RAP: Role-Aware Joint Prediction and Planning in Autonomous Driving

Xiaolong Tang, *Graduate Student Member, IEEE*, Meina Kan, *Member, IEEE*, Shiguang Shan, *Fellow, IEEE*, and Xilin Chen, *Fellow, IEEE*

***Abstract*—Planning the trajectory for an autonomous vehicle (AV) requires considering both the AV's objectives and the behaviors of surrounding agents, which interact in a dynamic process. Joint prediction and planning (JPP) methods have been proposed to model this bidirectional interaction and generate compatible joint futures. However, existing methods often overlook the fundamental differences between the two tasks and treat them almost identically, thus limiting the effect of planning. In this work, we revisit JPP from a role-aware perspective and identify three inherent asymmetries between prediction and planning: information, objective, and feedback. Based on these insights, we propose the Role-Aware Joint Prediction and Planning (RAP) framework, which explicitly differentiates the treatment of the ego vehicle and surrounding agents through three lightweight asymmetric designs. Specifically, RAP introduces a route–identity token pairing mechanism for route-aware yet agent-agnostic conditioning, ego-only vectorized auxiliary losses for safety-constrained planning, and dropout and perturbation applied to the ego's history and kinematic state to improve closed-loop robustness. Comprehensive experiments on the nuPlan benchmark demonstrate that RAP achieves state-of-the-art performance on both open-loop and closed-loop metrics, validating the effectiveness of asymmetric modeling in unified prediction and planning.**

***Index Terms*—Autonomous vehicle navigation, motion and path planning, imitation learning.**

I. Introduction

Autonomous driving has emerged as a transformative technology with the potential to significantly enhance transportation safety and efficiency. The core of an autonomous vehicle (AV) is trajectory planning, which aims to generate safe and efficient paths in dynamic, multi-agent environments populated by vehicles, cyclists, and pedestrians. In these multi-agent systems, each participant's behavior depends not only on their own intentions but also on the anticipated reactions of others. Therefore, an effective AV must not only predict the future motions of surrounding agents but also plan its own trajectory while accounting for this mutual influence.

Classical sequential prediction and planning pipelines [1], [2] decouple the two tasks, resulting in unidirectional information flow and often producing overly conservative or aggressive behaviors. Recent joint prediction and planning (JPP) methods [3], [4] alleviate this issue by modeling bidirectional interactions between the ego vehicle and surrounding agents, yielding more consistent joint futures. However, despite these advances, most existing JPP methods treat prediction and planning almost equivalently, overlooking their fundamentally distinct roles and objectives [5]. Such symmetric treatment blurs the boundary between perception-driven prediction and goal-driven planning, leading to less accurate agent forecasts and suboptimal ego trajectories.

Our key insight is that prediction and planning, although related, are inherently different tasks and should be modeled asymmetrically. We identify three core mismatches between them: (1) information asymmetry, where surrounding-agent prediction is goal-agnostic while the planner has access to a known navigation route; (2) objective asymmetry, where prediction emphasizes distributional accuracy while planning focuses on safety and feasibility; and (3) feedback asymmetry, where prediction is evaluated in an open-loop setting while planning operates in a closed-loop manner in which errors accumulate over time. How to seamlessly integrate prediction and planning under these asymmetric conditions remains an open and underexplored problem.

To address this challenge, we propose a Role-Aware Joint Prediction and Planning (RAP) framework that explicitly differentiates the treatment of the ego vehicle and surrounding agents with minimal modifications. RAP introduces three asymmetric designs to resolve the above mismatches. First, to handle information asymmetry, we design a route–identity token pairing mechanism that enables the AV to recognize its navigation route while preventing other agents from over-focusing on it, achieving lightweight route conditioning without extra modules. Second, to address objective asymmetry, we design and apply ego-only vectorized auxiliary losses (i.e., on-road loss, obstacle-collision loss, and agent-collision loss) to improve planning safety and feasibility, while maintaining accurate agent prediction. Third, to mitigate feedback asymmetry, we employ dropout and perturbation on the ego's history and kinematic states [6],

Received 27 October 2025; accepted 22 November 2025. Date of publication 9 December 2025; date of current version 23 December 2025. This article was recommended for publication by Associate Editor A. H. Qureshi and Editor A. Bera upon evaluation of the reviewers' comments. This work was supported by the National Natural Science Foundation of China under Grant 62495084 and Grant 62461160331. *(Corresponding author: Meina Kan.)*

Xiaolong Tang, Meina Kan, and Xilin Chen are with the State Key Laboratory of AI Safety, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China, and also with the University of Chinese Academy of Sciences, Beijing 100049, China (e-mail: tangxiaolong22s@ict.ac.cn; kanmeina@ict.ac.cn; xlchen@ict.ac.cn).

Shiguang Shan is with the State Key Laboratory of AI Safety, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China, and with the University of Chinese Academy of Sciences, Beijing 100049, China, and also with Peng Cheng National Laboratory, Shenzhen 518055, China (e-mail: sgshan@ict.ac.cn).

This article has supplementary downloadable material available at https://doi.org/10.1109/LRA.2025.3641815, provided by the authors.

Digital Object Identifier 10.1109/LRA.2025.3641815

[7] to prevent shortcut dependencies and enhance closed-loop robustness. Together, these components form a principled role-aware framework that unifies prediction and planning under differentiated information, objectives, and feedback dynamics. Experiments on the nuPlan benchmark demonstrate that RAP achieves state-of-the-art performance on both open-loop and closed-loop metrics, validating the effectiveness of asymmetric modeling for unified prediction and planning. The main contributions of this work are summarized as follows:

- We identify and formalize three fundamental asymmetries between prediction and planning, namely information, objective, and feedback asymmetries.
- We design a role-aware framework with lightweight components including route–identity token pairing mechanism, ego-only vectorized auxiliary losses, and dropout and perturbation applied to the ego's history and kinematic states to explicitly resolve these asymmetries.
- We validate the our approach on the nuPlan benchmark, where RAP achieves state-of-the-art performance on both open-loop and closed-loop metrics, demonstrating that addressing role asymmetry leads to safer and more consistent joint prediction–planning outcomes.

