Teaching Vision-Language-Action Models What to See and Where to Look

Yuguang Yang*[1,3], Canyu Chen*[2,3], Zhewen Tan*[6], Yizhi Wang[7], Zichao Feng[8], Chunyang Liu[4], Kehua Sheng[4], Juan Zhang[8], Linlin Yang[5], Baochang Zhang‡[8], Yan Wang†[3], Bo Zhang†[4], and Xianbin Cao†[1]

[1] School of Electronic Information Engineering, Beihang University
[2] National College for Excellent Engineers, Beihang University
[3] Institute for AI Industry Research (AIR), Tsinghua University
[4] DiDi
[5] State Key Laboratory of Media Convergence and Communication, Communication University of China
[6] School of Computer Science and Engineering, Beihang University
[7] School of Cyber Science and Technology, Beihang University
[8] School of Artificial Intelligence, Beihang University

Abstract. Vision-Language-Action (VLA) models have emerged as a promising paradigm for end-to-end autonomous driving. However, existing VLAs' traffic relies heavily on text-centric visual question answering and chain-of-thought reasoning data, which emphasizes linguistic reasoning rather than action-grounded planning. As a result, the learned representations capture semantic knowledge but lack spatial dependencies crucial for reliable trajectory prediction. We propose DriveTeach-VLA, a framework that explicitly teaches VLAs what to see and where to look. Driving-aware Vision Distillation (DVD) injects driving-specific perceptual priors into the vision encoder, while 2D Trajectory-Guided Prompts (2D-TGP) provide spatial conditioning aligned with feasible driving trajectories. Together they form a vision-guided learning pipeline: what to see (DVD pretraining) → where to look (TGP-guided SFT) → how to act (TGP-guided GRPO). DriveTeach-VLA achieves the state-of-the-art performance on NAVSIM and nuScenes. Our code is available at: `https://github.com/ShivaTeam/DriveTeach-VLA`.

Keywords: Vision-Language-Action model · Driving-aware Vision Distillation · 2D Trajectory-Guided Prompting

1 Introduction

Autonomous Driving (AD) has long been a challenging application [3]. The previous dominant paradigm relied on a modular design, decomposing the driving

‡ Project Lead.
* Equal contribution: {guangbuaa,chencanyu,tanzhewen}@buaa.edu.cn
† Corresponding authors: xbcao@buaa.edu.cn, wangyan@air.tsinghua.edu.cn, zhangbo@didiglobal.com

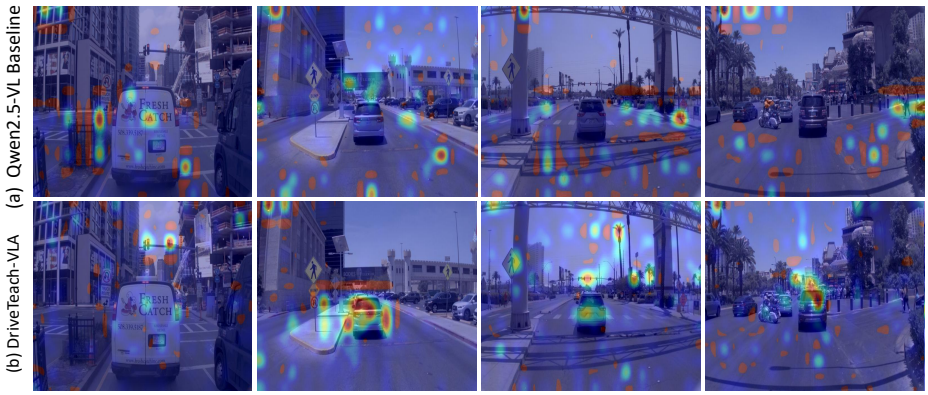

Fig. 1: Visualization of attention maps during autoregressive decoding on Qwen2.5-VL SFT baseline and DriveTeach-VLA.

process into perception [31, 33, 50], prediction [17, 46, 56], and planning [18, 19, 36] modules. While this structure offers interpretability and controllability, it suffers from error propagation and limited cross-task reasoning [55]. Recent advances in Multi-modal Large Language Models (MLLMs) [2, 7, 37] have demonstrated strong generalization and multi-task capabilities, inspiring a shift toward unified models that map raw sensor inputs to feasible trajectory planning results directly.

In this trend, Vision–Language–Action (VLA) models have emerged as a powerful framework, integrating perception, prediction, and planning in a unified manner [12, 27, 29, 38, 54, 55]. These models typically take RGB camera images as visual inputs and predict the vehicle's future trajectory over several seconds at 0.5 s intervals. In terms of model design, the paradigm formulates trajectory prediction as a decoding task, where the MLLM-based planner either autoregressing text-based waypoints [8, 21, 40, 45, 52, 55], or generating trajectories via an auxiliary planner, *e.g.*, a diffusion model, [12, 29, 43, 53]. In terms of training, recent advanced AD-VLA models [29,40,43,44,47,54] typically adopt a three-stage training pipeline: (*i*) Pretraining on Visual Question Answer (VQA) data [8, 47] to inject driving priors; (*ii*) Imitation Learning (IL) with Supervised Fine Tuning (SFT) on Chain-of-Thought (CoT) reasoning data and expert trajectories. These two stages are common across nearly all multi-stage VLA frameworks. (*iii*) More recently, the Reinforcement Learning (RL) stage with GRPO has been introduced by currently advanced works [23, 29, 40, 55] to further align AD-VLA's predicted trajectories with human driving preferences.

