MOJITO: Modal Joint Learning for Unified End-to-End Autonomous Driving

Zhijing Cheng[1,2*†], Xuancheng Zhang[2†], Donglin Di[2], Lei Fan[3], Baorui Ma[2], Hao Li[2], and Xun Yang[1✉]

[1] University of Science and Technology of China, China
[2] Li Auto, China
[3] University of New South Wales, Australia
mumucc@mail.ustc.edu.cn, xyang21@ustc.edu.cn,
{zhangxuancheng,didonglin,mabaorui,lihao43}@lixiang.com

Abstract. End-to-end autonomous driving systems commonly follow a cascaded two-stage pipeline where a perception stage compresses multi-modal sensor inputs into a compact context and a downstream planner predicts trajectories conditioned on this context. We argue that this one-way perception-to-planning interface forces sensor inputs into a compact representation, losing the fine-grained details critical for planning. Moreover, by constraining the planner to this compressed context, it is difficult to leverage the rich representations offered by modern vision foundation models. To address these issues, we propose **MOJITO**, a unified sensor-to-action framework for end-to-end autonomous driving built on modal joint learning. MOJITO removes the cascaded interface and instead performs block-wise Modal Joint Attention that simultaneously updates action, image, and LiDAR features, allowing the planner to directly access multi-modal features during action generation. MOJITO achieves 88.9 PDMS on the NAVSIM v1 dataset and 88.4 EPDMS on the more challenging NAVSIM v2 dataset, setting a new state-of-the-art. Extensive experiments further demonstrate strong scalability, instruction following, and diverse trajectory generation. Code and models are available at https://github.com/mumucc01/MOJITO.

Keywords: Autonomous Driving · Multi-modal Joint Learning · End-to-End · Representation Learning

1 Introduction

Autonomous driving has witnessed significant progress with the evolution of deep learning [2, 6, 17, 30, 38]. A long-standing goal is to map sensory observations directly to driving outputs, motivating increasing interest in end-to-end models. Unlike traditional modular stacks, these approaches optimize perception and

* Work done during an internship at Li Auto.
† Equal contribution.
✉ Corresponding author.

planning jointly. A core challenge is bridging low-level perception with high-level decision-making. To address this, most methods adopt a cascaded two-stage pipeline. A perception stage extracts intermediate context, and a planning stage generates the action based on it. Typically, this relies on specific encoders like CNNs [7, 18, 30] to produce compact representations [19]. More recently, Vision-Language-Action (VLA) models [26, 39, 42, 55] have advanced this stage by leveraging the rich semantic reasoning of Vision-Language Models (VLMs). However, these advanced methods still compress vision-language information into coarse tokens for planning. While semantically enriched, this compression sacrifices the fine-grained geometric precision required for trajectory prediction and is processed by separate heads like MLPs [24] or diffusion models [21, 30].

This two-stage design limits the potential of end-to-end driving in three ways. First, as shown in Fig. 1 (a), it imposes a one-way "perception-to-planning" interface which brings an information bottleneck. This compression process discards fine-grained details, resulting in the planner lacking access to the sensor data for decision-making. Second, relying exclusively on latent context necessitates manual constraints, which require costly annotations and limit scalability. As the planner is conditioned on compressed representations, the supervision from the trajectory is too weak to guide the context representation learning. To compensate, methods require auxiliary tasks like 3D box prediction [7, 23, 53] or introduce predefined anchors [28, 30] to stabilize generation. Third, this structure creates a compatibility gap. Modern foundation models are built on standard Transformers [36, 45], yet existing methods restrict them to the role of perception encoders [55]. This prevents the planner from leveraging the full expressive power of Vision Transformers (ViTs). As a result, the rich representation ability from pre-trained weights is not fully utilized for decision-making.

Motivated by these limitations, we aim to build a direct "vision-to-action" framework that avoids the information bottleneck and heavy auxiliary supervision while maintaining compatibility with vision foundation models. To this end, we propose MOJITO (**MO**dal **J**o**I**nt learning for unified end-to-end au**TO**nomous driving). As illustrated in Fig. 1 (b), our framework consists of three parallel branches: an Image branch (ViT), a LiDAR branch following a ViT-style design that utilizes a PillarGroup tokenizer, and an Action Planner branch built on a Diffusion Transformer (DiT) [35]. Crucially, instead of isolating the planner behind a compressed latent context, we introduce **Modal Joint Attention**. This mechanism allows action tokens to interact with dense sensor tokens at each block, enabling the planner to directly attend to fine-grained visual details. This deep interaction naturally constrains the diffusion process to feasible regions, removing the need for hand-crafted anchors or auxiliary supervised tasks.

To validate our approach, we evaluate MOJITO on the **NAVSIM** benchmark. Without relying on predefined anchors or heavy auxiliary supervision, our unified architecture achieves **88.9 PDMS** on the NAVSIM-V1 *navtest* split, significantly outperforming recent strong E2E and VLA baselines such as DiffusionDrive [30] (88.1 PDMS) and ReCogDrive [26] (86.5 PDMS). Furthermore, MOJITO establishes a new state-of-the-art performance (**88.4 EPDMS**) on the

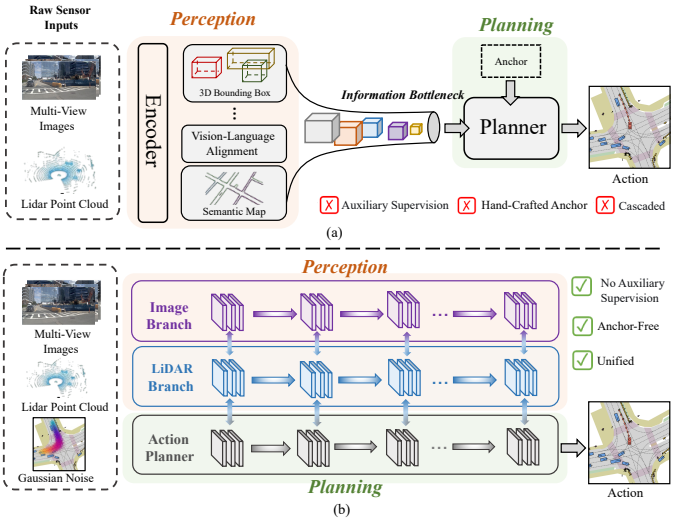

Fig. 1: This figure outlines the core differences between a cascaded two-stage framework and our unified E2E architecture. (a) shows that the "perception-to-planning" methods compress the fine-grained sensor information and suffer from the information bottleneck. (b) We propose a unified multimodal architecture comprising three parallel branches. Benefiting from the multi-modal fusion, MOJITO can generate feasible and stable actions without relying on predefined anchors or auxiliary supervision.

more challenging NAVSIM-V2 *navtest* split. Beyond standard metrics, extensive experiments also highlight MOJITO's scalability, instruction following, and diverse trajectory generation capabilities. These results demonstrate that directly bridging raw multi-modal sensor inputs with the action planner not only simplifies the architectural pipeline but also effectively improves planning quality and driving stability in complex scenarios.

