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

EgoFSD: Ego-Centric Fully Sparse Paradigm with Uncertainty Denoising and Iterative Refinement for End-to-End Self-Driving

Haisheng Su[1], Wei Wu[2], Zhenjie Yang[1] and Isabel Guan[†,3]

***Abstract*— Current End-to-End Autonomous Driving (E2E-AD) methods resort to unifying modular designs for various tasks (e.g. perception, prediction and planning). Although optimized with a fully differentiable framework in a planning-oriented manner, existing end-to-end driving systems lacking ego-centric designs still suffer from unsatisfactory performance and inferior efficiency, due to rasterized scene representation learning and redundant information transmission. In this paper, we propose an ego-centric fully sparse paradigm, named EgoFSD, for end-to-end self-driving. Specifically, EgoFSD consists of sparse perception, hierarchical interaction and iterative motion planner. The sparse perception module performs detection and online mapping based on sparse representation of the driving scene. The hierarchical interaction module aims to select the Closest In-Path Vehicle / Stationary (CIPV / CIPS) from coarse to fine, benefiting from an additional geometric prior. As for the iterative motion planner, both selected interactive agents and ego-vehicle are considered for joint motion prediction, where the output multi-modal ego-trajectories are optimized in an iterative fashion. In addition, position-level motion diffusion and trajectory-level planning denoising are introduced for uncertainty modeling, thereby enhancing the training stability and convergence speed. Extensive experiments are conducted on nuScenes and Bench2Drive datasets, which significantly reduces the average L2 error by 59% and collision rate by 92% than UniAD while achieves 6.9× faster running efficiency.**

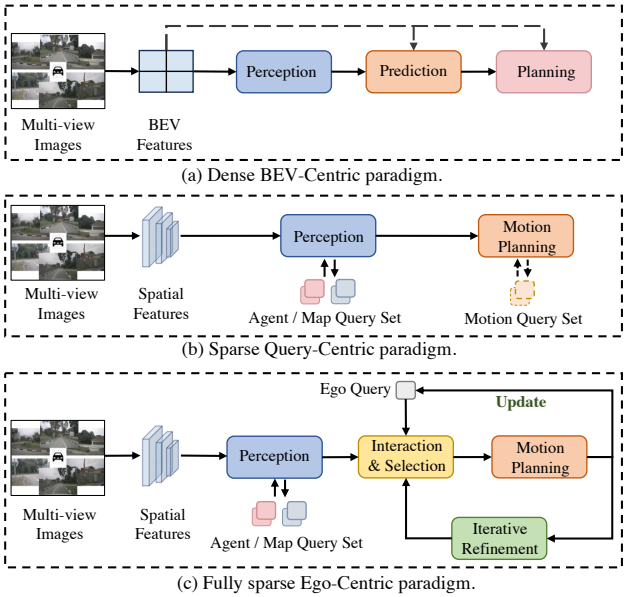


Fig. 1: Comparison of E2E-AD paradigms. (a) The dense **BEV-Centric** paradigm [12], [16]. (b) The sparse **Query-Centric** paradigm [46], [37]. (c) The proposed fully sparse **Ego-Centric** paradigm.

I. INTRODUCTION

Autonomous driving has experienced notable progress in recent years. Traditional driving systems are commonly decoupled into several standalone tasks, e.g. perception, prediction and planning. However, the well-established modular systems, which heavily rely on hand-crafted post-processing, suffer from information loss and error accumulation across sequential modules. Recently, end-to-end paradigm integrates all tasks into a unified model for planning-oriented optimization, showcasing great potential in pushing the limit of autonomous driving performance.

Literally, existing end-to-end models [12], [43], [16], [37], [41], [40], [42] designed for reliable trajectory planning can be classified into two mainstreams as summarized in Fig. 1(a) and (b). The dense BEV-Centric paradigm [12], [43] performs perception, prediction and planning consecutively upon the shared BEV (Bird's Eye View) features, which are computationally expensive leading to inferior efficiency. The sparse Query-Centric paradigm [37] utilizes sparse representation to achieve scene understanding and joint motion planning, thus improving the overall efficiency. However, object-intensive motion prediction inevitably causes computational redundancy. And it violates the driving habits of human drivers, who usually only concentrate on the Closest In-Path Vehicle / Stationary (CIPV / CIPS), which are more likely to affect the driving intention and trajectory planning of ego-vehicle. Meanwhile, excessive interaction with irrelevant agents will be conversely adverse to the ego-planning. Therefore, the planning performance remains unsatisfactory in both planning safety, comfort and personification.

To this end, we propose EgoFSD, an Ego-Centric fully sparse paradigm as shown in Fig. 1(c). Specifically, EgoFSD consists of sparse perception, hierarchical interaction and iterative motion planner. In the sparse perception module, multi-scale image features extracted from visual encoder are adopted for object detection and online mapping simultaneously in a sparse manner. Then the hierarchical interaction performs ego-centric and object-centric dual interaction to select the CIPV / CIPS with the help of an additional geometric prior. Thus the interactive queries can be selected gradually from coarse to fine. As for the motion planner,

† Corresponding author : eeguan@ust.hk. The SJTU authors were in part supported by Scientific Research Innovation Capability Support Project for Young Faculty (U40) of the Ministry of Education of China (SRICSPYF-ZY2025019).

[1] H. Su and Z. Yang are with School of Computer Science, Shanghai Jiao Tong University, Shanghai, 200240, China. {suhaisheng, yangzhengjie}@sjtu.edu.cn

[2] W. Wu is with SenseAuto, Beijing, 100080, China.

[3] I. Guan is with The Hong Kong University of Science and Technology

the mutual information between sparse interactive queries and ego-query is considered for motion prediction in a joint decoder, which is neglected in previous methods [12], [16] but is essential especially in the scenarios like intersections. To ensure the planning rationality and selection accuracy of interactive queries, the iterative planning optimization is further applied to the multi-modal proposal ego-trajectories, through continually updating the reference line and ego-query. Moreover, both position-level motion diffusion and trajectory-level planning denoising are introduced for stable training and fast convergence. It can not only model the uncertain positions of interactive agents for motion prediction, but also enhance the quality of trajectory refinement with arbitrary offsets. With above elaborate designs, EgoFSD exhibits the great potential of fully sparse paradigm for end-to-end autonomous driving, which significantly reduces the average L2 error by **59%** and collision rate by **92%** than UniAD [12] respectively. Notably, our EgoFSD-B achieves **6.9**$\times$ faster running efficiency as well. In sum, the main contributions of our work are as follows:

- We propose an **Ego-Centric Fully Sparse** paradigm for end-to-end self-**D**riving, named **EgoFSD**, without dense representation learning and redundant environmental modeling, which is proven to be effective for ego-planning.
- We introduce a geometric prior through intention-guided attention, where the **Closest In-Path Vehicle / Stationary (CIPV / CIPS)** are gradually picked out through ego-centric cross attention and selection. Besides, both **position-level diffusion** of interactive agents and **trajectory-level denoising** of ego-vehicle are adopted for uncertainty motion modeling.
- Extensive experiments are conducted on nuScenes [1] and Bench2Drive [14] for planning evaluation, which demonstrate the superiority and prominent efficiency of our EgoFSD, revealing the great potential of the proposed ego-centric fully sparse paradigm.

