2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

Proactive Risk-Aware Trajectory Planning for Autonomous Driving in Unstructured Environments via Reinforcement Learning with Adaptive Reward Design

Jiawei Du[1,†], Weiming Qu[1,†], Shenghai Yuan[3], Jia Wang[1], Qifei Bai[4], Xinbo Fang[4], Xihong Wu[1,2], Dingsheng Luo[1,2,*]

***Abstract*— Trajectory planning for autonomous driving in dynamic unstructured traffic remains a fundamental challenge. Existing methods are often reactive, i.e., they only respond to observed situations without explicitly anticipating future risks. Moreover, most reinforcement learning based approaches rely on manually crafted reward functions, which limits their adaptability and generalization across complex driving scenarios. In this paper, we propose a novel RL-based trajectory planning framework that integrates proactive obstacle avoidance and adaptive reward learning. Specifically, our planner predicts the future trajectories of surrounding traffic participants as well as potential ghost-probe risk zones, and proactively avoids these high-risk regions during planning. In addition, we introduce a large-model agent that dynamically adjusts the reward signals according to evolving traffic contexts, enabling more adaptive and robust policy learning compared with fixed reward designs. To evaluate our method, we build a high-fidelity simulation environment based on the Peking University campus, which provides realistic unstructured traffic scenarios. Extensive experiments demonstrate that our method significantly improves safety, efficiency, and generalization over state-of-the-art baselines, particularly in scenarios with occlusions and unpredictable behaviors.**

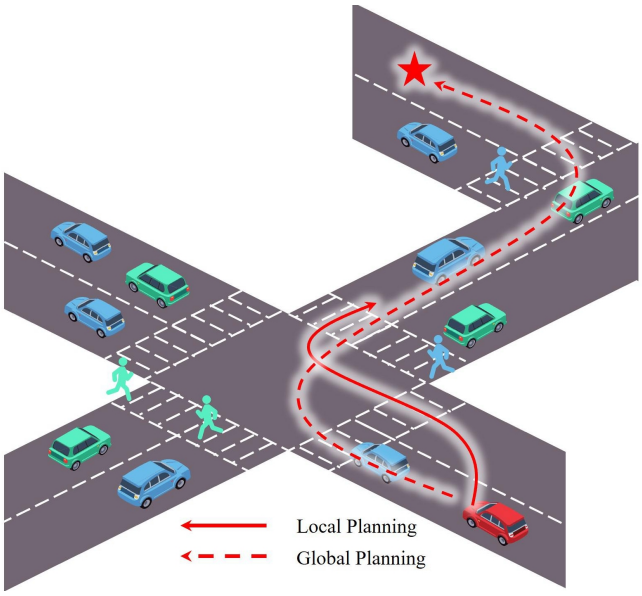


Fig. 1: Schematic diagram of trajectory planning

I. INTRODUCTION

Trajectory planning is a core component of autonomous driving, as it directly determines the safety, efficiency, and comfort of vehicle navigation in complex traffic environments. The trajectory planning problem can be broadly defined as generating a feasible, safe, and efficient trajectory that guides the ego vehicle toward its navigation goal while satisfying dynamic and environmental constraints, as illustrated in Fig. 1.

In this paper, we focus on local trajectory planning, which emphasizes short-horizon motion generation under real-time interaction with surrounding traffic participants. Unlike global planning, which primarily considers road-level routing, local trajectory planning must explicitly account for dynamic obstacles, occlusions, and uncertainties, making it particularly critical for safe autonomous driving in complex traffic. Despite remarkable progress in both optimization-based [1], [2] and learning-based [3], [4] approaches, existing methods remain largely reactive [5], [6], i.e., they generate feasible trajectories in response to observed situations but **rarely anticipate future risks and proactively reserve safe margins**. In addition, reinforcement learning (RL)-based planners [7] typically **rely on manually designed reward functions**, which are difficult to generalize across diverse driving contexts and often fail to capture the nuanced trade-offs required in real-world traffic. These challenges hinder the robustness and scalability of current trajectory planning systems, especially in unstructured environments where traffic participants exhibit highly uncertain and heterogeneous behaviors.

To address these challenges, we propose a proactive risk-aware trajectory planning framework based on RL with adaptive reward design. Our method predicts the future trajectories of surrounding traffic participants as well as potential ghost-probes risk zones, enabling the planner to proactively avoid high-risk regions by reserving adequate safety margins. Furthermore, we leverage a large-model agent to dynamically adjust reward signals according to evolving traffic contexts, thereby improving adaptability and generalization beyond fixed reward designs. To validate our framework, we construct a high-fidelity simulation environment based on the Peking University campus, providing realistic unstructured

* Corresponding author: Dingsheng Luo (e-mail: dsluo@pku.edu.cn).
† Equal contribution.
[1] National Key Laboratory of General Artificial Intelligence, Key Laboratory of Machine Perception (MoE), School of Intelligence Science and Technology, Peking University, Beijing 100871, China.
[2] PKU-WUHAN Institute for Artificial Intelligence, Wuhan, China.
[3] Nanyang Technological University, Singapore 639798, Singapore.
[4] China Automotive Innovation Corporation, Nanjing, China.

driving scenarios for evaluation.

In summary, the main contributions of this paper are summarized as follows:

- **Proactive obstacle avoidance:** We design a trajectory planning framework that explicitly predicts surrounding traffic participants' future trajectories and potential hidden pedestrian risk zones, enabling proactive avoidance of high-risk areas rather than purely reactive planning.
- **Adaptive reward learning:** We introduce a large-model agent that dynamically adjusts reward signals in reinforcement learning, eliminating the reliance on manually crafted reward functions and improving adaptability in diverse driving contexts.
- **Campus-scale unstructured scenario validation:** We build a high-fidelity simulation environment based on the Peking University campus to study autonomous driving in unstructured traffic, and demonstrate that our method achieves superior safety, efficiency, and generalization compared with state-of-the-art baselines.

