AlignDrive: Aligned Lateral-Longitudinal Planning for End-to-End Autonomous Driving

Yanhao Wu[1,2,*] **Haoyang Zhang**[2,‡] **Fei He**[2] **Rui Wu**[2]
Yanhu Shan[2] **Congpei Qiu**[1] **Liang Gao**[1] **Wei Ke**[1] **Tong Zhang** †[3]
[1] School of Software Engineering, XJTU [2] Horizon Robotics
[3] University of Chinese Academy of Sciences

Abstract

Practical autonomous driving requires models that generalize by reasoning through spatial-temporal possibilities to exclude unsafe outcomes. While state-of-the-art (SOTA) methods use parallel planning architectures, they fail to explicitly couple speed decisions with agent behavior along the driving path, leading to suboptimal coordination. To address this, we propose a cascaded framework that transforms longitudinal planning from an independent prediction task into a path-conditioned reasoning process. On the model side, we introduce an anchor-based regression design that conditions longitudinal prediction on the lateral drive path, and reformulate longitudinal planning as 1D displacement prediction along the path. This reduces geometric uncertainty and sharpens the model's focus on interaction-driven dynamics. On the data side, we introduce a planning-oriented data augmentation strategy that simulates rare safety-critical events by programmatically inserting agents and relabeling longitudinal targets to enforce collision avoidance. Evaluated on the challenging Bench2Drive benchmark, our method achieves SOTA performance with a driving score of 89.07 and a success rate of 73.18%, demonstrating significantly improved coordination and safety. Further evaluation on Fail2Drive confirms strong generalization to rare edge cases where parallel formulations typically fail. Project page: https://yanhaowu.github.io/AlignDrive/ [1]

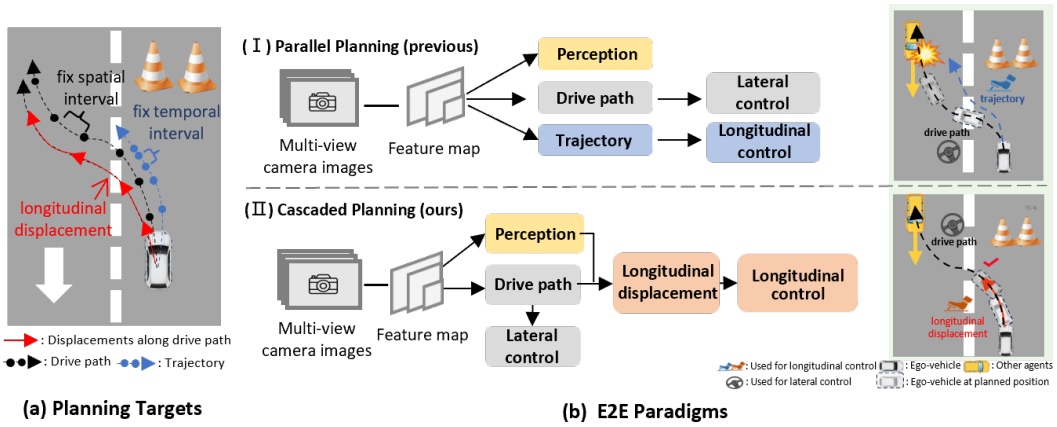
1 Introduction

End-to-end autonomous driving has rapidly advanced, enabling joint perception and planning in increasingly complex environments [28, 10, 15, 27, 12, 31]. A key requirement for practical deployment is generalization: the ability to reason through a wide range of spatial-temporal possibilities and systematically exclude unsafe outcomes, particularly in rare, safety-critical scenarios that are underrepresented in standard training data.

A promising direction in recent work is to decouple planning into separate lateral and longitudinal prediction tasks [13, 26, 29]. In this paradigm, lateral planning predicts a drive path—waypoints sampled at fixed spatial intervals—as the target for steering control, while longitudinal planning predicts a trajectory—waypoints sampled at fixed temporal intervals—as the target for speed control. This disentanglement has proven effective for improving lateral planning performance. However, reliable longitudinal planning remains a major challenge. In particular, SOTA methods such as HiP-AD [29] employ a parallel architecture in which drive path and trajectory queries are decoded by independent heads simultaneously. While this design is effective overall, the parallel formulation does not explicitly couple longitudinal decisions with surrounding agent behavior along the driving path. As a result, even when predicted drive paths are geometrically accurate, the corresponding

[1]*: Work done during internship at Horizon Robotics. †: Corresponding author. ‡: Project leader.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).



(a) Planning Targets **(b) E2E Paradigms**

Figure 1: (a) Drive path (black), trajectory (blue), and longitudinal displacement (red). Path waypoints are sampled spatially, trajectory waypoints temporally, and displacements represent traveled distance along the path at fixed time intervals. (b) Comparison of E2E paradigms. Parallel planning predicts the drive path and longitudinal trajectory independently, which can lead to potential coordination inconsistencies. In the example on the right, the independently predicted longitudinal trajectory is collision-free, but applying its speed along a separately predicted lateral path could cause a collision. In contrast, our cascaded paradigm first predicts the drive path and then regresses path-conditioned longitudinal displacements. With the path prior, the model identifies the potential conflict and outputs shorter displacements, yielding to avoid collision. Perception inputs are omitted for clarity.

speed profiles can become unsafe in complex interactive scenarios—for instance, failing to brake in response to a sudden cut-in or a decelerating lead vehicle (see Fig. 1(b)).

We argue that reliable longitudinal planning plays an important role in robust driving generalization, as interaction-heavy safety-critical scenarios critically depend on accurate speed reasoning conditioned on surrounding agent behavior. This motivates explicitly aligning longitudinal motion reasoning with the dynamics of surrounding agents along the intended driving path. To this end, we address the problem from two complementary perspectives: model design and data construction.

On the model side, we propose a cascaded framework that conditions longitudinal planning on the lateral drive path via an anchor-based regression design. Rather than predicting full 2D trajectory waypoints independently, our formulation first establishes the drive path as a structured geometric prior, then predicts longitudinal motion as 1D displacement values along that path (see Fig. 1(a)). This cascaded design offers two key benefits. First, anchoring longitudinal reasoning to the drive path reduces geometric uncertainty and provides a coarse but explicit alignment between longitudinal motion decisions and the spatial context of surrounding agents. Second, reformulating longitudinal planning as 1D displacement removes unnecessary geometric degrees of freedom, sharpening the model's focus on interaction-driven dynamics and resulting in more collision-aware longitudinal planning (see Fig. 1(b)).

On the data side, we introduce a planning-oriented data augmentation strategy to improve generalization. By programmatically inserting agents and relabeling 1D displacement targets to enforce collision avoidance, we simulate safety-critical events that are severely underrepresented in real-world logs. Whereas parallel 2D models fail to learn effectively from such augmentation because steering and speed are geometrically entangled, our 1D formulation **untangles** these variables. This structural clarity allows the model to learn causal collision-avoidance logic rather than merely memorizing expert patterns, enabling robust generalization to unseen interactive edge cases.

Building on these insights, we develop an E2E driving framework AlignDrive that conditions longitudinal planning on the drive path and leverages this formulation to enable effective data augmentation, with code and models to be publicly released. Overall, our contributions are threefold:

- We propose a novel cascaded planning paradigm where longitudinal planning is explicitly conditioned on a predicted lateral drive path. This method establishes a tight coupling between the two tasks, using the path as geometric priors for subsequent longitudinal planning.

- Based on this paradigm, we reformulate the longitudinal planning task as a simpler 1D displacement prediction problem along the drive path. This allows the model to focus its capacity on crucial dynamic interactions rather than redundantly encoding static geometry.

- We introduce an effective, planning-oriented data augmentation strategy. By programmatically modifying only the 1D displacement labels in response to inserted agents, we can generate diverse and realistic safety-critical training scenarios that are rare in logged data.

We primarily evaluate our method on the popular closed-loop benchmark Bench2Drive [14], where AlignDrive demonstrates clear improvements over state-of-the-art techniques. To further verify generalization, we additionally evaluate on the recently proposed Fail2Drive benchmark [9], as well as report open-loop results on the nuScenes dataset [2]. The results consistently demonstrate the strong performance, generalization ability, and robustness of our method. Together, these experiments provide a comprehensive validation of the model's capabilities.

2 Related work

2.1 End-to-end autonomous driving

End-to-end autonomous driving methods [30, 29, 7, 28, 12, 16, 34] have rapidly advanced in recent years, with trajectory planning playing a central role in predicting the ego vehicle's future states. One line of work, exemplified by SparseDrive [28, 27], directly predicts trajectories in an end-to-end manner. While effective in nominal scenarios, this joint prediction paradigm struggles to achieve accurate lateral and longitudinal planning simultaneously. TF++ [13] predicts the drive path and instantaneous vehicle speed in parallel, with speed treated as a classification task. However, the coarse discretization of velocity limits planning accuracy. More recent approaches, including HiP-AD [29] and Carllava [26], instead decouple path and trajectory prediction. HiP-AD employs independent heads, while Carllava, built on a LLaVA-like architecture [22], generates the drive path and trajectory sequentially as output tokens. However, these methods still rely on predicting full waypoints rather than explicit longitudinal displacements, limiting alignment between longitudinal planning and surrounding agents in challenging scenarios such as sharp turns or dynamic interactions. In contrast, we propose a cascaded, anchor-based formulation that first predicts the drive path and then forecasts a sequence of future longitudinal displacements along it. Our approach regresses offsets from predefined anchors using a two-stage design with dedicated modules for path and displacement prediction, rather than jointly generating them as tokens. This naturally enforces lateral–longitudinal consistency, simplifies reasoning about dynamic interactions, and improves path-following safety.

