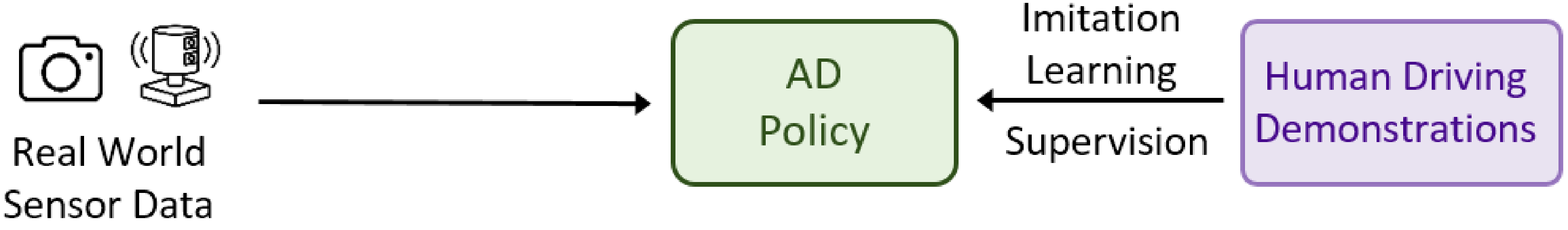
End-to-End Autonomous Driving Training Paradigm

(a) Imitation Learning for End-to-End AD



Limitations

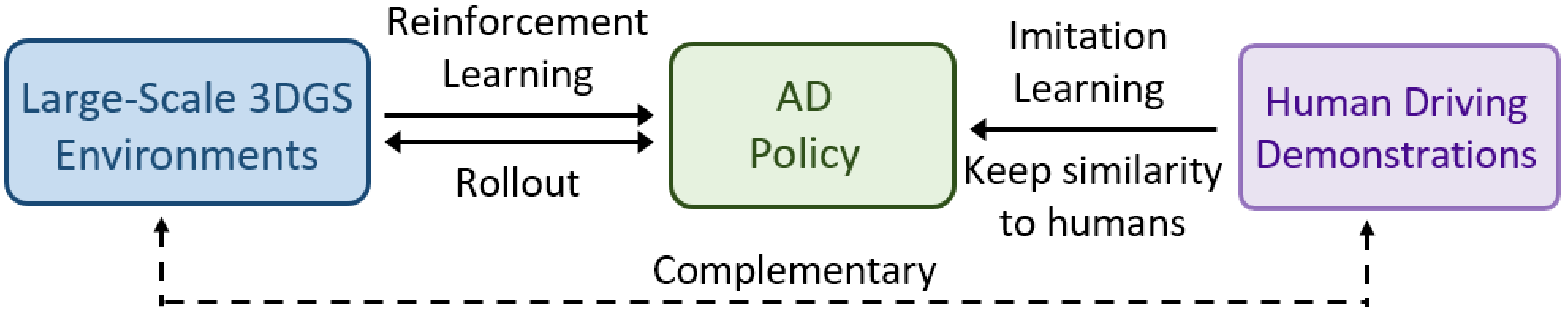
- Gap between open-loop training and closed-loop deployment
- Causal confusion

Solving above issues requires closed-loop reinforcement learning with an interactive driving environment.

Difficulty : The need for an environment with sensor data rendering ability render novel view according to ego vehicle’s position and orientation.

- Real-world driving environment
  - High safety risks and operational costs
- Simulator like CARLA
  - Gap between game-engine simulation data and real-world data
  - Naive actor behavior

RAD: 3DGS-based Reinforcement Learning with Imitation Learning for End-to-End AD


Large-Scale 3DGS Environments
Reinforcement Learning
Rollout
AD Policy
Imitation Learning
Keep similarity to humans
Human Driving Demonstrations
Complementary

