PIE: <u>P</u>erception and <u>I</u>nteraction <u>E</u>nhanced End-to-End Motion Planning for Autonomous Driving

Chengran Yuan[1*], Zijian Lu[1*], Zhanqi Zhang[1*], Yimin Zhao[2], Zefan Huang[1], Shuo Sun[1],
Jiawei Sun[1], Jiahui Li[1], Christina Dao Wen Lee[1], Dongen Li[1] and Marcelo H. Ang Jr.[1]

Abstract— **End-to-end motion planning is promising for simplifying complex autonomous driving pipelines. However, challenges such as scene understanding and effective prediction for decision-making continue to present substantial obstacles to its large-scale deployment. In this paper, we present *PIE*, a pioneering framework that integrates advanced perception, reasoning, and intention modeling to dynamically capture interactions between the ego vehicle and surrounding agents. It incorporates a bidirectional Mamba fusion that addresses data compression losses in multimodal fusion of camera and LiDAR inputs, alongside a novel reasoning-enhanced decoder integrating Mamba and Mixture-of-Experts to facilitate scene-compliant anchor selection and optimize adaptive trajectory inference. *PIE* adopts an action-motion interaction module to effectively utilize state predictions of surrounding agents to refine ego planning. The proposed framework is thoroughly validated on the NAVSIM benchmark. *PIE*, without using any ensemble and data augmentation techniques, achieves an 88.9 PDM score and 85.6 EPDM score, surpassing the performance of prior state-of-the-art methods. Comprehensive quantitative and qualitative analyses demonstrate that PIE is capable of reliably generating feasible and high-quality ego trajectories.**

I. INTRODUCTION

End-to-end motion planning has emerged as a promising paradigm for general robotic systems, including autonomous vehicles (AVs). This data-driven approach has the potential to enable AVs to handle complex and previously unseen scenarios, a capability that becomes increasingly critical as urban environments grow denser and more intricate. By leveraging sensor data directly, end-to-end methods [1]–[3] aim to consolidate the traditionally segmented autonomy pipeline—encompassing perception, prediction, and planning—into a single, cohesive framework.

Despite the promising performance of end-to-end methods, several key challenges remain. First, fusing multimodal data (e.g., image and LiDAR inputs) often leads to compression-induced losses when reducing historical information or fusing features from different sources. Second, while data-driven approaches offer the potential for enhancing environmental understanding, the complexity of real-world driving requires more sophisticated models that are capable of both reasoning and dynamically adjusting their strategies. Third, incorporating the predictions of other traffic participants into

the end-to-end planning pipeline often introduces substantial computational overhead. Developing efficient methods to seamlessly integrate these predictions into the planning process remains an open challenge, presenting considerable opportunities for further advancements.

To address these issues, we present *PIE*, an encoder-decoder framework designed to model the interaction between the action of ego vehicle and the motion of nearby agents and to enable more nuanced reasoning about the driving environment. Our approach mitigates data loss and integrates prediction and planning effectively. The contributions of this work are threefold:

1) **Bidirectional Mamba Fusion** We introduce a bidirectional Mamba fusion that effectively improves the multimodal data fusion between camera and LiDAR. A notable improvement of 1.9 PDM score can be achieved by merely employing this fusion approach based on the Transfuser backbone.
2) **Reasoning-Enhanced Decoder** To improve scene reasoning in complex driving scenarios, we design an efficient decoder integrating the MoE, harnessing Mamba to enhance trajectory generation.
3) **Action-Motion Interaction** We propose an action-motion interaction module via a shared cross-attention that directly integrates the velocity predictions of surrounding agents into ego action to model the dynamic interactions between traffic users.

Our approach surpasses the previous state-of-the-art DiffusionDrive [4] by achieving an 88.9 PDM score and 85.6 EPDM score on the NAVSIM `navtest` split, demonstrating the superiority and effectiveness of the proposed modules.

II. RELATED WORK

Aiming to unify perception and planning within a single framework, end-to-end learning paradigms have attracted increasing attention in the field of autonomous driving. UniAD [5], as a seminal effort, improves planning performance by integrating multiple perception tasks, thereby demonstrating the potential of end-to-end autonomous driving. VAD [6] further explores compact vectorized scene representations to enhance computational efficiency. Subsequently, a series of studies [7]–[11] adopts a single-trajectory planning paradigm to further boost planning quality. VADv2 [2] is the first to shift toward multi-modal planning by scoring and sampling from a large, fixed vocabulary of anchor trajectories. HydraMDP [12] refines VADv2's scoring mechanism by introducing additional supervision from a rule-based scorer.

* Joint first author

[1]Department of Mechanical Engineering, National University of Singapore, Singapore 119077 (e-mail: {chengran.yuan}@u.nus.edu; mpeangh@nus.edu.sg).

[2]Department of Civil and Environmental Engineering, National University of Singapore, Singapore 119077

SparseDrive [13] investigates an alternative solution that dispenses with an intermediate BEV representation. DiffusionDrive [4] employs a truncated diffusion policy for multi-modal trajectory planning. WoTE [14] proposes online trajectory evaluation using a Bird's-Eye-View (BEV) world model.

Recent efforts to improve computational efficiency in Transformer-based architectures have led to innovations like Mamba-2 [15], which reduces the quadratic complexity associated with attention computations. Mamba-based architectures have been successfully applied to language modeling [16]–[18], time-series forecasting [19], [20], human motion generation [21], and vision tasks [22], [23]. Notably, DRAMA [24] utilized Mamba to fuse image and LiDAR data and use it as a part of the decoder for AV applications. In this work, we propose a bidirectional Mamba fusion approach to address the memory forgetting inherent in Mamba and enhance perception performance.