The main contributions of this work are as follows:

- We propose MOJITO, a fully unified end-to-end driving framework that integrates multi-modal sensors with a diffusion-based planner via a novel Modal Joint Attention mechanism. This design resolves the information bottleneck caused by cascaded pipelines by allowing action tokens to interact deeply with dense sensor features at each block.
- We develop a simplified, anchor-free architecture that removes the dependency on auxiliary supervision and predefined anchors. By following standard ViT and DiT design protocols, our approach maintains full compatibility with foundation models, allowing the planner to utilize pre-trained representations directly for decision-making.
- We achieve leading performance on both NAVSIM-v1 and v2 benchmarks, surpassing strong existing baselines. Furthermore, extensive experiments demonstrate MOJITO's robust capabilities in scalability, instruction following, and diverse trajectory generation.

2 Related Work

2.1 Architectures for Autonomous Driving

Recent years have witnessed significant progress in end-to-end autonomous driving, moving from modular pipelines to integrated systems [29, 30, 33, 37]. Pioneering frameworks like UniAD [18] defined the standard E2E paradigm, where perception features are extracted and then fed into a planning head. Following this, works like SparseDrive [38] and HydraMDP [27] refined feature representations but maintained this two-stage design. Subsequently, generative approaches such as DiffusionDrive [30] and GTRS [28] replaced conventional planners with diffusion models, but still relied on perception features as conditions and predefined trajectory anchors for stable generation. In parallel, recent advances in multimodal learning have demonstrated the potential of MLLMs for visual language understanding, semantic reasoning, and robust visual representation learning [44, 47–49]. Inspired by these advances, Vision-Language-Action (VLA) methods [11, 21, 24] enhance perception through semantic reasoning, yet continue to pass compressed embeddings to a separate planner. Overall, whether utilizing traditional MLPs, diffusion processes, or VLA architectures, the cascaded design enforces a one-way information flow. This structural separation limits the mutual interaction between scene understanding and action planning, motivating our fully unified, tokenized approach.

2.2 Diffusion Models for Planning

Diffusion models have shown strong capability in generating diverse and high-quality samples across various domains [5,31,41], motivating researchers to transfer this strength to autonomous driving to improve trajectory diversity and enable capabilities like instruction following and driving style adaptation [30, 40, 43, 46]. Motion Diffuser [22] represents an early effort in applying diffusion models to action generation, learning a joint distribution of multi-agent trajectories and enabling diverse future predictions without predefined anchors. Diffusion Planner [53] extends diffusion modeling to closed-loop planning, producing diverse ego trajectories and supporting joint prediction-planning tasks. Another approach, Diffusion ES [46], validated the importance of trajectory diversity by achieving instruction-following behaviors through optimization with black-box rewards, even generating novel actions not present in the training data. Despite these advancements, most diffusion-based planners still rely on vectorized environmental representations and cannot directly process sensor data, which restricts their integration into end-to-end driving.

2.3 Unified Multimodal Architectures

Unified multimodal architectures aim to jointly model diverse modalities and tasks within a single framework [13]. Existing methods [1, 10] achieve unification by sharing self-attention layers between understanding and generation experts.

In autonomous driving, recent works like UniUGP [33] propose a hybrid expert architecture to synergize scene understanding, future video generation, and trajectory planning. They essentially unify multiple generative branches (e.g., video generation), our method instead couples discriminative perception (DINOv3, Uni3D) with generative planning (DiT). Unlike existing pipelines [20], our method learns optimal continuous action representations within the sparse perceptual space, with trajectory supervision only.

3 Problem Formulation and Challenges

3.1 Task Definition

The goal of end-to-end autonomous driving is to generate a sequence of future waypoints for the ego vehicle directly from raw sensor inputs. At each planning step, the model receives multi-view camera images $\mathcal{I} = \{I_1, I_2, \ldots, I_N\}$ and a LiDAR point cloud $\mathcal{L}$, capturing the semantic and geometric context of the scene. The output is a trajectory represented as a sequence of waypoints $\tau = \{w_t\}_{t=1}^{T_f}$, where each waypoint $w_t = [x_t, y_t, \theta_t]$ contains the 2D position and heading angle of the ego vehicle over a planning horizon T_f (in our case, $T_f = 8$).

The task is defined as learning a mapping $f_\theta : (\mathcal{I}, \mathcal{L}) \rightarrow \tau$ that predicts smooth and feasible trajectories from raw visual inputs. In our framework, this mapping is realized through a diffusion process, enabling flexible, data-driven trajectory generation without hand-crafted anchors or auxiliary supervised tasks.

3.2 Challenges in End-to-End Planning

While recent end-to-end driving models have shown promising results, unifying multi-modal perception and trajectory planning remains difficult due to two primary challenges:

Challenge 1: Unifying Perception and Planning Architectures. Most existing end-to-end frameworks [7, 30] treat perception and planning as isolated stages connected by a bottleneck. They typically employ CNN-based backbones (like ResNet34) to extract a compact "context feature" map, which is then passed to a separate planning module as a condition. This paradigm prevents the fine-grained feature-level interaction between the planned trajectory and the perceived environment. The planner only passively receives the compressed context, rather than actively querying the sensor information during the action planning. A unified architecture that allows bidirectional information flow between perception and planning is required.

Challenge 2: Preserving Multi-Modal Information. Autonomous driving relies on heterogeneous data: images provide dense semantic information, LiDAR provides sparse but accurate 3D geometry [12], and trajectories are 2D sequential coordinates. Previous methods often force these modalities into a single format, such as projecting 3D point clouds into 2D Bird's-Eye-View (BEV) images. This rough transformation discards fine-grained geometric details and

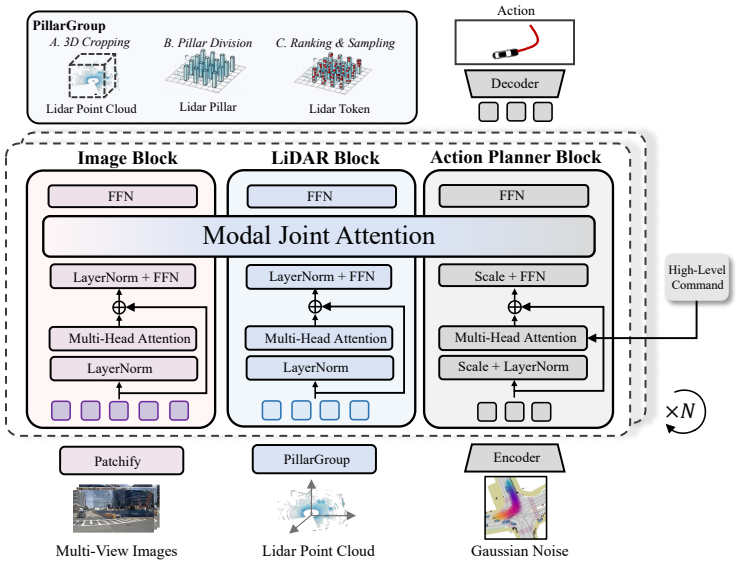

Fig. 2: MOJITO Architecture. To avoid the information bottleneck inherent in cascaded frameworks, we propose MOJITO, a unified multi-modal framework with three branches: Image, LiDAR, and the Action Planner branch. To preserve the 3D spatial understanding of the driving scene, we introduce a BEV Grid Patchify module named PillarGroup. The Modal Joint Attention mechanism enables the deep interaction between the planned trajectory and the perceived environment, achieving a bidirectional flow of rich geometric and semantic information during the generation process.

height information. Meanwhile, point-based methods like Farthest Point Sampling (FPS) [14] and K-Nearest Neighbors (KNN) [16, 52], commonly used in shape-level tasks, struggle in large-scale driving scenes because they lack absolute physical scale. Moreover, they struggle with uneven LiDAR density, splitting dense objects across multiple tokens while merging sparse ones. This instability degrades multi-modal alignment. A tokenization strategy that preserves the absolute metric scale of LiDAR is essential for effective multi-modal fusion.