Advantages

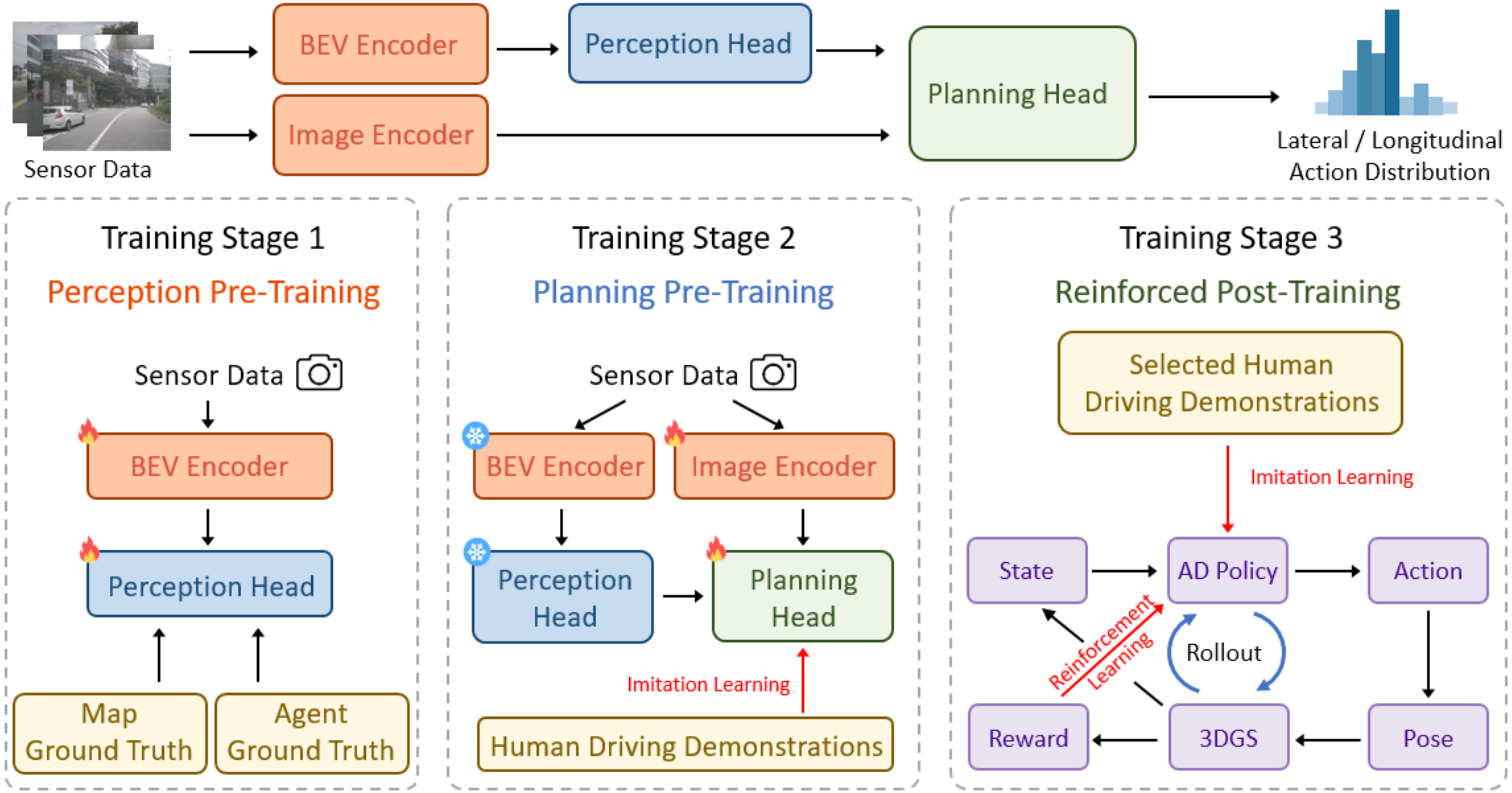
Realistic digital world

Model the causations

Narrow open-loop gap

Hao Gao, Shaoyu Chen, Bo Jiang, Bencheng Liao, Yiang Shi, Xiaoyang Guo, Yuechuan Pu, Haoran Yin, Xiangyu Li, Xinbang Zhang, et al. **RAD: Training an end-to-end driving policy via large-scale3dgs-based reinforcement learning. NeurIPS** 2025.

