite optimization objective integrates the PPO policy gradient with auxiliary guidance:

$$\mathcal{L}(\theta) = \mathcal{L}^{\text{PPO}}(\theta) + \lambda_1 \mathcal{L}_{\text{dc}}(\theta) + \lambda_2 \mathcal{L}_{\text{sc}}(\theta) + \lambda_3 \mathcal{L}_{\text{pd}}(\theta) + \lambda_4 \mathcal{L}_{\text{hd}}(\theta), \tag{8}$$

where λ_1, λ_2, λ_3, and λ_4 are weighting coefficients that balance the contributions of each auxiliary objective.

Table 2: Impact of different reward components on performance.

ID	Dyn. Col.	Sta. Col.	Pos. Dev.	Head. Dev.	CR↓	DCR↓	SCR↓	DR↓	PDR↓	HDR↓	ADD↓	Long. Jerk↓	Lat. Jerk↓
1	✓				0.172	0.154	0.018	0.092	0.033	0.059	0.259	4.211	0.095
2		✓	✓	✓	0.238	0.217	0.021	0.090	0.045	0.045	**0.241**	3.937	0.098
3	✓		✓	✓	0.146	0.128	0.018	0.060	**0.030**	0.030	0.263	3.729	0.083
4	✓	✓		✓	0.151	0.142	0.009	0.069	0.042	0.027	0.303	3.938	0.079
5	✓	✓	✓		0.166	0.157	0.009	**0.048**	0.036	**0.012**	0.243	**3.334**	**0.067**
6	✓	✓	✓	✓	**0.089**	**0.080**	**0.009**	0.063	0.042	0.021	0.257	4.495	0.082

Table 3: Impact of different auxiliary objectives on performance.

ID	PPO Obj.	Dyn. Col. Aux. Obj.	Sta. Col. Aux. Obj.	Pos. Dev. Aux. Obj.	Head. Dev. Aux. Obj.	CR↓	DCR↓	SCR↓	DR↓	PDR↓	HDR↓	ADD↓	Long. Jerk↓	Lat. Jerk↓
1	✓					0.249	0.223	0.026	0.077	0.047	0.030	0.266	4.209	0.104
2	✓	✓				0.178	0.163	0.015	0.151	0.101	0.050	0.301	3.906	0.085
3	✓		✓	✓	✓	0.137	0.125	0.012	0.157	0.145	0.012	0.296	**3.419**	**0.071**
4	✓	✓		✓	✓	0.169	0.151	0.018	0.075	0.042	0.033	0.254	4.450	0.098
5	✓	✓	✓		✓	0.149	0.134	0.015	0.063	0.057	**0.006**	0.324	3.980	0.086
6	✓	✓	✓	✓		0.128	0.119	0.009	0.066	0.030	0.036	**0.254**	4.102	0.092
7		✓	✓	✓	✓	0.187	0.175	0.012	0.077	0.056	0.021	0.309	5.014	0.112
8	✓	✓	✓	✓	✓	**0.089**	**0.080**	**0.009**	**0.063**	**0.042**	0.021	0.257	4.495	0.082

4 Experiments

4.1 Experimental Settings

Dataset and Benchmark. We collect 2000 hours of expert driving demonstrations in real-world conditions and generate map and agent annotations via an automated pipeline for perception pre-training. Ego-vehicle odometry is employed for planning pre-training. For reinforcement learning, we

https://doi.org/10.52202/085713-1092

Table 4: Closed-loop quantitative comparisons with other IL-based methods on the 3DGS evaluation benchmark.

Method	CR↓	DCR↓	SCR↓	DR↓	PDR↓	HDR↓	ADD↓	Long. Jerk↓	Lat. Jerk↓
TransFuser [27]	0.320	0.273	0.047	0.235	0.188	0.047	0.263	4.538	0.142
VAD [2]	0.335	0.273	0.062	0.314	0.255	0.059	0.304	5.284	0.550
GenAD [3]	0.341	0.299	0.042	0.291	0.160	0.131	0.265	11.37	0.320
VADv2 [6]	0.270	0.240	0.030	0.243	0.139	0.104	0.273	7.782	0.171
RAD	**0.089**	**0.080**	**0.009**	**0.063**	**0.042**	**0.021**	**0.257**	**4.495**	**0.082**

select 4305 real-world driving scenes covering diverse road types, traffic densities, and agent behaviors to ensure environmental diversity. These scenes are first reconstructed into 3DGS environments, from each of which a fixed-length 8-second clip is extracted. Among these clips, 3968 are used for RL training, and the other 337 are used as closed-loop evaluation benchmarks. More training details are in the Appendix A.5.

Metric. We evaluate the performance of the AD policy using nine key metrics. Dynamic Collision Ratio (DCR) and Static Collision Ratio (SCR) quantify the frequency of collisions with dynamic and static obstacles, respectively, with their sum represented as the Collision Ratio (CR). Positional Deviation Ratio (PDR) measures the ego vehicle's deviation from the expert trajectory with respect to position, while Heading Deviation Ratio (HDR) evaluates the ego vehicle's consistency to the expert trajectory with respect to the forward direction. The overall deviation is quantified by the Deviation Ratio (DR), defined as the sum of PDR and HDR. Average Deviation Distance (ADD) quantifies the mean closest distance between the ego vehicle and the expert trajectory before any collisions or deviations occur. Additionally, Longitudinal Jerk (Long. Jerk) and Lateral Jerk (Lat. Jerk) assess driving smoothness by measuring acceleration changes in the longitudinal and lateral directions. CR, DCR, and SCR mainly reflect the policy's safety, and ADD reflects the trajectory consistency between the AD policy and human drivers. More details are provided in Appendix A.6.