The rapid advancement of LLMs has also been driven by the Mixture-of-Experts (MoE). Scaling model capacity without commensurate computational overhead, MoE architectures have achieved state-of-the-art results not only in NLP [25], [26] but also in domains like vision and time-series analysis [27]–[29]. We investigate the application of the MoE within the domain of autonomous driving and integrate it into the decoder, leveraging its distinctive capability to enhance planning performance with minimal computational overhead.

Motion prediction is also crucial for autonomous driving. While recent approaches [30]–[32] have improved prediction accuracy, they often remain computationally expensive and struggle to achieve satisfactory training efficiency. Their integration into fully end-to-end pipelines remains an area requiring further exploration. These gaps, coupled with the need for enhanced inference, reduced complexity, and effective multimodal fusion, motivate our exploration of efficient and effective frameworks. Given these constraints, we propose predicting the simplified motion of agents within the current frame and employing a cross-attention to integrate ego action and agents' motions, thereby achieving both efficient and resilient planning.

III. METHODOLOGY

The overall pipeline of PIE is illustrated in Figure 1. PIE introduces three key modules: Bidirectional Mamba Fusion, Reasoning-Enhanced Decoder (RED), and Intention-Motion Prediction. Initially, multi-view camera images and LiDAR BEV inputs are fused through the bidirectional Mamba fusion module. The fused features, along with ego-vehicle status, are then processed by the reasoning-enhanced decoder, which integrates two core mechanisms: (1) using sequence modeling of Mamba-2 to refine trajectory features; (2) the MoE, which dynamically routes features to specialized expert networks for managing complex driving scenarios. The action-motion interaction module incorporates the state information (bounding box and velocity) of surrounding agents to improve planning accuracy and reliability. The last reason-enhanced decoder layer outputs the ego action and agents' state feature for further action-motion cross attention in the subsequent trajectory head.

A. Bidirectional Mamba Fusion

According to Mamba-2 [15], for input sequence $x = (x_0, x_1, \dots) \in \mathbb{R}^L$, the process can be written by:

$$y_t = \sum_{s=0}^{t} C_t^\top A_{t:s}^\times B_s x_s, \tag{1}$$

where $A_{t:s}^\times$ denotes $A_t \times \cdots \times A_{s+1}, s \leq t$, and $A \in \mathbb{R}^{N \times N}$ is parameter matrix and $B \in \mathbb{R}^N, C \in \mathbb{R}^N$ are parameter vectors and Eq. (1) can be reformulated as:

$$y = \text{SSM}(A, B, C)(x) = Mx, \tag{2}$$

and the matrix M is defined as follows:

$$M_{ji} = C_j^\top A_j \cdots A_{i+1} B_i \tag{3}$$

Mamba-2 exhibits remarkable proficiency in processing long sequences. Nonetheless, as delineated in Eq. (1), (2), and (3), Mamba-2 leverages the matrix M to compress and retain as much historical data as possible, some loss of information in earlier features is still inevitable due to the limitations inherent in the compression process. To address this limitation, we propose a novel bidirectional fusion method, as illustrated in Figure 2, to effectively integrate image and LiDAR modalities. This method comprises two key components as follows:

1) LiDAR-centric Fusion: Image features are concatenated before LiDAR features along the feature dimension, resulting in a concatenated feature tensor. When processed through Mamba-2, according to its sequential processing mechanism as Eq. (1), the latter part of the sequence (LiDAR features) inherently captures contextual information from the preceding image features. Due to the natural attenuation of earlier information in the feature sequence, this configuration emphasizes LiDAR features while retaining relevant image context.

2) Image-centric Fusion: The concatenation order is reversed, with LiDAR features preceding image features. This allows the image features to incorporate contextual information from the LiDAR features while preserving a stronger emphasis on visual information.

The final fused representation is derived through the element-wise addition of these complementary features, resulting in a balanced integration that leverages the strengths of both modalities.

B. Reasoning-enhanced Decoder

The trajectory planning process shown in Figure 3 utilizes the Mamba-2 architecture combined with a cross-attention mechanism to achieve iterative refinement. Through the first cross-attention, the model generates the trajectory and bounding box feature, and then the second cross-attention layer integrates the output of the first cross-attention layer
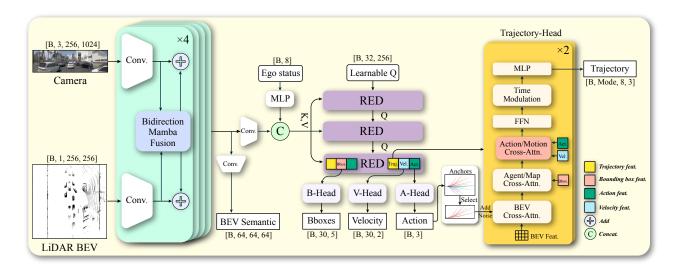
Fig. 1. **Pipeline overview of PIE.** PIE integrates camera and LiDAR BEV images in the feature space, leveraging the Bidirectional Mamba Fusion module for effective fusion in different orders. The fused feature is concatenated with ego status information and passed into the decoder. This decoder, utilizing multiple reasoning-enhanced decoder (RED) layers and specialized heads, outputs a trajectory for autonomous vehicle navigation. Additionally, it predicts the ego's action and agents' bounding boxes and velocities, which are used for auxiliary loss to enhance training and overall model performance.
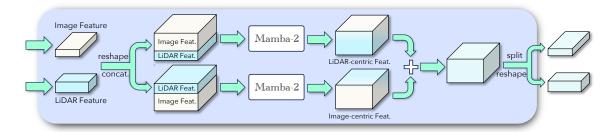

Fig. 2. **Bidirectional Mamba Fusion.** In this approach, image features and LiDAR features are first reshaped and then concatenated in two different orders. These concatenated features are separately processed by two Mamba-2 modules to generate LiDAR-centric features and Image-centric features, respectively. The outputs of these modules are then combined through an addition operation. Finally, the fused features are split and reshaped back to their original dimensions.
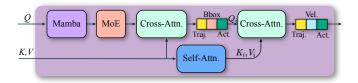

Fig. 3. **Reasoning-Enhanced Decoder (RED).** The learnable Q is processed through the Mamba module and MoE, followed by cross-attention with features from the encoder to generate an intermediate feature, part of which is used for bounding box generation. This intermediate feature undergoes further cross-attention with the feature derived through self-attention on the encoder features. The final outputs are the trajectory, velocity, and action features.

with environmental and traffic information from the self-attention. We discuss this in more detail in Section III-C.2.