2.2 Data augmentation

Data augmentation is widely employed in multiple fields [36, 24, 33, 21]. In autonomous driving, it is commonly applied to augment image data through techniques such as cropping, flipping, and color jittering [28], which improve the model's ability to generalize across varying visual conditions and strengthen perception robustness. Pluto [3] employs agent drop and insertion as data augmentation strategies to generate both positive and negative scene samples. These augmented samples are utilized in a contrastive learning framework to enhance the model's scene representation capabilities. However, these augmentations primarily affect perception and influence planning only indirectly. TF++ [13] introduced an auxiliary camera in the simulation environment, which is randomly repositioned at each time step to increase data diversity. This approach relies on additional simulator equipment and focuses primarily on lateral recovery, providing limited guidance for longitudinal planning or dynamic interaction reasoning. In contrast, our planning-oriented augmentation is directly coupled with longitudinal planning, operating on the percepted agents and adjusting longitudinal displacements. This forces the model to focus explicitly on dynamic agent interactions, enabling path-consistent and collision-aware planning in rare, safety-critical scenarios.

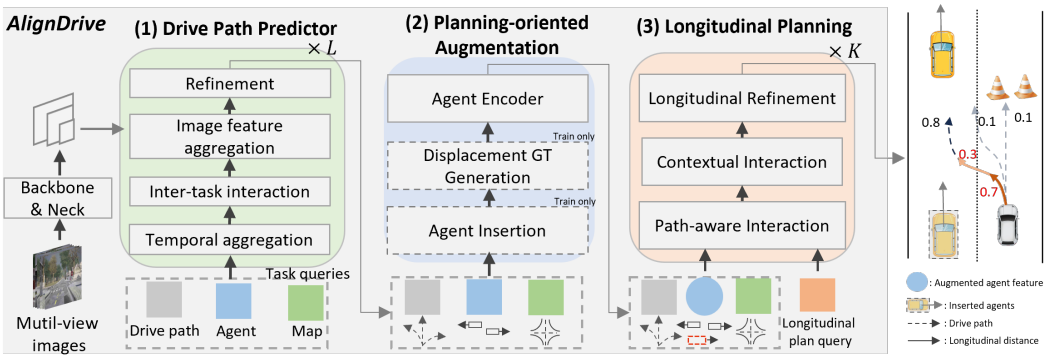


Figure 2: Overview of the proposed AlignDrive system, which consists of three components. The Drive Path Predictor refines queries through cross-attention with image features to encode the drive path, maps, and agents. The Planning-oriented Data Augmentation enriches scenarios by inserting additional agents and relabeling longitudinal displacements. Finally, the Longitudinal Planning Module predicts forward displacements along the drive path; combined with the path, these displacements yield the final trajectory that is both collision-aware and spatially consistent. On the right side of the figure, the black numbers denote the scores of predicted drive paths, while the red numbers represent the scores of the corresponding longitudinal planning for each drive path.

3 Method

3.1 Overview

Figure 2 provides an overview of AlignDrive, which consists of three main components. The **Drive Path Predictor** refines queries via cross-attention with image features [21], producing representations of the drive path, map, and dynamic agents. The **Planning-oriented Data Augmentation** module decodes agent queries into bounding boxes, re-encodes them as structured features, and enables insertion of synthetic agents with relabeled longitudinal displacements for consistent supervision. The **Longitudinal Planning** module then predicts displacements along the drive path from enriched queries, ensuring spatial consistency and collision awareness. This design preserves end-to-end training while supporting robust planning. We discuss components below.

3.2 Drive Path Predictor

Let us denote multi-scale features from V camera views as $\{\mathbf{f}_i\}_{i=1}^{V}$. Based on training data, we cluster ground-truth annotations to obtain anchors, which differ by task modality: bounding boxes for agents, and typical polylines for map elements and drive paths. We denote them as $\mathbf{A}_a \in \mathbb{R}^{N_a \times D_a}$, $\mathbf{A}_m \in \mathbb{R}^{N_m \times D_m}$, and $\mathbf{A}_d \in \mathbb{R}^{N_d \times D_d}$, where D_a, D_m, D_d are the dimensions of each type. Based on these anchors, we initialize three sets of task queries: agent queries $\mathbf{Q}_a \in \mathbb{R}^{N_a \times C}$, map queries $\mathbf{Q}_m \in \mathbb{R}^{N_m \times C}$, and drive path queries $\mathbf{Q}_d \in \mathbb{R}^{N_d \times C}$, where C is the feature dimension. The Drive Path Predictor consists of L stacked blocks, within which queries interact with image features, historical information, and each other. Through these interactions, the corresponding anchors are iteratively refined across blocks, yielding progressively updated estimates.

Temporal Aggregation. To incorporate historical information, each query interacts with retained queries from previous frames via a top-k strategy:

$$\mathbf{Q}_d \leftarrow \text{Cross-attention}\left(\mathbf{Q}=\mathbf{Q}_d, \mathbf{K}=\mathbf{Q}_d^{t-T_p:t-1}, \mathbf{V}=\mathbf{Q}_d^{t-T_p:t-1}\right), \tag{1}$$

where $\mathbf{Q}_d$ is the current drive path queries, and $\mathbf{Q}_d^{t-T_p:t-1}$ are historical ones. Map queries $\mathbf{Q}_m$ and agent queries $\mathbf{Q}_a$ are updated similarly.

Inter-Task Interaction. We enable interactions among drive path, agent, and map queries through cross-attention, allowing path queries to be contextually aware of agents and maps, and constraining agent behaviors with map information.

$$\begin{aligned}\mathbf{Q}_d &\leftarrow \text{Cross-attention}\left(\mathbf{Q}=\mathbf{Q}_d, \mathbf{K}=[\mathbf{Q}_a \| \mathbf{Q}_m], \mathbf{V}=[\mathbf{Q}_a \| \mathbf{Q}_m]\right), \\ \mathbf{Q}_a &\leftarrow \text{Cross-attention}\left(\mathbf{Q}=\mathbf{Q}_a, \mathbf{K}=\mathbf{Q}_m, \mathbf{V}=\mathbf{Q}_m\right),\end{aligned} \tag{2}$$

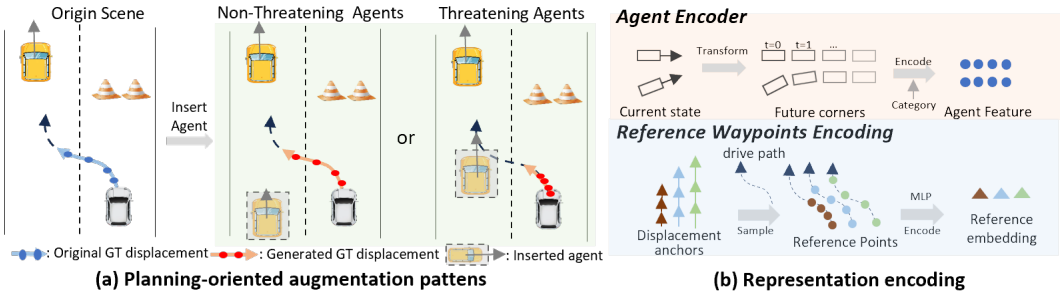


Figure 3: (a) Planning-oriented augmentation. Non-threatening agents are inserted at a distance with unchanged GT displacements, while threatening agents are placed nearby and cause adaptive shortening of GT displacements. (b) Representation encoding. Inserted agents are projected into future positions, transformed to corner representations, and encoded via a Fourier encoder (top). Reference points are sampled by displacement anchors and encoded with MLPs. For clarity, although multiple drive paths are predicted in practice, only one representative path is illustrated here (bottom).

where $[\cdot\|\cdot]$ denotes concatenation along the token dimension.

Image feature aggregation. To fuse image features, anchors are projected onto multi-view images, and their sampled features are aggregated via deformable attention. For drive path queries:

$$\mathbf{Q}_d \leftarrow \text{DA}\left(\mathbf{Q}=\mathbf{Q}_d, \mathbf{K}=\mathcal{P}(\mathbf{A}_d, \{\mathbf{F}_i\}_{i=1}^V), \mathbf{V}=\mathcal{P}(\mathbf{A}_d, \{\mathbf{F}_i\}_{i=1}^V)\right), \tag{3}$$

where DA is deformable attention and $\mathcal{P}(\cdot)$ denotes projection and sampling. Map queries $\mathbf{Q}_m$ and agent queries $\mathbf{Q}_a$ are enhanced in the same way using their anchors $\mathbf{A}_m, \mathbf{A}_a$.

Refinement. The model iteratively refines its predictions across the L blocks. In the refinement stage of each block, for a given anchor $\mathbf{A}_d$, we first generate a feature embedding using a task-specific encoder, $MLP_{enc}(A_d)$. This embedding is fused with the corresponding query $\mathbf{Q}_d$ and fed into an MLP to predict a corrective offset, $\Delta\mathbf{Y}_d$. The anchor is then updated by applying this offset. This process allows the model to progressively improve its estimate from a coarse anchor to a precise prediction.

$$\Delta\mathbf{Y}_d = \text{MLP}\left(\mathbf{Q}_d + \text{MLP}_{enc}(\mathbf{A}_d)\right), \quad \mathbf{A}_d \leftarrow \mathbf{A}_d + \Delta\mathbf{Y}_d, \tag{4}$$

where $\Delta\mathbf{Y}_d \in \mathbb{R}^{N_d \times D_d}$ is the predicted offset for each drive path anchor, and $D_d = P \times 2$ corresponds to P future waypoints. The refined waypoints are obtained as $\hat{\mathbf{Y}}_d = \mathbf{A}_d$ after iteratively updating the anchors through all L blocks. A separate MLP head is applied to the drive path query to predict confidence scores $\mathbf{S}_d \in \mathbb{R}^{N_d \times 1}$ for candidate drive paths. Map anchors are refined in the same iterative manner using $\mathbf{Q}_m$ and $\mathbf{A}_m$.

For agents, static attributes (e.g., position, size, orientation) are predicted using an MLP applied to the queries combined with anchor features, while dynamic motion is predicted directly from queries without anchor-based refinement:

$$\hat{\mathbf{Y}}_a^{state} = \text{MLP}_{state}\left(\mathbf{Q}_a + E(\mathbf{A}_a)\right), \quad \hat{\mathbf{Y}}_a^{motion} = \text{MLP}_{motion}\left(\mathbf{Q}_a\right), \tag{5}$$

where $\hat{\mathbf{Y}}_a^{state} \in \mathbb{R}^{N_a \times S}$ contains S attributes for each agent, and $\hat{\mathbf{Y}}_a^{motion} \in \mathbb{R}^{N_a \times T \times 2}$ contains the predicted future trajectories.