II. Related Work

A. Traditional Trajectory Planning Methods

Traditional trajectory planning approaches can be broadly divided into three categories: sampling-based, graph search-based, and artificial potential field methods.

Sampling-based approaches explore the configuration space by randomly generating samples and checking their feasibility through collision detection. They can be further classified into single-query algorithms [8], such as Rapidly-exploring Random Trees (RRT), and asymptotically optimal algorithms [9], such as RRT*. While RRT emphasizes efficiency by quickly finding feasible paths, the resulting trajectories are often suboptimal and suffer from redundant exploration due to the lack of directional guidance. Moreover, the randomness of sampling may lead to discontinuous paths with sharp turns. In contrast, RRT* improves optimality through iterative refinement, but at the cost of higher computational overhead.

Graph search-based methods discretize the environment and leverage search algorithms such as Dijkstra [10] or A* [11]. These approaches rely on prior knowledge of the road network topology and typically provide feasible but suboptimal solutions, as discretization inevitably limits smoothness and accuracy. Increasing the graph size or grid resolution improves trajectory quality but significantly increases computational cost, which hinders real-time applicability.

Artificial potential field (APF) methods [12], [13] model attractive forces from goals and repulsive forces from obstacles, guiding the vehicle along the negative gradient of the combined potential field. While conceptually simple and computationally efficient, APF methods often suffer from local minima and goal inaccessibility problems, particularly in environments with multiple obstacles or conflicting forces.

B. Imitation Learning-Based Methods

Imitation learning (IL) leverages expert demonstrations to learn driving policies. A dataset of expert state-action pairs is collected, from which models are trained to approximate the underlying mapping.

Behavior cloning (BC) [14], [15] directly trains a classifier or regressor to mimic expert policies from offline data. Although effective when expert demonstrations are diverse and accurate, BC suffers from compounding errors in unseen states due to distributional shift. Direct policy learning (DPL) addresses this limitation by iteratively collecting new data through environment interaction and refining policies using expert corrections, as exemplified by methods such as DAgger [16], [17], [18] and Observational IL (OIL) [19]. In this way, DPL improves robustness to previously unseen states.

Inverse reinforcement learning (IRL) also relies on expert demonstrations but focuses on inferring the underlying reward function, which is then used to optimize the agent's policy. Notable approaches include maximum margin [20] and maximum entropy IRL [21].

Despite these advances, IL methods still have limitations. They require large-scale, high-quality expert data, yet often struggle with long-tail scenarios and corner cases. Moreover, human drivers may exhibit inconsistent or contradictory behaviors in the same traffic situation, introducing uncertainty and reducing generalization ability.

C. Reinforcement Learning-Based Methods

Reinforcement learning (RL) offers an alternative by enabling agents to learn through trial-and-error interaction with the environment, without relying on extensive labeled expert data. The goal is to maximize cumulative rewards, leading to adaptive decision-making in complex and uncertain environments.

Value-based methods [7], [22], [23], [24] estimate the expected return for each state-action pair and select the action that maximizes value. Policy-based methods [25], [26], [27], by contrast, directly parameterize the policy and optimize it via gradient ascent. Hierarchical reinforcement learning (HRL) [28], [29], [30] decomposes complex driving tasks into multi-level subproblems, thereby improving scalability and decision efficiency—for example, high-level policies deciding lane-change maneuvers while low-level controllers execute detailed trajectories.

However, RL-based approaches face several challenges: they demand substantial computational resources and training time, and their performance heavily depends on the fidelity of the simulation environment. Furthermore, most existing RL planners remain reactive in nature, focusing only on responding to observed states, and rely on manually designed reward functions, which limits adaptability to complex and dynamic driving scenarios.

In summary, classical trajectory planning methods offer computational efficiency and feasibility but depend on prior environment information and often fail to achieve optimal and safe trajectories in complex dynamic scenarios. Imitation learning methods can leverage expert knowledge to directly learn policies, but they heavily rely on large-scale labeled data and exhibit limited generalization in long-tail or corner