Mixture of Experts To equip the planner with the capability to handle complex traffic situations efficiently without incurring excessive computation, we incorporate a Mixture of Experts (MoE) layer immediately following the output layer of Mamba-2. Specifically, the MoE layer comprises two components: a set of experts $E_1, E_2, \ldots, E_n$ and a gating network G. The MoE layer maintains consistency in the input and output dimensions, serving to refine and enhance the

results produced by Mamba-2. For input x, the final output y of MoE is denoted as follows:

$$y = \sum_{i=0}^{n} G(x)_i E_i(x) \qquad (4)$$

Our model uses *Top-K Gating* to select the first k optimal results and integrate results with *Softmax* function.

$$G(x) = Softmax(TopK(x \cdot W_g, k)) \qquad (5)$$

$$TopK(v, k)_i = \begin{cases} v_i & \text{if } v_i \text{ is in the top } k \text{ elements of } v \\ -\infty & \text{otherwise} \end{cases} \qquad (6)$$

To mitigate overfitting, the gating network normalizes the top k outputs and applies random dropout to the secondary outputs G_i. To ensure balanced utilization of the experts, the gating network redistributes any weight exceeding the maximum capacity to the next level of *Top-K* results. In our experimental setting, the number of experts is fixed to 3, and the hidden dimension of the MoE module is set to 768.

Anchor Selection Planning anchors are vital in generating trajectories that adhere to driving commands and comply

with scene constraints. We propose an innovative anchor selection mechanism designed to dynamically identify anchors aligned with the driving command. We cluster three distinct anchor types: turning left, going straight, and turning right, each comprising 20 trajectories. The anchor selection leverages the output of the action head to select the appropriate anchor type.

C. Action-Motion Interaction

To effectively model the high-level interactions between the ego vehicle and nearby agents without incurring intensive computation, we propose a shared cross-attention designed to enhance planning performance by capturing the dynamic interplay between ego actions and the motions of surrounding agents. As depicted in Figure 4, the ego action feature is initially incorporated as key and value within the cross-attention framework, followed by the integration of velocity features from surrounding agents, which are similarly utilized as key and value inputs to the same mechanism.
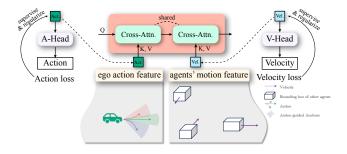

Fig. 4. **Action-Motion Cross-Attention.** The ego action feature (Act.) and the agents' motion features (Vel.) are sequentially input into a single shared cross-attention. Each feature is passed to its dedicated decoder for supervision, thereby regularizing its respective latent representation.

1) Rethink Trajectory Prediction for Planning: Modern prediction models attempt to predict precise trajectories of all agents in the environment. However, such detailed prediction is often unnecessary for real-world expert drivers. Instead, human drivers typically rely on real-time positional and velocity information of nearby agents to make split-second decisions.

In addition, predicting future states over extended time horizons inherently introduces increased uncertainty, particularly in autonomous driving scenarios. Empirical studies have shown that the variance and multimodality of predicted vehicle trajectories escalate significantly as the prediction horizon lengthens [33]. Moreover, scenarios characterized by sparse or long-tail data further exacerbate predictive uncertainty, compelling planning algorithms to adopt overly cautious strategies to maintain safety, thereby compromising efficiency [34]. Consequently, these compounded uncertainties inevitably degrade confidence in long-range predictions, directly impairing the accuracy and reliability of the associated planning processes. Given these research findings and drawing inspiration from human decision-making, in addition to bounding box prediction, we also propose to introduce real-time velocity estimations to enhance planning efficacy.

2) Velocity Prediction: To maintain efficiency, we restrict our predictions to the instantaneous velocities of surrounding agents at the current time step. The learnable query processed by the Mamba and MoE module, computes as query vectors Q for cross-attention with BEV features encoded as keys K and values V, as shown in Figure 3. Subsequently, the BEV features undergo a self-attention process to extract more velocity information features from the environment, represented as K_i and V_i. The intermediate features after the first cross-attention, which are crucial for bounding box generation, serve as query Q_i in the second cross-attention module. The second cross-attention generates the final features, which are then processed by the trajectory and velocity heads in the last RED layer to produce the final trajectory and velocity predictions. The bounding box latent feature after the second cross-attention will be used as velocity feature, hence, these two latent representations will be internally regularized during training, resulting in the velocity feature incorporating the bounding box information. In this context, we designate the velocity feature input into the action-motion cross-attention as the agents' motion feature.