RAD: 3DGS-based Reinforcement Learning with Imitation Learning for End-to-End AD

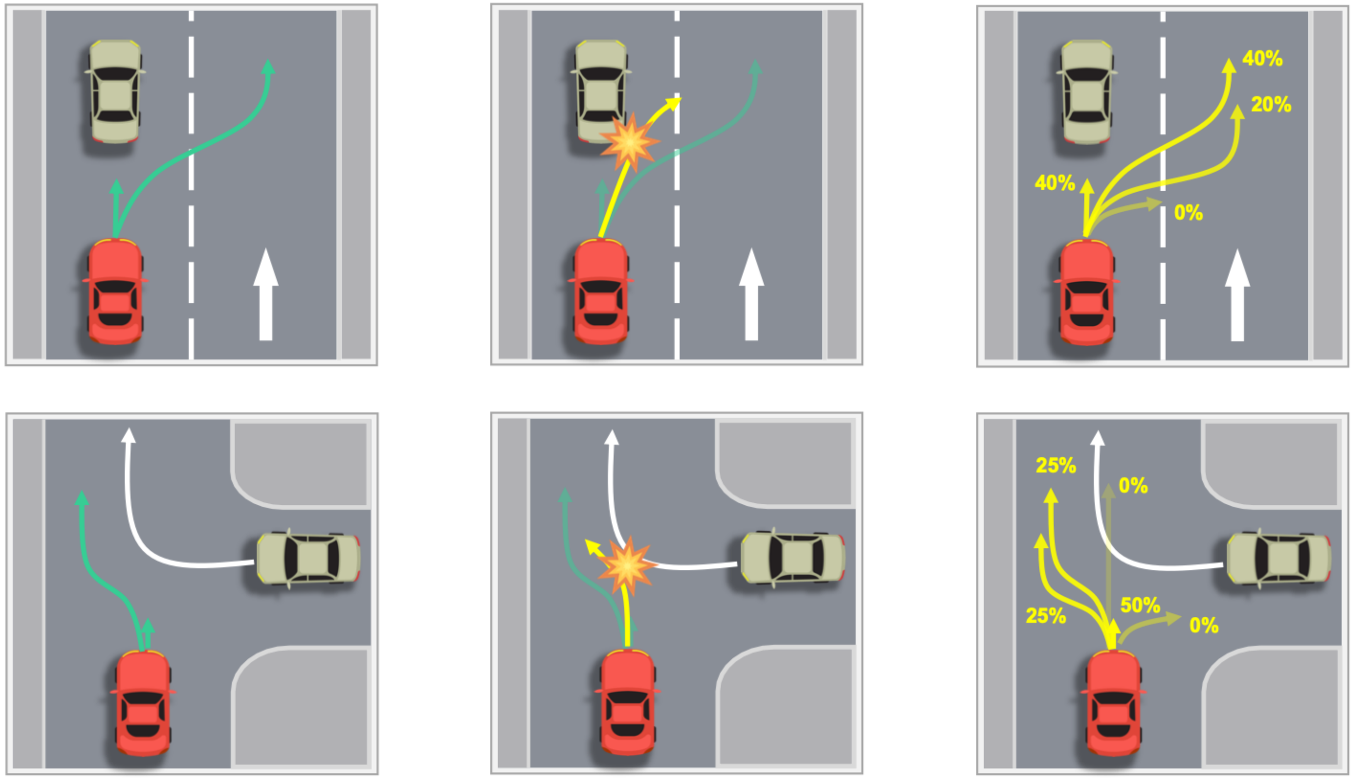


Overall framework of RAD. RAD adopts a three-stage training paradigm: (1) Perception pre-training – using map and agent ground truths to guide instance-level token encoding; (2) Planning pre-training – initializing the action distribution from large-scale driving demonstrations; (3) Reinforced post-training – synergistically fine-tuning the policy with RL and IL.

Three core problems:

1. How to model appropriately for reinforcement learning exploration?
2. How to design Efficient Training framework and Interaction mechanism between the environment and AD policy?
3. What kind of reward is suitable?