II. Related Work

Object Detection. Recent years witness remarkable progress achieved in multi-view 3D detection, which mainly build elaborate designs upon the dense BEV (Bird's Eye View) or sparse query features. To generate BEV features, LSS [33] lifts 2D image features to 3D space using depth estimation results, which are then splatted into BEV plane. Follow-up works apply such operation to perform view transform for 3D detection task [13], [21], [7], [17], [18]. Differently, some works [22], [39] project a series of predefined BEV queries in 3D space to the image domain for feature sampling. As for the sparse fashion, current methods [28], [27], [26], [35], [36] adopt a set of sparse queries to integrate spatial-temporal aggregations from multi-view image feature sequence for iterative anchor refinement.

Online Mapping. Maps could provide important static scenario information to ensure driving safety. Current works [20], [29], [25], [44] manage to construct online maps with on-board sensors, instead of using HD-Map which is labor intensive and expensive. HDMapNet [20] achieves this aim through BEV semantic segmentation and heuristic post-processing to generate map instances. VectorMapNet [29] introduces a two-stage auto-regressive transformer to refine map elements consecutively. MapTR [25] regards map elements as a set of points with equivalent permutations, while StreamMapNet [44] adopts a temporal fusion strategy to enhance the performance. However, all of them rely on dense BEV features for online map construction, which is computationally intensive and thus inefficient.

Motion Prediction. Predicting agent future trajectories is essential for the autonomous vehicle to understand motion intention of surrounding agents. FaF [32] predicts both current and future boxes from images using a single deep network. IntentNet [2] attempts to reason high-level behavior and long-term trajectories simultaneously. PnPNet [24] aggregate trajectory-level features for motion prediction through an online tracking module. ViP3D [6] takes images and HD-Map as input, and adopts agent queries to conduct tracking and prediction. PIP [15] further proposes to replace HD-Map with local vectorized map.

End-to-End Planning. End-to-end planning paradigm either unites modules of perception and prediction [16], [47], [43], or adopts a direct optimization on planning without intermediate tasks [3], [4], which lack interpretability and are hard to optimize. Recently, UniAD [12] presents a planning-oriented model which integrates various tasks in the dense BEV-Centric paradigm, achieving convincing performance. VAD [16] learns vectorized scene representations and improves planning safety with explicit constraints. GraphAD [47] constructs the interaction scene graph to model both dynamic and static relations. SparseDrive [37] introduces the symmetric sparse perception for parallel motion planner, which consumes more computation cost due to the repeated query projection and deformable feature aggregation, without a shared 3D feature [12], [16]. Besides, using straightforward designs and exhaustive modeling without ego-centric interaction, will inevitably lead to unsatisfactory planning performance and inferior efficiency.

III. Our Approach

A. Overview

The overall framework of EgoFSD, which addresses the end-to-end planning task in an ego-centric fully sparse paradigm, is illustrated in Fig. 2. Specifically, EgoFSD consists of four components: visual encoder, sparse perception, hierarchical interaction and iterative motion planner. First, the visual encoder extracts multi-scale spatial features from multi-view images. Then the sparse perception takes the encoded features as input to perform detection and online mapping simultaneously. In the hierarchical interaction module, an ego query equipped with a geometric prior is introduced to select the interactive queries through ego-centric cross-attention and hierarchical selection. In the iterative motion planner, both interactive agents and ego-vehicle are considered for joint motion prediction with iterative refinement. Meanwhile, both position-level diffusion

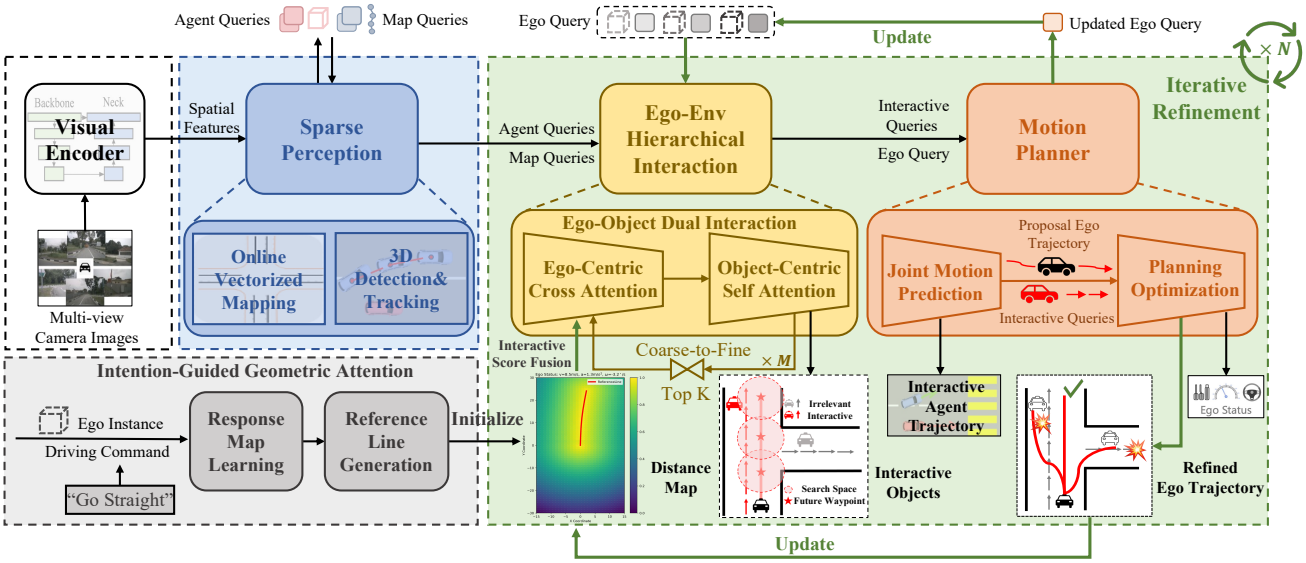


Fig. 2: Overview of our proposed framework. EgoFSD first extracts multi-scale image features from multi-view images using an off-the-shelf visual encoder, then perceives both dynamic and static elements in a sparse manner. The Ego-Env hierarchical interaction module is presented to select the interactive queries from coarse to fine according to different driving intentions of ego vehicle, which are leveraged for joint motion planner with iterative refinement. An additional geometric prior is introduced for high-quality query ranking through intention-guided attention. Besides, both position-level agent diffusion and trajectory-level ego-vehicle denoising are conducted for uncertainty modeling of the end-to-end driving system.

of interactive agents and trajectory-level denoising of ego-vehicle are conducted for uncertainty modeling of motion and planning tasks respectively.