To adapt the model for better velocity prediction, we incorporate the velocity prediction loss within the auxiliary loss framework. This loss component is calculated as the L1 loss between the predicted velocities of agents V^p and their corresponding ground truth values V^{gt}.

$$\mathcal{L}_{Vel} = \mathcal{L}_1(V^p, V^{gt}) \tag{7}$$

3) Action Re-extraction: We use the driving command as supervision to re-extract the ego action. The action head, referred to as **A-Head** in Figure 1, is specifically designed to generate general action intentions, including directives such as left, straight, or right. Beyond its role as an input to the action-motion cross-attention, this action additionally functions as a signal to select scenario-specific anchors tailored to the context. The action is supervised using cross-entropy loss and the loss $\mathcal{L}_{Act}$ is defined as follow:

$$\mathcal{L}_{Act} = \text{CrossEntropy}(Action, DrivingCmd[:3])$$

The reasons for not directly using the driving command are twofold: Firstly, we aim to constrain the action output to a set of three directives, namely, left, straight, and right, explicitly excluding the **unknown** command, thereby enabling scenario-specific anchor selection. Secondly, we intend for this supervision to enhance the learning of action features, resulting in this latent representation being well leveraged to facilitate the action-motion interaction.

D. Loss

We follow the loss design $\mathcal{L}_{DD}$ as implemented in [4] and introduce two auxiliary loss components, velocity loss $\mathcal{L}_{Vel}$ and action loss $\mathcal{L}_{Act}$. The overall loss function for our method can be formulated as follows:

$$\mathcal{L} = \mathcal{L}_{DD} + \lambda_v \mathcal{L}_{Vel} + \lambda_a \mathcal{L}_{Act},$$

where λ_* represents the weighting coefficients assigned to each respective loss term.

IV. EXPERIMENTS

A. Experimental Setup

*1) **Dataset:*** Our study utilizes the NAVSIM dataset [35], which is based on OpenScene [36], and serves as a streamlined version of the nuPlan dataset [37]. This dataset includes extensive driving logs totaling 120 hours and the primary data inputs for the agent comprise high-resolution images (1920×1080 pixels) from multiple viewpoints, coupled with consolidated LiDAR point cloud data derived from five distinct sensors. The input data ensemble includes the current frame along with, optionally, the three preceding frames, spanning a total duration of 1.5 seconds at a 2Hz sampling rate.

*2) **Metrics:*** The NAVSIM employs a specialized metric, the Predictive Driver Model score (PDM score), to assess the overall performance of end-to-end motion planners. The formula for PDM score is structured as follows:

$$\text{PDMS} = \left(\frac{5 \times \text{EP} + 5 \times \text{TTC} + 2 \times \text{C}}{12} \right) \times \text{NC} \times \text{DAC}, \qquad (8)$$

where Ego Progress (EP) quantifies the vehicle's progression along its intended route, focusing on advancement efficiency. Time-to-collision (TTC) assesses safety margins by measuring the temporal distance to potential collision points with other vehicles, thereby evaluating collision risk. Additionally, comfort (C) examines the trajectory's acceleration and jerk relative to established comfort thresholds, assessing the ride quality. Lastly, NC represents no collision with traffic participants, and DAC denotes no infraction regarding Drivable Area Compliance. This scoring system assigns a PDM score value of zero in instances of collisions or noncompliance with drivable area regulations. We also evaluate our performance with the extended PDM score (EPDMS):

$$\begin{aligned} \text{EPDMS} = &\left(\frac{5 \times \text{EP} + 5 \times \text{TTC} + 2 \times \text{HC} + 2 \times \text{LK} + 2 \times \text{EC}}{16} \right) \\ &\times \text{NC} \times \text{DAC} \times \text{DDC} \times \text{TLC}. \end{aligned} \qquad (9)$$

The newly introduced subscores are History Comfort (HC), Lane Keeping (LK), Extended Comfort (EC), Driving Direction Compliance (DDC), and Traffic Light Compliance (TLC), each designed to address specific aspects of driving performance evaluation.

*3) **Implementation Details:*** Our models are trained using the NAVSIM `navtrain` split and evaluated on the NAVSIM `navtest` split. Training and testing are conducted on an NVIDIA RTX 3090 Ti GPU with batch size 32, and the training epoch is 100. We use the AdamW optimizer, setting the learning rate and weight decay at 2e-4 and 0.01, respectively. The visual inputs are derived from camera images captured at the front-left and front-right positions, which are center-cropped and subsequently merged with the front-view image, forming a unified input dimension of 256×1024 pixels. The LiDAR-based BEV image is constructed by projecting LiDAR points onto a BEV plane. We only utilize the contemporaneous camera and LiDAR imagery as inputs, without historical frames and

data augmentation. Additionally, the model integrates realtime data on the ego vehicle's dynamics, such as velocity and acceleration, along with navigational commands like turning, lane changing, and following. The final output is an 8-point trajectory over a 4-second horizon, sampled at 2 Hz, where each waypoint includes coordinates for x, y, and orientation.

B. Qualitative Result

The proposed planner's performance is illustrated through ten diverse traffic scenarios in Figure 5, with several examples (highlighted in orange block) surpassing human driving behavior.

In subfigure (a), the planner ensures safer left turns by maintaining a trajectory closer to the turn lane's centerline, enhancing safety over human drivers. Subfigure (b) demonstrates the planner's ability to perceive complex intersections, halting for pedestrians in compliance with traffic laws. Subfigure (c) addresses a challenging scenario with barricades and pedestrians. The planner identifies obstacles and executes a lane change to maintain safety and speed.

In subfigure (f), the planner correctly identifies that crossing pedestrians do not compromise safety, maintaining velocity without unnecessary deceleration. Subfigures (g) and (h) showcase robust planning capabilities, with (g) dynamically navigating a T-junction to avoid congestion and narrow roads and (h) executing smooth lane changes at road forks.

Subfigures (d), (e), (i), and (j) further showcase superior decision-making. Subfigure (d) achieves better lane positioning, maintaining a safer distance from the lane edge. In subfigure (e), the planner adopts a cautious approach, yielding to pedestrians in a U-shaped road segment. Subfigure (i) optimizes parking near crosswalks, and subfigure (j) ensures safer right turns by slowing down.