4.2 Ablation Study

To evaluate the impact of different design choices in RAD, we conduct three ablation studies. These studies highlight the importance of combining RL and IL, the role of different reward sources, and the effect of auxiliary objectives.

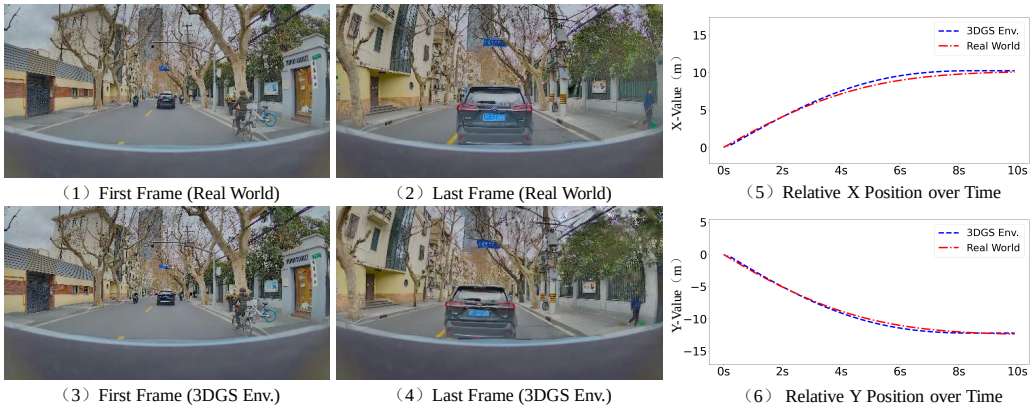
Training Strategy Comparison. We compare pure IL, pure RL, and our mixed IL+RL strategy in reinforced post-training (Tab. 1). Pure IL achieves low deviation (ADD 0.238) but suffers from high collision risk (CR 0.229). Pure RL improves safety (CR 0.143) but deviates more from expert behavior (ADD 0.345). Our IL+RL strategy achieves the best balance (CR 0.089, ADD 0.257), demonstrating improved safety without sacrificing behavioral fidelity.

Reward Source Analysis. We analyze the influence of different reward components (Tab. 2). Policies trained with only partial reward terms (e.g., ID 1, 2, 3, 4, 5) exhibit higher collision rates (CR) compared to the full reward setup (ID 6), which achieves the lowest CR (0.089) while maintaining a stable ADD (0.257). This demonstrates that a well-balanced reward function, incorporating all reward terms, effectively enhances both safety and trajectory consistency. Among the partial reward configurations, ID 2, which omits the dynamic collision reward term, exhibits the highest CR (0.238), indicating that the absence of this term significantly impairs the model's ability to avoid dynamic obstacles, resulting in a higher collision rate.

Auxiliary Objective Analysis. We examine the impact of auxiliary objectives (Tab. 3). Compared to the full auxiliary setup (ID 8), omitting any auxiliary objective increases CR, with a significant rise observed when all auxiliary objectives are removed. This highlights their collective role in enhancing safety. Notably, ID 1, which retains all auxiliary objectives but excludes the PPO objective, results in a CR of 0.187. This value is higher than that of ID 8, indicating that while auxiliary objectives help reduce collisions, they are most effective when combined with the PPO objective.

4.3 Comparisons with Existing Methods

As presented in Tab. 4, we compare RAD with other end-to-end autonomous driving methods in the proposed 3DGS-based closed-loop evaluation. For fair comparisons, all the methods are trained with the same amount of human driving demonstrations. The 3DGS environments for the RL training in



（1）First Frame (Real World)　（2）Last Frame (Real World)　（5）Relative X Position over Time

（3）First Frame (3DGS Env.)　（4）Last Frame (3DGS Env.)　（6） Relative Y Position over Time

Figure 5: Consistency analysis between 3DGS environment and real-world environment. We compare the driving behaviors of the same driving policy in both environments. Subfigure (1)–(4) present the first and last frames during closed-loop evaluation in both environments. Subfigure (5)-(6) depict the temporal evolution of the ego vehicle's position.

RAD are also based on these data. RAD achieves better performance compared to IL-based methods in most metrics. Especially in terms of CR, RAD achieves 3× lower collision rate, demonstrating that RL helps the AD policy learn general collision avoidance ability.

4.4 Consistency Analysis

To demonstrate the consistency between 3DGS environment and real-world environment, we compare the driving behaviors of the same driving policy in both environments. As illustrated in Fig. 5, subfigure (1)–(4) present the first and last frames during closed-loop evaluation in both environments. Subfigure (5)-(6) depict the temporal evolution of the ego vehicle's position. Both qualitative and quantitative results indicate a high degree of behavioral consistency between the 3DGS environment and the real world environment. It shows 3DGS-based closed-loop evaluation can reflect the real-world driving performance.



Figure 6: Closed-loop qualitative comparisons between the IL-only policy and RAD in a Yield to Pedestrians scenario. The IL-only policy fails to yield (Row 1), while RAD successfully yields to pedestrians (Row 2).