Although AD-VLAs can explain their intents, meta-behaviors, *e.g.*, turning left/right, and even their trajectory planning decisions in a linguistically interpretable manner, a fundamental gap between perception and planning exists: Although VQA or CoT reasoning data introduces general traffic priors, textual contexts alone can not truly equip VLAs with the spatial and behavioral knowledge required for autonomous driving. This is due to their supervision being inherently *text-centric*, focusing on semantic question answering rather than

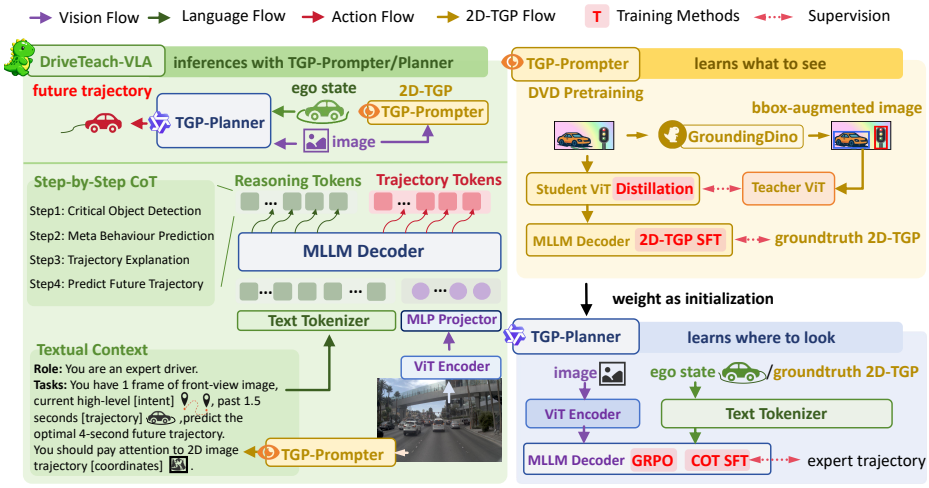

Fig. 2: DriveTeach-VLA Architecture. **Left**: DriveTeach-VLA operates in a dual-model manner consisting of a TGP-Prompter and a TGP-Planner. The TGP-Prompter first predicts the 2D-TGP, which is then used to condition the TGP-Planner for trajectory generation. **Right-Up**: The TGP-Prompter is trained via DVD, supervised by critical-object bounding-box–augmented images' features and ground-truth 2D-TGP. **Right-Down**: The TGP-Planner is supervised by expert trajectories during IL with SFT and further optimized by RL with GRPO. During both IL and RL, ground-truth 2D-TGP is provided as contextual guidance (teacher forcing) to provide the precise planning-aware spatial guidance for trajectory learning.

planning-related perception. Consequently, the learned representations emphasize linguistic semantics instead of motion and spatial dependencies essential for trajectory planning. Ultimately, the model learns to name *what is seen* rather than understand *what should see* and *where should look* for reliable trajectory generation. As shown in Figure 1(a), attention learned from purely textual traffic-knowledge SFT is scattered across the scene with no clear spatial grounding, making it difficult to interpret. Quantitative analysis is further provided in Analysis 1 of Section 4.3.

In this paper, we propose DriveTeach-VLA, a framework that operates in a dual-model manner consisting of a TGP-Prompter and a TGP-Planner, both built upon Qwen2.5-VL-3B. The overall pipeline of DriveTeach-VLA is shown in Figure 2. The TGP-Prompter undergoes vision-enhanced pretraining and is responsible for generating feasible driving region guidance, while TGP-Planner is conditioned on the guidance to predict future trajectory. To achieve **"teach what to see"**, *i.e.*, which objects are relevant for driving, the TGP-Prompter is trained with Driving-aware Vision Distillation (DVD). Specifically, DVD first employs Grounding DINO [39] to detect traffic-critical objects. The detected bounding boxes are then overlaid on the images as visual prompts, forming so-called *bbox-augmented images*. Then we apply self-distillation [4] on the ViT encoder of

the TGP-Prompter: a teacher ViT processes bbox-augmented images while a student ViT processes the corresponding raw images. Their representations are aligned in the latent space, enabling the student encoder to internalize traffic-relevant visual cues without relying on textual supervision. To achieve **"teach where to look"**, *i.e.*, which spatial region is most relevant for planning in the current scene, we further introduce a 2D image Trajectory-Guided Prompt (2D-TGP), which projects trajectories from world (BEV) coordinates onto the image plane with a pinhole camera model [14]. These 2D-TGP trajectories can provide spatial guidance about feasible driving regions for the AD-VLA. Thus, the TGP-Prompter's decoder output is supervised with ground-truth 2D-TGP trajectory derived from expert trajectories via SFT.