These examples underscore the planner's effectiveness in navigating complex urban traffic, demonstrating advanced perception, reasoning, and adaptability in scenarios that challenge even skilled human drivers.

In Figure 6, we conduct a qualitative comparison between DiffusionDrive and our proposed method. We evaluate three distinct driving scenarios: left turns, going straight, and right turns. Compared with DiffusionDrive, our planning results exhibit better feasibility and adherence to navigation information, ascribing to our innovative anchor selection mechanism. Especially shown in the right-turn scenario, our generated multimodal trajectory candidates demonstrate superior compliance with driving commands while effectively mitigating mode collapse.

C. Quantitative Results

In this section, we provide a quantitative comparison between our method and existing approaches, followed by an analysis of the efficacy of the proposed modules and ablation studies on different experimental settings.

To enable a fair comparison, we selected other advanced end-to-end driving models that use the same image backbone. On the NAVSIM navtest split, PIE achieves state-of-the-art performance across both PDMS and EPDMS metrics.
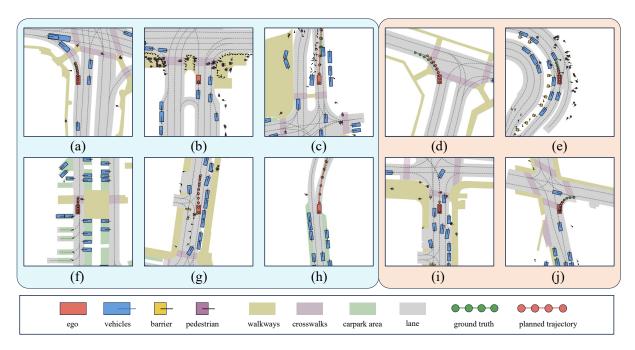
Fig. 5. Visualizations of the planning results of PIE across various scenarios: (a) Following a vehicle through a turn (b) Waiting at a red light (c) Lane change for obstacle avoidance (d) Turn left at a T-junction (e) Yielding to pedestrians (f) Crossing a crosswalk with pedestrian (g) Lane change at a T-junction (h) Turning and changing lanes at a fork in the road (i) Slow driving and stopping (j) Yielding right of way to oncoming vehicles when turning right.

TABLE I: PERFORMANCE COMPARISON ON NAVTEST SPLIT

Method	Inputs	Img.Backbone	NC ↑	DAC ↑	EP ↑	TTC ↑	C ↑	PDMS ↑
UniAD [5]	Camera	ResNet-34	97.8	91.9	78.8	92.9	**100**	83.4
PARA-Drive [10]	Camera	ResNet-34	97.9	92.4	79.3	93.0	99.8	84.0
LTF [7]	Camera	ResNet-34	97.4	92.8	79.0	92.4	**100**	83.8
Transfuser [7]	Camera & LiDAR	ResNet-34	97.7	92.8	79.2	92.8	**100**	84.0
DRAMA [24]	Camera & LiDAR	ResNet-34	98.0	93.1	80.1	94.8	**100**	85.5
Hydra-MDP [38]	Camera & LiDAR	ResNet-34	<u>98.3</u>	96.0	78.7	94.6	**100**	86.5
DiffusionDrive [4]	Camera & LiDAR	ResNet-34	98.2	96.2	<u>82.2</u>	94.7	**100**	88.1
WoTE [14]	Camera & LiDAR	ResNet-34	**98.5**	<u>96.8</u>	81.9	<u>94.9</u>	<u>99.9</u>	<u>88.3</u>
PIE (Ours)	Camera & LiDAR	ResNet-34	<u>98.3</u>	**96.9**	**83.0**	**95.0**	**100**	**88.9**

TABLE II: PERFORMANCE COMPARISON ON NAVTEST SPLIT WITH EXTENDED PDMS

Method	NC ↑	DAC ↑	EP ↑	TTC ↑	DDC ↑	LK ↑	EPDMS ↑
Transfuser [7]	97.7	92.8	79.2	92.8	98.3	67.6	77.8
VADv2 [2]	97.3	91.7	77.6	92.7	98.2	66.0	76.6
Hydra-MDP [38]	97.5	96.3	80.1	93.0	98.3	65.5	79.8
Hydra-MDP++ [39]	97.9	96.5	79.2	93.4	98.9	67.2	80.6
ARTEMIS [40]	**98.3**	95.1	81.5	<u>97.4</u>	98.6	96.5	83.1
DiffusionDrive [4]	<u>98.2</u>	96.2	**87.6**	97.3	98.6	97.0	84.0
GaussianFusion [41]	**98.3**	**97.3**	<u>87.5</u>	<u>97.4</u>	<u>99.0</u>	**97.4**	<u>85.0</u>
PIE (Ours)	**98.3**	<u>96.9</u>	**87.6**	**97.6**	**99.5**	<u>97.2</u>	**85.6**

Under the rapid-iteration release of the NAVSIM benchmark, the subscore naming has been somewhat ambiguous. For clarity of comparison, we additionally report the following subscores for our method: TL = 99.8, HC = 98.3, and EC = 88.4.