II. Related Work

Imitation-based Trajectory Planning: Learning-based planning methods, particularly imitation learning approaches (e.g., behavior cloning), have gained significant attention for learning driving strategies by mimicking human behavior. Early studies [7], [8], [9], [10] demonstrated the feasibility of imitation-based planning in real-world scenarios, while also highlighting two key challenges.

The first challenge is the covariate shift problem, where the distribution of states during testing differs from the training data. DAgger [11] addresses this by continuously combining model-generated data with expert feedback, ensuring the model adapts to various scenarios. ChauffeurNet [7] and PlanTF [6] simplify this by adding random perturbations to the AV's states and applying random dropout to its history, enabling the model to learn to correct errors without expert feedback. Building on this, [10] replaces random perturbations with a feedback synthesizer that generates "error states" for more precise policy adjustment. Following the work of ChauffeurNet [7] and PlanTF [6], we apply random perturbations and dropout to AV to address this issue.

The second challenge is that imitation-based models struggle to fully capture traffic norms implicit in expert trajectories, such as collision avoidance. To improve safety, previous works [7], [10], [12], [13], [14] introduced auxiliary losses (e.g., collision loss and on-road loss) to guide the models in learning these norms. For instance, SafetyNet [9] penalizes large jerk and curvature values to reduce abrupt movements and improve driving comfort. PLUTO [13] generates a rasterized map of the drivable area, projects trajectory points onto the map, and calculates losses via bilinear interpolation to ensure that the trajectory adheres to drivable area constraints. Going a step further, we design vectorized auxiliary losses rather than rasterized ones, calculating losses directly in continuous space. Additionally, we consider the bounding boxes of traffic participants, rather than treating them as points. This enables more precise auxiliary loss calculations and significantly enhances the safety of the planned trajectories.

Joint Prediction and Planninng: Traditional sequential prediction and planning methods generally follow one of two approaches: human-leader or AV-leader [15]. In the human-leader approach [1], [16], the future trajectories of surrounding agents are predicted first, and the AV's trajectory is then planned accordingly. However, this often leads to overly conservative behavior, and can even result in the "freezing robot" problem. In contrast, the AV-leader approach [17], [18] first plans multiple potential trajectories for the AV, then predicts the reactions of surrounding agents, selecting the best planning trajectory. However, this approach is based on the incorrect assumption that agents will always react to the AV's planning results, leading to overconfidence and potentially resulting in collisions.

To address these issues, researchers have proposed joint prediction and planning methods [3], [4], [19], [20], which replace the conditional modeling in sequential methods with joint modeling that predicts compatible futures for both the AV and surrounding agents. Gameformer [3] uses a hierarchical game-theoretic framework (level-k game theory) to iteratively model interactions between all agents, including the AV. In each iteration, the planned trajectory of the AV and the predicted trajectories of surrounding agents are generated simultaneously based on the results from the previous iteration. GenAD [4] learns a uniform prior for the future trajectories of the AV and surrounding agents and uses variational autoencoders (VAE) to generate the joint future of both. However, most of these methods overlook the significant differences between prediction and planning tasks, treating the AV and surrounding agents identically, which reduces the quality of the planning results. For example, Gameformer [3] applies collision losses to both the AV and surrounding agents, resulting in suboptimal performance. In contrast, our proposed RAP treats the AV and surrounding agents differently according to the task-specific differences between prediction and planning and achieves state-of-the-art performance on both open-loop and closed-loop planning in the nuPlan benchmark.

III. Method

In our proposed role-aware joint prediction and planning framework (RAP), we aim to predict the joint multimodal future trajectories of both the AV and the surrounding agents, given the AV's navigation route:

$$\{L_{i,k}\}_{i\in[0:N],k\in[1:K]} = f(A_0, A_{1:N}, \mathcal{M}, R|\phi), \tag{1}$$

where A_0 denotes the AV, $A_{1:N}$ represents the N surrounding dynamic agents, $\mathcal{M}$ denotes a high-definition map containing road elements and traffic light information, R is the navigation route of the AV, indicating its goal, f represents the whole neural network of our RAP, with ϕ as its model parameters, $L_{i,k} = \{l_{1,i,k}, l_{2,i,k}, \ldots, l_{F,i,k}\}$ denotes the predicted trajectory in the future time horizon F for agent A_i (including $i = 0$ for the AV)

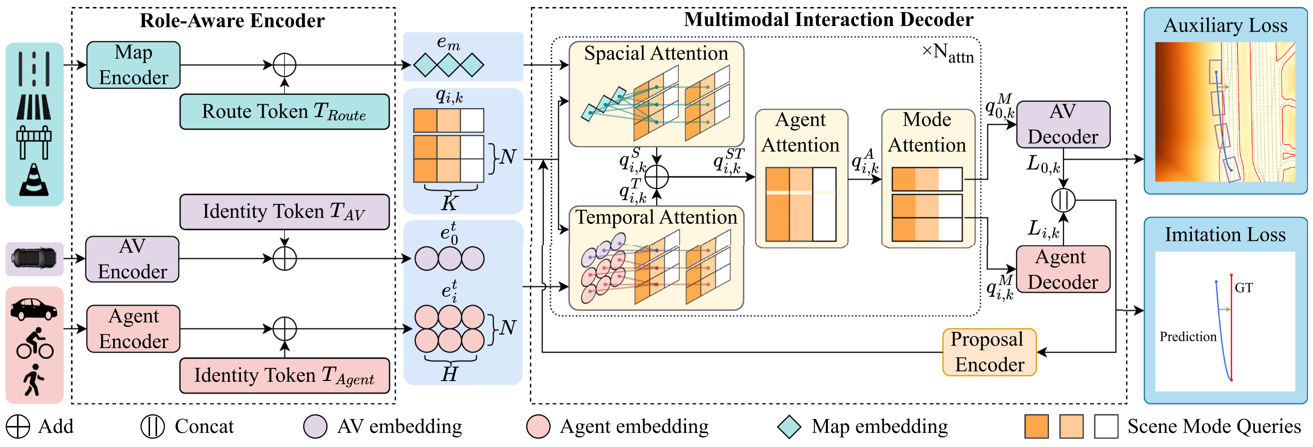


Fig. 1. An overview of RAP.

in the k-th possible joint future. RAP predicts K joint futures to capture interaction-induced multi-modality.