B. Sparse Perception

After extracting the multi-view visual features $\mathbf{F}$ from sensor images using the visual encoder [8], sparse query-based perception method proposed in [26], [28], [38], [36] is extended to perform unified detection and online mapping in parallel with the symmetric architecture as adopted in [37].

Detection. Following [26], [27], surrounding agents can be represented by a group of instance features $\mathbf{F}_a \in \mathbb{R}^{N_a \times C}$ and anchor boxes $\mathbf{B}_a \in \mathbb{R}^{N_a \times 11}$ respectively. And each anchor box b_a can be denoted:

$$b_a = \{x, y, z, ln(w), ln(h), ln(l), sin(\theta), con(\theta), v_x, v_y, v_z\}, \quad (1)$$

which contains location, dimension, yaw angle as well as velocity respectively. Taking $\mathbf{F}_a$, $\mathbf{B}_a$ and the visual features $\mathbf{F}$ as input, N_{dec} decoders are adopted to consecutively refine the anchor boxes and update the instance features through iterative decoding. The updated instance features are adopted to predict the classification scores and box offsets respectively. Refer to [38], hybrid attention between current propagated queries and history memory queue is conducted for temporal modeling. Besides, temporal instance denoising is introduced to improve model stability.

Online Mapping. Similarly, we adopt an additional detection branch of same structure for online mapping. Differently, the geometric anchor of each static map element is denoted as N_p points. Therefore, surrounding maps can be represented by a group of map instance features $\mathbf{F}_m \in \mathbb{R}^{N_m \times C}$ and anchor polylines $\mathbf{B}_m \in \mathbb{R}^{N_m \times N_p \times 2}$.

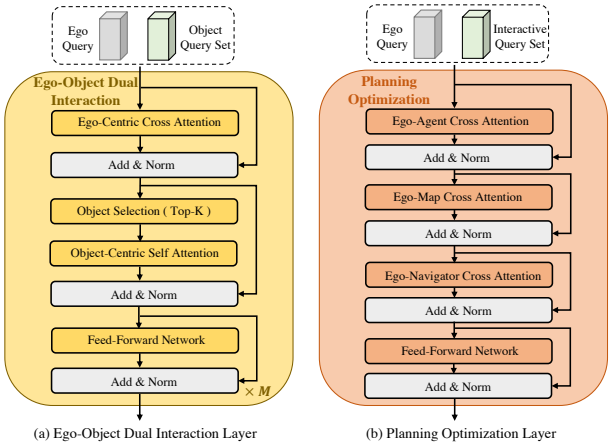
C. Ego-Env Hierarchical Interaction

We continue to perform hierarchical interaction between the ego-vehicle and surrounding objects. As shown in Fig. 2, the hierarchical interaction module mainly consists of three parts: *Ego-Object Dual Interaction*, *Intention-guided Geometric Attention* and *Coarse-to-Fine Selection*.

Ego-Object Dual Interaction. As shown in Fig. 3, a learnable embedding $\mathbf{F}_e \in \mathbb{R}^{1 \times C}$ is randomly initialized to serve as ego query, along with an ego anchor box $\mathbf{B}_e \in \mathbb{R}^{1 \times 11}$ together to represent the ego-vehicle. Both ego-centric cross attention with surrounding objects $\mathbf{F}_o \in \mathbb{R}^{N_o \times C}$ ($N_o = N_a + N_m$) and object-centric self attention are conducted consecutively to capture the mutual information comprehensively. During the attention calculation process, we combine positional embedding and query feature in a concatenated manner instead of an additive approach, which can effectively retain both semantic and geometric clues for interaction modeling.

Intention-Guided Geometric Attention. To enhance the accuracy and explainability of query ranking to facilitate selection, we introduce an ego-centric geometric prior additionally. As shown in Fig. 2, the intention-guided attention module is adopted to assess the importance of surrounding agent and map queries, which mainly consists of three steps: **response map learning**, **reference line generation** and **interactive score fusion**.

Specifically, we use four MLPs to encode the ego-intention respectively, including velocity, acceleration, angular velocity and driving command. Then we concatenate these embeddings to obtain ego-intention features $\mathbf{I}_e \in \mathbb{R}^{1 \times C}$, which are further concatenated with the position embeddings



(a) Ego-Object Dual Interaction Layer (b) Planning Optimization Layer

Fig. 3: Illustration of the **Dual Interaction** layer in the hierarchical interaction and **Planning Optimization** layer in the motion planner.

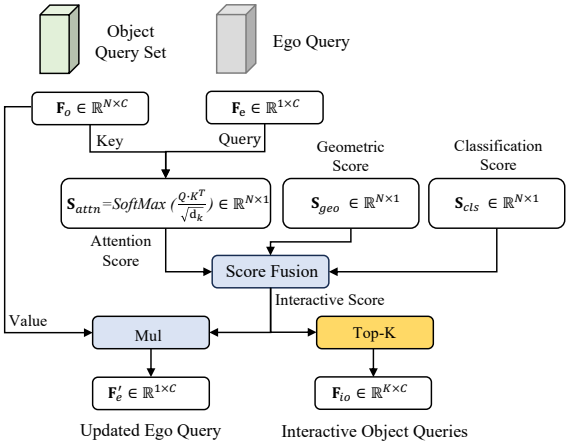


Fig. 4: Details of the **Interactive Score Fusion** process used for the geometric attended query selection.