The TGP-Planner is initialized from the TGP-Prompter to obtain the traffic prior and trained to predict trajectories conditioned on the 2D-TGP. Following [8, 45, 55], we annotate step-by-step CoT reasoning data with Qwen2.5-VL-72B. The TGP-Planner learns to generate intermediate CoT reasoning steps and produce the future trajectory under the guidance of 2D-TGP. Training is performed with SFT followed by GRPO. To provide stable spatial guidance during training, we employ teacher forcing by supplying ground-truth 2D-TGP as the condition for the TGP-Planner in training. While in test, to avoid the information leak, the 2D-TGP is first generated by the TGP-Prompter and subsequently fed into the TGP-Planner to produce the future trajectory.

DriveTeach-VLA fully harnesses the intrinsic visual understanding ability of MLLMs and aligns it with driving-aware spatial reasoning for trajectory planning, achieving more precise trajectory prediction. Our main contributions are summarized as follows:

- We identify that text-centric training in existing VLA models often produces spatially ungrounded attention.
- DriveTeach-VLA introduces DVD and 2D-TGP to explicitly teach VLAs what to see and where to look. For improving what to see, DVD transfers traffic visual cues via self-distillation from bbox-augmented images. For improving where to look, 2D-TGP projects expert trajectories onto the image plane to provide spatial guidance for planning.
- DriveTeach-VLA achieves the State-of-The-Art (SoTA) performance on Navsim and nuScene benchmarks.

2 Related Work

AD-VLA Model. Recent AD-VLAs formulate trajectory prediction as a decoding task, where a MLLM planner generates a future few seconds' trajectory conditioned on visual–language context. The paradigm follows two major branches. The first relies on VLM itself's autoregressive scheme, where the model sequentially regresses the trajectory. ImpromptuVLA [8], AdaThinkDrive [40], and other previous models [21, 52] regress text form trajectory directly. Poutine [45] further explores a more suitable prompt and CoT reasoning steps for VLA's text

waypoint trajectory prediction. AutoVLA [55] follows this scheme but instead replaces text-form trajectory with action tokens that discretize each dimension of the trajectory into several bins. The second relies on an additional planner to produce continuous trajectories conditioned on the latent reasoning features of the LLM. AsyncDriver [6], RecogDrive [29], and Orion [12] are built upon VLM's encoded features and introduce (query-based) MLP/Diffusion to predict the future trajectory. More recent works, such as DriveMoe [53] and DriveVLA-W0 [27], employ a Mixture-of-Expert (MoE) architecture that pairs the VLM with an additional action-expert transformer by mutual attention to process multi-view images or reduce the inference time. The proposed DriveTeach-VLA follows the text-waypoint trajectory autoregressive scheme to explore the native capabilities of the MLLM backbone in the autonomous driving domain.

AD-VLA Training. Some AD-VLAs directly finetune MLLMs for trajectory prediction [21, 52]. To achieve stronger generalization, recent studies have converged toward a multi-stage training pipeline including: *(i) VQA Pretraining.* VQA pairs are firstly pseudo-labeled by MLLM, such as Qwen2.5-VL-72B and used to pretrain VLA with instruction learning. These VQA pairs involve multiple traffic-relevant tasks including scene description, traffic condition reasoning, and action explanation. DriveLM [47] systematically organizes these tasks into a Perception–Prediction–Explanation–Behavior framework, which has since become the foundation for many subsequent VLA studies [29, 40, 44, 54]. *(ii) SFT-IL.* The model is supervised by expert trajectories and pseudo-labeled CoT reasoning steps with SFT to enable end-to-end trajectory generation from sensory inputs. *(iii) RL.* AutoVLA [55], RecogDrive [29], and AdaThinkDrive [40] introduce GRPO-based [13] reinforcement optimization. Especially, EVADrive [23] further proposes multi-objective RL framework, *i.e.*, APO, for trajectory planning. The RL-stage aligns model prediction with human driving preference, including comfort, safety, and compliance with traffic regulations. DriveTeach-VLA replaces text-centric VQA pretraining stage with vision-centric DVD pretraining and augments IL and RL stages with 2D-TGP, enabling vision-grounded reasoning for reliable trajectory planning.

3 Method

In this section, we first formulate the pipeline of the proposed method in Section 3.1. Then, we introduce DVD pretraining and TGP-Guided learning/inference in Section 3.2 and Section 3.3, respectively.

3.1 Preliminary

BEV trajectory, 2D-TGP trajectory, and pinhole camera model. An expert trajectory with T time-steps in world (BEV) coordinates is denoted as $\mathcal{T}_w = \{\mathbf{a}_t = (x_t, y_t, \psi_t) \mid t = 1, 2, \ldots, T\}$, where x_t and y_t denote the position offsets in the ego-centric coordinate frame and ψ_t denotes the yaw (heading)

angle of the ego vehicle. Given the front camera intrinsic matrix K and extrinsic matrix $[R \mid t]$, the (x_t, y_t) in ego-centric coordinates can be projected onto the 2D image plane (x_t^I, y_t^I) using the pinhole camera model $\pi(\cdot)$ [14]:

$$\begin{aligned} &(x_t^I, y_t^I) = \pi(K, [R \mid t], (x_t, y_t)), \mathcal{T}_I = \{\mathbf{a}_t^I = (x_t^I, y_t^I, \psi_t) \mid t = 1, 2, \ldots, T\}, \\ &P_I = \text{Text}\left[(x_1^I, y_1^I), (x_2^I, y_2^I), \ldots, (x_T^I, y_T^I)\right], \end{aligned} \tag{1}$$

where P_I is named ground-truth 2D-TGP, and $\mathcal{T}_I$, combining the projected offset points with yaw angle, is named ground-truth 2D-TGP trajectory. The yaw angle ψ_t is retained from $\mathcal{T}_w$ so that the complete BEV trajectory can be recovered from the 2D-TGP trajectory via the inverse projection π^{-1} (See Eq. 2 below).