RAD: Probabilistic Modeling with Decoupled Lateral and Longitudinal Actions



Driving Demonstrations **Deterministic Planning** **Probabilistic Planning**

1. **How to model appropriately for reinforcement learning exploration?**

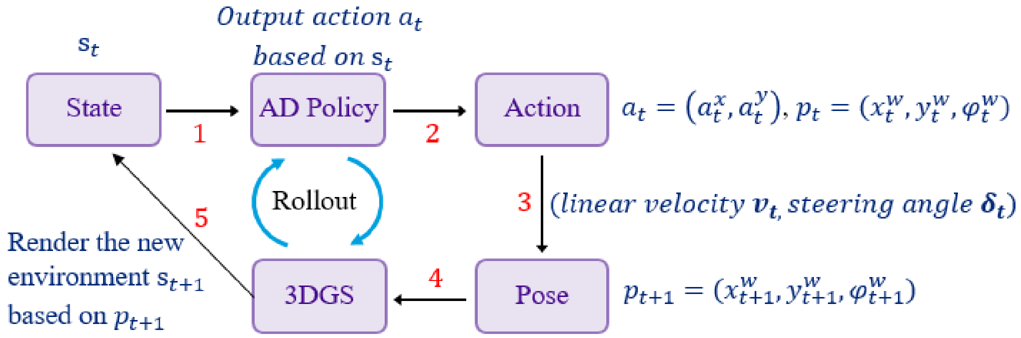
Reinforcement learning explores in uncertain environments, while probabilistic modeling formalizes such uncertainty. Our previous work VADv2 modeled 4,096 3s trajectories and achieved strong closed-loop results in CARLA, but the large discrete action space made RL exploration and convergence difficult!

RAD abandons the large discrete action space of VADv2 and decomposes each 0.5 s driving action into two independent dimensions: lateral (steering) and longitudinal (acceleration/braking), each with 61 discrete options, with softmax applied to each dimension to form a decoupled, small action space, probabilistic action modeling scheme.

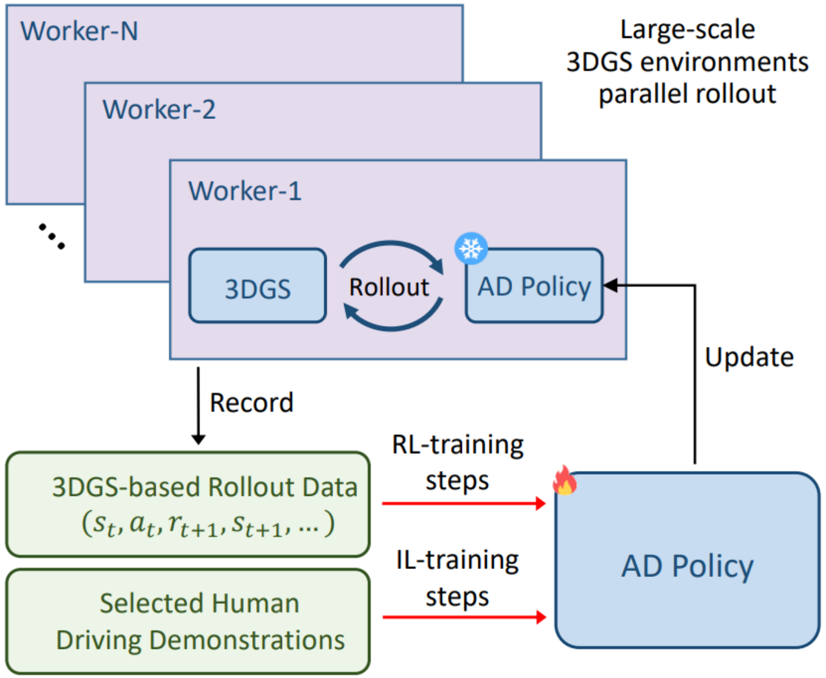
Shaoyu Chen, Bo Jiang, Hao Gao, Bencheng Liao, Qing Xu, Qian Zhang, Chang Huang, Wenyu Liu, Xinggang Wang*. **VADv2: End-to-End Vectorized Autonomous Driving via Probabilistic Planning.** arXiv:2402.13243, 2024.