$\mathbf{F}_p \in \mathbb{R}^{H\times W\times C}$ of a group of pre-defined locations $\mathbf{P} \in \mathbb{R}^{H\times W\times 2}$ to cover densely distributed grid cells in the BEV plane. Finally, the concatenated geometric features are fed to a single SE [9] block to learn response map $\mathbf{M}_r \in \mathbb{R}^{H\times W\times 1}$, which is supervised by the normalized minimum distance from $\mathbf{P}$ to interpolated future trajectory ψ' with T'_e waypoints. Thus, for each grid i, the response value M^i_r can be formulated as:

$$M^i_r = 1 - \min(\{||\mathbf{P}_i - \psi'_j||_2\}_{j=1}^{T'_e}) / \max(\{\min(||\mathbf{P}_i - \psi'||_2)\}_{i=1}^{H\times W}, \quad (2)$$

where T'_e is set to 30 for trajectory interpolation empirically. *The motivation is that the Closest In-Path Vehicle / Stationary are prone to affect the ego-intention, and vice versa.*

With the predicted $\mathbf{M}_r$, we first generate the reference line through row-wise thresholding, which are further used to generate the normalized distance map $\mathbf{M}_d$ (See Fig. 2). Then we can obtain the geometric score $\mathbf{S}_{geo}$ for each object query by referring to the $\mathbf{M}_d$. **The reason why we don't get the geometric score from $\mathbf{M}_r$ directly is that the imbalanced distribution of ego-intention and future waypoints [23] may lead to the inferior quality of $\mathbf{M}_r$.**

Finally, as shown in Fig. 4, we perform interactive score fusion through multiplying the attention, geometric and classification scores during the ego-centric cross attention:

$$\mathbf{S}_{inter} = \underbrace{Softmax(\mathbf{F}_e \cdot \mathbf{F}_o^T / \sqrt{d_k})}_{\mathbf{S}_{attn}\in\mathbb{R}^{N\times 1}} \odot \mathbf{S}_{geo} \odot \mathbf{S}_{cls}, \quad (3)$$

where the distance-prior is weighted with the attention score $\mathbf{S}_{attn}$ for both interaction and selection. $\cdot$ is inner product, $\odot$ is Hadamard product, and d_k is the channel dimension.

Coarse-to-Fine Selection. To capture the interaction information from coarse to fine, we stack M dual-interaction layers in a cascaded manner, where a top-K operation is appended between each two consecutive layers, thus the interactive objects can be gradually selected for latter prediction and planning usages. We claim that only a few interactive objects need to be considered for motion prediction, which are enough yet efficient for ego-centric path planning, instead of all detected agents existing in the driving scene. M is set to 3 and $K = \{10\%, 5\%, 2\%\}$ empirically.

D. Iterative Motion Planner

As shown in Fig. 2, the iterative motion planner is designed to conduct motion prediction for both interactive agents and ego-vehicle, and then optimize the ego-trajectory with both safety and kinematic constrains iteratively.

Joint Motion Prediction. With regard to the trajectory prediction, both surrounding agents and ego-vehicle are adopted for motion modeling in a joint decoder, unlike previous works [12], [16], [43] which neglect the crucial interactions between near agents and ego-vehicle when making motion predictions, especially in the common scenarios like intersections. Another difference is that only the interactive objects $\mathbf{F}_{io}$ (CIPV) sparsely selected in the former module are considered, instead of all detected agents in the driving scene which maybe irrelevant to the ego-vehicle planning. As for the joint motion decoder, we prepare three copies of ego query $\mathbf{F}'_e$ to indicate different driving intentions (*i.e.,* turn left, turn right and keep forward), which are combined with $\mathbf{F}_{io}$ to conduct agent-level self-attention and agent-map cross attention respectively. And then we concatenate these enhanced features to predict multi-modal trajectories $\tau_a \in \mathbb{R}^{N_a\times K_a\times T_a\times 2}$, $\tau_e \in \mathbb{R}^{N_e\times K_e\times T_e\times 2}$ and classification scores $\mathbf{S}_a \in \mathbb{R}^{N_a\times K_a}$, $\mathbf{S}_e \in \mathbb{R}^{N_e\times K_e}$ for both agents and ego-vehicle, where $N_e = 3$ is the number of driving command for planning, $K_a = K_e = 6$ are the mode number, $T_a = T_e = 6$ are the future timestamps.

Planning Optimization. With the predicted multi-intention and multi-modal trajectories of ego-vehicle, we can select the most probable proposal trajectory with the input driving command and classification score $\mathbf{S}_e$. As shown in Fig. 3(b), ego-agent, ego-map and ego-navigator cross attentions are conducted consecutively for planning optimization. And the offsets for each future waypoint are predicted upon the proposal trajectory respectively with several planning constraints proposed in [16] to ensure safety.

Iterative Refinement. To ensure the interaction quality and

selection accuracy of interactive queries, an additional iterative refinement strategy is proposed to continuously update the reference line and distance map $\mathbf{M}_d$ with refined ego trajectory as illustrated in Fig. 2.

E. Uncertainty Denoising

Due to the planning-oriented modular design, output uncertainty from each individual module will be inevitably introduced and passed through to the downstream tasks, leading to inferior and fragile system. Under this circumstance, we propose a two-level uncertainty modeling strategy to further stabilize the whole framework.

On one hand, **position-level diffusion process** is performed on Top-K boxes of interactive agents $\mathbf{B}_i \in \mathbb{R}^{K\times 11}$ for additional trajectory prediction of noisy agents:

$$\mathbf{B}_n = \mathbf{B}_i + \Delta\mathbf{B}_{pos} \in \mathbb{R}^{G\times K\times 11}, \tag{4}$$

which are equipped with G groups of random noises following uniform distributions. ΔB_{pos} locates within two different ranges of $\{-s, s\}$ and $\{-2s, -s\} \cup \{s, 2s\}$ to indicate positives and negatives respectively, where s indicates the noise scale. This process aims to promote the stability of motion forecasting for interactive agents with uncertain detected positions, scales and velocities.

On the other hand, **trajectory-level denoising process** is also introduced for robust offset prediction of proposal trajectory of ego-vehicle in the planning optimization stage. Different from the position diffusion of agent query on the purpose of detection and motion, we apply the random noise $\Delta\psi_{traj}$ to the ground-truth trajectory ψ_{gt} of ego-vehicle:

$$\psi_n = \psi_{gt} + \Delta\psi_{traj} \in \mathbb{R}^{G\times T_e\times 2}, \tag{5}$$

where ψ_n indicate noisy trajectory proposals and the noise scale s depends on the Final Displacement (FD) of ψ_{gt}.

F. End-to-End Learning

Multi-stage Training. To facilitate the model convergence and training performance, we divide the training process into two stages. In stage-1, the sparse perception, hierarchical interaction and joint motion prediction tasks are trained from scratch to learn sparse scene representation, interaction and motion capability respectively. Note that no selection operation is adopted in stage-1, namely all detected agents are considered for motion forecasting to make full use of annotations. In stage-2, the geometric attention module and the iterative planning optimizer are added to train jointly for overall optimization with uncertainty modeling.