In general, the $\pi(\cdot)$ is not invertible due to the loss of depth information. However, in AD scenarios, trajectory points are typically assumed to lie on the ground plane ($z = 0$) and thus the projection becomes invertible:

$$(x_t, y_t) = \pi^{-1}\big(K, [R|t], (x_t^I, y_t^I)\big) \quad \text{s.t.} \quad z = 0, \tag{2}$$

which enables recovering a unique BEV position from the image-space trajectory and can be used to verify the quality of the estimated 2D-TGP trajectory (ψ_t should be predicted separately for 2D-TGP trajectory evaluation).

To establish a more intuitive understanding, we visualize the 2D-TGP trajectory on the image in Figure 4. By providing the MLLM with 2D-TGP coordinates, we can better align its intrinsic grounding ability with the VLA driving task.

Task formulation. Given observed camera images C, ego-state S, language instruction L, AD-VLA models output BEV trajectory estimation $\hat{\mathcal{T}}_w$ for planning:

$$\hat{\mathcal{T}}_w = \text{AD-VLA}(C, S, L). \tag{3}$$

Different from previous AD-VLAs formulating trajectory prediction as Eq. 3, DriveTeach-VLA consists of two models, including TGP-Prompter and TGP-Planner, both of which are built upon Qwen2.5-VL-3B. TGP-Prompter firstly output 2D-TGP estimation $\hat{\mathcal{T}}_I$ as spatial guidance, serving as the textual condition for TGP-Planner to estimate BEV trajectory:

$$\hat{P}_I, \hat{\mathcal{T}}_I = \text{TGP-Prompter}(C, S, L_{\text{TGP}}), \quad \hat{\mathcal{T}}_w = \text{TGP-Planner}(C, S, [L; \hat{P}_I]), \tag{4}$$

where L_{TGP} and L is the language instruction for TGP-Prompter and TGP-Planner, detailed in Supp. Figure 8 and Supp. Figure 9, respectively. Such textual coordinates conditions $\hat{P}_I$ are reasonable due to the intrinsic grounding ability of MLLMs, as 2D coordinates are naturally interpretable to the MLLM model.

Furthermore, we decouple the TGP-Prompter and TGP-Planner into two separate models to avoid instruction confusion, as 2D-TGP generation and BEV trajectory prediction have distinct optimization objectives despite sharing similar input modalities.

3.2 TGP-Prompter: DVD Pretraining

DVD pretraining is specially designed for the TGP-Prompter to inject traffic priors without reliance on VQA pairs. Existing VLA models typically employ

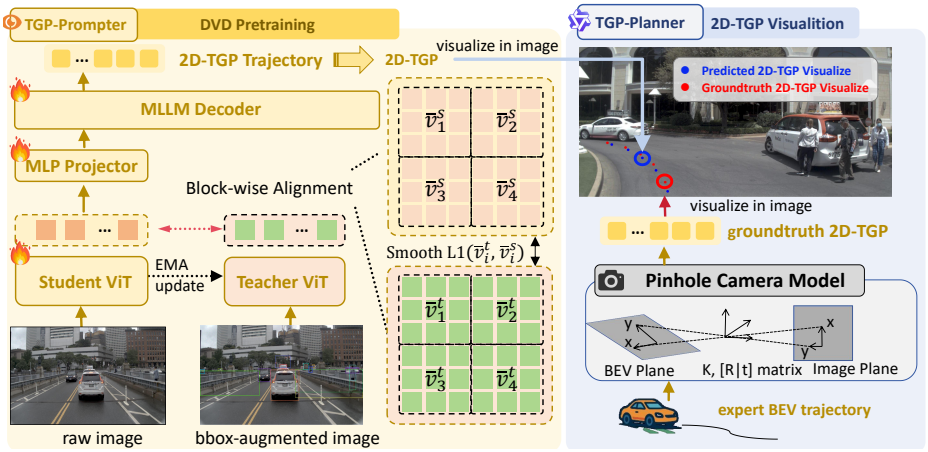

Fig. 3: DriveTeach-VLA schemes. **Left**: Traffic critical object priors are injected via bbox-augmented image self-distillation. **Right**: The visualized 2D-TGP is highly related to the driving behavior (turn left), and the 2D-TGP conditions TGP-Planner in text form of a sequence of 2D coordinates, which is interpretable for MLLM.