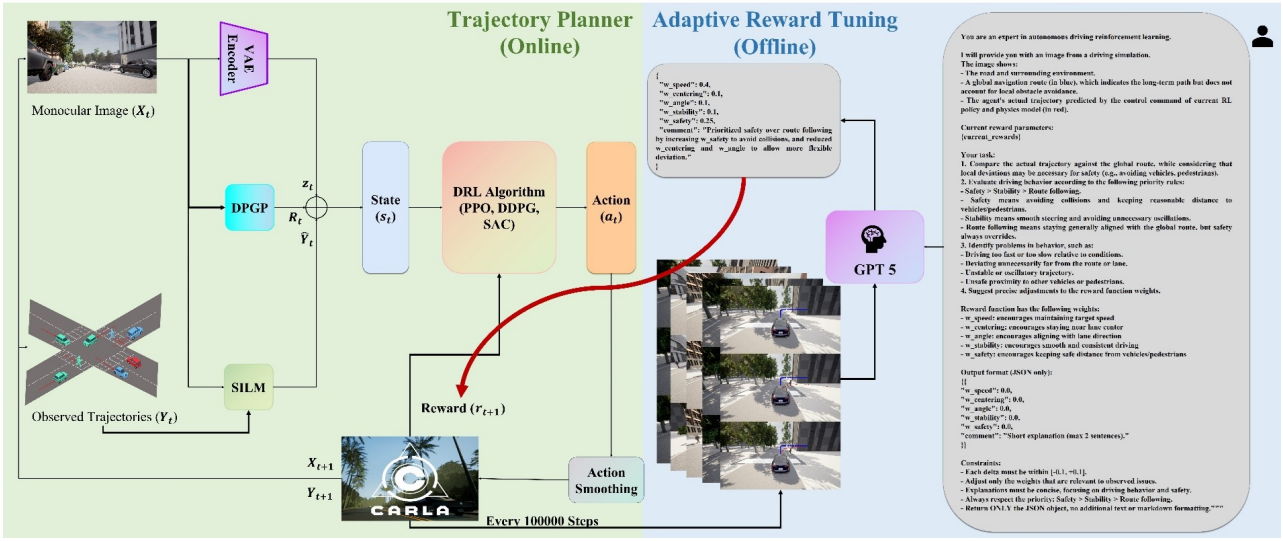


Fig. 2: Overview of the proposed trajectory planner. The overall training process follows a reinforcement learning loop, where the agent interacts with the CARLA simulator to collect transitions consisting of states, actions, rewards, and next states. Policy parameters are updated based on these experiences using the selected DRL algorithm. The integration of SILM and DPGP ensures that the agent proactively considers future risks, while the periodic guidance from the large-model reward agent dynamically refines the reward function.

cases. Reinforcement learning methods provide adaptability and exploration capabilities, yet existing approaches are mostly reactive and depend on manually designed reward functions, limiting their performance in complex traffic environments.

III. Problem Formulation

We formulate trajectory planning as a constrained optimization problem in dynamic and unstructured environments. Let the ego vehicle state at time step t be

$$x_t = [p_t, v_t, a_t, \theta_t], \tag{1}$$

where $p_t \in \mathbb{R}^2$ denotes the position, v_t the velocity, a_t the acceleration, and θ_t the heading. A trajectory is defined as a finite sequence of states within a planning horizon H:

$$\tau = \{x_t, x_{t+1}, \ldots, x_{t+H}\}. \tag{2}$$

The objective of local trajectory planning is to generate a trajectory τ that ensures safety, efficiency, and comfort, while satisfying both vehicle dynamics and environmental constraints. Formally, we aim to solve

$$\tau^* = \arg\min_{\tau} J(\tau), \tag{3}$$

subject to the following constraints:

$$x_{t+1} = f(x_t, u_t), \quad u_t \in \mathcal{U}, \tag{4}$$

$$\text{Dist}(p_t, \mathcal{O}_t) \geq d_{\text{safe}}, \quad \forall t \in [0, H], \tag{5}$$

$$x_t \in \mathcal{X}_{\text{valid}}, \quad \forall t, \tag{6}$$

where $J(\tau)$ is a cost function balancing efficiency, smoothness, and safety; $f(\cdot)$ denotes the vehicle dynamics model; u_t is the control input constrained by the feasible set $\mathcal{U}$; $\mathcal{O}_t$ represents the set of dynamic obstacles at time t; and d_{safe} is the minimum safety distance.

To incorporate proactive risk awareness, we extend the formulation by predicting: 1) the future trajectories of surrounding traffic participants, and 2) the ghost-probe zones, i.e., occluded areas where pedestrians or vehicles may suddenly emerge.

Accordingly, we impose the additional constraint:

$$\text{Dist}(p_t, \mathcal{R}_t) \geq d_{\text{risk}}, \quad \forall t \in [0, H], \tag{7}$$

where $\mathcal{R}_t$ denotes the set of predicted risk regions and d_{risk} is the safety margin proactively reserved for such high-risk areas.

IV. Methods

A. Framework Overview

The overall workflow of our framework is illustrated in Fig. 2. At each time step, the system receives raw sensory inputs, processes them into a compact representation, predicts both the future trajectories of traffic participants and potential ghost-probe zones, and fuses these outputs into a unified state representation. This state is then fed into a reinforcement learning (RL)-based planner, which generates candidate actions. An action smoothing module ensures that these actions are executed smoothly in the environment. Meanwhile, the training pipeline follows the standard RL loop, augmented by an adaptive reward tuning mechanism guided by a large-model agent.

B. State Representation

We take a monocular image of the environment and the observed trajectories of surrounding traffic participants as input. The image is encoded by a variational autoencoder (VAE), which produces compact latent features that preserve high-level semantics while reducing dimensionality. To enhance risk awareness, two predictive modules are employed. The

SILM module [31] forecasts the possible future trajectories of surrounding agents, capturing multimodal motion patterns. In parallel, the DPGP module [32] identifies occluded regions where pedestrians or vehicles might suddenly appear. The outputs of these components, together with the VAE features, are fused into a rich state representation that captures both the current environment and potential future risks. The fused state representation can be denoted as

$$s_t = \text{concat}(z_t, \hat{Y}_t, \mathcal{R}_t), \tag{8}$$

where z_t is the VAE latent feature, $\hat{Y}_t$ the predicted trajectories from SILM, and $\mathcal{R}_t$ the ghost-probe risk map from DPGP.

C. Trajectory Planner

The fused state is fed into a deep reinforcement learning (DRL)-based planner, such as PPO [33], DDPG [34], and SAC [35]. We implement PPO to train the policy. The policy π_ϕ outputs an action a_t:

$$a_t \sim \pi_\phi(a_t \mid s_t). \tag{9}$$

The actions should balance three key objectives:

- Safety, by avoiding collision and high-risk areas;
- Efficiency, by reducing unnecessary detours and ensuring timely navigation;
- Comfort, by minimizing abrupt accelerations and oscillatory maneuvers.