4.5 Qualitative Comparisons

We provide qualitative comparisons between the IL-only AD policy (without reinforced post-training) and RAD, as shown in Fig. 6. The IL-only method struggles in dynamic environments, frequently failing to avoid collisions with moving obstacles or manage complex traffic situations. In contrast, RAD consistently performs well, effectively avoiding dynamic obstacles and handling challenging tasks. These results highlight the benefits of closed-loop training in the hybrid method, which enables better handling of dynamic environments. Additional visualizations are included in Fig. 7.

https://doi.org/10.52202/085713-1092

5 Conclusion

In this work, we propose **RAD**, the first 3DGS-based closed-loop reinforcement learning framework for end-to-end autonomous driving. We combine RL and IL, with RL complementing IL to model the causations and narrow the open-loop gap, and IL complementing RL in terms of human alignment. Complemented by a targeted reward system and auxiliary objectives, RAD achieves 3× lower collision rates than state-of-the-art IL methods, with strong performance in challenging scenarios like unprotected left-turns and dense traffic.

Limitations and future work. The effect of 3DGS still has room for improvement, particularly in rendering non-rigid pedestrians, unobserved views, and low-light scenarios. Future works will focus on addressing these issues and scaling up RL to the next level.

Acknowledgement: This work was partially supported by National Natural Science Foundation of China (No. 62376102).

References

[1] 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.

[2] 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.

[3] Wenzhao Zheng, Ruiqi Song, Xianda Guo, Chenming Zhang, and Long Chen. Genad: Generative end-to-end autonomous driving. In *ECCV*, 2024.

[4] 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.

[5] Zhenxin Li, Kailin Li, Shihao Wang, Shiyi Lan, Zhiding Yu, Yishen Ji, Zhiqi Li, Ziyue Zhu, Jan Kautz, Zuxuan Wu, et al. Hydra-mdp: End-to-end multimodal planning with multi-target hydra-distillation. *arXiv preprint arXiv:2406.06978*, 2024.

[6] Shaoyu Chen, Bo Jiang, Hao Gao, Bencheng Liao, Qing Xu, Qian Zhang, Chang Huang, Wenyu Liu, and Xinggang Wang. Vadv2: End-to-end vectorized autonomous driving via probabilistic planning. *arXiv preprint arXiv:2402.13243*, 2024.

[7] Wenchao Sun, Xuewu Lin, Yining Shi, Chuang Zhang, Haoran Wu, and Sifa Zheng. Sparsedrive: End-to-end autonomous driving via sparse scene representation. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8795–8801. IEEE, 2025.

[8] 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.

[9] Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2020.

[10] Zhiqi Li, Zhiding Yu, Shiyi Lan, Jiahan Li, Jan Kautz, Tong Lu, and Jose M Alvarez. Is ego status all you need for open-loop end-to-end autonomous driving? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14864–14873, 2024.

[11] Jiang-Tian Zhai, Ze Feng, Jihao Du, Yongqiang Mao, Jiang-Jiang Liu, Zichang Tan, Yifu Zhang, Xiaoqing Ye, and Jingdong Wang. Rethinking the open-loop evaluation of end-to-end autonomous driving in nuscenes. *arXiv preprint arXiv:2305.10430*, 2023.

[12] 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.

[13] Applied Intuition. Carsim. `https://www.carsim.com/`, 2023.

[14] Epic Games. Unreal engine. `https://www.unrealengine.com/`, 1998.

[15] Unity Technologies. Unity. `https://https://unity.com/`, 2005.

[16] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023.

[17] Ze Yang, Yun Chen, Jingkang Wang, Sivabalan Manivasagam, Wei-Chiu Ma, Anqi Joyce Yang, and Raquel Urtasun. Unisim: A neural closed-loop sensor simulator. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1389–1399, 2023.

[18] Zirui Wu, Tianyu Liu, Liyi Luo, Zhide Zhong, Jianteng Chen, Hongmin Xiao, Chao Hou, Haozhe Lou, Yuantao Chen, Runyi Yang, et al. Mars: An instance-aware, modular and realistic simulator for autonomous driving. In *CAAI International Conference on Artificial Intelligence*, pages 3–15. Springer, 2023.

[19] Adam Tonderski, Carl Lindström, Georg Hess, William Ljungbergh, Lennart Svensson, and Christoffer Petersson. Neurad: Neural rendering for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14895–14904, 2024.

[20] Yunzhi Yan, Haotong Lin, Chenxu Zhou, Weijie Wang, Haiyang Sun, Kun Zhan, Xianpeng Lang, Xiaowei Zhou, and Sida Peng. Street gaussians: Modeling dynamic urban scenes with gaussian splatting. In *European Conference on Computer Vision*, pages 156–173. Springer, 2024.

[21] Xiaoyu Zhou, Zhiwei Lin, Xiaojun Shan, Yongtao Wang, Deqing Sun, and Ming-Hsuan Yang. Drivinggaussian: Composite gaussian splatting for surrounding dynamic autonomous driving scenes. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 21634–21643, 2024.

[22] Hongyu Zhou, Longzhong Lin, Jiabao Wang, Yichong Lu, Dongfeng Bai, Bingbing Liu, Yue Wang, Andreas Geiger, and Yiyi Liao. Hugsim: A real-time, photo-realistic and closed-loop simulator for autonomous driving. *arXiv preprint arXiv:2412.01718*, 2024.

[23] Yingyan Li, Lue Fan, Jiawei He, Yuqi Wang, Yuntao Chen, Zhaoxiang Zhang, and Tieniu Tan. Enhancing end-to-end autonomous driving with latent world model. *arXiv preprint arXiv:2406.08481*, 2024.

[24] Yuqi Wang, Jiawei He, Lue Fan, Hongxin Li, Yuntao Chen, and Zhaoxiang Zhang. Driving into the future: Multiview visual forecasting and planning with world model for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14749–14759, 2024.