RAP follows a joint prediction architecture similar to QCNeXt [21] and HPNet [22], using a graph neural network with relative spatio-temporal encodings. Elements within the scene (e.g., the AV, agents, and lanes) are represented as nodes, which encode location-independent features, while the edges capture the relative spatio-temporal positions between elements. The specific structure of RAP is illustrated in Fig. 1, which comprises a Role-Aware Encoder and a Multimodal Interaction Decoder. During the encoding phase, edges and nodes are processed with separate encoders to capture their unique characteristics. In particular, each element type (the AV, agents, and lanes along the AV's intended route) is assigned distinct tokens to clarify their roles. In the decoding phase, elements interact through multiple attention mechanisms: Spatio-Temporal Attention, Agent Attention, and Mode Attention, to obtain compatible joint futures. Finally, imitation loss is applied to all predicted trajectories, while vectorized auxiliary losses are additionally imposed on the AV to enhance safety.

A. *Role-Aware Encoder*

Given that only planning involves a navigation goal, it is crucial for the AV to capture its own target without interfering with the prediction of surrounding agents. To address this issue, we propose a route–identity token pairing mechanism. We assign identity tokens T_{AV} and T_{Agent} to distinguish the AV and other agents, and route tokens T_{Route} to the lane segments on R. All tokens are learnable and category-shared. During attention computation, the AV's identity token implicitly aligns with the route tokens, enabling the AV to accurately capture its own intention, while the unpairing of agent identity token and route token enables agents to focus on prediction and avoid false dependence on route. Unlike previous methods, which either introduce additional modules to process navigation routes [23], [24] or rely solely on route tokens [13], [25] to highlight the route, our token pairing mechanism eliminates the need for additional modules while ensuring that surrounding agents do not place undue emphasis on the AV's route. Additionally, when navigation routes for other agents are available, our approach can be further extended by assigning distinct route and identity tokens to different agents, allowing each agent to effectively capture its own navigation goal.

Encoding Autonomous Vehicle (AV): The AV's observational features at time step t are $\{\boldsymbol{p}_0^t, \theta_0^t, \boldsymbol{v}_0^t, \alpha_0^t, c_0^t\}$, representing position, orientation, velocity, angular velocity, and semantic attributes (such as category and bounding box dimensions), respectively. Following ChauffeurNet [7], we randomly perturb the AV's historical trajectory by adding noise to $\{\boldsymbol{p}_0^t, \theta_0^t, \boldsymbol{v}_0^t, \alpha_0^t\}$ at a randomly selected time step, and then smooth the entire trajectory. This perturbation improves closed-loop robustness by helping the model learn how to recover from errors. To obtain translation and rotation invariance, the global velocity $\boldsymbol{v}_0^t$ and angular velocity α_0^t are transformed into a local coordinate frame defined by $(\boldsymbol{p}_0^t, \theta_0^t)$, yielding position-independent motion states $\{\bar{\boldsymbol{v}}_0^t, \bar{\alpha}_0^t\}$. These invariant features are encoded by a State Dropout Encoder (SDE) [6], which randomly drops subsets of kinematic states during training to prevent shortcut dependencies. An additional identity token T_{AV} is introduced to indicate the ego role, resulting in the final embedding:

$$e_0^t = \text{SDE}(\bar{\boldsymbol{v}}_0^t, \bar{\alpha}_0^t, c_0^t) + T_{\text{AV}}, \tag{2}$$

where $e_0^t \in \mathbb{R}^D$ and D is the embedding dimension.

Encoding Surrounding Agents: For each agent A_i at time step t, similar to the AV, its global velocity $\boldsymbol{v}_i^t$ and angular velocity α_i^t are transformed into a local coordinate system, resulting in the local motion states $\{\bar{\boldsymbol{v}}_i^t, \bar{\alpha}_i^t\}$. An MLP is then employed to obtain the agent embeddings, along with an Identity Token T_{Agent} to obtain the final embedding:

$$e_i^t = \text{MLP}(\bar{\boldsymbol{v}}_i^t, \bar{\alpha}_i^t, c_i^t) + T_{\text{Agent}}, \tag{3}$$

where $e_i^t \in \mathbb{R}^D$, c_i^t denotes the semantic attributes of A_i.

Encoding Vectorized Map: The map provides environmental information such as road structures, lane centerlines, crosswalks, and boundaries of drivable areas. Each map element is represented by a polyline p_m representing its topological structure and corresponding semantic attributes c_m, which include speed limits, type, and traffic light status. The map encoder $\mathrm{E}_{\mathrm{map}}$ in [22] is employed to encode these features and obtain the map embeddings. Then, a learnable Route Token T_{Route} is added to the lane segments on the AV's navigation route to signify the AV's intention, resulting in the final D-dimensional map embedding:

$$e_m = \mathrm{E}_{\mathrm{map}}(p_m, c_m) + T_{\mathrm{Route}}, \tag{4}$$

Encoding Relative Spatio-Temporal Position: The relative spatio-temporal positions between nodes serve as features for the edges. The edge features are represented as $\{d_r, \gamma_r, \psi_r, \delta_r\}$, which denotes the distance from the source node to the target node, the orientation of the edge relative to the target node's reference frame, the relative orientation between the source and target nodes, and the temporal difference between them, respectively. These edge features are then encoded by an MLP into edge embeddings:

$$e_r = \mathrm{MLP}(d_r, \gamma_r, \psi_r, \delta_r). \tag{5}$$

B. Multimodal Interaction Decoder

The decoder employs learnable scene mode queries to aggregate spatio-temporal context information, followed by Agent Attention and Mode Attention to model the interaction among agents and modes, respectively, and finally outputs multimodal role-aware joint futures.

Learnable Scene Mode Queries: Given the inherently multimodal nature of the future, we employ K learnable scene mode queries to decode diverse joint futures. Each query is a learnable embedding, assigned per agent at the current time step, including the AV. This results in mode queries of shape $(N+1) \times K$, denoted as $\{q_{i,k}\}_{i\in[0,N],\, k\in[1,K]}$. Each mode query is a node in the graph, sharing the same spatio-temporal location as its corresponding agent and representing a potential future for that agent.