3.3 Planning-oriented Data Augmentation

To enrich interactive scenarios, we insert a virtual agent into the detected agents with probability α.

Agent Insertion. The virtual agent is initialized with a randomly sampled state $\mathbf{X}_{vir}$ and a target point $\mathbf{P}^*$, selected on the ego vehicle's ground-truth drive path $\mathbf{Y}_d^{\text{GT}}$. Together, they determine the virtual agent's future motion. As illustrated in Fig. 3(a), the virtual agent randomly adopts one of two velocity patterns: gradually approaching from afar, making it a low-risk and safe maneuver, or approaching faster and closer, potentially colliding with the ego vehicle. To maintain the total number of agents, the original agent with the lowest confidence is removed, and the virtual agent is inserted in its place. The resulting augmented set of agent states and motions are denoted as $\tilde{\mathbf{Y}}^{state}$ and $\tilde{\mathbf{Y}}^{motion}$, ensuring exposure to challenging interactions while preserving scene consistency.

Displacement Ground-truth Generation. The ground-truth displacement label is defined as a sequence $\mathbf{Y}_d^{\text{GT}} \in \mathbb{R}^{T\times 1}$, representing T future longitudinal displacements of the ego vehicle along the drive path, each measured over a fixed temporal interval. When a virtual agent is inserted that would collide with the ego vehicle, we first determine the maximum total displacement D_{safe} the ego vehicle can travel over the T steps without causing a collision. Given the original total displacement $D_{\text{orig}} = \sum_{t=1}^{T} \mathbf{Y}_d^{\text{GT}}[t]$, we compute a safe scaling factor $\beta = D_{\text{safe}}/D_{\text{orig}}$ and uniformly scale the original sequence: $\tilde{\mathbf{Y}}_d^{\text{GT}} = \beta \cdot \mathbf{Y}_d^{\text{GT}}$. This procedure ensures that the displacement at each step is consistently reduced to avoid collisions, while maintaining a dynamically plausible motion profile.

Agent Encode. After augmentation, agents are represented using both their states and motions, denoted as $\tilde{\mathbf{Y}}^{\text{state}}$ and $\tilde{\mathbf{Y}}^{\text{motion}}$. These variables are converted into a unified corner-based representation:

$$\mathbf{C}_a = f_{\text{corner}}(\tilde{\mathbf{Y}}^{\text{state}}, \tilde{\mathbf{Y}}^{\text{motion}}), \quad \mathbf{C}_a \in \mathbb{R}^{N_a\times(T+1)\times 8}, \tag{6}$$

where $f_{\text{corner}}(\cdot)$ converts each agent's bounding box state and future trajectory into the coordinates of its four corners across all time steps, as illustrated in the top of Fig. 3(b). To construct agent features, we separately process geometry and category information. The corner representation $\mathbf{C}_a$ is first mapped using Fourier encoder Φ and passed through $\text{MLP}_{\text{corner}}$ [23], while the agent category (extracted from $\tilde{\mathbf{Y}}^{\text{state}}$) is mapped with Φ and processed through MLP_{cat}. Their outputs are then summed to form the unified agent embedding:

$$\mathbf{E}_a^{0:T} = \text{MLP}_{\text{corner}}(\Phi(\mathbf{C}_a)) + \text{MLP}_{\text{cat}}(\Phi(\text{cat})), \quad \mathbf{E}_a^{0:T} \in \mathbb{R}^{N_a\times(T+1)\times C}. \tag{7}$$

Here, $\mathbf{E}_a^t \in \mathbb{R}^{N_a\times C}$ denotes the features of all agents at time step t, providing a temporally consistent, category-aware representation. The detailed process can be found in the appendix.

3.4 Longitudinal Planning Module

The longitudinal planning module predicts a sequence of ego vehicle's future displacements along the drive path at fixed time intervals. It is implemented as K stacked blocks, where queries interact with agent and path features and are progressively refined. We formulate this as an *anchor-based offset regression* task: for each candidate path, M anchors are defined, each representing a sequence of longitudinal displacements for the current step and T future steps, yielding $M \times (T+1)$ learnable queries. These queries are responsible for predicting offsets relative to their anchors, thereby coupling the drive path geometry with agent interactions and enabling precise longitudinal planning.

$$\mathbf{A}_l^{0:T} \in \mathbb{R}^{N_d\times M\times(T+1)\times 1}, \quad \mathbf{Q}_l^{0:T} \in \mathbb{R}^{N_d\times M\times(T+1)\times C}, \tag{8}$$

where the superscript $0:T$ denotes the stacked sequence of time steps, covering from the current step ($t=0$) to T future steps. Here, $\mathbf{A}_l^{0:T}$ are the anchor displacements and $\mathbf{Q}_l^{0:T}$ their learnable queries. The final displacements are obtained by adding predicted offsets to the anchors.

Path-aware Interaction. Each longitudinal planning query is enhanced with reference waypoints sampled along the predicted drive path $\hat{\mathbf{Y}}_d$ at anchor displacements $\mathbf{A}_l^{0:T}$. Let sampled points $\mathbf{P}_l$ be

$$\mathbf{P}_l = \text{Interp}(\hat{\mathbf{Y}}_d, \mathbf{A}_l^{0:T}), \quad \mathbf{P}_l \in \mathbb{R}^{N_d\times M\times(T+1)\times 2}, \tag{9}$$

where $\text{Interp}(\cdot)$ denotes linear interpolation along the path according to cumulative anchor displacements. The points for each anchor are encoded jointly into a feature vector, as shown in Fig. 3(b) (bottom):

$$\mathbf{F}_l = \text{MLP}(\mathbf{P}_l), \quad \mathbf{F}_l \in \mathbb{R}^{N_d\times M\times C}. \tag{10}$$

Then, this feature is broadcasted across the T+1 time steps and added to the longitudinal query for cross-attention with the drive path queries $\mathbf{Q}_d$:

$$\mathbf{Q}_l^{0:T} \leftarrow \text{CrossAttn}(\mathbf{Q} = \mathbf{Q}_l^{0:T} + \mathbf{F}_l, \mathbf{K} = \mathbf{Q}_d, \mathbf{V} = \mathbf{Q}_d). \tag{11}$$

In this way, each longitudinal planning query incorporates both the geometry of the drive path and its anchor-based temporal reference.

Contextual Interaction. At each time step t, longitudinal planning queries interact with dynamic agents via cross-attention on the encoded agent features to capture agent-specific context:

$$\mathbf{Q}_l^t \leftarrow \text{CrossAttn}\big(\mathbf{Q} = \mathbf{Q}_l^t,\ \mathbf{K} = \mathbf{E}_a^t,\ \mathbf{V} = \mathbf{E}_a^t\big), \quad t = 0, \ldots, T. \tag{12}$$

The updated queries attend to map queries via cross-attention, incorporating static cues such as stop lines for longitudinal planning:

$$\mathbf{Q}_l^{0:T} \leftarrow \text{CrossAttn}\big(\mathbf{Q} = \mathbf{Q}_l^{0:T},\ \mathbf{K} = \mathbf{Q}_m,\ \mathbf{V} = \mathbf{Q}_m\big). \tag{13}$$

Finally, a temporal positional encoding is added to the queries, and causal self-attention is applied along the temporal dimension to enforce consistency:

$$\mathbf{Q}_l^{0:T} \leftarrow \text{CausalSelfAttn}\big(\mathbf{Q}_l^{0:T} + \text{PE}^{0:T}\big). \tag{14}$$

Longitudinal Refinement. After obtaining the updated longitudinal queries $\mathbf{Q}_l^{0:T}$ from path-aware and contextual interactions, we enhance them using path-aligned reference points $\mathbf{P}_l$, providing spatial grounding for predicting offsets relative to the anchors. Specifically, the reference points $\mathbf{P}_l$ are encoded by an encoder MLP_{ref} and fused with the queries to predict offsets:

$$\Delta\mathbf{Y}_l = \text{MLP}\Big(\mathbf{Q}_l^{0:T} + \text{MLP}_{ref}(\mathbf{P}_l)\Big), \quad \Delta\mathbf{Y}_l \in \mathbb{R}^{N_d \times M \times (T+1) \times 1}. \tag{15}$$

The final longitudinal displacements are obtained by adding offsets to the anchors:

$$\hat{\mathbf{Y}}_l = \mathbf{A}_l^{0:T} + \Delta\mathbf{Y}_l, \quad \hat{\mathbf{Y}}_l \in \mathbb{R}^{N_d \times M \times (T+1) \times 1}. \tag{16}$$

An auxiliary MLP head is applied to the average of $\mathbf{Q}_l^{0:T}$ over the $T+1$ time steps to predict a confidence score $\mathbf{S}_l \in \mathbb{R}^{N_d \times M \times 1}$ for candidate selection.

Our model outputs N_d candidate drive paths $\hat{\mathbf{Y}}_d$ and, for each drive path, M candidate longitudinal displacement sequences $\hat{\mathbf{Y}}_l$. A hierarchical selection strategy [28] chooses the candidate based on confidence scores $\mathbf{S}_l$ and $\mathbf{S}_d$. The selected planning is then tracked using PID controllers. Full implementation details are provided in the Appendix A.