RAD: Interaction Mechanism between the Environment and AD Policy and Training Framework

Interaction mechanism:


s_t
Output action a_t based on s_t
State
1
AD Policy
2
Action
$a_t = (a_t^x, a_t^y),\ p_t = (x_t^w, y_t^w, \varphi_t^w)$
Rollout
3
(linear velocity $\boldsymbol{v}_t$, steering angle $\boldsymbol{\delta}_t$)
5
Render the new environment s_{t+1} based on p_{t+1}
3DGS
4
Pose
$p_{t+1} = (x_{t+1}^w, y_{t+1}^w, \varphi_{t+1}^w)$

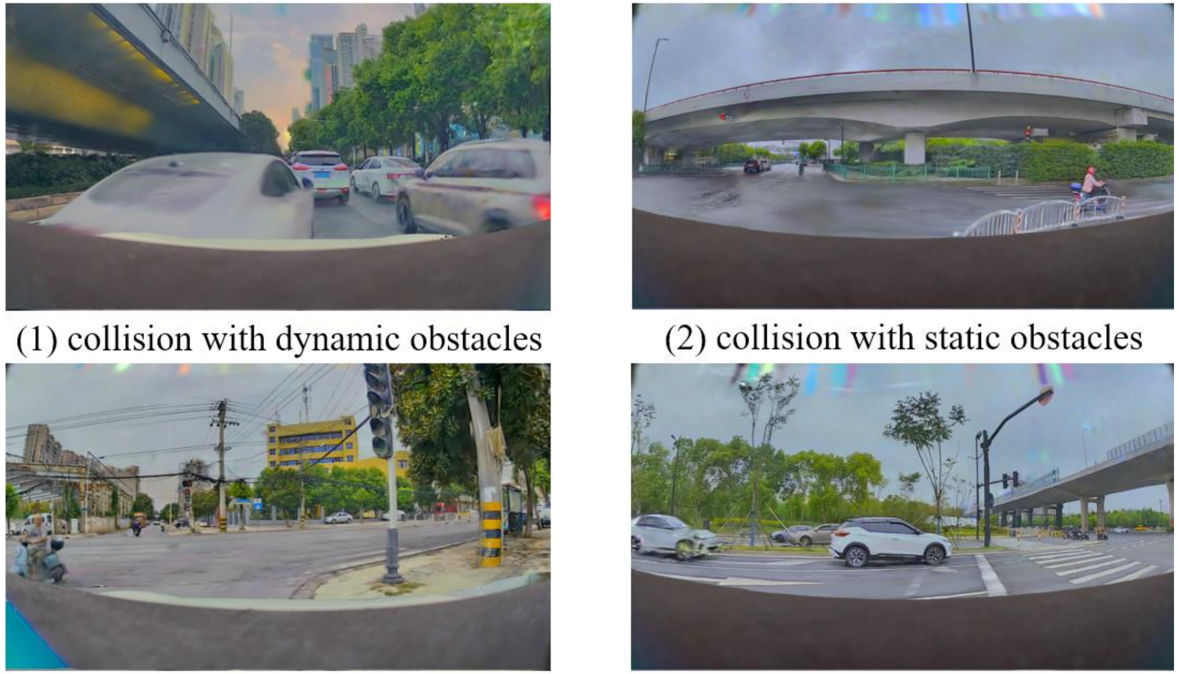
IL + RL co-optimization:


Worker-N
Worker-2
Worker-1
Large-scale 3DGS environments parallel rollout
3DGS
Rollout
AD Policy
Update
Record
3DGS-based Rollout Data $(s_t, a_t, r_{t+1}, s_{t+1}, \ldots)$
Selected Human Driving Demonstrations
RL-training steps
IL-training steps
AD Policy

2. **How to design Efficient Training framework and Interaction mechanism between the environment and AD policy?**

   1. In 3DGS environment, the AD policy outputs lateral and longitudinal actions each step, which are converted via a kinematic bicycle model to update the ego vehicle’s state; other agents replay logged trajectories, and the updated state is fed back for the next step.
   2. Three-stage training paradigm: (1) Perception pre-training; (2) Planning pre-training – initializing the action distribution from large-scale driving demonstrations; (3) Reinforced post-training – IL + RL co-optimization. RL explores rare scenarios; IL constrains actions to stay safe and human-like.