The planner is responsible for generating proactive and risk-aware local trajectories by optimizing long-term rewards rather than relying solely on reactive responses.

$$J(\phi; \theta_r) = \mathbb{E}_{\tau \sim \pi_\phi}\Big[\sum_{t=0}^{T} \gamma^t r(x_t, a_t; \theta_r)\Big], \tag{10}$$

To capture the diverse objectives of autonomous driving in unstructured environments, we propose a **hybrid reward function** that jointly considers *efficiency*, *stability*, *safety*, and *comfort*. At each time step t, the reward r_t is composed of four modules: *speed reward*, *lane-keeping reward*, *safety reward*, and a final *fusion module*. Their formulations and underlying motivations are detailed below.

a) Speed Reward ($r_{speed,t}$): Efficiency-Oriented: The speed reward encourages the ego-vehicle to maintain an appropriate cruising velocity while dynamically adapting to surrounding traffic. Let v_t denote the ego-vehicle's speed, v_{min} the minimum allowable speed, v_{max} the maximum safe speed, and $v_{\text{target,base}}$ the scenario-specific base target speed.

To account for interactions with preceding vehicles, we define a *dynamic target speed*:

$$v_{\text{target,dyn},t} = \begin{cases} \min(v_{\text{target,base}},\, v_{\text{front},t} + \Delta v_{\text{buffer}}), \\ \qquad d_{\text{front},t} < 2d_{\text{safe}}, \\ v_{\text{target,base}}, \quad \text{otherwise.} \end{cases} \tag{11}$$

where $v_{\text{front},t}$ is the speed of the closest leading vehicle, $d_{\text{front},t}$ the gap to that vehicle, d_{safe} the minimum safety distance, and Δv_{buffer} a small velocity margin (set to 2 km/h in experiments).

The reward is then defined as:

$$r_{\text{speed},t} = \begin{cases} \frac{v_t}{v_{\text{min}}}, & v_t < v_{\text{min}}, \\ 1.0 - \frac{v_t - v_{\text{target,dyn},t}}{v_{\text{max}} - v_{\text{target,dyn},t}}, & v_t > v_{\text{target,dyn},t}, \\ 1.0, & \text{otherwise.} \end{cases} \tag{12}$$

b) Lane-Keeping Reward ($r_{lane,t}$): Stability-Oriented: This module penalizes deviations from lane center, heading misalignment, and lateral oscillations, ensuring smooth and stable driving. It consists of three factors:

(i) Centering factor:

$$\gamma_{\text{center},t} = \max\left(1 - \frac{d_{\text{center},t}}{d_{\text{max}}}, 0\right), \tag{13}$$

(ii) Heading angle factor:

$$\gamma_{\text{angle},t} = \max\left(1 - \frac{|\theta_t|}{\theta_{\text{max}}}, 0\right), \tag{14}$$

(iii) Lateral stability factor:

$$\gamma_{\text{stability},t} = \max\left(1 - \frac{\sigma_{d,t}}{\sigma_{\text{max}}}, 0\right), \tag{15}$$

$$r_{\text{lane},t} = w_{\text{centering}} \cdot \gamma_{\text{center},t} + w_{\text{angle}} \cdot \gamma_{\text{angle},t} + w_{\text{stability}} \cdot \gamma_{\text{stability},t} \tag{16}$$

c) Safety Reward ($r_{safe,t}$): Collision Avoidance-Oriented: The safety reward enforces a safe separation from surrounding vehicles and pedestrians. Let $d_{\text{min},t}$ denote the minimum distance to any obstacle at time step t:

$$d_{\text{min},t} = \min\left(\min_{p_{\text{v}} \in \mathcal{P}_{\text{veh},t}} d(p_t, p\text{v}), \min_{p_{\text{p}} \in \mathcal{P}_{\text{ped},t}} d(p_t, p\text{p})\right), \tag{17}$$

where p_t is the position of the ego vehicle, $\mathcal{P}_{\text{veh},t}$ and $\mathcal{P}_{\text{ped},t}$ represent the sets of positions of surrounding vehicles and pedestrians respectively, and $d(\cdot,\cdot)$ is the Euclidean distance. The reward is then:

$$r_{\text{safe},t} = \begin{cases} \frac{d_{\text{min},t}}{d_{\text{safe}}}, & d_{\text{min},t} < d_{\text{safe}}, \\ 1.0, & \text{otherwise.} \end{cases} \tag{18}$$

d) Final Reward Fusion: The final reward combines a *dynamic term* to enforce joint satisfaction of all criteria and a fixed static weights term to alleviate sparsity:

$$r_t = (w_{\text{speed}} + \lambda_{\text{speed}}) \cdot r_{\text{speed},t} + r_{\text{lane},t} + (w_{\text{safe}} + \lambda_{\text{safe}}) \cdot r_{\text{safe},t} \tag{19}$$

In the event of a collision ($d_{\text{min},t} < 0.1$ m), a strong penalty is applied ($r_t = -100$), overriding the above formulation.

TABLE I: Key Parameters for the Hybrid Reward Function

Parameter	Description	Value
v_{min}	Minimum allowable speed	2 km/h
v_{max}	Maximum safe speed	20 km/h
$v_{\text{target,base}}$	Base target speed	15 km/h
Δv_{buffer}	Speed buffer for leading vehicles	2 km/h
d_{safe}	Minimum safety distance to obstacles	5 m
d_{max}	Maximum tolerable lateral deviation	1.5 m
θ_{max}	Maximum tolerable heading deviation	30° ($\pi/12$ rad)
σ_{max}	Maximum tolerable lateral stability std	0.5 m
λ_{speed}	Fixed base weight of speed reward term	0.15
λ_{safe}	Fixed base weight of safe reward term	0.25

Fig. 3: The sample of the representation frames. The desired reference trajectory is blue, and the planner-generated trajectory is red.