4 Method

To address the aforementioned challenges, we propose a fully unified Transformer-based architecture that integrates multi-view images, LiDAR point clouds, and action planning into a single framework.

4.1 Overall Architecture

As illustrated in Fig. 2, our model consists of three parallel branches: the Image Branch, the LiDAR Branch, and the Action Planner. These three branches adopt

structurally identical but independent Transformer architectures, each comprising 12 blocks with a hidden state dimension of 384.

Instead of extracting perception features first and passing them to the planner, our architecture processes all modalities simultaneously. The core of this design is the **Modal Joint Attention** mechanism. In each block, tokens from the image, LiDAR, and action planner branches are concatenated and processed through a shared self-attention layer. This allows every modality to directly interact with the others at every depth of the network. Moreover, to maintain spatial and sequential awareness across this mixed-modality sequence, a learnable positional embedding is dynamically generated and added within each layer.

4.2 Perception Branches

To enable the Modal Joint Attention, we first convert the raw sensor inputs into sequences of tokens with a unified dimension using dedicated perception branches.

Image Branch: The image feature is processed using a DINOv3-S+ architecture. For the initial tokenization, each multi-view image is divided into 16×16 patches. These patches are linearly projected into image tokens before being fed into the Transformer blocks to extract dense semantic features.

LiDAR Branch: For the LiDAR branch, we adopt the Uni3D-S architecture. Uni3D was originally designed for shape-level representation, relying on Farthest Point Sampling (FPS) and K-Nearest Neighbors (KNN) for tokenization. We observe that this tokenization design does not align well with the requirements of autonomous driving (AD) scenarios. FPS and KNN operate on relative spatial distances, discarding the absolute metric scale and global geometric layout required for scene-level navigation.

Inspired by grid-based BEV representations [19, 29], we introduce a BEV Grid Patchify module named **PillarGroup** for LiDAR tokenization. PillarGroup crops the 3D point cloud space and divides the space into a fixed grid (e.g., 32×32 pillars), covering a specific metric range. Points are assigned to their corresponding physical pillars. We select the top-K non-empty pillars ($K = 512$, padded if necessary) and sample a fixed number N of points per pillar ($N = 64$). Because each pillar covers a deterministic physical area, the generated tokens possess consistent semantic information and absolute physical coordinates.

4.3 Action Planner

The action planner generates future trajectories through a diffusion process parameterized directly within our unified Transformer architecture. Unlike traditional pipelines that extract perception features first and apply them as conditions, our approach treats action waypoints as native tokens that evolve jointly with perception tokens.

Formally, the ground-truth future trajectory is represented as:

$$\tau^{(0)} = [x_{\text{ego}}(1), x_{\text{ego}}(2), \ldots, x_{\text{ego}}(T)], \tag{1}$$

where each state $x_{\text{ego}}(t)$ encodes the ego vehicle's position and heading as $x_{\text{ego}}(t) = [x_t, y_t, \theta_t]$, and T is the predicted steps. During inference, the entire trajectory sequence is initialized as pure Gaussian noise. A forward process progressively corrupts the ground-truth states with noise, yielding a sequence of noisy trajectories $\tau^{(k)}$ at diffusion step k.

During training, our unified network ϵ_θ learns to recover the data distribution from noisy data with the following objective:

$$\mathcal{L}_\theta = \mathbb{E}_{x^{(0)},k\sim U(0,1),x^k\sim q_{k0}(x^{(k)}|x^{(0)})}\left[\|\epsilon_\theta(x^{(k)},k,C)-x^{(0)}\|^2\right]. \tag{2}$$

The diffusion step k is embedded via an MLP, combined with the high-level command (e.g., turn left and go straight), and injected into the action tokens using Adaptive Layer Normalization (adaLN). C consists of the perception tokens and the high-level command. The perception features and the action tokens are fused naturally through our modal joint attention mechanism.

4.4 Modal Joint Attention

To enable interaction across modalities, we introduce the Modal Joint Attention mechanism within each Transformer block. Let $X_I \in \mathbb{R}^{N_I\times D}$, $X_L \in \mathbb{R}^{N_L\times D}$, and $X_A \in \mathbb{R}^{N_A\times D}$ denote the token sequences from the image, LiDAR, and action branches, respectively, where $D = 384$ is the hidden dimension.

Within each block, we concatenate the tokens along the sequence dimension to form a unified representation:

$$X_{joint} = [X_I \parallel X_L \parallel X_A] \in \mathbb{R}^{(N_I+N_L+N_A)\times D}. \tag{3}$$

We add learnable positional embeddings to X_{joint} to retain spatial and sequential context. The joint sequence is then processed through a shared Multi-Head Self-Attention (MHSA) layer:

$$Y_{joint} = \text{MHSA}(\text{LN}(X_{joint}+PE)) + X_{joint}, \tag{4}$$

where the PE denotes the position embeddings, and the LN denotes Layer Normalization. This operation allows the action tokens to query geometric and semantic features directly from the perception tokens, while the perception branches adjust their representations based on the planning context. Finally, Y_{joint} is split back into the updated tokens Y_I, Y_L, and Y_A, which are passed to their respective Feed-Forward Networks (FFN).

This design embodies our core modality fusion philosophy. It preserves the distinct functional roles of perception and action experts without introducing task interference, while enabling effective cross-modal feature fusion. By projecting all modalities into a shared representational space, diverse pretrained knowledge from the vision and 3D domains naturally complements the action generation. During the self-attention operation, action tokens directly query the fine-grained geometric and semantic information from perception tokens, while the perception features adaptively update based on the planning intent. Finally, the updated joint sequence is split back into X_I, X_L, and X_A as the inputs of the subsequent block.

5 Experiment

5.1 Dataset

NAVSIM. NAVSIM [9] is a large-scale dataset for trajectory planning, based on OpenScene [8] and nuPlan [3]. It provides real-world driving sequences focused on decision-heavy scenarios, featuring synchronized 360° camera and LiDAR data sampled at 2 Hz. We evaluate planning in a non-reactive closed-loop setting, where the ego trajectory is simulated without environment feedback.