ViT encoders pretrained on natural images, which lack the domain knowledge about what should be seen during driving, *e.g.*, lanes, traffic participants, and scene layout. To address this issue, DVD firstly introduces the bbox-augmented image self-distillation for the ViT encoder of the VLA model, enabling it to learn driving-critical visual cues. In addition, to achieve precise driving-aware spatial perception regarding where to look, DVD pretraining also includes SFT that trains the VLA decoder to estimate the 2D-TGP trajectory. See Figure 3 (left) for the overall training framework, and detailed elaborations are as follows:

Bbox-augmented image self-distillation for ViT encoder. We firstly employ an open-vocabulary detector Grounding Dino [39] to detect traffic-critical objects, including *car, truck, bus, trailer, construction vehicle, pedestrian, motorcycle, bicycle, barrier, traffic element, and traffic light*. The complete grounding object lists can be found in the Supp. Figure 6. By overlaying the detected bounding boxes on the raw camera image C, we can obtain the bbox-augmented image C_{bbox}. The bbox in Figure 4 presents the object detection results. The self-distillation involves a pair of student and teacher ViT encoders, both initialized from the same MLLM's ViT encoder. The teacher ViT receives C_{bbox} while the student ViT is fed with C:

$$\text{ViT}_t(C_{\text{bbox}}) = \left[v_1^t, \cdots, v_N^t\right], \quad \text{ViT}_s(C) = \left[v_1^s, \cdots, v_N^s\right], \tag{5}$$

where N is the image patch token number, and ViT_s/v_i^s, and ViT_t/v_i^t denotes student and teacher encoder/their patch features, respectively. In terms of feature alignment between v_i^t and v_i^s , block-wise alignment is employed to capture broader contextual information brought by the bounding boxes. Specifically, we partition the feature maps into K non-overlapping blocks $\mathcal{B}_k$ (*i.e.*, dividing the

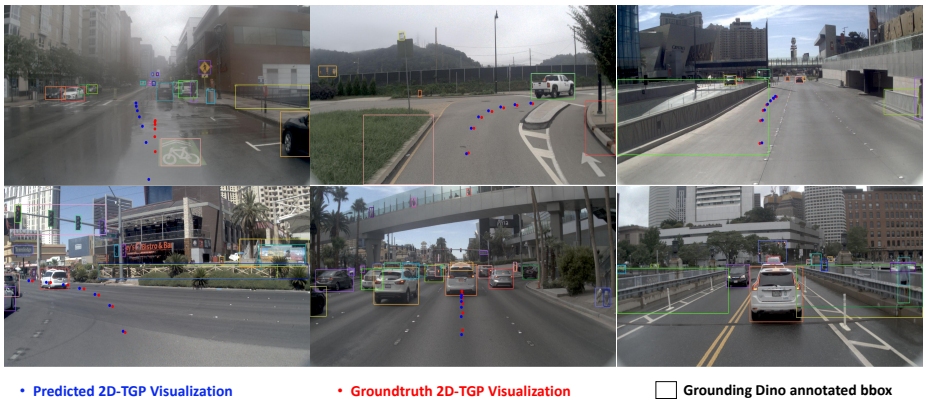

Fig. 4: Visualization of the predicted 2D-TGP trajectory, critical objects detected by Grounding Dino of DriveTeach-VLA.

feature map into a grid of rows × columns, e.g., 2 × 4 yields K=8 blocks) and compute block-wise alignment loss $\mathcal{L}_{\text{distill}}$:

$$\bar{v}_k^t = \frac{1}{|\mathcal{B}_k|} \sum_{i \in \mathcal{B}_k} v_i^t, \quad \bar{v}_k^s = \frac{1}{|\mathcal{B}_k|} \sum_{i \in \mathcal{B}_k} v_i^s, \quad \mathcal{L}_{\text{distill}} = \frac{1}{K} \sum_{k=1}^{K} \text{Smooth-L1}(\bar{v}_k^t, \bar{v}_k^s), \tag{6}$$

where Smooth-L1 loss provides robustness to local spatial outliers [35].

2D-TGP trajectory SFT for MLLM decoder. We first convert expert trajectories to 2D-TGP trajectories according to Eq. 1. Then the TGP-Prompter is supervised to predict the 2D-TGP trajectory via optimizing the SFT cross-entropy loss:

$$\mathcal{L}_{\text{TGP}} = -\frac{1}{T} \sum_{t=1}^{T} \log P_\theta(\mathcal{T}_I^{(t)} \mid C, L_{\text{TGP}}, \mathcal{T}_I^{(<t)}), \tag{7}$$

where $P(\cdot)$ is the TGP-Prompter policy, L_{TGP} is language instruction (detailed in Supp. Figure 8), and $\mathcal{T}_I^{(<t)}$ denotes the trajectory action sequence before step t, and $\mathcal{T}_I^{(t)}$ denotes the action at step t.

Training Objective. The overall training loss is defined as:

$$\mathcal{L}_{\text{DVD}} = \mathcal{L}_{\text{distill}} + \lambda_{\text{TGP}} \cdot \mathcal{L}_{\text{TGP}}, \tag{8}$$

where λ_{TGP} is a hyperparameter. During training, the encoded features from the student ViT are fed into the MLLM decoder to make a prediction. The weights of the student ViT and the MLLM decoder are updated through backpropagation, while the teacher ViT is updated with Exponential Moving Average (EMA) from student ViT following the classic self-distillation setup from DINO [4].