Loss Functions. The overall optimization function mainly includes five tasks, where each task can be optimized with both classification and regression losses. The overall loss function for end-to-end training can be formulated as:

$$\mathcal{L} = \mathcal{L}_{det} + \mathcal{L}_{map} + \mathcal{L}_{interact} + \sum_{i=1}^{N}(\mathcal{L}^i_{motion} + \mathcal{L}^i_{plan}), \tag{6}$$

where $\mathcal{L}_{interact}$ is a combination of binary classification loss and $\mathcal{L}2$ regression loss to learn geometric score, where the positive (interactive) samples are denoted as grid cells with geometric score $\mathbf{S}_{geo} \geq 0.9$ (within $3m$ for each future waypoint). An additional regression loss is included in $\mathcal{L}_{plan}$ for ego status prediction, instead of directly using it as input to the planner as [12], [43], [16], which will lead to information leakage as proven in [23]. Meanwhile, vectorized planning constrains identified in [16] such as collision, overstepping and direction are also included in $\mathcal{L}_{plan}$ for regularization. Besides, the weight terms λ_{task} of different losses are adjusted empirically to ensure the same magnitude. N is the number of motion planning stages.

IV. Experiments

A. Datasets and Metrics

Our experiments are first conducted on the challenging public nuScenes [1] dataset, which contains 1000 driving scenes lasting 20 seconds respectively. Over 1.4M 3D bounding boxes of 23 categories are provided in total, which are annotated at 2Hz. Following the conventions [12], [16], Collision Rate (%) and L2 Displacement Error (DE) (m) are adopted to measure the open-loop planning performance. To study the effect of various perception encoders, we also evaluate the 3D object detection and online mapping results using mAP and NDS metrics respectively. Besides, Bench2Drive [14] provides a comprehensive benchmarking for evaluating multiple abilities of end-to-end AD systems in a closed-loop manner, which collects 1000 clips covering 44 interactive scenarios, 23 weathers and 12 towns in CARLA v2 [5]. Following the official settings, we use 950 clips for training while leaving 50 clips for open-loop evaluation. As for the closed-loop evaluation, we run the trained model in CARLA with 220 official test routes and calculate the closed-loop metrics such as Driving Score (DS), Success Rate (SR) and Efficiency, respectively.

B. Implementation Details

EgoFSD plans a 3 seconds future ego-trajectory with 2 seconds history information as input, which has two variants, namely EgoFSD-S and EgoFSD-B. As for EgoFSD-S, both Perspective-View version and BEV version are all implemented. ResNet50 [8] is adopted as the default backbone network for visual encoding. The perception range is set to $60m\times 30m$ longitudinally and laterally. Input image size of EgoFSD-S is resized to 256×704. For EgoFSD-S (BEV), N_{dec} is 3, and the number of BEV query, map query, agent query are 100×100, 100×20 and 300, respectively. For EgoFSD-S (PV), N_a is 900 and N_m is 100 respectively. Each map element contains 20 map points. The feature dimension C is 256. The noise scale s is set to 2.0 and 0.2×FD for motion and planning respectively. G is set to 3. EgoFSD-B has deeper network (ResNet101, N_{dec} = 6) and larger image resolution (512×1408). We use AdamW [31] optimizer and Cosine Annealing [30] scheduler to train EgoFSD with weight decay 0.01 and initial learning rate 2×10^{-4}. EgoFSD is trained for 60 epochs and 6 epochs on nuScenes and Bench2Drive respectively, running on 8 NVIDIA Tesla A800 GPUs with total batch size 32 empirically.

TABLE I: **Open-Loop Planning Evaluation Results on the nuScenes val Dataset**. ∗: LiDAR-based method. †: Reproduced with official checkpoint. ‡: Using evaluation protocol proposed in [23], [45].

Protocol	Method	Backbone	L2 (*m*) ↓				Collision (%) ↓				Latency (*ms*) ↓	FPS ↑
			1*s*	2*s*	3*s*	Avg.	1*s*	2*s*	3*s*	Avg.		
ST-P3 Metrics	FF* [10]	-	0.55	1.20	2.54	1.43	0.06	0.17	1.07	0.43	-	-
	EO* [19]	-	0.67	1.36	2.78	1.60	0.04	0.09	0.88	0.33	-	-
	ST-P3 [11]	EfficientNet-b4	1.33	2.11	2.90	2.11	0.23	0.62	1.27	0.71	628.3	1.6
	VAD-Base [16]	ResNet50	0.41	0.70	1.05	0.72	0.07	0.17	0.41	0.22	224.3	4.5
	SparseDrive-S [37]†	ResNet50	0.30	0.58	0.95	0.61	0.47	0.47	0.69	0.54	111.1	9.0
	EgoFSD-S (BEV)	ResNet50	**0.16**	**0.33**	**0.59**	**0.35**	**0.00**	**0.04**	**0.18**	**0.07**	67.7	14.8
SparseDrive Metrics‡	UniAD [12]†	ResNet101-DCN	0.45	0.70	1.04	0.73	0.62	0.58	0.63	0.61	555.6	1.8
	VAD-Base [16]†	ResNet50	0.41	0.70	1.05	0.72	0.03	0.19	0.43	0.21	224.3	4.5
	SparseDrive-S [37]	ResNet50	0.29	0.58	0.96	0.61	0.01	0.05	0.18	0.08	111.1	9.0
	SparseDrive-B [37]	ResNet101	0.29	0.55	0.91	0.58	0.01	**0.02**	**0.13**	**0.06**	137.0	7.3
	EgoFSD-S (BEV)	ResNet50	0.16	0.33	0.59	0.35	0.03	0.07	0.21	0.10	67.7	14.8
	EgoFSD-S (PV)	ResNet50	0.15	0.31	0.56	0.33	**0.00**	0.06	0.19	0.08	**59.8**	**16.7**
	EgoFSD-B (PV)	ResNet101	**0.12**	**0.28**	**0.52**	**0.30**	**0.00**	0.04	0.15	**0.06**	80.5	12.4

TABLE II: **Results on the Bench2Drive [14] Benchmark**. Both Open-Loop and Closed-Loop metrics are reported. Avg. L2 is averaged over future 2 seconds under 2Hz.

Method	Open-loop Metric	Closed-loop Metrics		
	Avg. L2 ↓ (*m*)	Driving Score ↑	Success Rate ↑ (%)	Efficiency ↑
AD-MLP [45]	3.64	18.05	0.00	48.45
UniAD-Tiny [12]	0.80	40.73	13.18	123.92
UniAD-Base [12]	0.73	45.81	16.36	129.21
VAD [16]	0.91	42.35	15.00	157.94
GenAD [48]	-	44.81	15.90	-
MomAD [34]	0.82	44.54	16.71	170.21
EgoFSD-S	0.70	52.02	21.00	178.30
EgoFSD-B	**0.66**	**60.39**	**31.78**	**180.63**

TABLE III: Necessity of the ego-centric **Query Selector** and effect of the **Geometric Prior**.