The original NAVSIM-v1 evaluates planners using the **Predictive Driver Model Score (PDMS)** [9], which combines metrics like no at-fault collisions (NC), drivable area compliance (DAC), time-to-collision (TTC), comfort, and ego progress (EP). The NAVSIM-v2 update introduces a stricter evaluation with the **Extended PDMS (EPDMS)**, adding finer compliance checks such as Driving Direction Compliance (DDC), Traffic Light Compliance (TLC), Lane Keeping (LK), History Comfort (HC), and Extended Comfort (EC). NAVSIM-v2 additionally introduces the ***navhard*** split, which provides synthetic futures and stresses robustness in more challenging scenarios. Moreover, NAVSIM-v2 further defines a **two-stage** protocol: Stage-1 evaluates the planned trajectory on logged sequences, while Stage-2 leverages the synthetic follow-up observations to evaluate the planner under a stronger setting.

5.2 Implementation Details

We adopt a DINOv3-S+ [36] as our image branch and a Uni3D-S [54] as our LiDAR branch. We initialize both backbones from their originally released weights and do not perform extra pretraining on the autonomous-driving data. The action branch is modeled as an anchor-free diffusion process in the noise space, implemented with a Diffusion Transformer [35]. During training, the image and LiDAR branches are fine-tuned, while the action planner is trained from scratch.

During training and inference, we use three downscaled camera views (front, 60° left, 60° right), which are concatenated to a resolution of 1024×256, and the raw LiDAR points as input. We follow Transfuser [7] for image pre-processing. We train MOJITO on the *navtrain* using AdamW on 8 NVIDIA H200 GPUs with a total batch size of 512 and a learning rate of 6×10^{-4}. For both NAVSIM-v1 and v2, we predict an 8-waypoint trajectory over 4 seconds for evaluation.

5.3 Quantitative Comparison on NAVSIM-v1

To evaluate our proposed method, we compare it against two distinct categories of autonomous driving frameworks on the NAVSIM-v1 dataset: End-to-End methods and Vision-Language-Action (VLA) models.

Comparison with End-to-End Methods. Table 1 presents the comparison with representative end-to-end methods. Overall, our method achieves a leading PDMS score of 88.9, outperforming recent baselines such as DiffusionDrive (88.1 PDMS) and WoTE (88.3 PDMS). Many high-performing methods heavily rely on predefined anchors to constrain the planning space, such as VADv2

Table 1: Quantitative comparison with end-to-end planning methods on the planning-oriented NAVSIM-v1 *navtest* split. "Input" indicates sensor modalities, where "C & L" denotes using both camera and LiDAR. "Anchor" denotes the number of anchors used by anchor-based planners (0 for anchor-free). The best results are marked in bold.

Method	Input	Anchor	NC ↑	DAC ↑	TTC↑	Comf. ↑	EP ↑	PDMS↑
Constant Velocity	-	-	68.0	57.8	50.0	**100**	19.4	20.6
Ego Status MLP	-	-	93.0	77.3	83.6	**100**	62.8	65.6
UniAD [18]	Camera	0	97.8	91.9	92.9	**100**	78.8	83.4
Transfuser [7]	C & L	0	97.7	92.8	92.8	**100**	79.2	84.0
DRAMA [51]	C & L	0	98.0	93.1	94.8	**100**	80.1	85.5
VADv2 [4]	C & L	8192	97.2	89.1	91.6	**100**	76.0	80.9
Hydra-MDP [27]	C & L	8192	98.3	96.0	94.6	**100**	78.7	86.5
DiffusionDrive [30]	C & L	20	98.2	96.2	94.7	**100**	82.2	88.1
WoTE [25]	C & L	256	98.5	96.8	**94.9**	**100**	82.6	88.3
MOJITO (Ours)	C & L	0	**98.6**	**96.9**	94.5	**100**	**83.5**	**88.9**

Table 2: Quantitative comparison with VLA-based methods on NAVSIM-v1 (*navtest*) using non-reactive closed-loop evaluation. "VLM" denotes the vision-language model backbone used by each method, "RL" indicates whether reinforcement learning fine-tuning is applied, and "Model Size" reports the parameter count.

Method	VLM	RL	Model Size	PDMS ↑
ReCogDrive-Base-IL [26]	InternVL3		2B	86.5
ReCogDrive-Base-RL [26]	InternVL3	✓	2B	**90.8**
ReCogDrive-Large-IL [26]	InternVL3		8B	86.5
ReCogDrive-Large-RL [26]	InternVL3	✓	8B	90.4
AutoVLA-IL [55]	Qwen2.5-VL		3B	80.5
AutoVLA-RL [55]	Qwen2.5-VL	✓	3B	89.1
AdaThinkDrive-IL [34]	InternVL3		8B	86.2
AdaThinkDrive-RL [34]	InternVL3	✓	8B	90.3
MOJITO (Ours)	None		127M	88.9

and Hydra-MDP (8,192 anchors), WoTE (256 anchors), and DiffusionDrive (20 anchors). In contrast, our method completely eliminates this requirement. Despite using zero anchors, our approach surpasses DiffusionDrive, VADv2, and Hydra-MDP. This indicates that our architecture can generate superior and stable actions without being restricted by predefined anchors.

Comparison with VLA-based Methods. Recently, VLA models have introduced large foundation models to autonomous driving. As shown in Table 2, we compare with representative VLA frameworks built upon large foundation models like InternVL3 and Qwen2.5-VL. A notable characteristic of these VLA methods is their massive model sizes, ranging from 2B to 8B parameters. To achieve competitive driving performance, they heavily rely on Reinforcement Learning (RL) techniques. For instance, ReCogDrive-Base requires RL to improve its PDMS from 86.5 to 90.8.

Table 3: Quantitative comparison on NAVSIM-v2 *navtest* under non-reactive closed-loop evaluation. NAVSIM-v2 reports extended compliance metrics (e.g., DDC, TLC, LK, HC, and EC) in addition to the v1 metrics, and uses EPDMS as the overall score.

Method	NC ↑	DAC ↑	DDC ↑	TLC ↑	EP ↑	TTC ↑	LK ↑	HC ↑	EC ↑	EPDMS ↑
Ego Status MLP	93.1	77.9	92.7	99.6	86.0	91.5	89.4	98.3	85.4	64.0
Transfuser [7]	96.9	89.9	97.8	99.7	87.1	95.4	92.7	98.3	87.2	76.7
Hydra-MDP++ [27]	97.2	**97.5**	99.4	99.6	83.1	96.5	94.4	98.2	70.9	81.4
DriveSuprim [50]	97.5	96.5	99.4	99.6	88.4	96.6	95.5	98.3	77.0	83.1
ARTEMIS [15]	**98.3**	95.1	98.6	**99.8**	81.5	**97.4**	96.5	98.3	-	83.1
DiffusionDriveV2 [56]	97.7	96.6	99.2	**99.8**	**88.9**	97.2	96.0	97.8	**91.0**	85.5
MOJITO (Ours)	98.2	96.2	**99.5**	**99.8**	87.8	**97.4**	**97.3**	**98.4**	87.8	**88.4**

Table 4: Results on NAVSIM-v2 *navhard* under the two-stage evaluation protocol. We report Stage I and Stage II metrics and the overall EPDMS. Since *navhard* does not provide LiDAR for Stage II, we evaluate MOJITO under a camera-only setting.