[25] Xunjiang Gu, Guanyu Song, Igor Gilitschenski, Marco Pavone, and Boris Ivanovic. Producing and leveraging online map uncertainty in trajectory prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14521–14530, 2024.

[26] Zhili Chen, Maosheng Ye, Shuangjie Xu, Tongyi Cao, and Qifeng Chen. Ppad: Iterative interactions of prediction and planning for end-to-end autonomous driving. In *European Conference on Computer Vision*, pages 239–256. Springer, 2024.

[27] Aditya Prakash, Kashyap Chitta, and Andreas Geiger. Multi-modal fusion transformer for end-to-end autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7077–7087, 2021.

[28] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 2016.

 https://doi.org/10.52202/085713-1092

[29] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, Yutian Chen, Timothy P. Lillicrap, Fan Hui, Laurent Sifre, George van den Driessche, Thore Graepel, and Demis Hassabis. Mastering the game of go without human knowledge. *Nat.*, 2017.

[30] OpenAI. Openai o1. `https://openai.com/o1/`, 2024.

[31] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

[32] 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.

[33] Dian Chen, Vladlen Koltun, and Philipp Krähenbühl. Learning to drive from a world on rails. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15590–15599, 2021.

[34] Zhejun Zhang, Alexander Liniger, Dengxin Dai, Fisher Yu, and Luc Van Gool. End-to-end urban driving by imitating a reinforcement learning coach. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 15222–15232, 2021.

[35] Yiren Lu, Justin Fu, George Tucker, Xinlei Pan, Eli Bronstein, Rebecca Roelofs, Benjamin Sapp, Brandyn White, Aleksandra Faust, Shimon Whiteson, et al. Imitation is not enough: Robustifying imitation with reinforcement learning for challenging driving scenarios. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7553–7560. IEEE, 2023.

[36] Yihan Hu, Siqi Chai, Zhening Yang, Jingyu Qian, Kun Li, Wenxin Shao, Haichao Zhang, Wei Xu, and Qiang Liu. Solving motion planning tasks with a scalable generative model. In *European Conference on Computer Vision*, pages 386–404. Springer, 2024.

[37] Yinuo Zhao, Kun Wu, Zhiyuan Xu, Zhengping Che, Qi Lu, Jian Tang, and Chi Harold Liu. Cadre: A cascade deep reinforcement learning framework for vision-based autonomous urban driving. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pages 3481–3489, 2022.

[38] Xiaodan Liang, Tairui Wang, Luona Yang, and Eric Xing. Cirl: Controllable imitative reinforcement learning for vision-based self-driving. In *Proceedings of the European conference on computer vision (ECCV)*, pages 584–599, 2018.

[39] Zhiyu Huang, Jingda Wu, and Chen Lv. Efficient deep reinforcement learning with imitative expert priors for autonomous driving. *IEEE Transactions on Neural Networks and Learning Systems*, 34(10):7391–7403, 2022.

[40] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Qiao Yu, and Jifeng Dai. Bevformer: learning bird's-eye-view representation from lidar-camera via spatiotemporal transformers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[41] Bencheng Liao, Shaoyu Chen, Yunchi Zhang, Bo Jiang, Qian Zhang, Wenyu Liu, Chang Huang, and Xinggang Wang. Maptrv2: An end-to-end framework for online vectorized hd map construction. *International Journal of Computer Vision*, 133(3):1352–1374, 2025.

[42] Bencheng Liao, Shaoyu Chen, Xinggang Wang, Tianheng Cheng, Qian Zhang, Wenyu Liu, and Chang Huang. Maptr: Structured modeling and learning for online vectorized hd map construction. *arXiv preprint arXiv:2208.14437*, 2022.

[43] Bencheng Liao, Shaoyu Chen, Bo Jiang, Tianheng Cheng, Qian Zhang, Wenyu Liu, Chang Huang, and Xinggang Wang. Lane graph as path: Continuity-preserving path-wise modeling for online lane graph construction. In *European Conference on Computer Vision*, pages 334–351. Springer, 2024.

[44] Bo Jiang, Shaoyu Chen, Xinggang Wang, Bencheng Liao, Tianheng Cheng, Jiajie Chen, Helong Zhou, Qian Zhang, Wenyu Liu, and Chang Huang. Perceive, interact, predict: Learning dynamic and static clues for end-to-end motion prediction. *arXiv preprint arXiv:2212.02181*, 2022.

[45] Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[46] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.

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

[48] John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015.

[49] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, 2017.

[50] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[51] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

 https://doi.org/10.52202/085713-1092

A Technical Appendices

A.1 3DGS Reconstruction and Rendering Optimization

To support closed-loop training, we extend the StreetGaussian [20] framework, focusing on enhancing rendering realism and geometric accuracy, particularly for high-fidelity rendering in off-trajectory views.

First, we employ mesh modeling to constrain the geometry of the road surface, constraining Gaussian spheres to the mesh surface to ensure precise road geometry from any viewpoint. Additionally, we model the sky separately to prevent confusion with foreground objects, improving rendering realism under complex lighting conditions.

For foreground objects (e.g., vehicles, pedestrians), we optimize their poses during training and incorporate depth and normal consistency as supervision signals to further enhance geometric reconstruction accuracy and surface detail fidelity. These optimizations significantly improve rendering quality in novel viewpoints, especially in dynamic scenes, where object motion trajectories and surface details remain consistent with real-world observations.

These improvements allow the 3DGS environment to more effectively support closed-loop training, providing a foundation of high realism and geometric accuracy for large-scale trial-and-error learning in autonomous driving strategies.