3.5 Loss function

For planning tasks, we adopt a winner-takes-all strategy to determine which predictions are supervised. The winner is defined as the prediction whose corresponding anchor has the minimum L_2 distance from the ground truth. Other losses, including online mapping, agent detection, motion forecasting, and auxiliary tasks, follow [28]. The total loss is the weighted sum of all components. Details are described in the Appendix A.6.

$$\begin{aligned}\mathcal{L} = \lambda_{\text{map}}\mathcal{L}_{\text{map}} + \lambda_{\text{det}}\mathcal{L}_{\text{det}} + \lambda_{\text{motion}}\mathcal{L}_{\text{motion}} \\ + \lambda_{\text{drivepath}}\mathcal{L}_{\text{drivepath}} + \lambda_{\text{plan}}\mathcal{L}_{\text{plan}} + \lambda_{\text{aux}}\mathcal{L}_{\text{aux}}.\end{aligned} \tag{17}$$

4 Experiments

4.1 Dataset and metrics

Dataset. We utilize the Bench2Drive [14] benchmark for comprehensive evaluation of our model. Closed-loop performance is assessed on 220 standardized test routes to ensure a fair and reproducible comparison. To further evaluate generalization under distribution shifts, we additionally conduct experiments on the Fail2Drive benchmark [9], which introduces paired-route scenarios for testing robustness in unseen conditions. To assess open-loop performance in the real world, we also conduct experiments on the nuScenes dataset [2]. The dataset details can be found in Appendix A.2

Metrics. Driving Score (DS), Success Rate (SR), Driving Efficiency (DE), and Comfort. In addition, we introduce a Collision Rate metric, defined as the proportion of scenarios involving collisions with dynamic vehicles, to specifically assess the model's capability in handling interactive environments. For evaluation on Fail2Drive [9], we additionally report the Harmonic Mean (HM) of DS and SR, which provides a balanced metric by penalizing imbalanced performance between the two.

4.2 Main results

Closed-loop Planning Performance. As shown in Tab 1, our method achieves strong overall performance, with a Driving Score of 89.07 and a Success Rate of 73.18%, along with the highest Efficiency of 212.07. The Comfort score is 16.86. This is due to challenging scenarios, such as

Table 1: Closed-loop results of planning in Bench2Drive. **Bold** and underlined numbers indicate the best performance within different expert groups.

Method	Driving Score (↑)	SR (%) (↑)	Driving Efficiency (↑)	Comfort (↑)
Expert: PDM-Lite [1]				
SpaceDrive [17]	78.02	55.11	-	-
SimLingo [25]	86.02	67.27	259.23	33.67
Expert: Think2Drive [18]				
VAD [16]	42.35	15.00	157.94	46.01
SparseDrive [28]	44.54	16.71	170.21	**48.63**
DriveTransformer [15]	63.46	35.01	100.64	20.78
Hydra-NeXt [19]	73.86	50.00	197.76	20.68
HiP-AD [29]	86.77	69.09	203.12	19.36
AlignDrive(Ours)	**89.07**	**73.18**	**212.07**	16.86

Table 2: Multi-Ability Results in Bench2Drive.

Method	Ability (%) ↑					
	Mean	Merging	Overtaking	Emergency Brake	Give Way	Traffic Sign
UniAD-Base [12]	15.55	14.10	17.78	21.67	10.00	14.21
VAD [16]	18.07	8.11	24.44	18.64	20.00	19.15
DriveTransformer-Large [15]	38.60	17.57	35.00	48.36	40.00	52.10
HiP-AD [29]	65.98	50.00	**84.44**	**83.33**	40.00	72.10
AlignDrive	**70.06**	**75.00**	75.56	75.00	**50.00**	**74.74**

pedestrian crossings and vehicle cut-ins, which occasionally require abrupt braking or steering. Therefore, comparisons of Comfort are most meaningful among methods with similar Success Rates, as such maneuvers are necessary to ensure safe and successful navigation.

We also report the multi-ability scores in Tab 2. Our model achieves the highest overall performance, with a significantly superior average score. Most notably, it reaches a Merging score of 75, far surpassing the previous best of 50. Since merging scenarios involve challenging interactions such as consecutive lane changes and cut-ins, the improvement highlights our model's enhanced capability in handling dynamic interactions and avoiding collisions, directly validating our claim. Beyond the Carla-based benchmarks, we further evaluate AlignDrive on the nuScenes-reconstructed scenarios of HUGSIM [38], which supports reactive closed-loop driving; the results are provided in the Appendix B.

Table 3: Closed-loop results on Fail2Drive benchmark. The best results for RGB-only methods are **bolded**, and the best for Privileged/Multimodal methods are underlined.

Method	RGB	Lidar	Privilege	Language	In-Distribution			Generalization		
					DS (↑)	SR (↑)	HM (↑)	DS (↑)	SR (↑)	HM (↑)
Privileged Methods										
PlanT 2.0 [8]	×	×	✓	×	87.8	85.0	86.4	73.3	58.0	64.8
PDMLite-F2D [1]	×	×	✓	×	95.6	97.0	96.3	94.0	95.3	94.6
Multimodal Methods										
Orion [6]	✓	×	×	✓	53.0	52.0	52.5	51.2	46.0	48.5
SimLingo [25]	✓	×	×	✓	82.6	79.3	80.9	71.7	55.0	62.2
TF++ [13]	✓	✓	×	×	83.3	78.5	80.8	75.4	61.1	67.5
RGB-only Methods										
TCP [32]	✓	×	×	×	24.7	39.1	30.3	24.5	31.4	27.5
UniAD [12]	✓	×	×	×	47.5	36.3	41.2	44.0	27.6	33.9
HiP-AD [29]	✓	×	×	×	74.1	70.7	72.4	67.1	56.7	61.5
AlignDrive	✓	×	×	×	**76.0**	**80.0**	**78.0**	**71.4**	**62.0**	**66.4**

Table 4: Comparison of inference efficiency and driving performance. AlignDrive-Small is a lightweight variant with fewer decoder layers. Experiments are conducted on an RTX 3090 GPU.

Method	Parameters	Latency	Driving Score	Success Rate (%)
VAD-Base [16]	-	224.3 ms	42.35	15.00
DriveTransformer-Large [15]	646 M	221.7 ms	63.46	35.01
HiP-AD [29]	97.4 M	138.9 ms	86.77	69.09
AlignDrive	117.2 M	177.5 ms	**89.07**	**73.18**
AlignDrive-Small	**83.7 M**	**124.5 ms**	87.45	71.82

Table 5: Ablation study on AlignDrive components. LP: uses lateral path prediction to condition longitudinal planning; DP: formulates longitudinal planning as displacement regression along the drive path; DA: applies planning-oriented data augmentation

Variant	LP	DP	DA	Driving Score ↑	Success Rate (%)↑	Collision Rate (%) ↓
A				83.21	63.18	22.7
B	✓			84.85	65.45	19.5
C	✓	✓		85.82	66.81	16.3
D	✓		✓	86.54	68.92	15.7
E	✓	✓	✓	**89.07**	**73.18**	**11.4**

Generalization under Distribution Shift. As shown in Tab 3, we further evaluate our method on the Fail2Drive benchmark to assess its generalization capability, i.e., the ability to reason through unseen spatial-temporal driving scenarios and avoid unsafe outcomes in rare and challenging interactions. Our method achieves competitive performance among RGB-only approaches, with an In-Distribution DS of 76.0 and an SR of 80.0%, while maintaining strong generalization performance with a DS of 71.4 and SR of 62.0. Notably, it consistently outperforms prior RGB-only methods under the Generalization setting, demonstrating improved robustness in unseen interaction-heavy scenarios.

These results further validate that our approach not only performs competitively in in-distribution settings but also improves the ability to generalize to unseen spatial-temporal configurations and rare safety-critical interactions, which is essential for real-world autonomous driving systems.

Inference Efficiency. In addition to planning ability, we evaluate inference efficiency in table 4. Our method achieves the best Driving Score and Success Rate while maintaining lower latency than DriveTransformer and VAD. By reducing the number of stacked blocks, we further develop AlignDrive-Small, which is smaller and faster than HiP-AD yet still delivers superior performance, striking a better balance between accuracy and efficiency.

Open-loop Evaluation. We achieve the lowest collision rate in open-loop evaluation, and the full open-loop results are provided in the Appendix B.

Overall, our method consistently achieves strong planning performance and improved robustness under distribution shifts across multiple benchmarks, particularly in challenging interaction-heavy driving scenarios that require reliable longitudinal reasoning.

4.3 Ablation study

In this section, we perform ablation studies to verify the effectiveness of the key components proposed in AlignDrive, directly corresponding to our contributions.

Independent vs Path-Conditioned Longitudinal Planning. We compare Variant A, which predicts lateral drive path and longitudinal trajectories in parallel following prior SOTA methods [29], with Variant C, our proposed approach that predicts longitudinal displacements along the drive path. This cascaded, path-conditioned design couples lateral and longitudinal planning, resulting in more consistent and effective planning. As shown in Tab. 5, Variant C achieves a higher overall driving score and increases the Success Rate from 63.18% to 66.81%, demonstrating the effectiveness of path-conditioned longitudinal planning. In addition, Variant C reduces the Collision Rate from 22.7% to 16.3%, a 28.2% relative reduction. This improvement supports our claim that allow longitudinal planning condition on the drive path ensure the model to better focus on dynamic interactions, improving collision avoidance in complex scenarios.

Displacement vs Waypoint Prediction. We also evaluate Variant B, which predicts trajectory waypoints conditioned on the drive path at discrete future time steps, rather than predicting longitudinal displacements. Although both variants leverage the drive path as a lateral prior, displacements are more directly associated with dynamic interactions, whereas trajectory waypoints embed additional lateral variations that may dilute this focus. Tab. 5 shows that Variant C achieves higher Success Rate and lower Collision Rate, demonstrating that our displacement regression along the drive path is not only conceptually simpler but also empirically superior.

Planning-Oriented Data Augmentation. We evaluate the effectiveness of our planning-oriented data augmentation, which inserts synthetic traffic participants and adjusts longitudinal labels while keeping lateral paths unchanged. Variant C without augmentation is compared to Variant E with augmentation. As shown in Tab. 5, augmentation improves overall Driving Score from 85.82 to 89.07 and increases the Success Rate, demonstrating the effectiveness of our strategy. In addition, it reduces the Collision Rate from 16.3% to 11.4%, highlighting that augmentation helps the model better handle dynamic agents and improve safety.