To ensure the learned actions are dynamically feasible, the outputs of the RL policy (e.g., desired acceleration, steering angle) are refined by an action smoothing module. Let u_t^{new} be the raw policy output and u_t^{old} the action executed in the previous step; the smoothed action u_t^{smoothed} is:

$$u_t^{\text{smoothed}} = \alpha \cdot u_t^{\text{old}} + (1-\alpha) \cdot u_t^{\text{new}}, \tag{20}$$

where $\alpha = 0.75$ is the smoothing factor (balancing responsiveness and stability).

D. Adaptive Reward Tuning

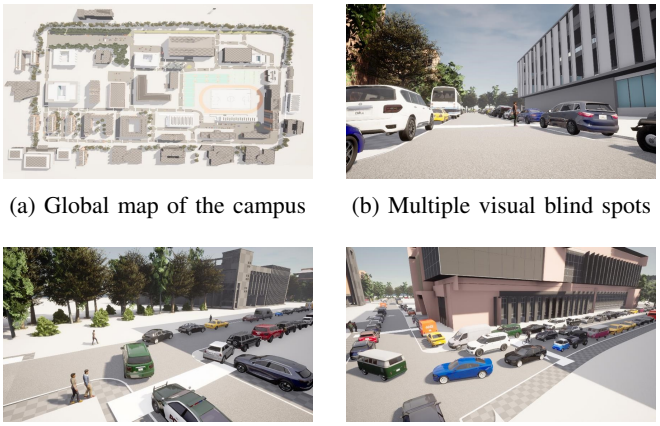
Traditional RL approaches rely on manually designed reward functions, which often fail to generalize across complex and uncertain driving scenarios. To address this limitation, we introduce an adaptive reward tuning mechanism guided by a vision language model (GPT-5) agent. Every 100,000 training steps, five groups of representative frames are sampled. Each group has three frames. Each frame includes the simulated environment, the desired reference trajectory, and the planner-generated trajectory with visual distinction for clarity, as is shown in Fig. 3. These frames are provided to the large-model reward agent, which analyzes potential safety concerns, deviations from expected behaviors, and comfort-related issues. Based on its analysis, the large-model agent suggests adjustments to the reward function, which are then incorporated into the training pipeline. In this paper, the static weights λ_{speed}, λ_{safe} remain fixed to preserve stability, while the dynamic weights $\{w_{\text{speed}}, w_{\text{safe}}, w_{\text{centering}}, w_{\text{angle}}, w_{\text{stability}}\}$ are dynamically adjusted by the large-model reward agent. This adaptive process enables the RL planner to continuously refine its policy, achieving improved robustness and adaptability in diverse traffic conditions.

V. Experiments

A. Environmental Setup

Simulation Environment: We construct a high-fidelity simulation environment based on the CARLA simulator [36]. The environment replicates the campus of Peking University, which serves as a representative example of a complex and unstructured traffic scenario. The campus environment exhibits several distinctive characteristics, including narrow roads, dense traffic participants, mixed traffic modes, frequent pedestrian activities, and numerous blind spots in drivers' field of view, as illustrated in Fig. 4. These features jointly introduce significant uncertainty and interaction complexity, making proactive trajectory planning essential.

Baselines Selections: We use available online implementations [3], [37], [38] for baseline comparisons.



(a) Global map of the campus (b) Multiple visual blind spots

(c) Mixed-traffic scene (d) Dense traffic participants

Fig. 4: Experiment Environment

Evaluation Metrics: To comprehensively evaluate the performance of the proposed proactive trajectory planning framework, we adopt the following metrics: **Average velocity** ($\bar{v}$) measures the mean longitudinal speed of the ego vehicle during a driving episode. This reflects the efficiency of the planner in maintaining reasonable driving progress. **Collision-free average distance** ($\bar{d}_{cf}$) defines the average distance that the vehicle can travel without collisions. A longer distance indicates higher robustness and safety of the planner. **Collision rate.** We compute the probability of collisions normalized by distance traveled. Specifically, we evaluate the collision rate at intervals of 1m, 3m, and 5m. This metric quantifies the risk level of the planner under different scales of operation. **Maximum and average acceleration.** The maximum acceleration (a_{max}) captures the extremity of the vehicle's motion, while the average acceleration ($\bar{a}$) reflects overall smoothness. Lower values typically correspond to more comfortable driving behavior. **Maximum steering angle change** ($angle_{max}$)**.** This metric records the largest instantaneous change in steering angle during the trajectory. It serves as an indicator of trajectory smoothness and driving stability. **Calculated time** ($time$)**.** The average runtime required for the planner to generate a trajectory at each decision step. This metric directly reflects the computational efficiency and real-time feasibility of the method.

B. Implementation Details

All our experiments were conducted on a single RTX 3090Ti GPU. The policy was trained using PPO with the Adam optimizer, and a learning rate scheduler that linearly decayed from 1×10^{-4} to 1×10^{-6} within 2 million steps. The discount factor γ was set to 0.98, and the GAE parameter λ was 0.95. The clipping range was 0.2, and the entropy coefficient was 0.05 to encourage exploration. Each policy update used 10 epochs with 1024 environment steps per rollout. The policy network adopted ReLU activation, with two fully connected layers of sizes [500, 300] for both the policy (π) and value function (V).

TABLE II: Comparison of different methods under varying density conditions.