TABLE III: EFFECTIVENESS EVALUATION OF THE PROPOSED MODULES

	Method	Para.	NC ↑	DAC ↑	EP ↑	TTC ↑	C ↑	PDMS ↑
Baseline	Transfuser(T)	56.0M	97.7	92.8	79.2	92.8	100	84.0
	DiffusionDrive(DD)	60.0M	98.2	96.2	82.2	94.7	100	88.1
Ours	T + RED	58.0M	97.4	93.6	79.5	92.6	100	84.4
	T + V.P.	56.7M	97.4	93.8	79.2	93.0	100	84.5
	T + BDM	57.7M	98.0	94.3	80.6	93.6	100	85.9
	DD + BDM	61.9M	98.2	96.6	82.7	94.4	100	88.3
	DD + BDM + RED	63.9M	98.2	96.6	82.7	94.7	100	88.5
	DD + AMI	64.8M	**98.3**	96.8	**83.0**	94.7	100	88.7
	PIE	68.5M	**98.3**	96.9	**83.0**	95.0	100	88.9

RED: Reasoning-Enhanced Decoder; V.P.: Velocity Prediction (only velocity prediction, no action-motion interaction); BDM: Bidirectional Mamba Fusion Module, using two stacked Mamba-2 layers for each data fusion branch; AMI: Action-Motion Interaction.

TABLE IV: Ablation study on Bidirectional Mamba Fusion

Method	Mamba layer(s) for single fusion	Para.	NC ↑	DAC ↑	EP ↑	TTC ↑	C ↑	PDMS ↑
Transfuser(T)	N.A.	56.0M	97.7	92.8	79.2	92.8	**100**	84.0
T + BDM (++)	1	53.1M	97.3	91.8	78.1	92.7	**100**	83.0 -1.0
T + BDM (++)	2	57.7M	97.5	92.5	78.7	92.5	**100**	83.6 -0.4
T + BDM (+-)	1	53.1M	**98.2**	93.8	80.0	**93.7**	**100**	85.4 +1.4
T + BDM (+-)	2	57.7M	98.0	**94.3**	**80.6**	93.6	**100**	**85.9 +1.9**

+ : LiDAR-centric fusion; - : Image-centric fusion. ++: two fusions are both LiDAR-centric; +-: one LiDAR-centric and one Image-centric fusion. Details of BDM design are illustrated in Figure 2.
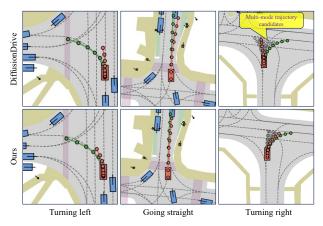

Fig. 6. **Qualitative comparison of DiffusionDrive and PIE.** With the anchor selection, our method generates a refined set of scene-compliant anchors, which effectively guide trajectory generation and facilitate more plausible and feasible waypoints.

TABLE V: Ablation study on whether to use shared cross-attention

Cross-Attn.	NC ↑	DAC ↑	EP ↑	TTC ↑	C ↑	PDMS ↑
Unshared	98.28	96.67	82.88	94.48	**100**	88.50
Shared	**98.32**	**96.92**	**82.99**	**95.00**	**100**	**88.89**

As shown in Table I, compared to DiffusionDrive, PIE achieves better performance on all metrics, especially on DAC (+0.7) and EP (+0.8). When using the extended PDM score (presented in Table II), our method achieves the highest DDC subscore (99.5), demonstrating the effectiveness of the anchor selection mechanism.

We evaluate the effectiveness of our proposed modules in Table III. Compared to the baseline Transfuser (T) with a PDM score of 84.0 and DiffusionDrive (DD) with a PDM score of 88.1, the integration of individual modules such as RED, V.P., BDM, and AMI into PIE yields quantitative enhancements. As illustrated in the table, the addition of the BDM fusion module to Transfuser significantly boosts the PDM score to 85.9. This substantial performance gain indicates the module effectively fuses LiDAR and image modalities through bidirectional fusion, preserving more modality information and strengthening the model's environmental perception. When all modules are combined in PIE, the model achieves the highest PDM score of 88.9, along with improvements in all key metrics (NC, DAC, EP, TTC).

To evaluate the effectiveness of our Bidirectional Mamba fusion (BDM) design, we conducted an ablation study, as presented in Table IV, in which we concatenated the image and LiDAR tensors into two sequences without reversing

TABLE VI: Ablation study on number of experts in MoE

Num of experts	Dim.	Para.	PDMS ↑	FPS ↑
2	512	66.5M	88.75	**39**
3	768	68.5M	**88.89**	37
4	1024	71.2M	88.78	34

The FPS results were obtained using a single NVIDIA RTX 4090 GPU.

one of them, thus removing the bidirectional property. Under these conditions, the performance declined across multiple PDM score subscores, indicating that using two LiDAR-centric fusion is detrimental. In contrast, using one LiDAR-centric and one Image-centric fusion resulted in a significant improvement in model performance across all metrics. The variant with two Mamba-2 layers in each fusion branch of the BDM achieved the highest score of 85.9.

Additionally, we conducted an ablation study in Table V to evaluate the impact of employing shared cross-attention during the action-motion interaction process. The adoption of a single shared cross-attention mechanism yielded a notable increase in PDM score, with an improvement of 0.39.

We evaluated the impact of changing the number of experts in TableVI. The results demonstrate that using three experts with a dimension of 768 achieves an optimal balance between performance and efficiency.

V. CONCLUSION

In this study, we propose PIE, an advanced framework designed to enhance perception and dynamic interaction for end-to-end autonomous driving. Its reasoning-enhanced decoder facilitates the selection of scene-compliant anchors and enhances the generation of feasible trajectories. The bidirectional Mamba fusion exhibits its efficacy through a significant performance enhancement (+1.9 PDM score), achieved by integrating it into the baseline Transfuser model. The proposed interaction mechanism leverages predictions from other agents to model the intention-level interaction between the ego and nearby agents, enhancing the planning quality. We assess the performance of PIE on the NAVSIM leaderboard, where it achieves a PDM score of 88.9 and an EPDM score of 85.6, outperforming the previous state-of-the-art methods, DiffusionDrive (PDMS 88.1) and GaussianFusion (EPDMS 85.0), respectively. The quantitative and qualitative results demonstrate the effectiveness and robustness of our method.