Displacement Formulation Better Fits Augmentation. We investigate how planning-oriented data augmentation interacts with different longitudinal representations. As shown in Tab. 5, applying augmentation to waypoint-based planning (Variant D) improves Driving Score from 84.85 to 86.54 (+1.69), Success Rate from 65.45% to 68.92% (+3.47), and reduces Collision Rate from 19.5% to 15.7% (-3.8). In contrast, augmentation paired with displacement-based planning (Variant E) boosts Driving Score from 85.82 to 89.07 (+3.25), Success Rate from 66.81% to 73.18% (+6.37), and lowers Collision Rate from 16.3% to 11.4% (-4.9). These results indicate that longitudinal displacement formulation better leverages augmentation, yielding larger gains in safety-critical scenarios.

Additional qualitative results are provided in the Appendix C, while video demonstrations are included in the supporting materials.

5 Conclusion

We propose AlignDrive, a novel cascaded planning paradigm that explicitly aligns longitudinal planning reasoning with surrounding agent dynamics along the intended driving path. From the model design perspective, longitudinal planning is explicitly conditioned on predicted drive paths, where the path geometry serves as a structured prior for speed reasoning. Building on this, we reformulate longitudinal planning as 1D displacement prediction along the drive path, allowing the model to focus on dynamic interactions rather than redundantly encoding static geometry. From the data construction perspective, we further introduce a planning-oriented augmentation strategy that generates diverse safety-critical interaction scenarios to strengthen longitudinal interaction generalization. Extensive evaluations demonstrate that AlignDrive achieves state-of-the-art performance and robust generalization across challenging driving benchmarks.

Acknowledgments and Disclosure of Funding

This work was supported by the National Natural Science Foundation of China under Grant No. 62376209.

The authors declare the following competing interests: some authors are employed by Horizon Robotics, a company developing autonomous driving technology related to the subject of this work. The remaining authors declare no competing interests.

References

[1] Beißwenger, J.: Pdm-lite: A rule-based planner for carla leaderboard 2.0. Univ. Tübingen (2024)

[2] Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O.: nuscenes: A multimodal dataset for autonomous driving. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11621–11631 (2020)

[3] Cheng, J., Chen, Y., Chen, Q.: Pluto: Pushing the limit of imitation learning-based planning for autonomous driving. arXiv preprint arXiv:2404.14327 (2024)

[4] Dauner, D., Hallgarten, M., Geiger, A., Chitta, K.: Parting with misconceptions about learning-based vehicle motion planning. In: Conference on Robot Learning. pp. 1268–1281. PMLR (2023)

[5] Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., Koltun, V.: Carla: An open urban driving simulator. In: Conference on robot learning. pp. 1–16. PMLR (2017)

[6] Fu, H., Zhang, D., Zhao, Z., Cui, J., Liang, D., Zhang, C., Zhang, D., Xie, H., Wang, B., Bai, X.: Orion: A holistic end-to-end autonomous driving framework by vision-language instructed action generation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 24823–24834 (2025)

[7] Gao, H., Chen, S., Jiang, B., Liao, B., Shi, Y., Guo, X., Pu, Y., Yin, H., Li, X., Zhang, X., et al.: Rad: Training an end-to-end driving policy via large-scale 3dgs-based reinforcement learning. arXiv preprint arXiv:2502.13144 (2025)

[8] Gerstenecker, S., Geiger, A., Renz, K.: Plant 2.0: Exposing biases and structural flaws in closed-loop driving. arXiv preprint arXiv:2511.07292 (2025)

[9] Gerstenecker, S., Geiger, A., Renz, K.: Fail2drive: Benchmarking closed-loop driving generalization. arXiv preprint arXiv:2604.08535 (2026)

[10] Guo, K., Liu, H., Wu, X., Pan, J., Lv, C.: ipad: Iterative proposal-centric end-to-end autonomous driving. arXiv preprint arXiv:2505.15111 (2025)

[11] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)

[12] Hu, Y., Yang, J., Chen, L., Li, K., Sima, C., Zhu, X., Chai, S., Du, S., Lin, T., Wang, W., et al.: Planning-oriented autonomous driving. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 17853–17862 (2023)

[13] Jaeger, B., Chitta, K., Geiger, A.: Hidden biases of end-to-end driving models. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 8240–8249 (2023)

[14] Jia, X., Yang, Z., Li, Q., Zhang, Z., Yan, J.: Bench2drive: Towards multi-ability benchmarking of closed-loop end-to-end autonomous driving. Advances in Neural Information Processing Systems **37**, 819–844 (2024)

[15] Jia, X., You, J., Zhang, Z., Yan, J.: Drivetransformer: Unified transformer for scalable end-to-end autonomous driving. arXiv preprint arXiv:2503.07656 (2025)

[16] Jiang, B., Chen, S., Xu, Q., Liao, B., Chen, J., Zhou, H., Zhang, Q., Liu, W., Huang, C., Wang, X.: Vad: Vectorized scene representation for efficient autonomous driving. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 8340–8350 (2023)

[17] Li, P., Zhang, Z., Holtz, D., Yu, H., Yang, Y., Lai, Y., Song, R., Geiger, A., Zell, A.: Spacedrive: Infusing spatial awareness into vlm-based autonomous driving. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 40096–40107 (2026)

[18] Li, Q., Jia, X., Wang, S., Yan, J.: Think2drive: Efficient reinforcement learning by thinking with latent world model for autonomous driving (in carla-v2). In: European Conference on Computer Vision. pp. 142–158. Springer (2024)

[19] Li, Z., Wang, S., Lan, S., Yu, Z., Wu, Z., Alvarez, J.M.: Hydra-next: Robust closed-loop driving with open-loop training. arXiv preprint arXiv:2503.12030 (2025)

[20] Li, Z., Yu, Z., Lan, S., Li, J., Kautz, J., Lu, T., Alvarez, J.M.: Is ego status all you need for open-loop end-to-end autonomous driving? In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14864–14873 (2024)

[21] Lin, X., Lin, T., Pei, Z., Huang, L., Su, Z.: Sparse4d: Multi-view 3d object detection with sparse spatial-temporal fusion. arXiv preprint arXiv:2211.10581 (2022)

[22] Liu, H., Li, C., Wu, Q., Lee, Y.J.: Visual instruction tuning. Advances in neural information processing systems **36**, 34892–34916 (2023)

[23] Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. Communications of the ACM **65**(1), 99–106 (2021)

[24] Qiu, C., Wu, Y., Ke, W., Bai, X., Zhang, T.: Refining clip's spatial awareness: A visual-centric perspective. arXiv preprint arXiv:2504.02328 (2025)

[25] Renz, K., Chen, L., Arani, E., Sinavski, O.: Simlingo: Vision-only closed-loop autonomous driving with language-action alignment. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 11993–12003 (2025)

[26] Renz, K., Chen, L., Marcu, A.M., Hünermann, J., Hanotte, B., Karnsund, A., Shotton, J., Arani, E., Sinavski, O.: Carllava: Vision language models for camera-only closed-loop driving. arXiv preprint arXiv:2406.10165 (2024)

[27] Song, Z., Jia, C., Liu, L., Pan, H., Zhang, Y., Wang, J., Zhang, X., Xu, S., Yang, L., Luo, Y.: Don't shake the wheel: Momentum-aware planning in end-to-end autonomous driving. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 22432–22441 (2025)

[28] Sun, W., Lin, X., Shi, Y., Zhang, C., Wu, H., Zheng, S.: Sparsedrive: End-to-end autonomous driving via sparse scene representation. arXiv preprint arXiv:2405.19620 (2024)

[29] Tang, Y., Xu, Z., Meng, Z., Cheng, E.: Hip-ad: Hierarchical and multi-granularity planning with deformable attention for autonomous driving in a single decoder. arXiv preprint arXiv:2503.08612 (2025)

[30] Wang, T., Zhang, C., Qu, X., Li, K., Liu, W., Huang, C.: Diffad: A unified diffusion modeling approach for autonomous driving. arXiv preprint arXiv:2503.12170 (2025)

[31] Weng, X., Ivanovic, B., Wang, Y., Wang, Y., Pavone, M.: Para-drive: Parallelized architecture for real-time autonomous driving. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 15449–15458 (2024)

[32] Wu, P., Jia, X., Chen, L., Yan, J., Li, H., Qiao, Y.: Trajectory-guided control prediction for end-to-end autonomous driving: A simple yet strong baseline. Advances in Neural Information Processing Systems **35**, 6119–6132 (2022)

[33] Wu, Y., Zhang, T., Ke, W., Süsstrunk, S., Salzmann, M.: Spatiotemporal self-supervised learning for point clouds in the wild. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5251–5260 (2023)

[34] Xing, Z., Zhang, X., Hu, Y., Jiang, B., He, T., Zhang, Q., Long, X., Yin, W.: Goalflow: Goal-driven flow matching for multimodal trajectories generation in end-to-end autonomous driving. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 1602–1611 (2025)

[35] Zhai, J.T., Feng, Z., Du, J., Mao, Y., Liu, J.J., Tan, Z., Zhang, Y., Ye, X., Wang, J.: Rethinking the open-loop evaluation of end-to-end autonomous driving in nuscenes. arXiv preprint arXiv:2305.10430 (2023)

[36] Zhang, T., Qiu, C., Ke, W., Süsstrunk, S., Salzmann, M.: Leverage your local and global representations: A new self-supervised learning strategy. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 16580–16589 (2022)

[37] Zheng, W., Song, R., Guo, X., Zhang, C., Chen, L.: Genad: Generative end-to-end autonomous driving. In: European Conference on Computer Vision. pp. 87–104. Springer (2024)

[38] Zhou, H., Lin, L., Wang, J., Lu, Y., Bai, D., Liu, B., Wang, Y., Geiger, A., Liao, Y.: Hugsim: A real-time, photo-realistic and closed-loop simulator for autonomous driving. IEEE Transactions on Pattern Analysis and Machine Intelligence (2025)

Technical appendices and supplementary material

We include the following supplementary content:

- Additional visualization demos: a set of videos demonstrating the effectiveness of our method (see the included `index.html` file).
- Original simulation results in the CarlaV2 simulator: detailed scores for each scenario (see `AlignDrive_Meraged_Bench2drive_Results.json`).