Interaction Layer: Each interaction layer comprises a Spatio-Temporal Attention module, an Agent Attention module, and a Mode Attention module. The Spatio-Temporal Attention module includes two parallel cross-attention mechanisms, which aggregate the historical features of the agents within the historical horizon H and interact with map elements (including the route) within spatial radius R_1 to capture the terrain structure and understand the AV's intentions as below:

$$q^S_{i,k} = \mathrm{MHA}(q_{i,k}, [e_m, e_r], [e_m, e_r]), \tag{6}$$

$$q^T_{i,k} = \mathrm{MHA}(q_{i,k}, [e_i^{-H\sim 0}, e_r], [e_i^{-H\sim 0}, e_r]), \tag{7}$$

where $\mathrm{MHA}(a, b, c)$ denotes the multi-head attention with a as query, b as key, and c as value.The results of the two cross-attention modules are then summed:

$$q^{ST}_{i,k} = q^S_{i,k} + q^T_{i,k}. \tag{8}$$

Subsequently, the queries undergo self-attention across the agent dimension:

$$q^A_{i,k} = \mathrm{MHA}(q^{ST}_{i,k}, [q^{ST}_{i',k}, e_r], [q^{ST}_{i',k}, e_r]), \tag{9}$$

where i' denotes all agents within a certain radius R_2 of agent A_i under the same mode. Agent Attention facilitates interactions between the future trajectories of different agents, allowing them to mutually attend to each other. This bidirectional attention ensures that agents negotiate their paths in a balanced manner rather than exerting a unidirectional influence, leading to coordinated future trajectories.

Finally, the queries perform Mode Attention, which models the interaction across different future trajectories, promoting future diversity:

$$q^M_{i,k} = \mathrm{MHA}(q^A_{i,k}, [q^A_{i,1\sim K}, e_r], [q^A_{i,1\sim K}, e_r]). \tag{10}$$

To facilitate comprehensive interactions, we employ N_{attn} interaction layers in decoder.

Multimodal Output: After sufficient interaction, two separate MLPs are used to decode the trajectories $L_{0,k}$ of the AV and the trajectories $L_{i,k}$ of surrounding agents:

$$L_{0,k} = \mathrm{MLP}(q^M_{0,k}), k \in [1, K], \tag{11}$$

$$L_{i,k} = \mathrm{MLP}(q^M_{i,k}), i \in [1, N], k \in [1, K], \tag{12}$$

where $L_{i,k} \in \mathbb{R}^{F\times 4}$ includes the future position and the sine and cosine values of the angle.

To further enhance prediction accuracy, following [22], [26], $\{L_{i,k}\}_{i\in[0,N],k\in[1,K]}$ are encoded as proposal-based queries. These queries replace the learnable mode queries and are re-injected into the decoder to produce the trajectory refinements $\{\Delta L_{i,k}\}_{i\in[0,N],k\in[1,K]}$. The final predicted trajectories are then obtained by summing the proposals and the refinements $\{L_{i,k} + \Delta L_{i,k}\}_{i\in[0,N],k\in[1,K]}$.

Additionally, max-pooling is applied to $q^M_{i,k}$ across the agent dimension, aggregating information from all agents within the same mode, inducing the final probability score $\{\pi_k\}_{k\in[1,K]}$ for each joint future. The AV trajectory in the joint future with the highest score will be used for planning.

C. Training Objective

Although both planning and prediction generate trajectories, their objectives differ: for the AV, the planned trajectory must align with its navigation route while ensuring safety, whereas for agents, accurate trajectory predictions are crucial for the AV to make informed decisions and avoid collisions—errors in prediction can lead to erroneous planning outcomes. To distinguish these objectives, we employ two types of losses: imitation loss and our newly designed vectorized auxiliary losses. The imitation loss guides both the AV and agents to learn expert driving behaviors, ensuring realistic motion patterns. Meanwhile, the vectorized auxiliary losses—including on-road loss, obstacle collision loss, and agent collision loss—are specifically designed to keep the AV's planned trajectory within the safe area and minimize potential collisions, enhancing overall safety. Notably, we impose auxiliary losses only to the AV. Experiments demonstrate that this differentiated treatment leads to better planning outcomes compared to applying auxiliary losses to all agents, as in GameFormer [3] or ChauffeurNet [7].

Imitation Loss: Imitation loss aims to minimize the difference between the predicted joint future and the ground truth, where the joint future refers to the predicted trajectories of the AV and surrounding agents under the same mode. The

$\hat{k}$-th mode to be optimized is determined based on the minimum total waypoint displacements between the predicted future $\{L_{i,k}\}_{i\in[0,N],k\in[1,K]}$ and the ground truth $\{G_i\}_{i\in[0,N]} = \{g_{1,i}, g_{2,i}, \ldots, g_{F,i}\}_{i\in[0,N]}$:

$$\hat{k} = \arg \min_{k\in[1,K]} \sum_{i=0}^{N}\sum_{t=1}^{F} \left\| l_{t,i,k} - g_{t,i} \right\|_1. \tag{13}$$

The huber loss and cross-entropy loss are employed to calculate the regression and classification losses, respectively:

$$\mathcal{L}_{reg1} = \frac{1}{N+1}\sum_{i=0}^{N} \mathcal{L}_{Huber}(L_{i,\hat{k}}, G_i), \tag{14}$$

$$\mathcal{L}_{reg2} = \frac{1}{N+1}\sum_{i=0}^{N} \mathcal{L}_{Huber}(L_{i,\hat{k}} + \Delta L_{i,\hat{k}}, G_i), \tag{15}$$

$$\mathcal{L}_{cls} = \mathcal{L}_{CE}(\pi_k, \hat{k}). \tag{16}$$

The total imitation loss is expressed as follows:

$$\mathcal{L}_{imitation} = \mathcal{L}_{reg1} + \mathcal{L}_{reg2} + \mathcal{L}_{cls}. \tag{17}$$

Vectorized Auxiliary Losses: As previous research [7], [10], [12], [13] has demonstrated, imitation loss is insufficient to capture the traffic rules (e.g. collision avoidance) implicit in expert trajectories. So auxiliary losses are necessary for the safety of planning. In this work, we designed vectorized on road loss and obstacle collision loss, as well as agent collision loss considering the shape of agents.