A.2 Action Space Details

Here, we provide a more comprehensive explanation of the action space design. To ensure stable control and efficient learning, we define the action space over a short time horizon of 0.5 seconds. The ego vehicle's movement is modeled using discrete displacements in both the lateral and longitudinal directions.

Lateral Displacement. The lateral displacement, denoted as a^x, represents the vehicle's movement in the lateral direction over the 0.5-second horizon. We discretize this dimension into N_x options, symmetrically distributed around zero to allow leftward and rightward movements, with an additional option to maintain the current trajectory. The set of possible lateral displacements is:

$$a^x \in \{d^x_{\text{min}}, \ldots, 0, \ldots, d^x_{\text{max}}\}. \tag{9}$$

In our implementation, we use $N_x = 61$, with $d^x_{\text{min}} = -0.75$ m, $d^x_{\text{max}} = 0.75$ m, and intermediate values sampled uniformly.

Longitudinal Displacement. The longitudinal displacement, denoted as a^y, represents the vehicle's movement in the forward direction over the 0.5-second horizon. Similar to the lateral component, we discretize this dimension into N_y options, covering a range of forward displacements, including an option to maintain the current position:

$$a^y \in \{0, \ldots, d^y_{\text{max}}\}. \tag{10}$$

In our setup, we use $N_y = 61$, with $d^y_{\text{max}} = 15$m, and intermediate values sampled uniformly.

A.3 Policy Optimization Details

GAE Computation Details. The advantage estimates $\hat{A}^x_t$ and $\hat{A}^y_t$ are then computed as follows:

$$\begin{aligned}
\delta^x_t &= r^x_t + \gamma V_x(s_{t+1}) - V_x(s_t), \\
\delta^y_t &= r^y_t + \gamma V_y(s_{t+1}) - V_y(s_t), \\
\hat{A}^x_t &= \sum_{l=0}^{\infty} (\gamma\lambda)^l \delta^x_{t+l}, \\
\hat{A}^y_t &= \sum_{l=0}^{\infty} (\gamma\lambda)^l \delta^y_{t+l},
\end{aligned} \tag{11}$$

where δ_t^x and δ_t^y are the temporal difference errors for the lateral and longitudinal dimensions, γ is the discount factor, and λ is the GAE parameter that controls the trade-off between bias and variance.

Complete PPO Objectives. The full clipping objectives are:

$$\begin{aligned}\mathcal{L}_x^{\text{PPO}}(\theta) &= \mathbb{E}_t\left[\min\left(\rho_t^x \hat{A}_t^x,\ \text{clip}(\rho_t^x, 1-\epsilon_x, 1+\epsilon_x)\hat{A}_t^x\right)\right], \\ \mathcal{L}_y^{\text{PPO}}(\theta) &= \mathbb{E}_t\left[\min\left(\rho_t^y \hat{A}_t^y,\ \text{clip}(\rho_t^y, 1-\epsilon_y, 1+\epsilon_y)\hat{A}_t^y\right)\right],\end{aligned} \tag{12}$$

where $\rho_t^x = \frac{\pi_\theta(a_t^x|s_t)}{\pi_{\theta_{\text{old}}}(a_t^x|s_t)}$ is the importance sampling ratio for the lateral dimension, $\rho_t^y = \frac{\pi_\theta(a_t^y|s_t)}{\pi_{\theta_{\text{old}}}(a_t^y|s_t)}$ is the importance sampling ratio for the longitudinal dimension, ϵ_x and ϵ_y are small constants that control the clipping range for the lateral and longitudinal dimensions, ensuring stable policy updates.

A.4 Auxiliary Objectives Details

Action Probability Decomposition. The auxiliary objectives are designed to penalize undesirable behaviors by incorporating specific reward sources, including dynamic collisions, static collisions, positional deviations, and heading deviations. These objectives are computed based on the actions $a_t^{x,\text{old}}$ and $a_t^{y,\text{old}}$ selected by the old AD policy $\pi_{\theta_{\text{old}}}$ at time step t. To facilitate the evaluation of these actions, we separate the probability distribution of the action into four parts:

$$\begin{aligned}\Delta\pi_y^{\text{dec}} &= \sum_{a_t^y < a_t^{y,\text{old}}} \pi_\theta(a_t^y \mid s_t), \\ \Delta\pi_y^{\text{acc}} &= \sum_{a_t^y > a_t^{y,\text{old}}} \pi_\theta(a_t^y \mid s_t), \\ \Delta\pi_x^{\text{left}} &= \sum_{a_t^x < a_t^{x,\text{old}}} \pi_\theta(a_t^x \mid s_t), \\ \Delta\pi_x^{\text{right}} &= \sum_{a_t^x > a_t^{x,\text{old}}} \pi_\theta(a_t^x \mid s_t).\end{aligned} \tag{13}$$

Here, $\Delta\pi_y^{\text{dec}}$ represents the total probability of deceleration actions, $\Delta\pi_y^{\text{acc}}$ represents the total probability of acceleration actions, $\Delta\pi_x^{\text{left}}$ represents the total probability of leftward steering actions, and $\Delta\pi_x^{\text{right}}$ represents the total probability of rightward steering actions.