We include the following content in the appendix:

- Additional experiments, including further ablation studies.
- Detailed model designs, with full training parameters and algorithmic specifications.

A Implementation details

A.1 Implementation details

The model employs 900 agent queries, 100 map queries, 6 drive path queries, and 5 longitudinal queries. The supervision signal for the drive path is derived from the ego vehicle's ground truth trajectory, sampled at 2-meter intervals. For longitudinal planning, the ground truth is defined as the displacements traveled along the trajectory at a 5Hz sampling rate. The longitudinal planning module employs five constant displacement anchors along the drive path, positioned at 0.25, 1.7, 4.0, 6.0, and 8.5 meters ahead of the current vehicle position. These anchors serve as reference points for predicting future longitudinal displacements. In practice, we set $\alpha = 0.1$, meaning that a virtual agent is inserted in 10% of training samples.

A.2 Dataset details

Bench2Drive [14]. This dataset consists of 1000 short video clips uniformly sampled from 44 interactive scenarios in CARLA v2 [5]. Following the official split, we use 950 clips for training and 50 for validation. Closed-loop performance is evaluated on 220 standardized test routes to ensure fair and reproducible comparison.

nuScenes [2]. nuScenes is a large-scale real-world autonomous driving dataset consisting of 1000 scenes, which are split into 700 training, 150 validation, and 150 test scenes. Each scene contains multi-sensor data collected in complex urban driving environments.

Fail2Drive [9]. Fail2Drive does not provide a training set. Following the standard protocol, we directly evaluate using models trained on Bench2Drive. Specifically, we use the weights trained on the Bench2Drive training split without further finetuning. This setup ensures a fair assessment of generalization ability under distribution shifts and unseen paired-route scenarios.

A.3 Training details

During training, the Longitudinal Planning module is initially frozen while the Drive Path predictor is trained for 12 epochs. The Longitudinal Planning module is then unfrozen, and the entire system is trained jointly, with the full training process spanning 36 epochs. Training is conducted on 32 NVIDIA RTX 4090 GPUs with a total batch size of 256. We use the AdamW optimizer with weight decay and set the initial learning rate to 1×10^{-4}. Planning-oriented data augmentation is introduced after 24 epochs to enrich interactive scenarios with virtual agents.

Our model predicts the next $T = 15$ drive path waypoints $\{\hat{\mathbf{Y}}_d^t\}_{t=1}^T$ at 2-meter intervals and longitudinal displacements $\{\hat{\mathbf{Y}}_l^t\}_{t=1}^T$ at 5 Hz. Supervision is applied using a weighted L1 loss:

$$\mathcal{L}_{\text{drivepath}} = \sum_{t=1}^{T} w_t^{\text{DP}} \|\hat{\mathbf{Y}}_d^t - \mathbf{Y}_d^t\|_1, \tag{18}$$

$$\mathcal{L}_{\text{plan}} = \sum_{t=1}^{T} w_t^{\text{long}} |\hat{\mathbf{Y}}_l^t - \mathbf{Y}_l^t|, \tag{19}$$

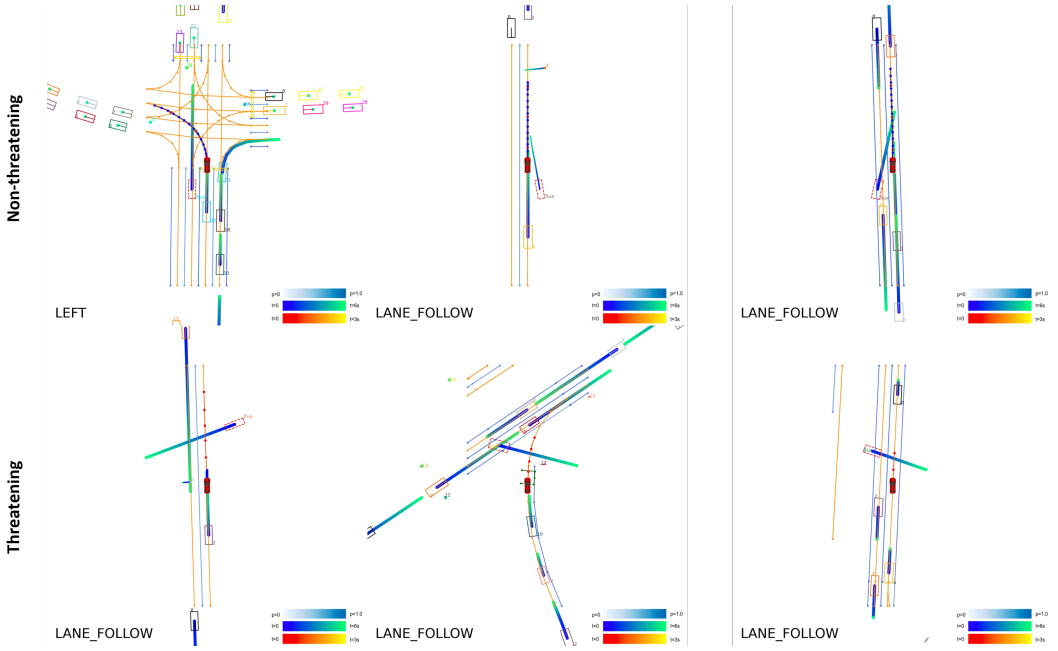


Figure 4: Visualization of planning-oriented data augmentation. The top row shows non-threatening agents, while the bottom row shows threatening agents. Inserted synthetic agents are indicated with dashed boxes. Red points denote the ego vehicle's original trajectory, and blue lines represent the adjusted longitudinal displacements after augmentation.

where the weights assign higher importance to more critical predictions. For the Drive Path waypoints, closer points receive larger weights: $w_t^{\text{DP}} = 1.0$ for $t = 1$–5, 0.6 for $t = 6$–11, and 0.4 for $t = 12$–15. A similar time-based weighting w_t^{long} is applied to longitudinal displacements. This design encourages the model to prioritize predictions that are most critical for immediate planning and safe driving.

The weights for each component of the training objective are set as follows: $\lambda_{\text{map}} = 1$, $\lambda_{\text{det}} = 1$, $\lambda_{\text{motion}} = 1$, $\lambda_{\text{drivepath}} = 2$, $\lambda_{\text{plan}} = 2$, and $\lambda_{\text{aux}} = 1$.

A.4 Model architecture

We implement our model using a ResNet-50 backbone [11] with an input image size of 640×352. Target waypoints and high-level commands are encoded into plan queries via an MLP. During training, noise is injected into the target waypoints and commands with a certain probability to improve robustness. In the standard version of our model, we employ L = 6 layers in the Drive Path Predictor and K = 6 layers in the Longitudinal Planning module. For the AlignDrive-Small variant, we use L = 4 and K = 3.

A.5 Planning-oriented data augmentation

Agent Insertion. Our planning-oriented data augmentation begins with the insertion of synthetic agents (see Algorithm 1). For each training frame, we first compute the ego vehicle's displacement over the next 3 seconds. If this displacement is below a predefined threshold δ, indicating that the ego vehicle is effectively stationary or moving very slowly, no augmentation is performed for that frame (line 10 of Algorithm 1).

For frames satisfying the displacement criterion, synthetic agents are inserted via a two-step process: selecting an initial position and generating a trajectory. The initial position depends on the agent type: threatening agents are sampled near the ego vehicle, while non-threatening agents are sampled from a distant range. Trajectories are determined by three parameters: the starting position $\mathbf{p}_{start}$, a waypoint $\mathbf{w}$ along the ego vehicle's future drive path, and the arrival time $t_{arrival}$ at the waypoint. Assuming constant velocity, the agent's position at each timestep is computed along the straight-line path connecting the start and waypoint. For threatening agents, the arrival time is chosen to potentially

induce a collision, whereas non-threatening agents have arrival times that avoid interference. This formulation allows continuous modeling of interactions between the ego vehicle and synthetic agents.

Algorithm 1 Planning-Oriented Agent Insertion

Input:
  $\mathbf{Y}_{ego}^{Traj}$: Ego vehicle future trajectory
  $\mathbf{Y}_{ego}^{DrivePath}$: Ego vehicle fßuture trajectory
  δ: Displacement threshold
  α: Insertion probability
Output:
  $\hat{\mathbf{Y}}_{a}^{motion}$: Synthetic agent trajectories
for each training frame **do**
  $D \leftarrow$ ComputeEgoDisplacement($\mathbf{Y}_{ego}^{future}$, 3s)
  if $D < \delta$ **then**
    continue
  end if
  if Random$(0, 1) \leq \alpha$ **then**
    $agentRole \leftarrow$ SelectAgentRole()
    if $agentRole =$ threatening **then**
      $\mathbf{p}_{start} \leftarrow$ SampleNearPosition($\mathbf{Y}_{ego}^{Traj}$)
    else
      $\mathbf{p}_{start} \leftarrow$ SampleFarPosition()
    end if
    $\mathbf{w} \leftarrow$ SelectWaypoint($\mathbf{Y}_{ego}^{DrivePath}$)
    $t_{arrival} \leftarrow$ SampleArrivalTime()
    $\hat{\mathbf{Y}}_{a}^{motion} \leftarrow$ GenerateTrajectory($\mathbf{p}_{start}, \mathbf{w}, t_{arrival}$)
  end if
end for

Displacement Ground-truth Generation. With the synthetic agent trajectory inserted, we adjust the ego vehicle's longitudinal ground-truth displacements to ensure safety (see Algorithm 2). Specifically, we measure the distance between the ego's predicted positions and the agent at each future timestep within 3 seconds. The last point that satisfies the minimum safety distance is chosen as the new terminal point. We then compute a scaling factor as the ratio between the ego's travel distance to the adjusted terminal point and the distance to the original terminal point. This factor is used to proportionally shrink the longitudinal displacements between consecutive waypoints, preserving trajectory smoothness while guaranteeing collision-free behavior. The effectiveness of this relabeling procedure is demonstrated in our ablation results (see table 5).