Object Selection	Geometric Attention	Planning L2 (*m*) ↓				Planning Coll. (%) ↓			
		1*s*	2*s*	3*s*	Avg.	1*s*	2*s*	3*s*	Avg.
100%	✗	0.27	0.47	0.74	0.49	0.10	0.21	0.37	0.22
Random (5%)	✗	0.28	0.49	0.79	0.52	0.08	0.17	0.38	0.21
Random (2%)	✗	0.33	0.57	0.87	0.59	0.18	0.30	0.51	0.33
0%	✗	2.25	3.75	5.26	3.75	2.82	5.42	6.39	4.88
Attn (5%)	✗	0.16	0.34	0.63	0.38	0.07	0.09	0.31	0.16
Attn (2%)	✗	0.16	0.34	0.61	**0.37**	0.06	0.08	0.24	**0.12**
Attn (2%)	Random	0.17	0.36	0.67	0.40	0.07	0.10	0.34	0.17
Attn (2%)	GroundTruth	0.14	0.23	0.33	0.23	0.07	0.08	0.10	0.07
Attn (2%)	✓	0.16	0.33	0.59	**0.35** (-5%)	0.00	0.04	0.18	**0.07** (-42%)

C. Main Results

Open-Loop Planning Evaluation. As shown in Tab. I, EgoFSD demonstrates significant advantages in both performance and efficiency compared to previous works. On one hand, EgoFSD-S achieves the minimum L2 error despite its lightweight visual backbone and inferior BEV perception. Specifically, compared with BEVFormer-based end-to-end methods [12], [16], EgoFSD-S (BEV) reduces the average L2 error by a great margin ($0.38m$ and $0.37m$, separately), while significantly reducing the average collision rates by 84% and 52% respectively. Equipped with deeper visual backbone and advanced sparse detectors from Perspective View (PV), the average L2 error and collision rates can be further reduced to $0.30m$ and to 0.06% respectively. On the other hand, benefiting from the ego-centric hierarchical interaction, *only sparse interactive agents (2%) are considered for motion planning*. Hence, EgoFSD-S can achieve great efficiency with 16.7 FPS, $9.3\times$ and $3.7\times$ faster than UniAD [12] and VAD [16] respectively.

Closed-Loop Planning Evaluation. We further validate the closed-loop performance in Bench2Drive [14], which has been proposed recently for comprehensive benchmarking of end-to-end planning methods. As shown in Tab. II, AD-MLP [45] has a high L2 error and bad closed-loop planning performance using merely ego status as input, which is different from findings in nuScenes [1], demonstrating the behavior diversity in Bench2Drive. UniAD [12] has a lower L2 error compared to VAD [16] but with worse closed-loop planning performance as discussed in [23]. Notably, EgoFSD-S achieves both the lowest L2 error and best closed-loop performance with great efficiency, showcasing the superiority and generalizability of our proposed method.

D. Ablation Study

Necessity of Geometric Prior. We claim that the Closest In-Path Vehicle as well as Stationary (CIPV / CIPS) are more likely to interact with the ego-vehicle. To verify the necessity of such geometric prior, we conduct exhaustive ablations of the ego-centric query selector as show in Tab. III. Without ego-centric selection, fewer objects randomly selected can result in worse planning results. While using the ego-centric cross attention, only 2% of surrounding queries are enough for achieving convincing planning performance. Besides, introducing the geometric prior through attention can dramatically reduce the collision rate by 42% especially, thanks to the superior interactive planning. Meanwhile, when utilizing the Ground-Truth geometric score for upper-limit evaluation, we can obtain the extremely excellent planning performance ($0.23m$ average L2 error). **The proposed ego-centric query selector equipped with geometric attention is nontrivial for motion planner.**

Effect of designs in Hierarchical Interaction. Tab. IV shows the effectiveness of our elaborate designs in the hierarchical interaction module, which contains three main designs such as Dual Interaction (DI), Geometric Attention (GA) and Coarse-to-Fine Selection (CFS). DI models both ego-centric and object-centric interactions respectively, which ensures the query selection quality for interactive planning. GA facilitates the query selection process as discussed in

TABLE IV: Ablation for designs in the **Hierarchical Interaction**. "DI" means dual interaction; "GA" means geometric attention; "CFS" means coarse-to-fine selection.

DI	GA	CFS	Planning L2 (m) ↓				Planning Coll. (%) ↓			
			1s	2s	3s	Avg.	1s	2s	3s	Avg.
✗	✓	✓	0.16	0.33	0.59	0.36	0.01	0.08	0.23	0.11
✓	✗	✓	0.16	0.34	0.61	0.37	0.06	0.08	0.24	0.12
✓	✓	✗	0.19	0.37	0.64	0.40	0.09	0.12	0.23	0.14
✓	✓	✓	0.16	0.33	0.59	**0.35**	0.00	0.04	0.18	**0.07**

TABLE V: Ablation for designs in the **Motion Planner**. "JMP" means joint motion prediction; "PO" means planning optimization; "IR" means iterative refinement. "UD" means uncertainty denoising.

ID	JMP	PO	IR	UD	Planning L2 (m) ↓				Planning Coll. (%) ↓			
					1s	2s	3s	Avg.	1s	2s	3s	Avg.
1	✓	✗	✗	✓	0.23	0.48	0.83	0.51	0.08	0.13	0.35	0.18
2	✓	✓	✗	✓	0.16	0.33	0.61	0.37	0.01	0.08	0.23	0.11
3	✓	✓	✓	✗	0.16	0.34	0.64	0.38	0.07	0.07	0.17	0.10
4	✓	✓	✓	✓	0.16	0.33	0.59	**0.35**	0.00	0.04	0.18	**0.07**

Tab. III, which reduces the collision rate obviously owing to the specialized attention on CIPV / CIPS. And CFS contributes to the multi-granularity interaction modeling through hierarchical receptive fields from global to local. All of these three designs combined together can achieve overall convincing planning performance.

Effect of designs in Motion Planner. As for motion planner in EgoFSD, Joint Motion Prediction (JMP), Planning Optimization (PO) as well as Iterative Refinement (IR) makes up the planning pipeline of ego-vehicle. Besides, Uncertain Denoising (UD) contributes to the system stability and training convergence. Tab. V explores the effect of each design exhaustively. ID-1 indicates evaluating the proposal trajectory of ego-vehicle predicted together with interactive agents, which achieves competitive L2 error but is easier to collide with surrounding agents. ID-2 improves the collision rate greatly by 38.9% with the help of PO and planning constraints [16] during training phase. ID-4 emphasizes the importance of IR in improving the quality of ego-planning trajectory (average 5.4% L2 error and 36.3% collision rate reduction respectively). ID-3 reflects the benefit of UD used for end-to-end training compared to ID-4.

Effect of Iterative Refinement stages. We continue to study the number of refinement stages in Tab. VI. We can observe that our EgoFSD can obtain superior planning performance with one additional refinement stage (36.3% collision rate reduction), which becomes saturated when introducing more stages. Hence, two-stage interacted motion planner is enough for achieving convincing results.