Dynamic Collision Auxiliary Objective. The dynamic collision auxiliary objective adjusts the longitudinal control action a_t^y based on the location of potential collisions relative to the ego vehicle. If a collision is detected ahead, the policy prioritizes deceleration actions ($a_t^y < a_t^{y,\text{old}}$); if a collision is detected behind, it encourages acceleration actions ($a_t^y > a_t^{y,\text{old}}$). To formalize this behavior, we define a directional factor f_{dc}:

$$f_{\text{dc}} = \begin{cases} 1 & \text{if the collision is ahead,} \\ -1 & \text{if the collision is behind.} \end{cases} \tag{14}$$

The auxiliary objective for dynamic collision avoidance is defined as:

$$\mathcal{L}_{\text{dc}}(\theta) = \mathbb{E}_t\left[\hat{A}_t^{\text{dc}} \cdot f_{\text{dc}} \cdot (\Delta\pi_y^{\text{dec}} - \Delta\pi_y^{\text{acc}})\right], \tag{15}$$

where $\hat{A}_t^{\text{dc}}$ is the advantage estimate for dynamic collision avoidance.

Static Collision Auxiliary Objective. The static collision auxiliary objective adjusts the steering control action a_t^x based on the proximity to static obstacles. If the static obstacle is detected on the left side, the policy promotes rightward steering actions ($a_t^x > a_t^{x,\text{old}}$); if the static obstacle is detected on the right side, it promotes leftward steering actions ($a_t^x < a_t^{x,\text{old}}$). To formalize this behavior, we define a directional factor f_{sc}:

$$f_{\text{sc}} = \begin{cases} 1 & \text{if the static obstacle is on the left,} \\ -1 & \text{if the static obstacle is on the right.} \end{cases} \tag{16}$$

https://doi.org/10.52202/085713-1092

The auxiliary objective for static collision avoidance is defined as:

$$\mathcal{L}_{\text{sc}}(\theta) = \mathbb{E}_t \left[\hat{A}_t^{\text{sc}} \cdot f_{\text{sc}} \cdot (\Delta\pi_x^{\text{right}} - \Delta\pi_x^{\text{left}}) \right], \tag{17}$$

where $\hat{A}_t^{\text{sc}}$ is the advantage estimate for static collision avoidance.

Positional Deviation Auxiliary Objective. The positional deviation auxiliary objective adjusts the steering control action a_t^x based on the ego vehicle's lateral deviation from the expert trajectory. If the ego vehicle deviates leftward, the policy promotes rightward corrections ($a_t^x > a_t^{x,\text{old}}$); if it deviates rightward, it promotes leftward corrections ($a_t^x < a_t^{x,\text{old}}$). We formalize this with a directional factor f_{pd}:

$$f_{\text{pd}} = \begin{cases} 1 & \text{if ego vehicle deviates leftward,} \\ -1 & \text{if ego vehicle deviates rightward.} \end{cases} \tag{18}$$

The auxiliary objective for positional deviation correction is:

$$\mathcal{L}_{\text{pd}}(\theta) = \mathbb{E}_t \left[\hat{A}_t^{\text{pd}} \cdot f_{\text{pd}} \cdot (\Delta\pi_x^{\text{right}} - \Delta\pi_x^{\text{left}}) \right], \tag{19}$$

where $\hat{A}_t^{\text{pd}}$ estimates the advantage of trajectory alignment.

Heading Deviation Auxiliary Objective. The heading deviation auxiliary objective adjusts the steering control action a_t^x based on the angular difference between the ego vehicle's current heading and the expert's reference heading. If the ego vehicle deviates counterclockwise, the policy promotes clockwise corrections ($a_t^x > a_t^{x,\text{old}}$); if it deviates clockwise, it promotes counterclockwise corrections ($a_t^x < a_t^{x,\text{old}}$). To formalize this behavior, we define a directional factor f_{hd}:

$$f_{\text{hd}} = \begin{cases} 1 & \text{if ego vehicle deviates clockwise,} \\ -1 & \text{if ego vehicle deviates counterclockwise.} \end{cases} \tag{20}$$

The auxiliary objective for heading deviation correction is then defined as:

$$\mathcal{L}_{\text{hd}}(\theta) = \mathbb{E}_t \left[\hat{A}_t^{\text{hd}} \cdot f_{\text{hd}} \cdot (\Delta\pi_x^{\text{right}} - \Delta\pi_x^{\text{left}}) \right], \tag{21}$$

where $\hat{A}_t^{\text{hd}}$ is the advantage estimate for heading alignment.

A.5 Implementation Details

In this section, we summarize the training settings, configurations, and hyperparameters used in our approach.

Planning Pre-Training. The action space is discretized using predefined anchors $\mathcal{A} = \{(a_i^x, a_j^y)\}_{i=1,j=1}^{N_x,N_y}$. Each anchor corresponds to a specific steering-speed combination within the 0.5-second planning horizon. Given the ground truth vehicle position at $t = 0.5$ s denoted as $p_{\text{gt}} = (p_{\text{gt}}^x, p_{\text{gt}}^y)$, we implement normalized nearest-neighbor matching over predefined anchor positions:

$$\begin{aligned} \hat{i} &= \arg\min_i \left\| \frac{\mathbf{a}_i^x - d_{\text{min}}^x}{d_{\text{max}}^x - d_{\text{min}}^x} - \frac{p_{\text{gt}}^x - d_{\text{min}}^x}{d_{\text{max}}^x - d_{\text{min}}^x} \right\|_2, \\ \hat{j} &= \arg\min_j \left\| \frac{\mathbf{a}_j^y - 0}{d_{\text{max}}^y - 0} - \frac{p_{\text{gt}}^y - 0}{d_{\text{max}}^y - 0} \right\|_2. \end{aligned} \tag{22}$$

Based on the matched anchor indices $(\hat{i}, \hat{j})$, we formulate the imitation learning objective as a dual focal loss [49]:

$$\mathcal{L}_{\text{IL}} = \mathcal{L}_{\text{focal}}(\pi(a^x \mid s), \hat{i}_t) + \mathcal{L}_{\text{focal}}(\pi(a^y \mid s), \hat{j}_t), \tag{23}$$

where $\mathcal{L}_{\text{focal}}$ is focal loss for discrete action classification.