REFERENCES

[1] Y. Hu, J. Yang, L. Chen, K. Li, C. Sima, X. Zhu, S. Chai, S. Du, T. Lin, and W. Wang, "Planning-oriented autonomous driving," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 17 853–17 862.

[2] S. Chen, B. Jiang, H. Gao, B. Liao, Q. Xu, Q. Zhang, C. Huang, W. Liu, and X. Wang, "Vadv2: End-to-end vectorized autonomous driving via probabilistic planning," *ArXiv*, vol. abs/2402.13243, 2024. [Online]. Available: https://api.semanticscholar.org/CorpusID: 267760114

[3] K. Chitta, A. Prakash, B. Jaeger, Z. Yu, K. Renz, and A. Geiger, "Transfuser: Imitation with transformer-based sensor fusion for autonomous driving," *Pattern Analysis and Machine Intelligence (PAMI)*, 2023.

[4] B. Liao, S. Chen, H. Yin, B. Jiang, C. Wang, S. Yan, X. Zhang, X. Li, Y. Zhang, Q. Zhang, and X. Wang, "Diffusiondrive: Truncated diffusion model for end-to-end autonomous driving," 2025. [Online]. Available: https://arxiv.org/abs/2411.15139

[5] Y. Hu, J. Yang, L. Chen, K. Li, C. Sima, X. Zhu, S. Chai, S. Du, T. Lin, W. Wang, L. Lu, X. Jia, Q. Liu, J. Dai, Y. Qiao, and H. Li, "Planning-oriented autonomous driving," 2023. [Online]. Available: https://arxiv.org/abs/2212.10156

[6] B. Jiang, S. Chen, Q. Xu, B. Liao, J. Chen, H. Zhou, Q. Zhang, W. Liu, C. Huang, and X. Wang, "Vad: Vectorized scene representation for efficient autonomous driving," 2023. [Online]. Available: https://arxiv.org/abs/2303.12077

[7] K. Chitta, A. Prakash, B. Jaeger, Z. Yu, K. Renz, and A. Geiger, "Transfuser: Imitation with transformer-based sensor fusion for autonomous driving," 2022. [Online]. Available: https://arxiv.org/abs/2205.15997

[8] Z. Li, Z. Yu, S. Lan, J. Li, J. Kautz, T. Lu, and J. M. Alvarez, "Is ego status all you need for open-loop end-to-end autonomous driving?" 2024. [Online]. Available: https://arxiv.org/abs/2312.03031

[9] Y. Wang, J. He, L. Fan, H. Li, Y. Chen, and Z. Zhang, "Driving into the future: Multiview visual forecasting and planning with world model for autonomous driving," 2023. [Online]. Available: https://arxiv.org/abs/2311.17918

[10] X. Weng, B. Ivanovic, Y. Wang, Y. Wang, and M. Pavone, "Paradrive: Parallelized architecture for real-time autonomous driving," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2024, pp. 15449–15458.

[11] W. Zheng, R. Song, X. Guo, C. Zhang, and L. Chen, "Genad: Generative end-to-end autonomous driving," 2024. [Online]. Available: https://arxiv.org/abs/2402.11502

[12] Z. Li, K. Li, S. Wang, S. Lan, Z. Yu, Y. Ji, Z. Li, Z. Zhu, J. Kautz, Z. Wu, Y.-G. Jiang, and J. M. Alvarez, "Hydra-mdp: End-to-end multimodal planning with multi-target hydra-distillation," 2024. [Online]. Available: https://arxiv.org/abs/2406.06978

[13] W. Sun, X. Lin, Y. Shi, C. Zhang, H. Wu, and S. Zheng, "Sparsedrive: End-to-end autonomous driving via sparse scene representation," *ArXiv*, vol. abs/2405.19620, 2024. [Online]. Available: https://api.semanticscholar.org/CorpusID:270123261

[14] Y. Li, Y. Wang, Y. Liu, J. He, L. Fan, and Z. Zhang, "End-to-end driving with online trajectory evaluation via bev world model," 2025. [Online]. Available: https://arxiv.org/abs/2504.01941

[15] T. Dao and A. Gu, "Transformers are ssms: Generalized models and efficient algorithms through structured state space duality," 2024. [Online]. Available: https://arxiv.org/abs/2405.21060

[16] O. Lieber, B. Lenz, H. Bata, G. Cohen, J. Osin, I. Dalmedigos, E. Safahi, S. Meirom, Y. Belinkov, S. Shalev-Shwartz *et al.*, "Jamba: A hybrid transformer-mamba language model," *arXiv preprint arXiv:2403.19887*, 2024.

[17] Q. Anthony, Y. Tokpanov, P. Glorioso, and B. Millidge, "Blackmamba: Mixture of experts for state-space models," *arXiv preprint arXiv:2402.01771*, 2024.

[18] W. He, K. Han, Y. Tang, C. Wang, Y. Yang, T. Guo, and Y. Wang, "Densemamba: State space models with dense hidden connection for efficient large language models," *arXiv preprint arXiv:2403.00818*, 2024.

[19] X. Xu, Y. Liang, B. Huang, Z. Lan, and K. Shu, "Integrating mamba and transformer for long-short range time series forecasting," *arXiv preprint arXiv:2404.14757*, 2024.

[20] M. A. Ahamed and Q. Cheng, "Timemachine: A time series is worth 4 mambas for long-term forecasting," *arXiv preprint arXiv:2403.09898*, 2024.