Method	Stage	NC↑	DAC↑	DDC↑	TLC↑	EP↑	TTC↑	LK↑	HC↑	EC↑	EPDMS↑
LTF [7]	Stage I	96.2	79.5	99.1	99.5	83.1	95.1	94.2	97.5	79.1	23.1
	Stage II	77.7	70.2	84.2	98.0	85.1	75.6	45.4	95.7	75.9	
DiffusionDrive [30]	Stage I	96.0	79.7	97.4	99.5	81.3	93.1	90.8	96.8	73.8	24.2
	Stage II	82.1	72.2	88.5	98.7	85.1	78.8	49.2	89.3	71.2	
GTRS-DP [28]	Stage I	94.7	78.8	96.1	99.5	83.0	94.4	92.0	97.5	72.8	23.8
	Stage II	80.3	74.4	84.9	98.0	81.9	78.8	45.4	96.7	70.1	
GuideFlow [32]	Stage I	96.6	80.5	96.3	99.3	82.3	94.9	91.5	97.7	67.8	27.1
	Stage II	87.3	76.7	88.8	99.2	84.3	85.1	49.7	93.1	44.5	
MOJITO Camera-Only	Stage I	96.2	82.4	98.1	99.6	84.1	94.7	94.2	97.6	80.0	**29.0**
	Stage II	78.0	69.4	85.0	97.9	85.5	75.4	48.5	96.0	71.6	

In contrast, our method operates with a compact model size of 127M parameters. Without utilizing Vision-Language Models or complex Reinforcement Learning pipelines, our approach achieves a PDMS of 88.9, achieving a competitive score compared to VLM-RL-based methods. This demonstrates the exceptional parameter efficiency and learning capability of our proposed architecture.

5.4 Quantitative Comparison on NAVSIM-v2

We further test our method under the stricter NAVSIM-v2 protocol using the same *navtest* split. Because the *navtest* split lacks synthetic follow-up observations required for Stage-2 causal evaluation, we focus our comparison on the Stage-1 overall score (EPDMS). Table 3 summarizes the results. Our method achieves the best EPDMS of 88.4, outperforming DiffusionDriveV2 (85.5), ARTEMIS (83.1), and DriveSuprim (83.1). It also performs strongly on key compliance metrics, with 99.5 DDC, 97.3 LK, and 98.4 HC.

Fig. 3: This figure presents a qualitative comparison between our method and existing methods on NAVSIM-v1. Our proposed MOJITO achieves more stable and reliable decision-making in complex scenarios. The green line denotes the ground-truth trajectory, while the red line represents the trajectory generated by DiffusionDrive, ReCogDrive, and MOJITO, respectively.

We additionally report results on NAVSIM-v2 *navhard*, which supports both Stage-1 and Stage-2 evaluation. As *navhard* does not provide LiDAR for Stage-2, we remove the LiDAR branch of our model and evaluate under a camera-only setting. This causes a performance drop compared to using both modalities, but the results remain strong overall. As shown in Table 4, our method reaches 29.0 EPDMS, exceeding GuideFlow (27.1), DiffusionDrive (24.2), and LTF (23.1). Stage-2 is consistently harder than Stage-1 for all methods under this setting.

5.5 Qualitative Comparison on NAVSIM-v1

To demonstrate the superiority of our proposed MOJITO, we conduct a qualitative comparison on NAVSIM-v1, as shown in Fig. 3. The first row illustrates a challenging roundabout scenario with multiple drivable routes. MOJITO generates a precise, smooth, and stable trajectory. The second row depicts a road bifurcation with slight directional differences, and MOJITO successfully distinguishes the correct lane. The third row shows that other methods tend to produce overly aggressive turning behaviors, potentially driving into the undrivable area. MOJITO achieves a stronger understanding of scene geometry and depth, enabling it to generate more stable and reliable trajectories.

5.6 Ablation Study

In this section, we conduct ablation experiments on the NAVSIM-v1 navtest split, as reported in Table 5. We vary the input sensor modalities, the LiDAR tokenization strategy, and the attention type in our Modal Joint Attention blocks.

Table 5: Ablations on sensor modalities, LiDAR tokenization, and the attention formulation on NAVSIM-v1 *navtest*.

ID	Image	LiDAR	Sample	Attn.	NC↑	DAC↑	TTC ↑	Comf.↑	EP ↑	PDMS ↑
1	✓	✗	✗	Self-Attn.	98.0	95.3	93.6	**100**	81.6	86.8
2	✓	✓	FPS+KNN	Self-Attn.	97.8	94.8	93.3	**100**	81.0	86.1
3	✓	✓	Pillar	Cross-Attn.	97.8	94.4	93.1	**100**	80.7	85.7
4	✓	✓	Pillar	Self-Attn.	**98.6**	**96.9**	**94.5**	**100**	**83.5**	**88.9**

Table 6: Scaling study on NAVSIM-v1 (*navtest*). We increase the aligned blocks in the vision, LiDAR, and action planner branches, initializing the perception branches from the corresponding pretrained blocks. The best results are marked in bold.

ID	Blocks	Model Size	NC↑	DAC↑	TTC↑	Comf.↑	EP ↑	PDMS ↑
1	3	33.4M	96.6	90.4	90.7	**100**	76.1	80.5
2	6	64.4M	96.7	90.2	90.7	**100**	76.1	80.5
3	9	95.5M	97.1	92.3	91.8	**100**	78.4	83.0
4	12	127M	**98.6**	**96.9**	**94.5**	**100**	**83.5**	**88.9**

These ablations provide insights into the effectiveness of our unified architecture and the necessity of each design choice.

ID 1 evaluates a camera-only variant of our model, which still achieves a competitive PDMS of 86.8, indicating that directly fusing trajectory tokens with image tokens provides strong planning guidance. In ID 2, we incorporate LiDAR inputs using an FPS+KNN tokenization strategy, which is commonly adopted in shape-level point cloud representation learning. This tokenization does not improve performance, and we attribute this degradation to the incompatibility with large-scale autonomous driving scenes.

In ID 3, we prevent the sensor tokens from attending to the action tokens by modifying the attention map within the Joint Modal Attention. Cross-attention allows action tokens to understand sensor inputs, while the sensor tokens do not adapt to the current denoising state. Our self-attention mechanism fuses image, LiDAR, and action tokens simultaneously, enabling the bidirectional interaction between perception and action planning.

5.7 Scalability Study

Table 6 studies how performance scales with model depth. The "Blocks" denote aligned blocks that are increased jointly in the image, LiDAR, and action planner branches. For the image and LiDAR branches, we load the corresponding pre-trained weights for the selected number of blocks and train the model in an end-to-end manner. Even the smallest model variant achieves reasonable performance, and scaling up the number of blocks consistently improves PDMS. This trend indicates that our architecture is easy to scale in model size and can benefit from larger capacity, suggesting further gains when trained on larger-scale data. Moreover, we also train the action planner from scratch for a fairer comparison, and the scaling trend remains consistent.