RAD: Reward Design – Safe Driving and Error Correction Along Human Routes



(1) collision with dynamic obstacles

(2) collision with static obstacles

(3) positional deviation from the expert trajectory

(4) heading deviation from the expert trajectory

Example of four reward sources.
(1) Collision with a dynamic obstacle → dynamic collision reward;
(2) Collision with a static obstacle → static collision reward;
(3) Exceeding position deviation threshold → position deviation reward;
(4) Exceeding heading deviation threshold → heading deviation reward.

3. **What kind of reward is suitable?**

In autonomous driving, safety is paramount. Dynamic and static collision penalties prevent RL from learning unsafe policies that ignore obstacles, while position and orientation deviation penalties use human-validated safe routes to correct errors early, encouraging human-like trajectories. Overall, the design prioritizes mitigating collisions and potential hazards.

RAD: Loss Functions

Policy Optimization:

We propagate rewards using generalized advantage estimation (GAE), compute the advantage estimates, and, following the proximal policy optimization (PPO) framework, define separate objective functions to optimize the lateral and longitudinal policy dimensions.

Advantage Estimation：

$$
\begin{aligned}
\delta_t^x &= r_t^x + \gamma V_x(s_{t+1}) - V_x(s_t), \\
\delta_t^y &= r_t^y + \gamma V_y(s_{t+1}) - V_y(s_t), \\
\hat{A}_t^x &= \sum_{l=0}^{\infty} (\gamma\lambda)^l \delta_{t+l}^x \,, \\
\hat{A}_t^y &= \sum_{l=0}^{\infty} (\gamma\lambda)^l \delta_{t+l}^y \,.
\end{aligned}
$$

Policy Optimization Objective:

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

Auxiliary Objectives:

To address the sparse reward problem in RL, we introduce auxiliary objectives that penalize undesirable behaviors by adjusting the action probability distribution. For example, if there is a collision risk ahead, the probability of acceleration is reduced while that of deceleration is increased.

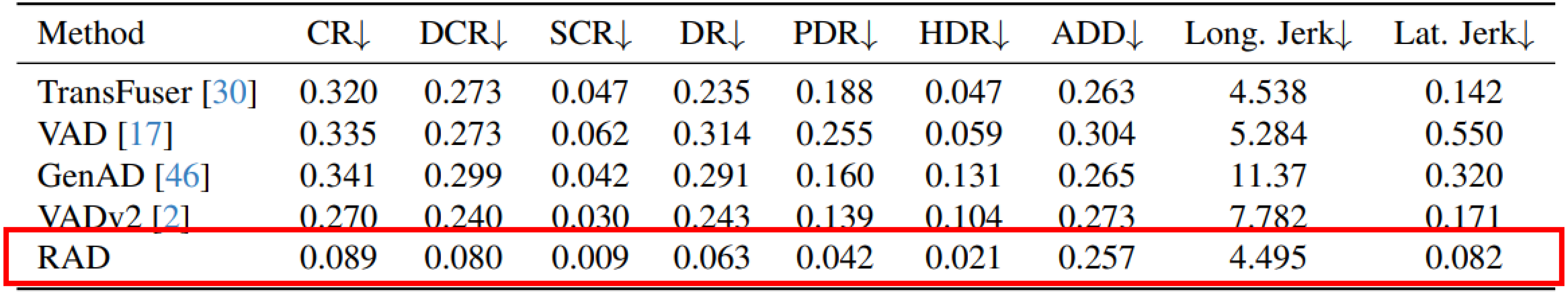
Sum of Action Probabilities:

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

Auxiliary Optimization Objective:

$$
\begin{aligned}
\mathcal{L}_{\mathrm{dc}}(\theta_y) &= \mathbb{E}_t\left[\hat{A}_t^{\mathrm{dc}} \cdot f_{\mathrm{dc}} \cdot \left(\Delta\pi_y^{\mathrm{dec}} - \Delta\pi_y^{\mathrm{acc}}\right)\right], \\
\mathcal{L}_{\mathrm{sc}}(\theta_x) &= \mathbb{E}_t\left[\hat{A}_t^{\mathrm{sc}} \cdot f_{\mathrm{sc}} \cdot \left(\Delta\pi_x^{right} - \Delta\pi_x^{\mathrm{left}}\right)\right], \\
\mathcal{L}_{\mathrm{pd}}(\theta_x) &= \mathbb{E}_t\left[\hat{A}_t^{\mathrm{pd}} \cdot f_{\mathrm{pd}} \cdot \left(\Delta\pi_x^{right} - \Delta\pi_x^{\mathrm{left}}\right)\right], \\
\mathcal{L}_{\mathrm{hd}}(\theta_x) &= \mathbb{E}_t\left[\hat{A}_t^{\mathrm{hd}} \cdot f_{\mathrm{hd}} \cdot \left(\Delta\pi_x^{right} - \Delta\pi_x^{\mathrm{left}}\right)\right].
\end{aligned}
$$