Density	Method	$\bar{v}$ (km/h)	$\bar{d}_{cf}$ (m)	Collision Rate (%)			$time$ (ms)	a_{max} (m/s^{-2})	$\bar{a}$ (m/s^2)	$angle_{max}$ (°)
				1m	3m	5m				
Crowded	VAD	**6.93**	30.75	2.31	6.81	11.12	**205.22**	4.48	2.91	27.63
	UniAD-Base	5.12	34.12	1.66	4.90	8.04	518.62	**3.49**	**2.00**	25.49
	UniAD-Tiny	5.94	32.68	1.98	5.86	9.59	324.06	4.21	*2.03*	26.81
	Ours (w/o agent)	*6.71*	**57.26**	*1.64*	*4.83*	*7.92*	*223.16*	6.92	3.21	**21.45**
	Ours	5.36	*55.37*	**1.53**	**4.44**	**7.14**	*223.16*	*3.96*	3.13	*23.36*
Normal	VAD	*18.26*	35.28	0.89	2.67	4.48	**206.07**	4.92	3.01	24.14
	UniAD-Base	16.32	38.44	0.95	2.86	4.78	514.52	*3.12*	*2.02*	25.15
	UniAD-Tiny	16.71	35.56	1.10	3.30	5.57	322.46	3.13	**1.97**	19.34
	Ours (w/o agent)	**18.73**	**58.47**	*0.67*	*2.02*	*3.32*	*220.18*	6.80	3.08	*18.66*
	Ours	18.24	*57.98*	**0.65**	**1.98**	**3.30**	*220.18*	**2.96**	2.50	**10.56**
Uncrowded	VAD	**22.18**	50.23	0.56	1.68	2.80	**201.18**	4.01	2.14	27.83
	UniAD-Base	16.95	52.85	0.61	1.84	3.08	498.17	*3.07*	**1.95**	23.77
	UniAD-Tiny	16.51	49.54	0.63	1.90	3.19	307.39	3.25	2.09	16.78
	Ours (w/o agent)	19.86	**60.86**	*0.34*	*1.01*	*1.68*	*210.85*	4.61	2.97	*10.32*
	Ours	*19.92*	*58.05*	**0.32**	**0.96**	**1.62**	*210.85*	**2.58**	*2.02*	**8.21**

Note: Best results are **bolded**, and Second results are *Italic*.

For representation learning, we employed a variational autoencoder (VAE) with a latent dimension of 64, denoted as *vae_64*. This model was pre-trained in the CARLA simulation environment to extract latent features from monocular RGB images. The input images were first resized to a resolution of 640×640 before being encoded into the latent space.

In parallel, the DPGP module produced a ghost risk map feature of 32 dimensions, which was concatenated with the VAE latent code to enrich the state representation. The trajectory prediction results were generated by the SILM module, with a prediction horizon of 20 steps and a prediction interval of 0.1 seconds. And the sensing radius is 25m. Action smoothing factor $\alpha = 0.75$.

C. Quantitative Results

Table II reports the quantitative comparison between our method and three representative baselines under varying traffic density conditions (Crowded, Normal, Uncrowded). Several key observations can be made: Across all density conditions, our method achieves the lowest collision rates at different travel distances. Our planner also achieves superior mean collision-free distance and competitive mean velocity. In terms of motion smoothness, our approach consistently lowers maximum acceleration and maximum steering angle change compared to most baselines. The average runtime of our method is comparable to VAD, and significantly faster than UniAD-Base. This demonstrates that the proposed planner is computationally feasible for real-time applications.

In summary, the results show that our method outperforms existing approaches across a wide range of traffic densities, achieving a favorable balance between **safety**, **efficiency**, **comfort**, and **real-time feasibility**.

D. Qualitative Results

To demonstrate the effectiveness of the proposed proactive trajectory planning framework, we present several representative qualitative examples. These scenarios highlight how the planner adapts its behavior under different traffic conditions, as illustrated in Fig. 5. **(a) Normal driving scenario.** In the absence of nearby traffic participants, the planner generates a smooth and efficient trajectory that closely follows the reference route. This demonstrates the baseline capability of the framework to ensure stable and comfortable driving under regular conditions. **(b) Scenario with surrounding participants.** When vehicles or pedestrians are present in the vicinity, the planner adjusts its trajectory proactively to maintain a safe distance. By leveraging trajectory predictions from SILM, the system can anticipate the motion of surrounding agents and plan accordingly, avoiding collisions while still making progress toward the goal. **(c) Ghost-probe scenario.** In situations where potential risks may arise from occluded regions—commonly known as ghost-probe zones—the planner proactively allocates a safety margin by adjusting its path and speed. This behavior is enabled by the DPGP module, which predicts high-risk blind areas where pedestrians or vehicles might suddenly emerge. As shown in the example, the ego vehicle decelerates and shifts its trajectory to ensure that even unexpected appearances from hidden zones can be safely avoided.

These qualitative results collectively demonstrate that the proposed framework is capable of not only handling standard driving situations but also proactively accounting for dynamic interactions and potential unseen risks. This proactive nature marks a key distinction from conventional reactive trajectory planning methods.

E. Ablation Studies

To further investigate the effectiveness of different components in our framework, we conduct ablation studies by selectively removing the modules of **SILM** (trajectory prediction), **DPGP** (ghost-probe zone prediction), and the **large-model reward agent**. Table III summarizes the results. When only SILM is integrated, the collision rate significantly increases compared to DPGP + planner, indicating that while trajectory prediction improves foresight, the lack of ghost-probe zone

Fig. 5: Qualitative examples of the proposed proactive trajectory planning framework: (a) Normal driving scenario, (b) Scenario with surrounding participants, (c) Ghost-probe scenario, the red bounding boxes indicate the potential ghost-probe zones.

modeling leads to safety risks in occluded environments. Comparing DPGP + SILM + Planner with and without the large-model reward agent shows that the agent helps balance safety and efficiency. Although the improvement in collision rate is relatively small (0.65 vs. 0.67 at 1m), the agent achieves a more stable trade-off by dynamically adjusting the reward function, preventing over-conservatism and ensuring a competitive mean velocity. Meanwhile, as shown in Fig. 6, the inclusion of the reward agent also leads to a more stable training process.