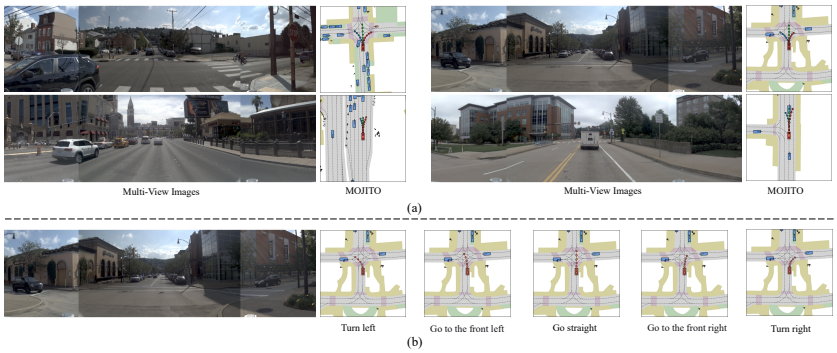

Fig. 4: This figure demonstrates the instruction-following and diverse trajectory generation capability of MOJITO. (a) The blue, green, and red lines represent the trajectories generated for the instructions "turn left", "go straight", and "turn right", respectively. (b) shows our ability to follow more complex instructions. Benefiting from our diffusion-based planner, our method can generate trajectories with varying turning angles, further validating the diversity of its trajectory generation.

5.8 Instruction-Following and Diverse Trajectory Generation

VLM-based driving methods often emphasize instruction-following through language understanding [46, 55]. MOJITO does not rely on a VLM backbone, but it supports instruction-following via a high-level command interface. In our examples, we construct a command vocabulary and use Qwen3-1.7B [45] to map language instructions to the closest commands, which are then fed into the model as the conditions. As shown in Fig. 4 (a), MOJITO accurately follows the specified command and produces lane-consistent trajectories toward the intended direction. The second row in Fig. 4 (a) shows that in the straight-lane scenario, even with left or right turn commands, MOJITO does not generate unreasonable sharp turns. As a diffusion-based planner, MOJITO enables diverse and smooth trajectory generation in a continuous noise space. As shown in Fig. 4 (b), as the turning angle implied by the instruction gradually varies, MOJITO consistently generates feasible and stable trajectories.

6 Cross Modal Learning

To enable the planning branch to effectively exploit visual semantics, we introduce the Modal Joint Attention that facilitates direct interaction between the action planner and the image branch. To better understand this interaction, we visualize the attention maps derived from the Modal Joint Attention across different transformer blocks. As shown in Fig. 5, the first block focuses on the forward drivable area; the middle blocks shift to large foreground objects and obstacles on both road sides; and the last blocks exhibit fine-grained attention to both the drivable area and roadside obstacles. This attention pattern validates that the planner dynamically adapts its visual attention to the driving context, contributing to more stable and precise trajectory planning.

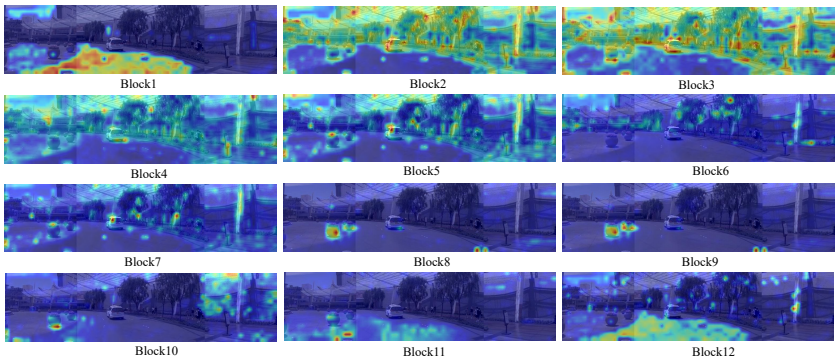

Fig. 5: Visualization of the trajectory-to-image attention maps. The attention weights are averaged over the 8 future waypoints to produce a holistic view of how the planning branch attends to the multi-view image patches.

7 Inference Latency

In this section, we measure per-sample latency on a single H200 GPU using BF16 precision. MOJITO employs 2 diffusion steps and achieves an inference latency of 187.65 ms, while Transfuser [7], DiffusionDrive [30], and ReCogDrive [26] have latencies of 88.15 ms, 123.22 ms, and 331.68 ms, respectively. Although our inference speed is slightly slower than methods such as Transfuser and DiffusionDrive, MOJITO significantly outperforms them on the NAVSIM-v1 and NAVSIM-v2 datasets. In practical autonomous driving scenarios, the safety and stability of driving decisions are of critical importance. Moreover, compared to VLA-based methods like ReCogDrive, MOJITO achieves much faster inference while maintaining superior performance. These results demonstrate that our method strikes a good balance between inference efficiency and trajectory accuracy.

8 Conclusion

In this work, we presented **MOJITO**, a unified end-to-end driving framework that avoids the information bottleneck induced by the cascaded pipelines. MOJITO adopts three parallel Transformer branches for image, LiDAR, and action planning, and introduces a block-wise Modal Joint Attention to fuse all modalities simultaneously. The action planner branch can leverage fine-grained sensor inputs without auxiliary supervision or predefined trajectory anchors. Extensive experiments on NAVSIM validate the effectiveness of our design, achieving 88.9 PDMS on NAVSIM-v1 *navtest* and 88.4 EPDMS on NAVSIM-v2 *navtest*.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (NSFC) under Grant U22A2094 and Grant 62272435.