[21] Z. Zhang, A. Liu, I. Reid, R. Hartley, B. Zhuang, and H. Tang, "Motion mamba: Efficient and long sequence motion generation," in *European Conference on Computer Vision*. Springer, 2025, pp. 265–282.

[22] L. Zhu, B. Liao, Q. Zhang, X. Wang, W. Liu, and X. Wang, "Vision mamba: Efficient visual representation learning with bidirectional state space model," 2024. [Online]. Available: https://arxiv.org/abs/2401.09417

[23] W. Dong, H. Zhu, S. Lin, X. Luo, Y. Shen, X. Liu, J. Zhang, G. Guo, and B. Zhang, "Fusion-mamba for cross-modality object detection," *arXiv preprint arXiv:2404.09146*, 2024.

[24] C. Yuan, Z. Zhang, J. Sun, S. Sun, Z. Huang, C. D. W. Lee, D. Li, Y. Han, A. Wong, K. P. Tee *et al.*, "Drama: An efficient end-to-end motion planner for autonomous driving with mamba," *arXiv preprint arXiv:2408.03601*, 2024.

[25] W. Fedus, B. Zoph, and N. Shazeer, "Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity," *J. Mach. Learn. Res.*, vol. 23, no. 1, pp. 120:5232–120:5270, Jan. 2022.

[26] D. Lepikhin, H. Lee, Y. Xu, D. Chen, O. Firat, Y. Huang, M. Krikun, N. Shazeer, and Z. Chen, "Gshard: Scaling giant models with conditional computation and automatic sharding," Jun. 2020.

[27] C. Riquelme, J. Puigcerver, B. Mustafa, M. Neumann, R. Jenatton, A. Susano Pinto, D. Keysers, and N. Houlsby, "Scaling vision with sparse mixture of experts," in *Advances in Neural Information Processing Systems*, vol. 34. Curran Associates, Inc., 2021, pp. 8583–8595.

[28] Y. Li, Y. Lin, L. Zhong, R. Yin, Y. Ji, C. T. Calafate, and C. Wu, "Boosting rare scenario perception in autonomous driving: An adaptive approach with moes and lora," *IEEE Internet of Things Journal*, pp. 1–1, 2024.

[29] S. Pini, C. S. Perone, A. Ahuja, A. S. R. Ferreira, M. Niendorf, and S. Zagoruyko, "Safe real-world autonomous driving by learning to predict and plan with a mixture of experts," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, May 2023, pp. 10069–10075.

[30] Z. Zhou, J. Wang, Y.-H. Li, and Y.-K. Huang, "Query-centric trajectory prediction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 17863–17873.

[31] S. Shi, L. Jiang, D. Dai, and B. Schiele, "MTR++: Multi-agent motion prediction with symmetric scene modeling and guided intention querying," jun 2023, arXiv:2306.17770 [cs].

[32] J. Sun, C. Yuan, S. Sun, S. Wang, Y. Han, S. Ma, Z. Huang, A. Wong, K. P. Tee, and M. H. Ang Jr, "Controlmtr: Control-guided motion transformer with scene-compliant intention points for feasible motion prediction," *arXiv preprint arXiv:2404.10295*, 2024.

[33] W. Shao, J. Xu, Z. Cao, H. Wang, and J. Li, "Uncertainty-aware prediction and application in planning for autonomous driving: Definitions, methods, and comparison," *arXiv preprint arXiv:2403.02297*, 2024.

[34] W. Zhou, Z. Cao, Y. Xu, N. Deng, X. Liu, K. Jiang, and D. Yang, "Long-tail prediction uncertainty aware trajectory planning for self-driving vehicles," in *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2022, pp. 1275–1282.

[35] D. Dauner, M. Hallgarten, T. Li, X. Weng, Z. Huang, Z. Yang, H. Li, I. Gilitschenski, B. Ivanovic, M. Pavone, A. Geiger, and K. Chitta, "Navsim: Data-driven non-reactive autonomous vehicle simulation and benchmarking," 2024. [Online]. Available: https://arxiv.org/abs/2406.15349

[36] O. Contributors, "Openscene: The largest up-to-date 3d occupancy prediction benchmark in autonomous driving," 2023.

[37] H. Caesar, J. Kabzan, K. S. Tan, W. K. Fong, E. Wolff, A. Lang, L. Fletcher, O. Beijbom, and S. Omari, "nuplan: A closed-loop ml-based planning benchmark for autonomous vehicles," *arXiv preprint arXiv:2106.11810*, 2021.

[38] Z. Li, K. Li, S. Wang, S. Lan, Z. Yu, Y. Ji, Z. Li, Z. Zhu, J. Kautz, Z. Wu, Y.-G. Jiang, and J. M. Alvarez, "Hydra-mdp: End-to-end multimodal planning with multi-target hydra-distillation," 2024. [Online]. Available: https://arxiv.org/abs/2406.06978

[39] K. Li, Z. Li, S. Lan, Y. Xie, Z. Zhang, J. Liu, Z. Wu, Z. Yu, and J. M. Alvarez, "Hydra-mdp++: Advancing end-to-end driving via expert-guided hydra-distillation," 2025. [Online]. Available: https://arxiv.org/abs/2503.12820

[40] R. Feng, N. Xi, D. Chu, R. Wang, Z. Deng, A. Wang, L. Lu, J. Wang, and Y. Huang, "Artemis: Autoregressive end-to-end trajectory planning with mixture of experts for autonomous driving," 2025. [Online]. Available: https://arxiv.org/abs/2504.19580

[41] S. Liu, Q. Liang, Z. Li, B. Li, and K. Huang, "Gaussianfusion: Gaussian-based multi-sensor fusion for end-to-end autonomous driving," 2025. [Online]. Available: https://arxiv.org/abs/2506.00034