Reinforced Post-Training. During the training process, we use a cycle where reinforcement learning (RL) and imitation learning (IL) alternate. In each full cycle, we run four rounds of RL training, followed by one round of IL training.

Each RL training round consists of 320 iterations. Our clips are 8 seconds long with a frame rate of 10Hz, meaning that 320 iterations cover four clips of data. We employ a sliding window mechanism that holds four clips of data, updated in a first-in, first-out manner. In each RL training round, data from one new clip is collected and the sliding window is updated accordingly. We find that an RL-to-IL ratio of $(320 \times 4) : 320 = 4 : 1$ yields the best results.

Training configurations. We provide detailed hyperparameters for the two main stages, Planning Pre-Training and Reinforced Post-Training, in Tab. 5 and Tab. 6, respectively.

Table 5: Hyperparameters used in RAD Planning Pre-Training stage.

config	Planning Pre-Training
learning rate	1e-4
learning rate schedule	cosine decay
optimizer	AdamW [50, 51]
optimizer hyper-parameters	β_1, β_2, ϵ = 0.9, 0.999, 1e-8
weight decay	1e-4
batch size	512
training steps	30k
planning head dim	256
Traning GPU	128 RTX4090

Table 6: Hyperparameters used in RAD Reinforced Post-Training stage.

config	Reinforced Post-Training
learning rate	5e-6
learning rate schedule	cosine decay
optimizer	AdamW [50, 51]
optimizer hyper-parameters	β_1, β_2, ϵ = 0.9, 0.999, 1e-8
weight decay	1e-4
RL worker number	32
RL batch size	32
IL batch size	128
GAE parameter	$\gamma = 0.9$, $\lambda = 0.95$
clipping thresholds	$\epsilon_x = 0.1$, $\epsilon_y = 0.2$
deviation threshold	$d_{\text{max}} = 2.0m$, $\psi_{\text{max}} = 40^\circ$
planning head dim	256
value function dim	256
Traning GPU	32 RTX4090

A.6 Metric Details

We evaluate the performance of the autonomous driving policy using nine key metrics.

Dynamic Collision Ratio (DCR). DCR quantifies the frequency of collisions with dynamic obstacles. It is defined as:

$$DCR = \frac{N_{dc}}{N_{total}}, \tag{24}$$

where N_{dc} is the number of clips in which collisions with dynamic obstacles occur, and N_{total} is the total number of clips.

Static Collision Ratio (SCR). SCR measures the frequency of collisions with static obstacles and is defined as:

$$SCR = \frac{N_{sc}}{N_{total}}, \tag{25}$$

where N_{sc} is the number of clips with static obstacle collisions.

https://doi.org/10.52202/085713-1092

Collision Ratio (CR). CR represents the total collision frequency, given by:

$$CR = DCR + SCR. \tag{26}$$

Positional Deviation Ratio (PDR). PDR evaluates the ego vehicle's adherence to the expert trajectory in terms of position. It is defined as:

$$PDR = \frac{N_{pd}}{N_{total}}, \tag{27}$$

where N_{pd} is the number of clips in which the positional deviation exceeds a predefined threshold.

Heading Deviation Ratio (HDR). HDR assesses orientation accuracy by computing the proportion of clips where heading deviations surpass a predefined threshold:

$$HDR = \frac{N_{hd}}{N_{total}}, \tag{28}$$

where N_{hd} is the number of clips where the heading deviation exceeds the threshold.

Deviation Ratio (DR). captures the overall deviation from the expert trajectory, given by:

$$DR = PDR + HDR. \tag{29}$$

Average Deviation Distance (ADD). ADD quantifies the mean closest distance between the ego vehicle and the expert trajectory during time steps when no collisions or deviations occur. It is defined as:

$$ADD = \frac{1}{T_{safe}} \sum_{t=1}^{T_{safe}} d_{min}(t), \tag{30}$$

where T_{safe} represents the total number of time steps in which the ego vehicle operates without collisions or deviations, and $d_{min}(t)$ denotes the minimum distance between the ego vehicle and the expert trajectory at time step t.

Finally, **Longitudinal Jerk (Long. Jerk)** and **Lateral Jerk (Lat. Jerk)** quantify the smoothness of vehicle motion by measuring acceleration changes. Longitudinal jerk is defined as:

$$J_{long} = \frac{d^2 v_{long}}{dt^2}, \tag{31}$$

where v_{long} represents the longitudinal velocity. Similarly, lateral jerk is defined as:

$$J_{lat} = \frac{d^2 v_{lat}}{dt^2}, \tag{32}$$

where v_{lat} is the lateral velocity. These metrics collectively capture abrupt changes in acceleration and steering, providing a comprehensive measure of passenger comfort and driving stability.

A.7 More Qualitative Results

Fig. 7 presents additional qualitative comparisons across various driving scenarios, including detours, crawling in dense traffic, traffic congestion, and U-turn maneuvers. The results highlight the effectiveness of our approach in generating smoother trajectories, enhancing collision avoidance, and improving adaptability in complex environments.