Overall, the ablation results highlight that DPGP, SILM, and the reward agent are all indispensable components, jointly contributing to proactive safety awareness, accurate trajectory forecasting, and adaptive decision-making.

TABLE III: Ablation study on the proposed framework

Planner	SLIM	DPGP	Agent	Collision Rate (%)			$\bar{v}$ (km/h)
				1m	3m	5m	
✓	✗	✓	✗	0.71	2.15	3.60	17.26
✓	✓	✗	✗	1.60	4.65	7.74	**19.35**
✓	✓	✓	✗	0.67	2.02	3.32	18.73
✓	✓	✓	✓	**0.65**	**1.98**	**3.30**	18.24

VI. Conclusion

In this paper, we proposed a proactive trajectory planning framework for autonomous driving in unstructured traffic environments. Unlike existing reactive planners, our approach jointly incorporates SILM-based trajectory prediction, DPGP-based ghost-probe zone modeling, and a large-model reward agent that adaptively adjusts the reinforcement learning objective. This enables the ego vehicle to proactively anticipate potential risks, achieve safer and smoother trajectories, and dynamically balance safety, efficiency, and comfort. We further constructed a high-fidelity campus-scale simulation environment based on the Peking University scenario, and extensive experiments under varying traffic densities demonstrated the superiority of our framework over baselines in terms of collision avoidance, trajectory quality, and computational feasibility. We believe this work opens a promising direction toward resilient and human-aligned trajectory planning for real-world autonomous driving.

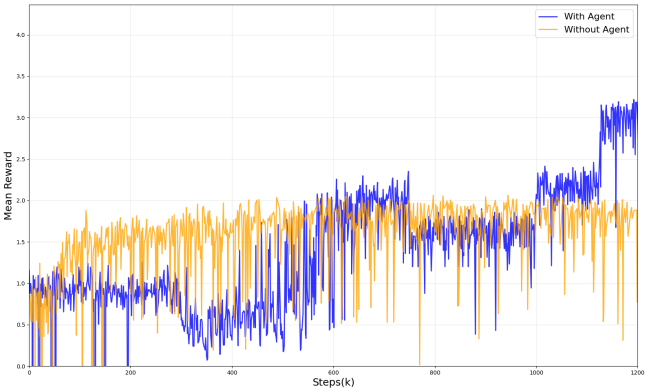


Fig. 6: Training Curve

Acknowledgments

The work is supported in part by the National Natural Science Foundation of China (No. 62176004, No. U1711327), Intelligent Robotics and Autonomous Vehicle Lab (RAV), and Wuhan East Lake High-Tech Development Zone National Comprehensive Experimental Base of Governance of Intelligent Society.

References

[1] Z. Li, B. Zhou, C. Hu, L. Xie, and H. Su, "A data-driven aggressive autonomous racing framework utilizing local trajectory planning with velocity prediction," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2025.

[2] T. Hu, S. Yuan, R. Bai, X. Xu, Y. Liao, F. Liu, and L. Xie, "Swept volume-aware trajectory planning and mpc tracking for multi-axle swerve-drive amrs," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*. IEEE, 2025.

[3] B. Jiang, S. Chen, Q. Xu, B. Liao, J. Chen, H. Zhou, Q. Zhang, W. Liu, C. Huang, and X. Wang, "Vad: Vectorized scene representation for efficient autonomous driving," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

[4] J. Zhang, C. Wang, J. Peng, H. Li, J. Ji, Y. Zhang, and Y. Zhang, "Cafe-ad: Cross-scenario adaptive feature enhancement for trajectory planning in autonomous driving," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2025.

[5] S. Chen, B. Jiang, H. Gao, B. Liao, Q. Xu, Q. Zhang, C. Huang, W. Liu, and X. Wang, "Vadv2: End-to-end vectorized autonomous driving via probabilistic planning," *arXiv preprint arXiv:2402.13243*, 2024.

[6] J. Wang, X. Zhang, Z. Xing, S. Gu, X. Guo, Y. Hu, Z. Song, Q. Zhang, X. Long, and W. Yin, "He-drive: Human-like end-to-end driving with vision language models," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*. IEEE, 2025.

[7] A. Parag, N. Mansard, and E. Misimi, "Optimizing complex control systems with differentiable simulators: A hybrid approach to reinforcement learning and trajectory planning," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2025.

[8] S. Lu, "Path tracking control algorithm for unmanned vehicles based on improved rrt algorithm," in *2022 IEEE 2nd International Conference on Electronic Technology, Communication and Information (ICETCI)*. IEEE, 2022, pp. 1201–1204.

[9] J. Wang, W. Chi, C. Li, C. Wang, and M.-H. Meng, "Neural rrt*: Learning-based optimal path planning," *IEEE Transactions on Automation Science and Engineering*, vol. 17, no. 4, pp. 1748–1758, 2020.

[10] L.-s. Liu, J.-f. Lin, J.-x. Yao, D.-w. He, J.-s. Zheng, J. Huang, and P. Shi, "Path planning for smart car based on dijkstra algorithm and dynamic window approach," *Wireless Communications and Mobile Computing*, 2021.