References

1. Bi, H., Tan, H., Xie, S., Wang, Z., Huang, S., Liu, H., Zhao, R., Feng, Y., Xiang, C., Rong, Y., et al.: Motus: A Unified Latent Action World Model. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 35101–35113 (2026)
2. Bojarski, M., Del Testa, D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., Jackel, L.D., Monfort, M., Muller, U., Zhang, J., et al.: End to End Learning for Self-Driving Cars. arXiv preprint arXiv:1604.07316 (2016)
3. Caesar, H., Kabzan, J., Tan, K.S., Fong, W.K., Wolff, E., Lang, A., Fletcher, L., Beijbom, O., Omari, S.: NuPlan: A closed-loop ML-based planning benchmark for autonomous vehicles. arXiv preprint arXiv:2106.11810 (2021)
4. Chen, S., Jiang, B., Gao, H., Liao, B., Xu, Q., Zhang, Q., Huang, C., Liu, W., Wang, X.: VaDv2: End-to-End Vectorized Autonomous Driving via Probabilistic Planning. arXiv preprint arXiv:2402.13243 (2024)
5. Cheng, Z., Zhang, X., Di, D., Wei, C., Li, H., Yang, X.: MoCa: Modeling Object Consistency for 3D Camera Control in Video Generation. In: The Fourteenth International Conference on Learning Representations
6. Chi, H., Qiu, D., Su, H., Liu, H., Li, Z., Zhang, H., Lv, C.: Driver-WM: A Driver-Centric Traffic-Conditioned Latent World Model for In-Cabin Dynamics Rollout. arXiv preprint arXiv:2605.05092 (2026)
7. Chitta, K., Prakash, A., Jaeger, B., Yu, Z., Renz, K., Geiger, A.: TransFuser: Imitation with Transformer-Based Sensor Fusion for Autonomous Driving. IEEE Transactions on Pattern Analysis and Machine Intelligence **45**(11), 12878–12895 (2022)
8. Contributors, O.: OpenScene: The Largest Up-to-Date 3D Occupancy Prediction Benchmark in Autonomous Driving. In: Proceedings of the Conference on Computer Vision and Pattern Recognition, Vancouver, Canada. pp. 18–22 (2023)
9. Dauner, D., Hallgarten, M., Li, T., Weng, X., Huang, Z., Yang, Z., Li, H., Gilitschenski, I., Ivanovic, B., Pavone, M., Geiger, A., Chitta, K.: NAVSIM: Data-Driven Non-Reactive Autonomous Vehicle Simulation and Benchmarking. In: Advances in Neural Information Processing Systems (2024)
10. Deng, C., Zhu, D., Li, K., Gou, C., Li, F., Wang, Z., Zhong, S., Yu, W., Nie, X., Song, Z., Shi, G., Fan, H.: Emerging Properties in Unified Multimodal Pretraining. arXiv preprint arXiv:2505.14683 (2025)
11. Deng, T., Chen, X., Chen, Y., Chen, Q., Xu, Y., Yang, L., Xu, L., Zhang, Y., Zhang, B., Huang, W., et al.: GaussianDWM: 3D Gaussian Driving World Model for Unified Scene Understanding and Multi-Modal Generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10656–10667 (2026)
12. Deng, T., Pan, Y., Yuan, S., Li, D., Wang, C., Li, M., Chen, L., Xie, L., Wang, D., Wang, J., et al.: What Is The Best 3D Scene Representation for Robotics? From Geometric to Foundation Models. arXiv preprint arXiv:2512.03422 (2025)
13. Du, S., Zou, Y., Wang, Z., Li, X., Li, Y., Shang, C., Shen, Q.: Unsupervised Hyperspectral Image Super-Resolution via Self-Supervised Modality Decoupling. International Journal of Computer Vision **134**(4), 152 (2026)
14. Eldar, Y., Lindenbaum, M., Porat, M., Zeevi, Y.Y.: The Farthest Point Strategy for Progressive Image Sampling. IEEE Transactions on Image Processing **6**(9), 1305–1315 (1997)

15. Feng, R., Xi, N., Chu, D., Wang, R., Deng, Z., Wang, A., Lu, L., Wang, J., Huang, Y.: ARTEMIS: Autoregressive End-to-End Trajectory Planning with Mixture of Experts for Autonomous Driving. IEEE Robotics and Automation Letters **11**(1), 226–233 (2025)
16. Guo, G., Wang, H., Bell, D., Bi, Y., Greer, K.: KNN Model-based Approach in Classification. In: OTM Confederated International Conferences" On the Move to Meaningful Internet Systems". pp. 986–996. Springer (2003)
17. Hu, S., Chen, L., Wu, P., Li, H., Yan, J., Tao, D.: ST-P3: End-to-end Vision-based Autonomous Driving via Spatial-Temporal Feature Learning. In: European Conference on Computer Vision. pp. 533–549. Springer (2022)
18. 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)
19. Huang, J., Huang, G.: BEVDet4D: Exploit Temporal Cues in Multi-camera 3D Object Detection. arXiv preprint arXiv:2203.17054 (2022)
20. Jia, X., You, J., Zhang, Z., Yan, J.: DriveTransformer: Unified Transformer for Scalable End-to-End Autonomous Driving. arXiv preprint arXiv:2503.07656 (2025)
21. Jiang, A., Gao, Y., Sun, Z., Wang, Y., Wang, J., Chai, J., Cao, Q., Heng, Y., Jiang, H., Dong, Y., et al.: DiffVLA: Vision-Language Guided Diffusion Planning for Autonomous Driving. arXiv preprint arXiv:2505.19381 (2025)
22. Jiang, C., Cornman, A., Park, C., Sapp, B., Zhou, Y., Anguelov, D., et al.: MotionDiffuser: Controllable Multi-Agent Motion Prediction using Diffusion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9644–9653 (2023)
23. Li, B., Ouyang, W., Sheng, L., Zeng, X., Wang, X.: GS3D: An Efficient 3D Object Detection Framework for Autonomous Driving. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1019–1028 (2019)
24. 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. arXiv preprint arXiv:2512.10719 **2** (2025)
25. Li, Y., Wang, Y., Liu, Y., He, J., Fan, L., Zhang, Z.: End-to-End Driving with Online Trajectory Evaluation via Bev World Model. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 27137–27146 (2025)
26. Li, Y., Xiong, K., Guo, X., Li, F., Yan, S., Xu, G., Zhou, L., Chen, L., Sun, H., Wang, B., et al.: ReCogDrive: A Reinforced Cognitive Framework for End-to-End Autonomous Driving. arXiv preprint arXiv:2506.08052 (2025)
27. Li, Z., Li, K., Wang, S., Lan, S., Yu, Z., Ji, Y., Li, Z., Zhu, Z., Kautz, J., Wu, Z., et al.: Hydra-MDP: End-to-end Multimodal Planning with Multi-target Hydra-Distillation. arXiv preprint arXiv:2406.06978 (2024)
28. Li, Z., Yao, W., Wang, Z., Sun, X., Chen, J., Chang, N., Shen, M., Wu, Z., Lan, S., Alvarez, J.M.: Generalized Trajectory Scoring for End-to-end Multimodal Planning. arXiv preprint arXiv:2506.06664 (2025)
29. Li, Z., Wang, W., Li, H., Xie, E., Sima, C., Lu, T., Yu, Q., Dai, J.: BEVFormer: Learning Bird's-Eye-View Representation from Lidar-Camera via Spatiotemporal Transformers. IEEE Transactions on Pattern Analysis and Machine Intelligence (2024)
30. Liao, B., Chen, S., Yin, H., Jiang, B., Wang, C., Yan, S., Zhang, X., Li, X., Zhang, Y., Zhang, Q., et al.: DiffusionDrive: Truncated Diffusion Model for End-to-End Autonomous Driving. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 12037–12047 (2025)