3.3 TGP-Planner: TGP-guided Learning and Reasoning

The TGP-Planner is firstly initialized from the TGP-Prompter to inherit driving-related priors. It is then trained to generate step-by-step CoT reasoning and predict future trajectories in BEV space, conditioned on the 2D-TGP, the observed camera image, and the driving instruction. Detailed processes are as follows.

TGP-guided CoT Reasoning. Following Poutine [45], we generate pseudo-labeled reasoning data using the Qwen2.5-VL-72B model. The first three sub-tasks are identical to those in Poutine, while the fourth is modified to suit the 2D-TGP condition:

Task 1: Critical Object Detection. Determine whether any critical instance from a predefined object class list (*e.g.*, hazards, cyclists, conflicting vehicles) may influence the ego vehicle's future trajectory.

Task 2: Natural Language Explanation. Provide a concise explanation of the expert trajectory and the reasoning behind the expert driver's behavior.

Task 3: Meta-Behavior Selection. Select a single behavior (*e.g.*, turning, acceleration/deceleration, lane-following) that best describes the trajectory.

Task 4: Future Trajectory Prediction. Different from prior work, we introduce spatial guidance via 2D-TGP, where projected trajectory keypoints $\{(x_1^I, y_1^I), \ldots, (x_8^I, y_8^I)\}$ indicate regions along the feasible driving path that the model should attend to. Task 4 is modified as: given the observed camera image, ego-state, reasoning outputs, and 2D-TGP keypoints, predict the optimal 4-second future trajectory.

Supp. Figure 7 and Supp. Figure 9 details the prompts used for pseudo-labeling and TGP-Planner, respectively. We supervise TGP-Planner with above CoT reasoning data by SFT. After CoT-SFT training, we further align TGP-Planner's trajectory prediction with human driving preferences using reinforcement learning, following AutoVLA [55]. At each rollout, the predicted trajectory $\hat{\mathcal{T}}_w$ is evaluated by the PDM-Score (PDMS) [10]:

$$\text{PDMS} = \prod_{c \in C} c \times \frac{\sum_{m \in M} w_m \cdot m}{\sum_{m \in M} w_m}, \tag{9}$$

where $C = \{\text{NC, DAC}\}$ (No-at-fault Collisions, Drivable Area Compliance) and $M = \{\text{EP, TTC, C}\}$ (Ego Progress, Time to Collision, Comfort) with weights $w_m = \{5, 5, 2\}$. We follow CuriousVLA [5] to conduct data filtering and rollout for GRPO [13].

Teacher forcing training and Inference pipeline. During CoT-SFT and GRPO-RL, to produce the most precise guidance for driving skills learning, we used the ground-truth 2D-TGP to condition TGP-Planner. While in test (inference), as shown in Figure 2 (left), the 2D-TGP condition is firstly generated by TGP-Prompter, which is subsequently fed into TGP-Planner for future trajectory prediction. Although this introduces a train–test gap, we show in Section 4.3 that the gap is negligible ($\sim$0.4 PDMS), confirming TGP-Prompter provides sufficiently accurate spatial guidance.

Table 1: PDMS Results on Navtest. Best results within each category are in **bold**. ∗ Curious-VLA is fully open-sourced, and we reproduce its results with the same GRPO setting with us (1 round).

Method	Pub.	Base VLM	NC	DAC	TTC	C	EP	PDMS
Human		-	100	100	100	99.9	87.5	94.8
End-to-End Methods								
UniAD [16]	CVPR'23	-	97.8	91.9	92.9	**100.0**	78.8	83.4
TransFuser [9]	TPAMI'22	-	97.7	92.8	92.8	**100.0**	79.2	84.0
Hydra-MDP [30]	CVPRW'24	-	98.3	96.0	94.6	**100.0**	78.7	86.5
DiffusionDrive [34]	CVPR'25	-	98.2	96.2	94.7	**100.0**	82.2	88.1
WoTE [28]	ICCV'25	-	**98.5**	96.8	94.4	99.9	81.9	88.3
ASSCG [1]	Arxiv'26	-	98.2	**98.3**	**94.8**	**100.0**	**87.5**	**91.4**
VLA-based Methods								
ReCogDrive [29]	ICLR'26	InternVL3-8B	98.2	97.8	95.2	99.8	83.5	89.6
ImagiDrive-S [25]	ICRA'26	InternVL2.5-4B	98.1	96.2	94.5	**100.0**	80.5	86.9
AutoVLA [55]	NeurIPS'25	Qwen2.5-VL-3B	98.4	95.6	**98.0**	99.9	81.9	89.1
CuriousVLA* [5]	CVPR'26	Qwen2.5-VL-3B	97.7	95.9	97.2	98.2	89.2	88.9
DriveTeach-VLA	-	Qwen2.5-VL-3B	**98.5**	**96.9**	97.9	98.2	**88.5**	**90.4**

4 Experiment

4.1 Experimental Setup

Benchmarks. We evaluate our method on two public autonomous-driving benchmarks: non-reactive closed-loop NAVSIM [10] and open-loop nuScenes [41]. NAVSIM predicts future 4-second trajectory and is designed for intention-changing scenarios where the future motion of the ego vehicle cannot be reliably predicted from history alone. Following RecogDrive [29], we report PDMS results of DriveTeach-VLA on NAVSIM's navtest. On nuScenes, following UniAD [16] and ST-P3 [15], we report the average L2 trajectory error and collision rate over a 3-second horizon.