Previous works [7], [13] rely on rasterized maps to compute the on-road loss, which suffer from precision loss at boundaries and require significant memory. To address these issues, we designed a vectorized on-road loss calculation method that operates directly in continuous position space, eliminating the need for rasterized maps. Specifically, we represent the drivable area boundary as a polygon formed by a series of continuous vector segments. Instead of treating the AV as a point, we consider its shape by accounting for its bounding box. For each trajectory point $l_{t,0,\hat{k}}$, we calculate the four corner points of the AV's bounding box and determine the minimum distance from each corner to the segments forming the drivable area boundary. The minimum signed distance $\{D_{road}^{t,c}\}_{c\in[1,4]}$ are then computed for four corners at future moment t. The positive distances indicate the corner point is outside the drivable area, and negative distances indicate it is inside. Finally, a hinge loss is applied to penalize the signed distance, ensuring that the AV remains within the drivable area:

$$\mathcal{L}_{road} = \frac{1}{4F}\sum_{t=1}^{F}\sum_{c=1}^{4} \max(0, D_{road}^{t,c} + \epsilon_{road}), \tag{18}$$

where ϵ_{road} is a safety threshold.

Similarly, we also designed an obstacle collision loss to avoid the planned trajectory colliding with static obstacles:

$$\mathcal{L}_{obstacle} = \frac{1}{4F}\sum_{t=1}^{F}\sum_{c=1}^{4} \max(0, D_{obstacle}^{t,c} + \epsilon_{obstacle}). \tag{19}$$

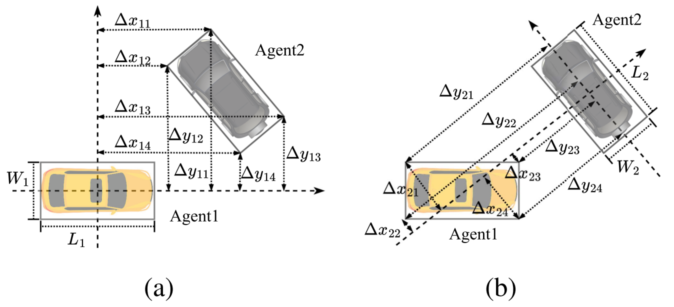


Fig. 2. Illustration of the calculation of agent collision loss.

Moreover, we design the agent collision loss $\mathcal{L}_{agent}$ based on the Separating Axis Theorem (SAT) [27] to prevent collisions between the AV and dynamic agents. Specifically, we approximate road participants—including pedestrians, cyclists, and vehicles—as rectangles. According to the SAT, for any two rectangles, if there is a separation axis, then there is a separation axis aligned with one of their four normal directions. Therefore, we calculate the overlap distance between the two agents along these axes, and then calculate agent collision loss by minimizing the overlap distance, as shown in Fig. 2 :

$$D_{t,2\to1,L} = \max\left\{\max\left(\frac{L_1}{2} + \epsilon_L - |\Delta x_{1j}|, 0\right), \forall j\right\},$$

$$D_{t,2\to1,W} = \max\left\{\max\left(\frac{W_1}{2} + \epsilon_W - |\Delta y_{1j}|, 0\right), \forall j\right\},$$

$$D_{t,1\to2,L} = \max\left\{\max\left(\frac{L_2}{2} + \epsilon_L - |\Delta x_{2j}|, 0\right), \forall j\right\},$$

$$D_{t,1\to2,W} = \max\left\{\max\left(\frac{W_2}{2} + \epsilon_W - |\Delta y_{2j}|, 0\right), \forall j\right\},$$

$$\mathcal{L}_{agent} = \frac{1}{NF}\sum_{t=1}^{F}\sum_{i=1}^{N} \min(D_{t,i\to0,L}, D_{t,i\to0,W}, D_{t,0\to i,L}, D_{t,0\to i,W}). \tag{20}$$

where ϵ_L and ϵ_W represent safety margins along the respective directions, and j indexes the four corners of the agent's bounding box.

The total auxiliary loss is finally expressed as follows:

$$\mathcal{L}_{auxiliary} = \mathcal{L}_{road} + \mathcal{L}_{obstacle} + \mathcal{L}_{agent}. \tag{21}$$

Total Loss: The total loss of our RAP consists of both imitation loss and auxiliary loss, expressed as:

$$\mathcal{L}_{total} = \mathcal{L}_{imitation} + \lambda_1 \cdot \mathcal{L}_{auxiliary}, \tag{22}$$

where λ_1 is a weighting factor that balances the contribution of the auxiliary loss in the overall optimization.

IV. Experiments

A. Experimental Setup

Implementation Details: Our model is trained on 8 RTX 4090 GPUs using the AdamW optimizer, with batch size, initial

TABLE I
COMPARISON OF RAP WITH THE STATE-OF-THE-ART METHODS. FOR EACH METRIC, THE BEST RESULT IS IN **BOLD** AND THE SECOND BEST RESULT IS UNDERLINED. ∗ REPRESENTS RULE-BASED POST-PROCESSING