[11] Y. Li, R. Jin, X. Xu, Y. Qian, H. Wang, S. Xu, and Z. Wang, "A mobile robot path planning algorithm based on improved a* algorithm and dynamic window approach," *IEEE Access*, 2022.

[12] H. Zhang, M. Li, and Z. Wu, "Path planning based on improved artificial potential field method," in *Proc. 33rd Chin. Control Decis. Conf. (CCDC)*, 2021.

[13] J. Luo, Z.-X. Wang, and K.-L. Pan, "Reliable path planning algorithm based on improved artificial potential field method," *IEEE Access*, 2022.

[14] F. Codevilla, M. Müller, A. López, V. Koltun, and A. Dosovitskiy, "End-to-end driving via conditional imitation learning," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2018.

[15] A. Sauer, N. Savinov, and A. Geiger, "Conditional affordance learning for driving in urban environments," in *Conference on Robot Learning*. PMLR, 2018, pp. 237–252.

[16] S. Ross, G. Gordon, and D. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics*. JMLR W&CP, 2011, pp. 627–635.

[17] J. Zhang and K. Cho, "Query-efficient imitation learning for end-to-end autonomous driving," *arXiv preprint arXiv:1605.06450*, 2016.

[18] R. Hoque, A. Balakrishna, E. Novoseller, A. Wilcox, D. Brown, and K. Goldberg, "Thriftydagger: Budget-aware novelty and risk gating for interactive imitation learning," *arXiv preprint arXiv:2109.08273*, 2021.

[19] G. Li, M. Mueller, V. Casser, N. Smith, D. Michels, and B. Ghanem, "Oil: Observational imitation learning," *arXiv preprint arXiv:1803.01129*, 2018.

[20] T. Phan-Minh, F. Howington, T.-S. Chu, M. S. Tomov, R. E. Beaudoin, S. U. Lee, N. Li, C. Dicle, S. Findler, F. Suarez-Ruiz, B. Yang, S. Omari, and E. M. Wolff, "Driveirl: Drive in real life with inverse reinforcement learning," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2023.

[21] K. Lee, D. Isele, E. Theodorou, and S. Bae, "Spatiotemporal costmap inference for mpc via deep inverse reinforcement learning," *IEEE Robot. Autom. Lett.*, 2022.

[22] A. Alizadeh, M. Moghadam, Y. Bicer, N. Ure, U. Yavas, and C. Kurtulus, "Automated lane change decision making using deep reinforcement learning in dynamic and uncertain highway environment," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2019, pp. 1399–1404.

[23] S. Mo, X. Pei, and C. Wu, "Safe reinforcement learning for autonomous vehicle using monte carlo tree search," *IEEE Trans. Intell. Transp. Syst.*, 2022.

[24] G. Li, Y. Yang, S. Li, X. Qu, N. Lyu, and S. Li, "Decision making of autonomous vehicles in lane change scenarios: Deep reinforcement learning approaches with risk awareness," *Transportation Research Part C: Emerging Technologies*, vol. 134, p. 103452, 2022.

[25] D. Saxena, S. Bae, A. Nakhaei, K. Fujimura, and M. Likhachev, "Driving in dense traffic with model-free reinforcement learning," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*. IEEE, 2020, pp. 5385–5392.

[26] Y. Tian, X. Cao, K. Huang, C. Fei, Z. Zheng, and X. Ji, "Learning to drive like human beings: A method based on deep reinforcement learning," *IEEE Trans. Intell. Transp. Syst.*, 2022.

[27] Z. Huang, J. Wu, and C. Lv, "Efficient deep reinforcement learning with imitative expert priors for autonomous driving," *IEEE Trans. Neural Netw. Learn. Syst.*, pp. 1–13, 2022.

[28] Y. Chen, C. Dong, P. Palanisamy, P. Mudalige, K. Muelling, and J. Dolan, "Attention-based hierarchical deep reinforcement learning for lane change behaviors in autonomous driving," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, 2019.

[29] Y. Lu, X. Xu, X. Zhang, L. Qian, and X. Zhou, "Hierarchical reinforcement learning for autonomous decision making and motion planning of intelligent vehicles," *IEEE Access*, 2020.

[30] L. Gao, Z. Gu, C. Qiu, L. Lei, S. Li, S. Zheng, W. Jing, and J. Chen, "Cola-hrl: Continuous-lattice hierarchical reinforcement learning for autonomous driving," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*. IEEE, 2022, pp. 13 143–13 150.

[31] W. Qu, J. Wang, J. Du, Y. Zhu, J. Yu, R. Xia, S. Cao, X. Wu, and D. Luo, "Silm: A subjective intent based low-latency framework for multiple traffic participants joint trajectory prediction," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2025.

[32] W. Qu, J. Du, S. Yuan, J. Wang, Y. Sun, S. Liu, Y. Zhu, J. Yu, S. Cao, R. Xia, X. Tang, X. Wu, and D. Luo, "Dpgp: A hybrid 2d-3d dual path potential ghost probe zone prediction framework for safe autonomous driving," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2025.

[33] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[34] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.

[35] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2018.

[36] A. Dosovitskiy, G. Ros, F. Codevilla, A. M. López, and V. Koltun, "Carla: An open urban driving simulator," in *Proc. Conf. Robot Learn. (CoRL)*, 2017.

[37] Y. Hu, J. Yang, L. Chen, K. Li, C. Sima, X. Zhu, S. Chai, S. Du, T. Lin, W. Wang, L. Lu, X. Jia, Q. Liu, J. Dai, Y. Qiao, and H. Li, "Planning-oriented autonomous driving," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2023.

[38] U. contributors, "Planning-oriented autonomous driving," https://github.com/OpenDriveLab/UniAD, 2023.