Implementation Details. We follow CuriousVLA [5] to use the step-wise normalized text trajectory. Training uses AdamW (lr 4×10^{-5}, weight decay 0.05) with a cosine schedule and 0.10 warm-up. Models are trained with batch size 16 on 8×H100 GPUs. DriveTeach-VLA is trained in three stages. *(i) DVD pretraining*:1 epoch, with block size set to 2×4 and trajectory-loss weight $\lambda_{\mathrm{TGP}} = 0.1$. EMA momentum is set to 0.996. *(ii) CoT-SFT*: 6 epochs using Chain-of-Thought annotations generated by Qwen2.5-VL-72B. *(iii) GRPO-RL*: 180 update steps with a group size of 8. Especially, for fair comparison, CoT and GRPO are not employed for nuScene.

Table 2: Open-loop trajectory prediction on nuScenes. Best results within each category are in **bold**, second best are underlined.

Method	Pub.	ST-P3 metrics		UniAD metrics	
		L2 (m) ↓	Collision (%) ↓	L2 (m) ↓	Collision (%) ↓
UniAD [16]	CVPR'23	0.69	**0.12**	1.03	**0.31**
ST-P3 [15]	ECCV'22	2.11	0.71	–	–
VAD [22]	ICCV'23	0.37	0.14	–	–
EMMA [21]	TMLR'24	0.32	–	–	–
OpenEMMA [52]	WCAC'25	2.81	–	–	–
AutoVLA [55]	NeurIPS'25	0.48	0.13	0.86	0.35
Impromptu VLA [8]	NeurIPS'25	0.33	0.13	0.67	0.38
DriveTeach-VLA	-	**0.30**	**0.12**	**0.60**	**0.31**

4.2 Comparison with the SoTA models.

NAVSIM Results. We benchmark against recently published end-to-end and VLA-based SoTA models, including UniAD, TransFuser, Hydra-MDP, Diffusion-Drive, WoTE, RecogDrive, ImagiDrive-S, and AutoVLA. Among them, AutoVLA is the most similar to DriveTeach-VLA, as both adopt an autoregressive trajectory prediction paradigm. We report results on the Navtest split using PDMS metrics, and results are exhibited in Table 1. DriveTeach-VLA achieves 90.4 PDMS on Navtest with a single inference pass.

NuScenes Results. Table 2 shows the comparison of DriveTeach-VLA with several SOTA methods using the L2 distance error (L2) and collision rate as the primary metrics. DriveTeach-VLA achieves the best performance on both metrics, with an L2 of 0.30 and a collision rate of 0.12, surpassing all prior methods. The model exhibits the highest precision in trajectory prediction with minimal collision occurrence, indicating superior performance in real-world driving scenarios.

4.3 Ablation and Analysis

Attention Analysis of Vision Encoders. We compare the attention maps of the TGP-Planner during trajectory decoding with a CoT-SFT trained Qwen2.5-VL-3B baseline. The qualitative observations in Figure 1 show that the baseline exhibits dispersed attention, while DriveTeach-VLA produces clear spatial grounding on traffic-critical objects. To quantify this phenomenon, we introduce a metric called *Attention Mass (AM)*. The AM score for an observed camera image C is calculated as follows. Firstly, we compute the attention maps between the decoded waypoint tokens and the visual patch tokens. The attention maps are averaged across all waypoint tokens to obtain an aggregated attention map A. Then, let $\mathcal{R}$ denote the set of bounding boxes obtained from the bbox-augmented image C_{bbox} (Section 3.2), and let $|r|$ denote the area of box r. We sum the

Table 3: Ablation on λ_{TGP} with the TGP-Prompter. A moderate weight yields the best results.

λ_{TGP}	NC	DAC	TTC	EP	PDMS
0.05	97.5	94.5	96.7	86.9	86.3
0.08	97.5	94.3	96.3	87.0	86.1
0.10	**98.0**	**95.5**	**97.1**	**87.1**	**87.6**
0.12	97.8	95.1	97.0	86.9	87.2
0.15	97.9	95.2	**97.1**	86.9	87.4

Table 4: Ablation on $\mathcal{B}_k$ with the TGP-Prompter. Both patch-/block-wise DVD improve performance.

$\mathcal{B}_k$	NC	DAC	TTC	PDMS
w/o DVD	-	-	-	86.2
2×4	**98.0**	**95.5**	**97.1**	**87.6**
4×7	97.7	95.1	96.9	87.1
Patch-by-patch	97.7	94.5	96.7	86.5

Table 5: Ablation of DVD pretraining and 2D-TGP on Navsim. † indicates we convert 2D-TGP trajectory generated by TGP-Prompter to BEV trajectory with Eq. 2.