Planners		Test14-random				Test14-hard			
Type	Method	OS↑	NRS↑	RS↑	**FS↑**	OS↑	NRS↑	RS↑	**FS↑**
Expert	Log-Replay	100.0	94.03	75.86	89.96	100.0	85.96	68.80	84.92
Rule-based & Hybrid	IDM [28]	34.15	70.39	72.42	58.99	20.07	56.16	62.26	46.16
	PDM-Closed* [29]	46.32	90.05	91.64	76.01	26.43	65.07	75.18	55.57
	PDM-Hybrid* [29]	**82.21**	90.20	91.56	**87.99**	**73.81**	65.95	75.79	**71.85**
	PEP* [2]	26.24	91.45	89.74	69.14	14.63	77.38	77.35	56.45
	PLUTO* [13]	-	92.23	90.29	-	-	**80.08**	76.88	-
	Diffusion Planner* [24]	-	**94.80**	91.75	-	-	78.87	**82.00**	-
	RAP* (Ours)	45.81	94.46	**92.98**	77.75	31.16	79.03	81.62	63.94
Learning-based	UrbanDriver [8]	82.44	63.27	61.02	68.91	76.90	51.54	49.07	59.17
	PDM-Open [29]	84.14	52.80	57.23	64.72	79.06	33.51	35.83	49.47
	Gameformer [3]	79.35	80.80	79.31	79.82	75.27	66.59	68.83	70.16
	PlanTF [6]	87.07	86.48	80.59	84.71	83.32	72.68	61.70	72.57
	PLUTO [13]	87.13	**90.42**	78.02	85.19	79.00	72.94	59.20	70.38
	PEP [2]	86.76	88.83	85.31	86.97	83.22	74.96	70.87	76.35
	BeTop [30]	87.6	90.2	85.7	87.8	84.0	77.1	68.8	76.6
	Diffusion Planner [24]	-	89.19	82.93	-	-	75.99	69.22	-
	RAP (Ours)	**88.86**	90.10	**87.82**	**88.92**	**86.43**	**77.60**	**75.23**	**79.75**

learning rate, weight decay, and dropout rate set to 32, 3×10^{-4}, 1×10^{-4}, and 0.1, respectively. The radii R_1 and R_2, historical horizon H, number of interaction layers N_{attn}, and dimension of latent variables D are set to 50, 50, 20, 3, and 128, respectively. The final model size is 6.4 M.

Dataset: The experiments are conducted on the nuPlan benchmark [5]. nuPlan is a large-scale, closed-loop planning benchmark for autonomous vehicles, comprising approximately 1,300 hours of real-world driving data across 75 labeled scenario types. The benchmark includes a closed-loop simulator, where each simulation rollout spans 15 seconds at a frequency of 10 Hz. In the simulation, the autonomous vehicle is modeled using a bicycle kinematic model with an LQR controller for trajectory tracking. Following PlanTF [6], 1 M samples are sampled across all scenario types for training, the Test14-random and Test14-hard splits are used for evaluation.

Metrics: We employ open-loop score (OS), non-reactive closed-loop score (NRS), reactive closed-loop score (RS) and final scores (FS) as evaluation metrics. OS measures distance between the predicted trajectory and GT, while NRS and RS assess closed-loop performance based on safety, rule adherence, and goal achievement. The difference is that RS controls surrounding agents using an Intelligent Driver Model (IDM) [28], while NRS simply replays their logged trajectories. FS, the average score of OS, NRS and RS, is used to evaluate the comprehensive ability of open-loop and closed-loop planning.

B. Comparison With State-of-The-Art

The comparison results with learning-based methods on the Test14 benchmark are shown in Table I. As shown, our RAP framework outperforms existing methods on nearly all metrics, especially on the more challenging Test14-hard, where it achieves the best performance across all metrics. Notably, RAP improves RS by a large margin (6.15%) higher than PEP [2], the best sequential prediction and planning method, and 9.30% higher than Gameformer [3], a representative joint prediction and planning approach. These results highlight both the effectiveness of joint prediction and planning methods, which explicitly capture the bidirectional interactions between the AV and agents, and the crucial role of differentiated treatment in improving performance based on task characteristics.

To ensure a comprehensive comparison, we also applied rule-based post-processing used by Diffusion Planner [24]. As shown in Table I, the post-processed RAP improves closed-loop metrics: on Test14-random, NRS and RS reach 94.46 and 92.98, respectively—slightly surpassing Diffusion Planner∗ and outperforming all other methods. This demonstrates that our proposed framework is compatible with rule-based refinements and can achieve SOTA closed-loop planning. However, such improvements come at a cost. First, OS degrades significantly, as the post-processing overrides the model's original trajectory. Second, the added module introduces non-negligible computational overhead (225.0 ms with vs. 40.2 ms without post-processing per planning step on a single RTX 4090), which may pose challenges for real-time deployment.

C. Ablation Study

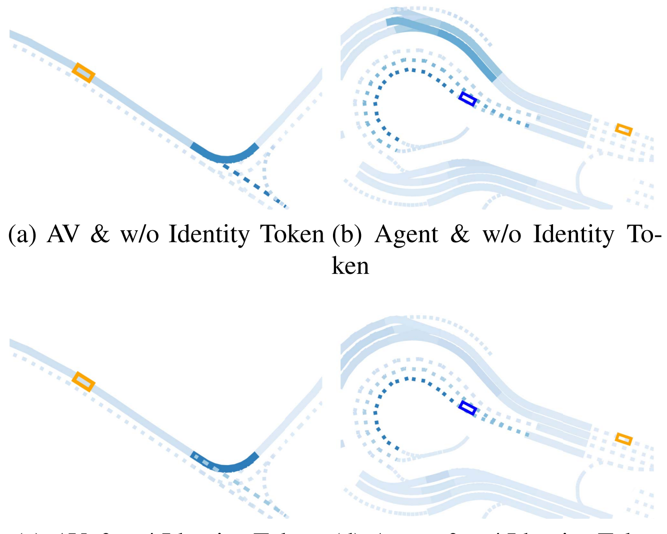
The Importance of Differentiated Treatment of AV and Surrounding Agents: To assess the contribution of each asymmetric component, we conduct ablation experiments on Test14-hard, as shown in Table II. Three components are evaluated: route–identity token pairing addressing information asymmetry, ego-only auxiliary losses addressing objective asymmetry and dropout and perturbation addressing feedback asymmetry. Removing the route–identity token pairing leads to a notable drop in OS (–2.42), confirming its key role in enabling the AV to focus on its navigation route without distracting surrounding agents. Excluding the ego-only auxiliary losses results in higher collision rates—particularly when surrounding agents behave differently from the training distribution (RS –1.98)—demonstrating their

TABLE II
ABLATION STUDY ON DIFFERENTIATED TREATMENT. RIT: ROUTE-IDENTITY TOKEN PAIRING; EAL: EGO-ONLY AUXILIARY LOSSES; DNP: DROPOUT AND PERTURBATION