Effect of Uncertainty Denoising. We also validate the effectiveness of uncertainty denoising strategy including position-level motion diffusion and trajectory-level planning denoising. As shown in Tab. VII, motion diffusion can improve the prediction stability with uncertain agent positions, while the planning denoising can also strengthen the trajectory regression precision of ego-vehicle.

E. Qualitative Results

We visualize the motion trajectories of sparse interactive agents as well as planning results of EgoFSD as illustrated in Fig. 5. Both surrounding camera images and prediction results on BEV are provided accordingly. Besides, we also project the planning trajectories to the front-view camera image. Only the top-3 trajectories of selected agents interacting with ego-vehicle are visualized for better understanding of EgoFSD motivation. EgoFSD outputs planning results based on the vectorized representation in an end-to-end manner, not requiring any dense interaction and redundant motion modeling, let alone hand-crafted post-processing.

TABLE VI: Ablation for **Iterative Refinement** stages.

Number of Stages	Planning L2 (m) ↓				Planning Coll. (%) ↓			
	1s	2s	3s	Avg.	1s	2s	3s	Avg.
1	0.16	0.33	0.61	0.37	0.01	0.08	0.23	0.11
2	0.16	0.33	0.59	**0.35**	0.00	0.04	0.18	**0.07**
3	0.16	0.33	0.60	0.36	0.01	0.40	0.22	0.09
4	0.16	0.33	0.61	0.36	0.00	0.04	0.20	0.08

TABLE VII: Ablation for **Uncertainty Denoising** on both position and trajectory aspects.

Position Diffusion	Trajectory Denoising	Planning L2 (m) ↓				Planning Coll. (%) ↓			
		1s	2s	3s	Avg.	1s	2s	3s	Avg.
✗	✗	0.16	0.34	0.64	0.38	0.07	0.07	0.17	0.10
✓	✗	0.16	0.34	0.63	0.37	0.02	0.04	0.15	0.07
✓	✓	0.16	0.33	0.59	**0.35**	0.00	0.04	0.18	**0.07**

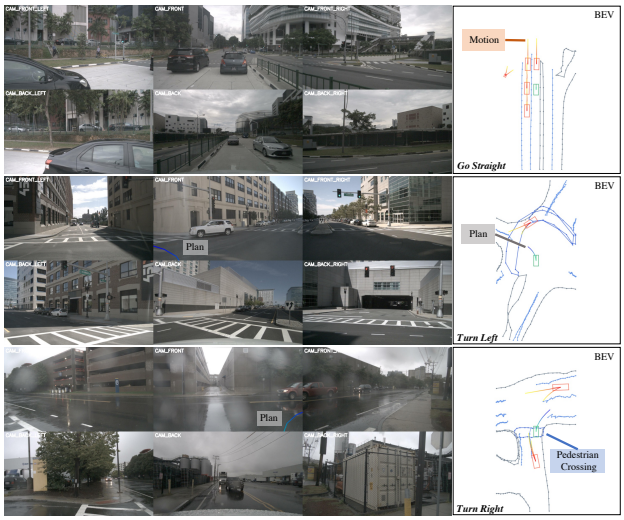


Fig. 5: Qualitative results of EgoFSD. We omit the map selection results for clarity of road structure details.

V. Conclusion

In this paper, we propose a **Fully Sparse Paradigm** for end-to-end self-driving in an **Ego-Centric** manner, termed as EgoFSD. EgoFSD conducts hierarchical interaction based on sparse representation and perception results. Only interactive agents are considered for joint motion prediction including the ego-vehicle. Iterative planning optimization strategy contributes to the driving safety with interactive decision. Besides, uncertainty modeling is conducted to improve the stability of end-to-end system. Extensive ablations and comparisons reveal the superiority and great potential of our ego-centric fully sparse paradigm.

REFERENCES

[1] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *CVPR*, 2020.

[2] S. Casas, W. Luo, and R. Urtasun. Intentnet: Learning to predict intention from raw sensor data. In *CoRL*, pages 947–956. PMLR, 2018.

[3] F. Codevilla, M. Müller, A. López, V. Koltun, and A. Dosovitskiy. End-to-end driving via conditional imitation learning. In *ICRA*, 2018.

[4] F. Codevilla, E. Santana, A. M. López, and A. Gaidon. Exploring the limitations of behavior cloning for autonomous driving. In *ICCV*, 2019.

[5] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun. Carla: An open urban driving simulator. In *CoRL*, pages 1–16. PMLR, 2017.

[6] J. Gu, C. Hu, T. Zhang, X. Chen, Y. Wang, Y. Wang, and H. Zhao. Vip3d: End-to-end visual trajectory prediction via 3d agent queries. In *CVPR*, pages 5496–5506, 2023.

[7] C. Han, J. Yang, J. Sun, Z. Ge, R. Dong, W. Mao, Y. Peng, and X. Zhang. Exploring recurrent long-term temporal fusion for multi-view 3d perception. *RAL*, 2024.

[8] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[9] J. Hu, L. Shen, and G. Sun. Squeeze-and-excitation networks. In *CVPR*, 2018.

[10] P. Hu, A. Huang, J. Dolan, D. Held, and D. Ramanan. Safe local motion planning with self-supervised freespace forecasting. In *CVPR*, pages 12732–12741, 2021.

[11] S. Hu, L. Chen, P. Wu, H. Li, J. Yan, and D. Tao. St-p3: End-to-end vision-based autonomous driving via spatial-temporal feature learning. In *ECCV*, 2022.

[12] Y. Hu, J. Yang, L. Chen, K. Li, C. Sima, X. Zhu, S. Chai, S. Du, T. Lin, W. Wang, et al. Planning-oriented autonomous driving. In *CVPR*, pages 17853–17862, 2023.

[13] J. Huang, G. Huang, Z. Zhu, Y. Ye, and D. Du. Bevdet: High-performance multi-camera 3d object detection in bird-eye-view. *arXiv preprint arXiv:2112.11790*, 2021.

[14] X. Jia, Z. Yang, Q. Li, Z. Zhang, and J. Yan. Bench2drive: Towards multi-ability benchmarking of closed-loop end-to-end autonomous driving. *arXiv preprint arXiv:2406.03877*, 2024.

[15] B. Jiang, S. Chen, X. Wang, B. Liao, T. Cheng, J. Chen, H. Zhou, Q. Zhang, W. Liu, and C. Huang. Perceive, interact, predict: Learning dynamic and static clues for end-to-end motion prediction. *arXiv preprint arXiv:2212.02181*, 2022.