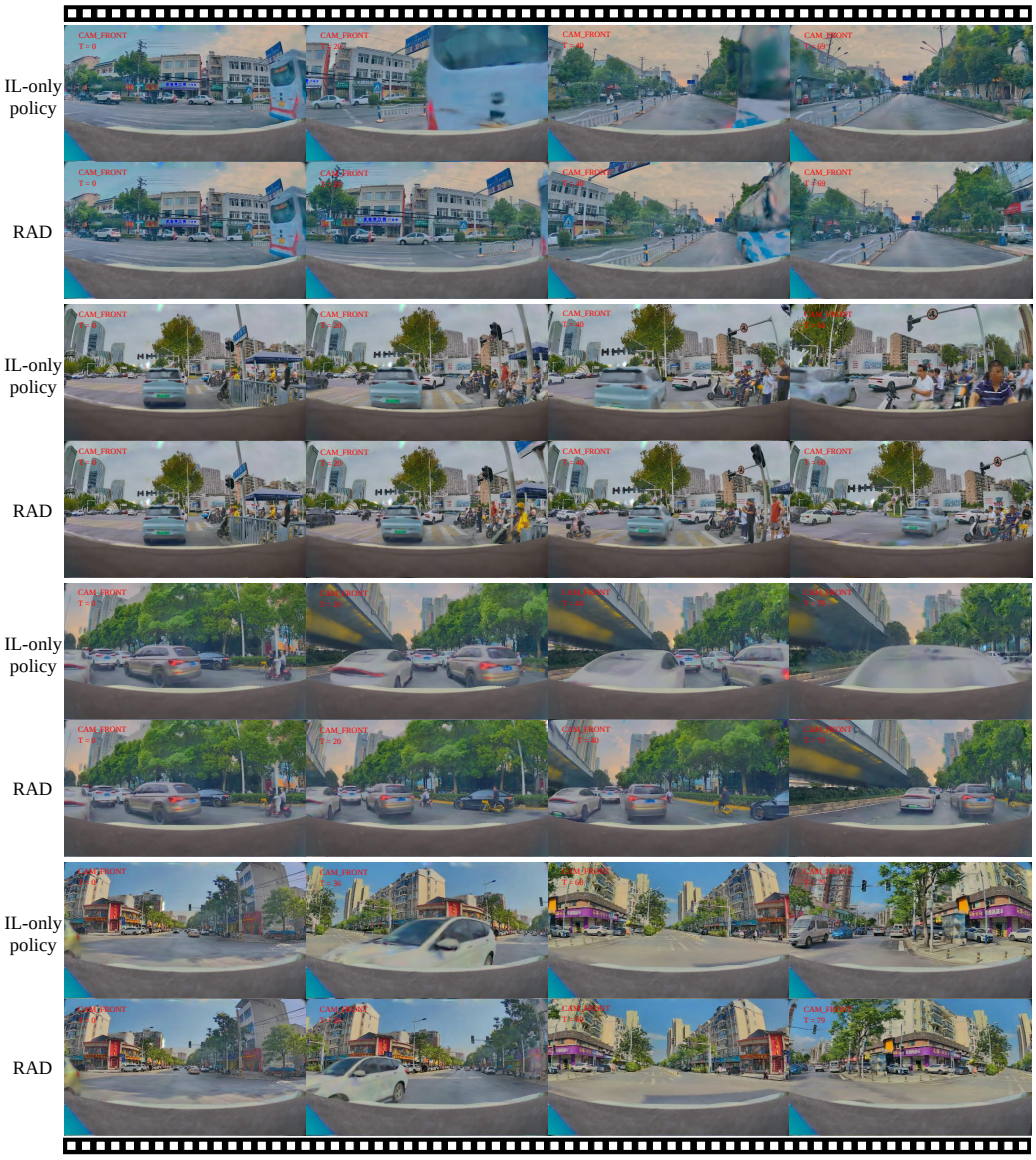


Figure 7: **More Qualitative Results.** Comparison between the IL-only policy and RAD in various driving scenarios: Detour (Rows 1-2), Crawl in Dense Traffic (Rows 3-4), Traffic Congestion (Rows 5-6), and U-turn(Rows 7-8).

https://doi.org/10.52202/085713-1092

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Yes, the main claims in the abstract and introduction accurately reflect the paper's contributions and scope, providing a clear and consistent overview of the methods and findings presented.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Yes, the paper includes a discussion of the limitations in Sec 5, outlining potential constraints and directions for future improvement.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: Yes, the paper provides a complete and correct proof for each theoretical result, along with the full set of underlying assumptions.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Yes, the paper fully discloses all necessary details to reproduce the experimental results, ensuring the validity of its claims and conclusions.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

https://doi.org/10.52202/085713-1092

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: Due to company confidentiality and data privacy policies, we are currently unable to release the dataset. However, we are committed to promoting reproducibility and plan to open-source the key components of our model implementation after the review process.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Yes, the paper clearly specifies all experimental settings necessary to understand and interpret the results.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Yes, the paper includes appropriate information on statistical significance to ensure the reliability of the experimental results.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Yes, the paper provides detailed information on the computational resources used to support reproducibility.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Yes, the research fully adheres to the NeurIPS Code of Ethics, ensuring ethical standards in data use, experimentation, and reporting.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Yes, the paper discusses the potential societal impacts of the work.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.

https://doi.org/10.52202/085713-1092

- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: This paper does not involve data or models that pose such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Yes, all creators and original owners of assets used in the paper are properly credited, and their licenses and terms of use are explicitly acknowledged and respected.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: Yes, all new assets introduced in the paper are well documented, with accompanying documentation provided to facilitate their use and understanding.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

https://doi.org/10.52202/085713-1092

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.