RIT	EAL	DnP	OS↑	NRS↑	RS↑	FS↑
			87.22	60.74	55.53	67.83
	✓	✓	84.01	75.63	72.22	77.29
✓		✓	86.54	77.44	73.25	79.08
✓	✓		**87.88**	62.25	64.49	71.54
✓	✓	✓	86.43	**77.60**	**75.23**	**79.75**

TABLE III
ABLATION STUDY ON TOKEN PAIRING MECHANISM. "P" IS PROGRESS ALONG EXPERT ROUTE

Route Token	Identity Token	OS↑	NRS↑	P↑	RS↑	P↑	FS↑
		84.01	75.63	85.95	72.22	74.13	77.29
✓		85.82	75.78	86.18	73.72	77.63	78.44
✓	✓	**86.43**	**77.60**	**87.94**	**75.23**	**78.79**	**79.75**


(a) AV & w/o Identity Token (b) Agent & w/o Identity Token

(c) AV & w/ Identity Token (d) Agent & w/ Identity Token

Fig. 3. Attention weight maps of the AV (orange) and the agent (blue) w/o and w/ Identity Token. The bold solid line is the AV's navigation route, dashed lines denote other lanes, with darker shades of blue indicating higher attention weight. Without identity token, both the AV and the agent may assign high attention weights to irrelevant lanes.

effectiveness in enhancing AV safety. Removing dropout and perturbation increases OS (+1.45) but severely degrades closed-loop performance (NRS −15.35), as the AV loses its ability to recover from accumulated feedback errors. Combining all three components yields the best overall performance (FS = 79.75), highlighting the importance of differentiating the AV and surrounding agents according to the characteristics of the planning task.

The Impact of Route-Identity Token Pairing Mechanism: As shown in Table III, the introduction of route tokens improved all improved all scores, and the progress (P) indicated that they help the AV capture its navigation route to some extent. However, route tokens alone are insufficient. As illustrated in Fig. 3, when

TABLE IV
ABLATION STUDY OF TOKEN PAIRING ON HPNET [22]

Setting	OS↑	NRS↑	RS↑	FS↑
HPNet	71.01	65.18	68.67	68.29
HPNet + Token Pairing	**74.90**	**66.06**	**71.66**	**70.87**

TABLE V
ABLATION STUDY ON AUXILIARY LOSSES. "C" AND "TTC" ARE EGO-AT-FAULT COLLISION AND TIME-TO-COLLISION SCORES

AV	Agents	OS↑	NRS↑	C↑	TTC↑	RS↑	C↑	TTC↑	FS↑
		86.54	77.44	90.63	79.78	73.25	90.26	80.88	79.08
✓		86.43	**77.60**	**90.81**	**81.99**	75.23	91.73	83.09	**79.75**
✓	✓	86.04	72.67	85.11	78.31	**75.50**	**92.28**	**83.46**	78.07

only route tokens are used, the AV may still focus on irrelevant lanes, and unrelated agents may pay excessive attention to the AV's route. The introduction of identity tokens, which distinguish the roles of the AV and agents, effectively resolves these issues. Identity tokens enable the AV to accurately attend to its navigation route while allowing agents to focus solely on their own trajectory prediction, undisturbed by the route. The results in Table III confirm that identity tokens further improve the progress and all scores, demonstrating the effectiveness of our token pairing mechanism.

To further assess the generalization of the proposed token pairing mechanism, we implemented it in HPNet [22], a representative joint prediction method. As shown in Table IV, introducing token pairing consistently improves all evaluation metrics, particularly OS and RS. These results suggest that the mechanism can be easily transferred to other joint trajectory prediction methods, helping to bridge the gap between pure prediction and joint prediction and planning, and enabling the generation of more accurate planning outcomes.

Comparison of Auxiliary Losses Applied to AV Only vs. All Agents: As shown in Table V, applying auxiliary losses to the AV slightly reduces OS but notably improves RS and collision score (C and TTC), indicating safer and more robust planning. These losses encourage the AV to adopt more defensive behaviors (e.g., yielding to avoid collisions), thereby enhancing closed-loop safety. When auxiliary losses are further applied to surrounding agents, OS drops even more, and NRS decreases 4.93 points. To investigate this effect, we evaluated the prediction accuracy of surrounding agents on the validation set. The results show that minADE increases from 0.565 to 0.567 and minFDE from 1.468 to 1.473, suggesting a slight degradation in prediction accuracy. Due to the tight coupling between prediction and planning, this degradation propagates to the planning process, lowering open-loop scores. More importantly, it increases the collision likelihood (C and TTC) in non-reactive simulations. One possible explanation is that auxiliary losses on agents make the model incorrectly assume cooperative behaviors of agents, leading the AV to plan overly aggressive maneuvers. Fig. 4 qualitatively illustrates this issue: the model predicts that the AV will turn right while the straight-going agent slows down to yield, but in the ground truth, the agent maintains its speed,

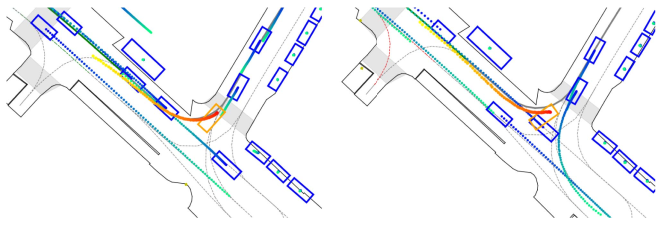


Fig. 4. An illustration showing that applying auxiliary losses to all agents can lead to a collision. The AV (orange box) incorrectly predicts that the straight-moving agent (blue box) will yield and proceeds to make a right turn (a). However, the agent does not yield, resulting in a collision (b).

resulting in a collision. A video demonstration of this example is provided in the supplementary material. Interestingly, C and TTC in reactive simulations still increases slightly. This is a reasonable result, since reactive evaluation assumes surrounding agents always yield to the AV.

In summary, auxiliary losses should be applied exclusively to the AV in joint prediction and planning. This design promotes safer behavior and yields better overall performance in both open-loop and closed-loop evaluations.

V. Conclusion

In this work, we proposed the Role-Aware Joint Prediction and Planning (RAP) framework, which differentiates the ego vehicle and surrounding agents to address the information, objective, and feedback asymmetries between prediction and planning. Experiments on the nuPlan benchmark show that RAP achieves SOTA performance, confirming the effectiveness of asymmetric, role-aware modeling for unified prediction and planning.