31. Ling, H., Chen, Z., Chen, Q., Di, D., Ma, Y., Li, H., Wei, C., Tao, Z., Yang, X.: EverybodyDance: Bipartite Graph–Based Identity Correspondence for Multi-Character Animation. Advances in Neural Information Processing Systems **38**, 14123–14151 (2026)
32. Liu, L., Jia, C., Yu, G., Song, Z., Li, J., Jia, F., Wu, P., Hao, X., Luo, Y.: GuideFlow: Constraint-Guided Flow Matching for Planning in End-to-End Autonomous Driving. arXiv preprint arXiv:2511.18729 (2025)
33. Lu, H., Liu, Z., Jiang, G., Luo, Y., Chen, S., Zhang, Y., Chen, Y.C.: UniUGP: Unifying Understanding, Generation, and Planing for End-to-End Autonomous Driving. arXiv preprint arXiv:2512.09864 (2025)
34. Luo, Y., Li, F., Xu, S., Lai, Z., Yang, L., Chen, Q., Luo, Z., Xie, Z., Jiang, S., Liu, J., et al.: AdaThinkDrive: Adaptive Thinking via Reinforcement Learning for Autonomous Driving. arXiv preprint arXiv:2509.13769 (2025)
35. Peebles, W., Xie, S.: Scalable Diffusion Models with Transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 4195–4205 (2023)
36. Siméoni, O., Vo, H.V., Seitzer, M., Baldassarre, F., Oquab, M., Jose, C., Khalidov, V., Szafraniec, M., Yi, S., Ramamonjisoa, M., Massa, F., Haziza, D., Wehrstedt, L., Wang, J., Darcet, T., Moutakanni, T., Sentana, L., Roberts, C., Vedaldi, A., Tolan, J., Brandt, J., Couprie, C., Mairal, J., Jégou, H., Labatut, P., Bojanowski, P.: DINOv3 (2025)
37. Su, H., Wu, W., Song, F., Zhang, J., Yang, Z., Yan, J.: DriveMamba: Task-Centric Scalable State Space Model for Efficient End-to-End Autonomous Driving. arXiv preprint arXiv:2602.13301 (2026)
38. 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)
39. Tian, X., Gu, J., Li, B., Liu, Y., Wang, Y., Zhao, Z., Zhan, K., Jia, P., Lang, X., Zhao, H.: DriveVLM: The Convergence of Autonomous Driving and Large Vision-Language Models. In: Conference on Robot Learning. pp. 4698–4726. PMLR (2025)
40. Tu, J., Ji, W., Zhao, H., Zhang, C., Zimmermann, R., Qian, H.: DriveDiTFit: Fine-tuning Diffusion Transformers for Autonomous Driving Data Generation. ACM Trans. Multim. Comput. Commun. Appl. pp. 85:1–85:29 (2025)
41. Wan, T., Wang, A., Ai, B., Wen, B., Mao, C., Xie, C.W., Chen, D., Yu, F., Zhao, H., Yang, J., et al.: Wan: Open and Advanced Large-Scale Video Generative Models. arXiv preprint arXiv:2503.20314 (2025)
42. Wang, S., Yu, Z., Jiang, X., Lan, S., Shi, M., Chang, N., Kautz, J., Li, Y., Alvarez, J.M.: OmniDrive: A Holistic Vision-Language Dataset for Autonomous Driving with Counterfactual Reasoning. In: Proceedings of the computer vision and pattern recognition conference. pp. 22442–22452 (2025)
43. 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)
44. Wang, X., Yang, X., Xu, Y., Wu, Y., Li, Z., Zhao, N.: AffordBot: 3D Fine-grained Embodied Reasoning via Multimodal Large Language Models. Advances in Neural Information Processing Systems **38**, 140367–140387 (2026)
45. Yang, A., Li, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Gao, C., Huang, C., Lv, C., Zheng, C., Liu, D., Zhou, F., Huang, F., Hu, F., Ge, H., Wei, H., Lin, H., Tang, J., Yang, J., Tu, J., Zhang, J., Yang, J., Yang, J., Zhou, J., Zhou, J., Lin, J., Dang, K., Bao, K., Yang, K., Yu, L., Deng, L., Li, M., Xue, M., Li, M., Zhang,

P., Wang, P., Zhu, Q., Men, R., Gao, R., Liu, S., Luo, S., Li, T., Tang, T., Yin, W., Ren, X., Wang, X., Zhang, X., Ren, X., Fan, Y., Su, Y., Zhang, Y., Zhang, Y., Wan, Y., Liu, Y., Wang, Z., Cui, Z., Zhang, Z., Zhou, Z., Qiu, Z.: Qwen3 Technical Report. arXiv preprint arXiv:2505.09388 (2025)
46. Yang, B., Su, H., Gkanatsios, N., Ke, T.W., Jain, A., Schneider, J., Fragkiadaki, K.: Diffusion-ES: Gradient-free Planning with Diffusion for Autonomous and Instruction-guided Driving. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 15342–15353 (2024)
47. Yang, X., Chang, T., Zhang, T., Wang, S., Hong, R., Wang, M.: Learning Hierarchical Visual Transformation for Domain Generalizable Visual Matching and Recognition. International Journal of Computer Vision **132**(11), 4823–4849 (2024)
48. Yang, X., Feng, F., Ji, W., Wang, M., Chua, T.S.: Deconfounded Video Moment Retrieval with Causal Intervention. In: Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval. pp. 1–10 (2021)
49. Yang, X., Pan, H., Wang, X., Cao, Y., Li, J., Wang, M.: FiNE: Fine-grained Neuron-level Model Editing for Reliable and Safe LLMs. IEEE Transactions on Pattern Analysis and Machine Intelligence (2026)
50. Yao, W., Li, Z., Lan, S., Wang, Z., Sun, X., Alvarez, J.M., Wu, Z.: DriveSuprim: Towards Precise Trajectory Selection for End-to-End Planning. arXiv preprint arXiv:2506.06659 (2025)
51. Yuan, C., Zhang, Z., Sun, J., Sun, S., Huang, Z., Lee, C.D.W., Li, D., Han, Y., Wong, A., Tee, K.P., et al.: DRAMA: An Efficient End-to-End Motion Planner for Autonomous Driving with Mamba. arXiv preprint arXiv:2408.03601 (2024)
52. Zhang, S., Li, X., Zong, M., Zhu, X., Cheng, D.: Learning K for Knn Classification. ACM Transactions on Intelligent Systems and Technology **8**(3), 1–19 (2017)
53. Zheng, Y., Liang, R., ZHENG, K., Zheng, J., Mao, L., Li, J., Gu, W., Ai, R., Li, S.E., Zhan, X., Liu, J.: Diffusion-Based Planning for Autonomous Driving with Flexible Guidance. In: The Thirteenth International Conference on Learning Representations (2025)
54. Zhou, J., Wang, J., Ma, B., Liu, Y.S., Huang, T., Wang, X.: Uni3D: Exploring Unified 3D Representation at Scale. In: International Conference on Learning Representations (2024)
55. Zhou, Z., Cai, T., Zhao, Seth Z.and Zhang, Y., Huang, Z., Zhou, B., Ma, J.: AutoVLA: A Vision-Language-Action Model for End-to-End Autonomous Driving with Adaptive Reasoning and Reinforcement Fine-Tuning. Advances in Neural Information Processing Systems (2025)
56. Zou, J., Chen, S., Liao, B., Zheng, Z., Song, Y., Zhang, L., Zhang, Q., Liu, W., Wang, X.: DiffusionDriveV2: Reinforcement Learning-Constrained Truncated Diffusion Modeling in End-to-End Autonomous Driving. arXiv preprint arXiv:2512.07745 (2025)