[16] B. Jiang, S. Chen, Q. Xu, B. Liao, J. Chen, H. Zhou, Q. Zhang, W. Liu, C. Huang, and X. Wang. Vad: Vectorized scene representation for efficient autonomous driving. In *ICCV*, pages 8340–8350, 2023.

[17] X. Jin, H. Su, K. Liu, C. Ma, W. Wu, F. Hui, and J. Yan. Unimamba: Unified spatial-channel representation learning with group-efficient mamba for lidar-based 3d object detection. In *CVPR*, pages 1407–1417, 2025.

[18] X. Jin, H. Su, C. Ma, K. Liu, W. Wu, F. Hui, and J. Yan. Geoformer: Geometry point encoder for 3d object detection with graph-based transformer. In *ICCV*, pages 26879–26889, 2025.

[19] T. Khurana, P. Hu, A. Dave, J. Ziglar, D. Held, and D. Ramanan. Differentiable raycasting for self-supervised occupancy forecasting. In *ECCV*, 2022.

[20] Q. Li, Y. Wang, Y. Wang, and H. Zhao. Hdmapnet: An online hd map construction and evaluation framework. In *ICRA*, 2022.

[21] Y. Li, Z. Ge, G. Yu, J. Yang, Z. Wang, Y. Shi, J. Sun, and Z. Li. Bevdepth: Acquisition of reliable depth for multi-view 3d object detection. In *AAAI*, volume 37, pages 1477–1485, 2023.

[22] Z. Li, W. Wang, H. Li, E. Xie, C. Sima, T. Lu, Y. Qiao, and J. Dai. Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. In *ECCV*, pages 1–18. Springer, 2022.

[23] 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? In *CVPR*, pages 14864–14873, 2024.

[24] M. Liang, B. Yang, W. Zeng, Y. Chen, R. Hu, S. Casas, and R. Urtasun. Pnpnet: End-to-end perception and prediction with tracking in the loop. In *CVPR*, pages 11553–11562, 2020.

[25] B. Liao, S. Chen, X. Wang, T. Cheng, Q. Zhang, W. Liu, and C. Huang. Maptr: Structured modeling and learning for online vectorized hd map construction. *arXiv preprint arXiv:2208.14437*, 2022.

[26] X. Lin, Z. Pei, T. Lin, L. Huang, and Z. Su. Sparse4d v3: Advancing end-to-end 3d detection and tracking. *arXiv preprint arXiv:2311.11722*, 2023.

[27] H. Liu, Y. Teng, T. Lu, H. Wang, and L. Wang. Sparsebev: High-performance sparse 3d object detection from multi-camera videos. In *ICCV*, pages 18580–18590, 2023.

[28] Y. Liu, T. Wang, X. Zhang, and J. Sun. Petr: Position embedding transformation for multi-view 3d object detection. In *ECCV*, pages 531–548. Springer, 2022.

[29] Y. Liu, T. Yuan, Y. Wang, Y. Wang, and H. Zhao. Vectormapnet: End-to-end vectorized hd map learning. In *ICML*, pages 22352–22369. PMLR, 2023.

[30] I. Loshchilov and F. Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

[31] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[32] W. Luo, B. Yang, and R. Urtasun. Fast and furious: Real time end-to-end 3d detection, tracking and motion forecasting with a single convolutional net. In *CVPR*, pages 3569–3577, 2018.

[33] J. Philion and S. Fidler. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In *ECCV*, 2020.

[34] Z. Song, C. Jia, L. Liu, H. Pan, Y. Zhang, J. Wang, X. Zhang, S. Xu, L. Yang, and Y. Luo. Don't shake the wheel: Momentum-aware planning in end-to-end autonomous driving. *arXiv preprint arXiv:2503.03125*, 2025.

[35] H. Su, F. Song, C. Ma, W. Wu, and J. Yan. Robosense: Large-scale dataset and benchmark for egocentric robot perception and navigation in crowded and unstructured environments. *arXiv preprint arXiv:2408.15503*, 2024.

[36] H. Su, J. Zhang, F. Song, S. Zhou, W. Wu, J. Yan, and N. Zheng. Freqpde: Rethinking positional depth embedding for multi-view 3d object detection transformers. In *ICCV*, pages 28145–28155, 2025.

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

[38] S. Wang, Y. Liu, T. Wang, Y. Li, and X. Zhang. Exploring object-centric temporal modeling for efficient multi-view 3d object detection. In *ICCV*, pages 3621–3631, 2023.

[39] C. Yang, Y. Chen, H. Tian, C. Tao, X. Zhu, Z. Zhang, G. Huang, H. Li, Y. Qiao, L. Lu, et al. Bevformer v2: Adapting modern image backbones to bird's-eye-view recognition via perspective supervision. In *CVPR*, pages 17830–17839, 2023.

[40] Z. Yang, Y. Chai, X. Jia, Q. Li, Y. Shao, X. Zhu, H. Su, and J. Yan. Drivemoe: Mixture-of-experts for vision-language-action model in end-to-end autonomous driving. *arXiv preprint arXiv:2505.16278*, 2025.

[41] Z. Yang, X. Jia, H. Li, and J. Yan. Llm4drive: A survey of large language models for autonomous driving. *arXiv preprint arXiv:2311.01043*, 2023.

[42] Z. Yang, X. Jia, Q. Li, X. Yang, M. Yao, and J. Yan. Raw2drive: Reinforcement learning with aligned world models for end-to-end autonomous driving (in carla v2). *arXiv preprint arXiv:2505.16394*, 2025.

[43] T. Ye, W. Jing, C. Hu, S. Huang, L. Gao, F. Li, J. Wang, K. Guo, W. Xiao, W. Mao, et al. Fusionad: Multi-modality fusion for prediction and planning tasks of autonomous driving. *arXiv preprint arXiv:2308.01006*, 2023.

[44] T. Yuan, Y. Liu, Y. Wang, Y. Wang, and H. Zhao. Streammapnet: Streaming mapping network for vectorized online hd map construction. In *WACV*, pages 7356–7365, 2024.

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

[46] D. Zhang, G. Wang, R. Zhu, J. Zhao, X. Chen, S. Zhang, J. Gong, Q. Zhou, W. Zhang, N. Wang, et al. Sparsead: Sparse query-centric paradigm for efficient end-to-end autonomous driving. *arXiv preprint arXiv:2404.06892*, 2024.

[47] Y. Zhang, D. Qian, D. Li, Y. Pan, Z. Zhang, S. Zhang, H. Li, M. Fu, et al. Graphad: Interaction scene graph for end-to-end autonomous driving. *arXiv preprint arXiv:2403.19098*, 2024.

[48] W. Zheng, R. Song, X. Guo, C. Zhang, and L. Chen. Genad: Generative end-to-end autonomous driving. *arXiv preprint arXiv: 2402.11502*, 2024.