References

[1] Y. Hu et al., "Planning-oriented autonomous driving," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 17853–17862.

[2] D. Zhang et al., "PEP: Policy-embedded trajectory planning for autonomous driving," *IEEE Robot. Autom. Lett.*, vol. 9, no. 12, pp. 11361–11368, Dec. 2024.

[3] Z. Huang, H. Liu, and C. Lv, "GameFormer: Game-theoretic modeling and learning of transformer-based interactive prediction and planning for autonomous driving," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2023, pp. 3903–3913.

[4] W. Zheng, R. Song, X. Guo, C. Zhang, and L. Chen, "GenAD: Generative end-to-end autonomous driving," in *Proc. Eur. Conf. Comput. Vis.*, 2024, pp. 87–104.

[5] H. Caesar et al., "NuPlan: A closed-loop ML-based planning benchmark for autonomous vehicles," 2021, *arXiv:2106.11810*.

[6] J. Cheng, Y. Chen, X. Mei, B. Yang, B. Li, and M. Liu, "Rethinking imitation-based planners for autonomous driving," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2024, pp. 14123–14130.

[7] M. Bansal, A. Krizhevsky, and A. Ogale, "ChauffeurNet: Learning to drive by imitating the best and synthesizing the worst," 2018, *arXiv:1812.03079*.

[8] O. Scheel, L. Bergamini, M. Wolczyk, B. Osiński, and P. Ondruska, "Urban driver: Learning to drive from real-world demonstrations using policy gradients," in *Proc. Conf. Robot Learn.*, 2022, pp. 718–728.

[9] M. Vitelli et al., "SafetyNet: Safe planning for real-world self-driving vehicles using machine-learned policies," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2022, pp. 897–904.

[10] J. Zhou et al., "Exploring imitation learning for autonomous driving with feedback synthesizer and differentiable rasterization," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2021, pp. 1450–1457.

[11] S. Ross, G. Gordon, and D. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *Proc. 14th Int. Conf. Artif. Intell. Statist.*, 2011, pp. 627–635.

[12] S. Pini, C. S. Perone, A. Ahuja, A. S. R. Ferreira, M. Niendorf, and S. Zagoruyko, "Safe real-world autonomous driving by learning to predict and plan with a mixture of experts," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2023, pp. 10069–10075.

[13] J. Cheng, Y. Chen, and Q. Chen, "PLUTO: Pushing the limit of imitation learning-based planning for autonomous driving," 2024, *arXiv:2404.14327*.

[14] Y. Chen, S. Tonkens, and M. Pavone, "Categorical traffic transformer: Interpretable and diverse behavior prediction with tokenized latent," 2023, in *Proc. IEEE Int. Conf. Robot. Automat.*, 2023, pp. 2423–2430.

[15] S. Hagedorn, M. Hallgarten, M. Stoll, and A. P. Condurache, "The integration of prediction and planning in deep learning automated driving systems: A review," *IEEE Trans. Intell. Veh.*, vol. 10, no. 5, pp. 3626–3643, May 2025.

[16] B. Jiang et al., "VAD: Vectorized scene representation for efficient autonomous driving," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2023, pp. 8340–8350.

[17] E. Tolstaya, R. Mahjourian, C. Downey, B. Vadarajan, B. Sapp, and D. Anguelov, "Identifying driver interactions via conditional behavior prediction," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2021, pp. 3473–3479.

[18] H. Song, W. Ding, Y. Chen, S. Shen, M. Y. Wang, and Q. Chen, "PiP: Planning-informed trajectory prediction for autonomous driving," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 598–614.

[19] Z. Chen, M. Ye, S. Xu, T. Cao, and Q. Chen, "PPAD: Iterative interactions of prediction and planning for end-to-end autonomous driving," in *Proc. Eur. Conf. Comput. Vis.*, 2024, pp. 239–256.

[20] Y. Chen, S. Veer, P. Karkus, and M. Pavone, "Interactive joint planning for autonomous vehicles," *IEEE Robot. Autom. Lett.*, vol. 9, no. 2, pp. 987–994, Feb. 2024.

[21] Z. Zhou, Z. Wen, J. Wang, Y.-H. Li, and Y.-K. Huang, "QCNeXt: A next-generation framework for joint multi-agent trajectory prediction," 2023, *arXiv:2306.10508*.

[22] X. Tang, M. Kan, S. Shan, Z. Ji, J. Bai, and X. Chen, "HPNet: Dynamic trajectory forecasting with historical prediction attention," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2024, pp. 15261–15270.

[23] Z. Huang et al., "Gen-drive: Enhancing diffusion generative driving policies with reward modeling and reinforcement learning fine-tuning," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2024, pp. 3445–3451.

[24] Y. Zheng et al., "Diffusion-based planning for autonomous driving with flexible guidance," in *Proc. Int. Conf. Learn. Represent.*, 2025.

[25] M. Hallgarten, M. Stoll, and A. Zell, "From prediction to planning with goal conditioned lane graph traversals," in *Proc. IEEE Int. Conf. Intell. Transp. Syst.*, 2023, pp. 951–958.

[26] Z. Zhou, J. Wang, Y.-H. Li, and Y.-K. Huang, "Query-centric trajectory prediction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 17863–17873.

[27] S. Boyd and L. Vandenberghe, *Convex Optimization.*, Cambridge, U.K.: Cambridge Univ. Press, 2004.

[28] M. Treiber, A. Hennecke, and D. Helbing, "Congested traffic states in empirical observations and microscopic simulations," *Phys. Rev. E*, vol. 62, no. 2, 2000, Art. no. 1805.

[29] D. Dauner, M. Hallgarten, A. Geiger, and K. Chitta, "Parting with misconceptions about learning-based vehicle motion planning," in *Proc. Conf. Robot Learn.*, 2023, pp. 1268–1281.

[30] H. Liu, L. Chen, Y. Qiao, C. Lv, and H. Li, "Reasoning multi-agent behavioral topology for interactive autonomous driving," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 37, 2024.