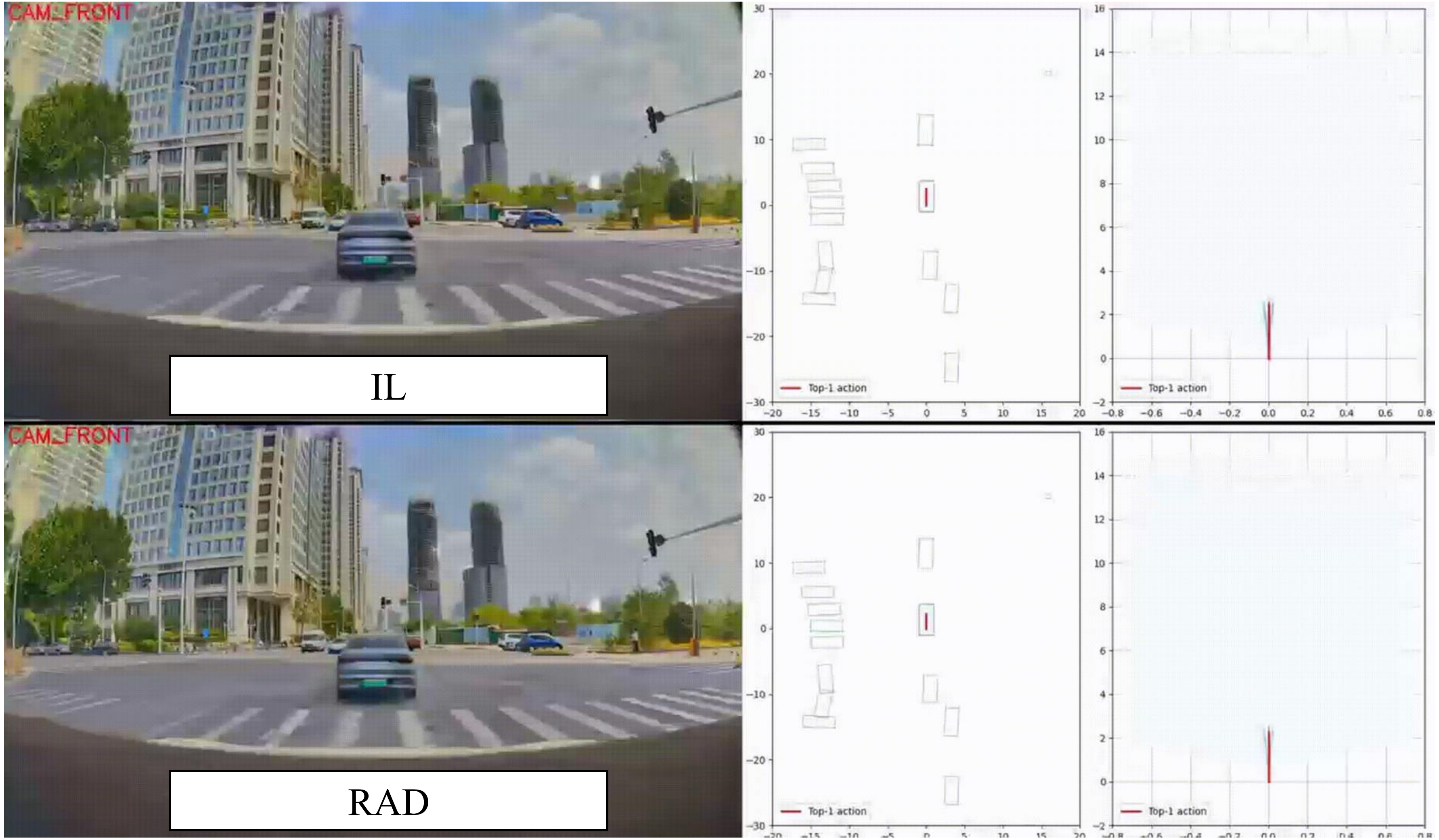
RAD: Superior Performance



Method	CR↓	DCR↓	SCR↓	DR↓	PDR↓	HDR↓	ADD↓	Long. Jerk↓	Lat. Jerk↓
TransFuser [30]	0.320	0.273	0.047	0.235	0.188	0.047	0.263	4.538	0.142
VAD [17]	0.335	0.273	0.062	0.314	0.255	0.059	0.304	5.284	0.550
GenAD [46]	0.341	0.299	0.042	0.291	0.160	0.131	0.265	11.37	0.320
VADv2 [2]	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

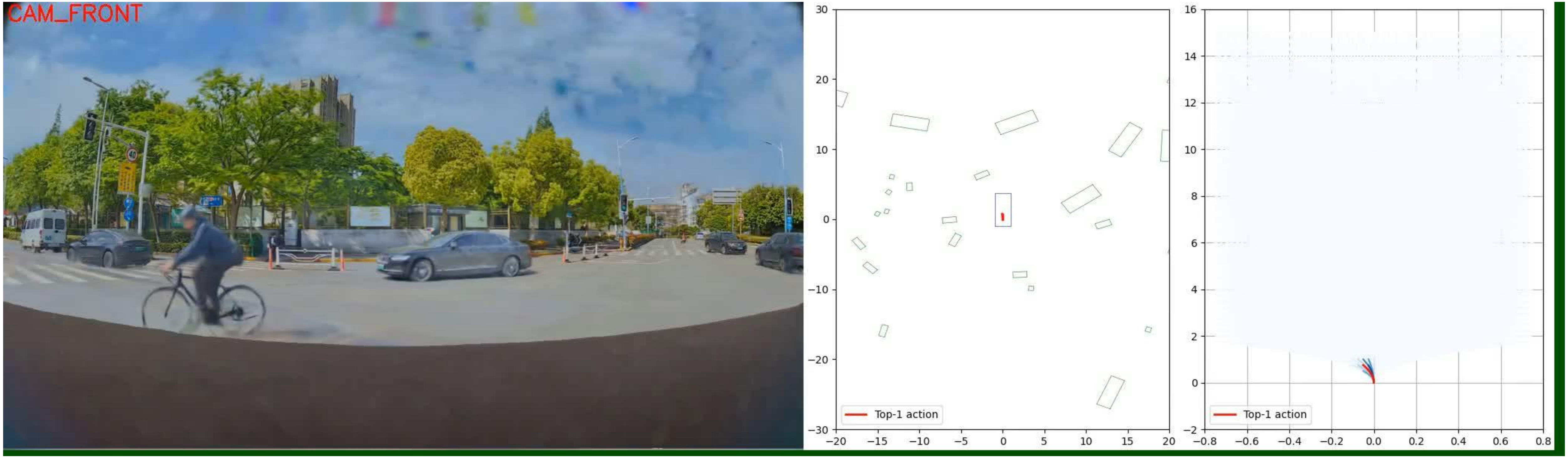
On the 3DGS-based closed-loop evaluation benchmark, RAD achieves state-of-the-art performance in end-to-end autonomous driving tasks, reducing the collision rate by three times compared to IL methods.

RAD: Effectively Avoids Collisions Compared with Policies Trained with Imitation Learning



RAD can effectively avoid collisions with dynamic and static obstacles in complex traffic scenarios, making safer and more reasonable decisions compared to policies trained with imitation learning.

RAD: Closed-Loop Visualization Video in the 3DGS Digital Twin Environment


CAM_FRONT
30
20
10
0
−10
−20
−30
−20
−15
−10
−5
0
5
10
15
20
Top-1 action
16
14
12
10
8
6
4
2
0
−2
−0.8
−0.6
−0.4
−0.2
0.0
0.2
0.4
0.6
0.8
Top-1 action