Figure 4 illustrates our planning-oriented data augmentation. The top row presents non-threatening agents, while the bottom row shows threatening agents. Inserted synthetic agents are highlighted with dashed boxes. Red points indicate the ego vehicle's original trajectory, and blue lines show the adjusted longitudinal displacements after augmentation.

It is worth noting that our agent insertion relies on minimal rule-based constraints and does not explicitly use road information. As a result, the trajectories of inserted agents may violate road rules. However, this does not negatively impact the longitudinal planning module, which primarily learns to reason about potential interactions with dynamic objects rather than strict road compliance. During training, further constraining inserted agents according to road elements represents a natural extension and a promising direction for future exploration. The hyperparameters used in our method are summarized in Table 6, and we will release our code to facilitate reproducibility.

A.6 Auxiliary tasks

We employ two primary auxiliary tasks to improve model learning. The first is ego-status prediction. Specifically, an MLP is used to predict the current ego-status of the vehicle from the plan queries, and supervision is applied using an L2 loss. The second task is inspired by the multi-granularity waypoint prediction used in HiP-AD [29]. In the Drive Path Predictor, we introduce three additional

Algorithm 2 Displacement Ground-truth Generation

Input: Ego future trajectory $\mathbf{Y}_{ego}^{future}$, synthetic agent trajectory $\hat{\mathbf{Y}}_a$, minimum safe distance d_{safe}
Output: Adjusted ego trajectory $\mathbf{Y}_{ego}^{adjusted}$
Determine all future timesteps where ego is at least d_{safe} away from the inserted agent
Let t_{new} be the last safe timestep
Set P_{new} as the ego position at t_{new} (new 3s terminal point)
Compute scaling factor s = (distance from start to P_{new}) / (distance from start to original terminal point)
for each consecutive pair of ego future trajectory points **do**
    Scale the longitudinal displacement between the points by s
end for
$\mathbf{Y}_{ego}^{adjusted} \leftarrow$ updated ego trajectory with scaled displacements

Table 6: Dataset-specific hyperparameters for nuScenes and Bench2Drive.

Hyperparameter	nuScenes	Bench2Drive
Displacement threshold δ	0.5 m	0.5 m
Insertion probability α	0.05	0.1
Safe distance d_{safe}	3.0 m	3.0 m
Ego trajectory horizon	3 s	3 s

types of queries that interact with the perceived environment in parallel with the drive path query. An Align-fusion strategy [29] is then applied, followed by separate heads to predict: (i) spatial waypoints at 5-meter intervals, (ii) temporal waypoints at 5Hz, and (iii) temporal waypoints at 2Hz. Each prediction is supervised independently. These auxiliary predictions are used only during training and do not participate in inference.

A.7 Selection and control

Selection The framework produces N_d candidate drive paths and, for each drive path, M longitudinal displacement sequences, representing $N_d \times M$ multimodal predictions that capture both lateral and longitudinal variations. First, the drive path with the highest confidence score $\mathbf{S}_d$ predicted by the Drive Path Predictor is selected, along with its corresponding longitudinal displacement candidates $\hat{\mathbf{Y}_\mathbf{l}}' \in \mathbb{R}^{M\times(T+1)\times 1}$. These candidates are further scored $\mathbf{S}_l$, penalizing those that would lead to collisions with predicted motions of other agents, following SparseDrive [28]. The candidate with the highest adjusted score is then chosen as the final output for downstream control. Importantly, we apply the same strategy to all variants to ensure a fair comparison in ablation studies.

Control. The selected candidates are executed using two independent PID controllers: one for steering and one for speed. The steering controller computes the desired heading based on the selected drive path, while the speed controller computes the desired velocity from the longitudinal displacements. Control signals for the vehicle—throttle, brake, and steering angle—are then calculated based on the difference between the desired and the current vehicle states.

B More experiments

Open-loop results on nuScene We report the open-loop results in table 7. Our method achieves the lowest collision rate, indicating stronger capability in handling dynamic interactions. Although the L2 distance is not the best, this is influenced by our data augmentation strategy, where inserting additional agents and adjusting the corresponding ground-truth trajectory can introduce discrepancies under an L2-based metric, while other approaches are more directly aligned with such supervision.

Notably, the very L2 of DriveTransformer-Large [15] is likely related to its substantially larger model capacity, as increased parameter scale can improve trajectory fitting ability under open-loop supervision. However, such improvements may not directly translate to better real-world planning performance.

Table 7: Open-loop planning evaluation results on the nuScenes validation dataset.

Method	Params	L2 (m) ↓				Collision (%) ↓			
		1s	2s	3s	Avg.	1s	2s	3s	Avg.
VAD-Base [16]	–	0.41	0.70	1.05	0.72	0.03	0.19	0.43	0.21
GenAD [37]	–	0.28	0.49	0.78	0.52	0.08	0.14	0.34	0.19
SparseDrive-S [28]	85 M	0.29	0.58	0.96	0.61	0.01	0.05	0.18	0.08
DriveTransformer-Large [15]	646 M	0.16	0.30	0.55	**0.33**	0.01	0.06	0.15	0.07
HiP-AD [29]	90 M	0.28	0.53	0.87	0.56	0.01	0.05	0.15	0.07
AlignDrive	147 M	0.31	0.56	0.93	0.60	0.00	0.02	0.17	**0.06**

Table 8: Ablation on longitudinal planning (LP), agent query decoding–re-encoding (RE), and planning-oriented data augmentation (DA). Decouple: no LP; LP + Original: LP with original agent queries; LP + Reencode: LP with decoded–re-encoded queries; Full (ours): LP + Reencode + DA.

Method	LP	RE	DA	Driving Score ↑	Success Rate (%) ↑	Collision Rate (%) ↓
Decouple				83.21	63.18	22.7
LP + Original	✓			87.47	68.18	15.4
LP + Reencode	✓	✓		85.82	66.81	16.3
Full (AlignDrive)	✓	✓	✓	**89.07**	**73.18**	**11.4**

Prior work has also noted that open-loop metrics may not fully reflect planning quality due to issues like distribution shift and causal confusion [35, 20, 4]. Consistent with this, our method achieves SOTA performance in the closed-loop CARLA evaluation, which offers a more faithful measure of real-world driving behavior.

Table 9: Evaluation on HUGSIM nuScenes-reconstructed Hard scenarios. All metrics are higher-is-better. LTF denotes Latent TransFuser.

Method	NC	DAC	TTC	Com	Rc	HD-Score
UniAD [12]	0.781	0.929	0.673	0.729	0.516	0.365
VAD [16]	0.516	0.941	0.376	1.000	0.383	0.204
LTF [13]	0.513	0.872	0.453	1.000	0.498	0.226
AlignDrive w/o Aug. w/o Cascade	0.826	0.995	0.810	0.962	0.441	0.391
AlignDrive w/o Aug.	0.868	0.993	0.843	0.981	0.472	0.422
AlignDrive	**0.889**	**0.995**	**0.880**	0.986	**0.519**	**0.491**

Evaluation on Reconstructed Real-World Scenarios. To complement the Carla-based closed-loop benchmarks with more realistic perception conditions, we further evaluate AlignDrive on the nuScenes-reconstructed Hard scenarios of HUGSIM [38], which lifts real-world driving scenes into 3D Gaussian representations while supporting reactive closed-loop evaluation. As shown in Tab. 9, AlignDrive achieves the best HD-Score (0.491), improving over UniAD (0.365) with higher NC (0.889) and TTC (0.880). Removing the planning-oriented augmentation reduces the HD-Score from 0.491 to 0.422, and further removing the cascaded formulation reduces it to 0.391, indicating that both components remain effective under reconstructed real-world interactions.

Effect of Re-encoding Agent Queries. Our planning-oriented augmentation requires agent queries to be decoded into bounding boxes and then re-encoded as structured features, which enables the insertion of synthetic agents. This design differs from directly attending to the original agent queries in the longitudinal planning (LP) module, and could potentially affect performance. To further disentangle these factors, we compare four variants: (i) **Decouple**, which excludes LP and predicts lateral and longitudinal trajectories independently; (ii) **No-Reencode**, which introduces LP but directly attends to original agent queries without decoding and re-encoding; (iii) **Reencode**, which uses LP with decoded–re-encoded agent features but without augmentation; and (iv) **Full** (AlignDrive), which combines LP, re-encoding, and planning-oriented augmentation.

As shown in Tab. 8, introducing LP (No-Reencode) already improves Driving Score and Success Rate over Decouple, demonstrating that conditioning longitudinal planning on the drive path is effective. Comparing No-Reencode and Reencode reveals a trade-off: directly using original agent queries yields stronger immediate interactions with dynamic agents, but re-encoding is necessary to support

Table 10: Multiple simulation runs of AlignDrive on Bench2Drive benchmarks. Driving Score, Success Rate, Driving Efficiency, and Comfort are reported for each run along with the average.

Run	Driving Score ↑	Success Rate (%) ↑	Driving Efficiency ↑	Comfort ↑
Run 1	89.07	73.18	212.07	16.86
Run 2	87.80	71.36	207.85	15.25
Run 3	88.05	70.00	210.08	17.10
Average	88.30	71.50	210.00	16.40

augmentation. With augmentation enabled, the Full model achieves the best performance, reducing collision rate most significantly, which confirms that data augmentation and displacement-based LP complement each other in improving robustness, particularly in safety-critical scenarios.

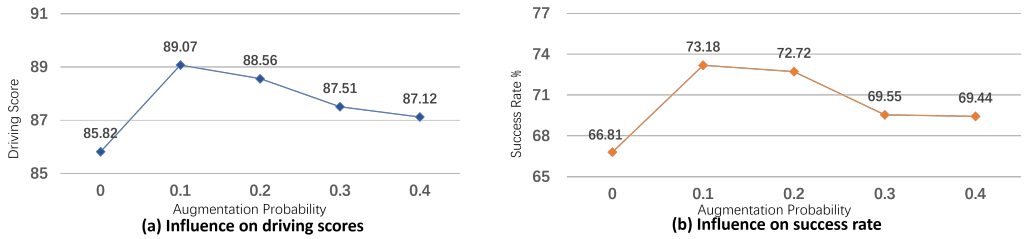


Figure 5: Effect of planning-oriented data augmentation on planning performance. All augmented variants ($p = 0.1, 0.2, 0.3, 0.4$) outperform the no-augmentation baseline.

Effect of Augmentation Probability. Fig. 5 further illustrates the impact of augmentation across different scenarios, showing that performance slightly declines when the augmentation probability exceeds 0.1, as excessive augmentation may encourage overly conservative driving. Overall, all augmented variants substantially outperform the no-augmentation baseline, demonstrating the benefit of our strategy.

Experimental Reproducibility. Due to the inherent stochasticity in the CARLA closed-loop simulator, the results of a single run may slightly. To provide a more comprehensive and reliable reference, we report multiple simulation runs of our base model and compute their average performance in Tab 10. Despite these fluctuations, all runs consistently achieve state-of-the-art results, demonstrating the robustness of our approach. This protocol ensures that the reported performance is representative and not an artifact of random variations in the simulation environment.



Figure 6: Red points are predicted drive paths, while blue points show longitudinal planning outputs (trajectory waypoints for the baseline, displacement sequences for ours). Relevant vehicles are highlighted in green. The baseline collides with cross-traffic, while our method avoids it.

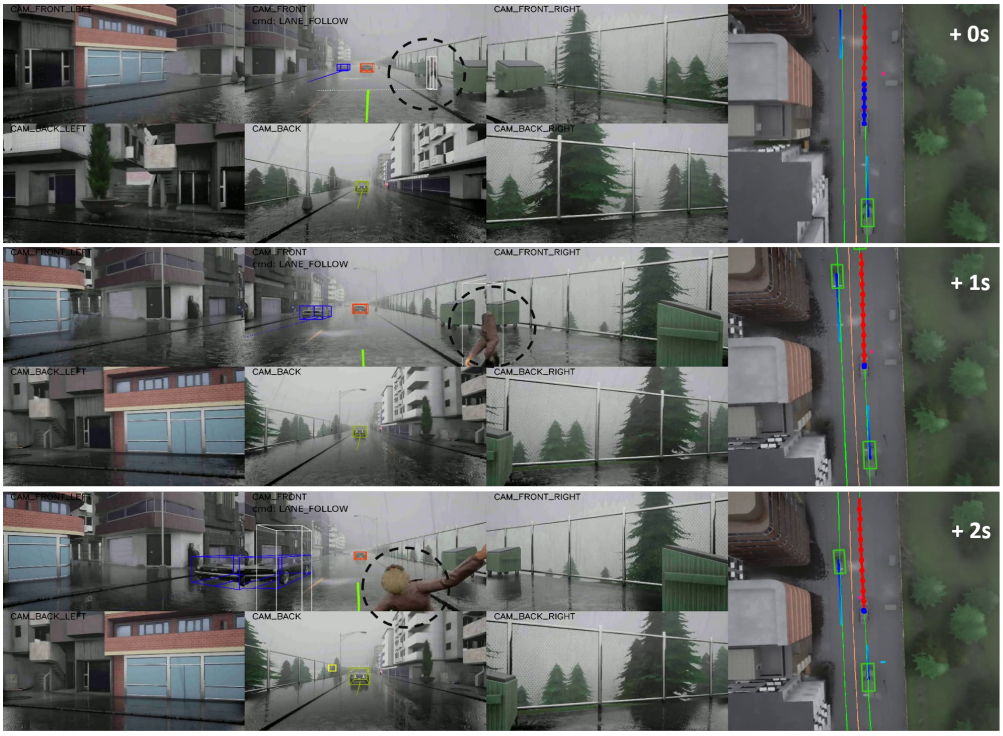
C Visulization

To better illustrate the effectiveness of our design, we compare our model with the baseline that predicts driving path and trajectory independently (Variant A in Tab. 5) under different challenging interactive scenarios. As shown in fig. 6, we consider a multi-vehicle intersection scenario where the ego vehicle intends to turn right while yielding to cross traffic. The baseline fails to react to the incoming vehicle (highlighted in purple), resulting in a conflict and eventual collision. In contrast, our method successfully anticipates the cross traffic, waits until it passes, and executes a safe maneuver.

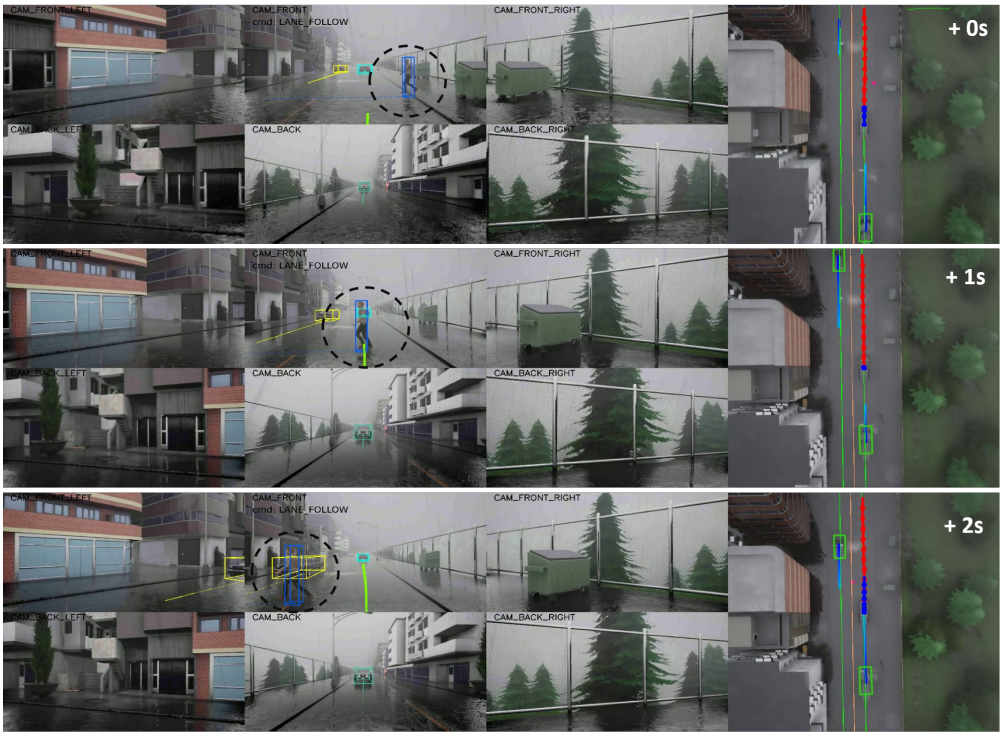
We further evaluate a more critical scenario where a pedestrian suddenly emerges onto the road. As shown in fig. 7, the baseline again fails to respond appropriately, leading to a severe safety incident, whereas our method promptly reacts to the pedestrian and avoids the collision. These qualitative results consistently demonstrate the superior safety-aware decision-making capability of our approach under dynamic and unexpected interactions. More visualization results and video demonstrations are provided in the supplementary material.

D Limitation

Recent advances in generative models, such as diffusion- and flow-matching-based approaches, have shown strong potential in producing more realistic trajectory distributions. However, how to incorporate explicit interaction-aware longitudinal reasoning into these generative frameworks remains an open question. In particular, most existing methods still focus on trajectory-level generation without explicitly modeling the coupling between longitudina decisions and surrounding agent dynamics along the driving path. We view integrating our path-conditioned interaction formulation into generative planning paradigms as a promising direction for future work.



(a) Baseline



(b) Ours

Figure 7: Comparison of Baseline (a) and Ours (b) in a pedestrian cut-in scenario. The baseline model fails to avoid the pedestrian, resulting in a collision, whereas our method promptly reacts and avoids the accident. The pedestrian is highlighted with a black dashed circle.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and introduction accurately reflect the paper's contributions, assumptions, and limitations, and are consistent with the theoretical and experimental results presented.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: The paper discusses limitations by analyzing the model's shortcomings in comfort and open-loop metrics in the experimental section, and also reports inference time as part of its efficiency considerations.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

Answer: [Yes]

Justification: The paper provides detailed experimental studies that validate each component of the method, and the experiments directly correspond to and support the claims made in the paper.

Guidelines:

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

4. **Experimental result reproducibility**

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

Answer: [Yes]

Justification: The paper provides a detailed description of the model design in the main text and includes training settings and strategy hyperparameters in the supplementary material. In addition, the code will be released, which further facilitates the reproducibility of the results.

Guidelines:

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

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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

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

   Answer: [No]

   Justification: The submitted version does not include open access to the code or data, although the authors plan to release the code in the near future, which would help facilitate reproducibility.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: The paper provides detailed experimental settings, including training and testing details, in both the main text and the appendix.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: [Yes] The appendix reports results from multiple repeated runs of the main experiments, which are used to demonstrate the variability of the proposed method.

   Guidelines:

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: The appendix provides detailed information about the hardware used for training, which is sufficient for understanding and reproducing the computational setup of the experiments.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: The research conforms to the NeurIPS Code of Ethics in all respects.

   Guidelines:

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

10. **Broader impacts**

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

Answer: [N/A]

Justification: The work is a purely algorithmic contribution and does not have direct societal impact discussed or analyzed in the paper.

Guidelines:

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

11. **Safeguards**

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

Answer: [N/A]

Justification: The work does not introduce or release any new data or models with misuse or dual-use risks, as it relies solely on publicly available datasets.

Guidelines:

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

12. **Licenses for existing assets**

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

Answer: [Yes]

Justification: The paper properly credits and cites the creators or original owners of the datasets and other existing assets used, and the use of these resources follows their respective licenses and terms of use.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [Yes]

    Justification: The paper introduces a model whose weights are planned to be released, and the documentation and relevant details of the model are described in the paper to support its future use.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing experiments or research with human subjects.

    Guidelines:

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

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

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

Answer: [N/A]

Justification: The paper does not involve human subjects research, and therefore no IRB approval or equivalent review is required.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

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

Answer: [No]

Justification: The LLM was only used for writing assistance (e.g., language polishing and formatting) and is not part of the core methodology or experimental design.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.