Method	Model	DAC	EP	TTC	PDMS
QwenVL-2.5-3B	Planner	93.2	85.8	97.3	84.8
+ VQA + CoT	Planner	94.0	87.2	96.7	86.4
+ DVD + CoT	Prompter †	94.9	87.5	96.7	87.3
+ DVD + CoT+GRPO	Prompter †	95.5	90.2	96.5	88.2
+ DVD + CoT	Planner	94.4	86.3	97.8	87.1
+ DVD + TGP + CoT	Planner	95.8	87.2	96.6	88.2
+ DVD + CoT +GRPO	Planner	95.8	**90.6**	96.7	89.1
+ DVD + TGP + CoT +GRPO	Planner	**96.9**	88.5	**97.9**	**90.4**

attention values of visual tokens lying inside all bounding boxes and normalize by the total box area:

$$\text{AM}(C) = \frac{1}{\sum_{r\in\mathcal{R}} |r|} \sum_{r\in\mathcal{R}} \sum_{(i,j)\in r} A_{i,j}, \tag{10}$$

where $A_{i,j}$ denotes the average attention value at spatial position (i, j) in the averaged attention map. We randomly sample 1k scenes from Navtrain and compute the mean and variance of their AM scores. Table 6 reports the results. The CoT-SFT baseline exhibits significantly lower AM than the TGP-Planner and also shows a smaller variance, indicating consistently weak attention on traffic-critical objects. This verifies that DVD pretraining effectively injects critical traffic-object priors into DriveTeach-VLA's TGP-Planner.

Hyperparameter analysis of λ_{TGP} and $\mathcal{B}_k$ As this evaluation only involves DVD pretraining, *i.e.*, TGP-Prompter only, we transform the predicted 2D-TGP trajectories into BEV trajectories using Eq. 2 and calculate PDMS on Navtest. Table 3 studies the hyperparameter λ_{TGP} balancing 2D-TGP learning and self-distillation in DVD pretraining. A small value (0.05) under-utilizes trajectory cues, while a large value (1.5) overemphasizes the auxiliary prediction task. A

Table 6: Attention Mass Statistics. AM-Score computed on 1k randomly sampled Navtrain scenes.

AM-Score	Mean ($\times 10^{-2}$)	Std ($\times 10^{-2}$)
Qwen2.5-VL-3B	2.28	3.12
TGP-Planner	3.19	3.77

Table 7: DVD robustness under detector noise. PDM-Score under random bbox drop and jitter perturbations.

Perturb.	w/o DVD	0%	20%	40%
Random drop	86.2	87.6	86.8	85.3
Jitter bbox	86.2	87.6	87.1	86.6

Table 8: Efficiency and Dual-model Design Ablation on H100. Only DVD and TGP-guided CoT-SFT is used. † indicates we convert 2D-TGP trajectory generated by TGP-Prompter to BEV trajectory with Eq. 2. Type column indicates the number of used model.

Method	**Setting**	**Type**	**Latency (s)**	**Mem. (GiB)**	**PDMS**
DriveTeach	Prompter-Planner	dual	3.03	17.2	88.2
Call 1	Prompter	–	1.14	8.2	–
Call 2	Planner	–	1.89	9.0	–
DriveTeach	Multi-turn QA	single	2.87	8.6	87.0
Turn 1	Prompter QA	–	1.11	–	–
Turn 2	Planner QA	–	1.75	–	–
Baseline	Prompter †	single	1.14	8.2	87.3

moderate value (0.1) achieves the best performance across NC, DAC, TTC, and PDMS.

Table 4 analyzes how the block participation strategy $\mathcal{B}_k$ in DVD affects planning quality. The 2×4 partition achieves the best PDMS among the tested configurations. This is likely because DVD relies on bbox-based visual prompts highlighting traffic-critical objects, and blocks with a larger receptive field during distillation better capture these highlighted regions and their surrounding context. In contrast, patch-level alignment dilutes the signal since most patches fall outside the highlighted bounding boxes.

Ablation on TGP-guided CoT and DVD. The incremental ablation results are reported in Table 5. Ablation begins based on a single Qwen2.5-VL-3B model (84.8 PDMS). We first establish a baseline using conventional VQA pretraining followed by CoT-SFT training (second row, 86.4 PDMS). Next, we introduce DVD pretraining combined with CoT-SFT. Under this setting, only the TGP-Prompter is trained: after completing DVD pretraining, the TGP-Planner model is initialized from TGP-Prompter and further optimized with CoT-SFT (87.1 PDMS). Subsequently, 2D-TGP guided reasoning is employed and brings performance to 88.2. Finally, the GRPO boosts performance to 90.4.

Furthermore, Table 5's 3-4 rows isolate the performance of TGP-Prompter, and 5-8 rows ablate TGP-guided CoT reasoning with and without GRPO.

Table 9: Train–test gap analysis of 2D-TGP. Using predicted $\hat{P}_I$ at inference yields comparable performance to the oracle ground-truth P_I setting.

Test-time 2D-TGP	DAC	EP	TTC	PDMS
Predicted $\hat{P}_I$	**96.9**	**88.5**	97.9	90.4
Ground-truth P_I	96.5	87.5	**99.3**	**90.8**

Table 10: Runtime comparison (seconds per sample) on L20 GPUs. AutoVLA's runtime is sourced from its paper [55].

Method	Trajectory	Runtime
ours	text waypoint	3.18 s
AutoVLA	physical action	3.95 s
AutoVLA	text waypoint	7.65 s

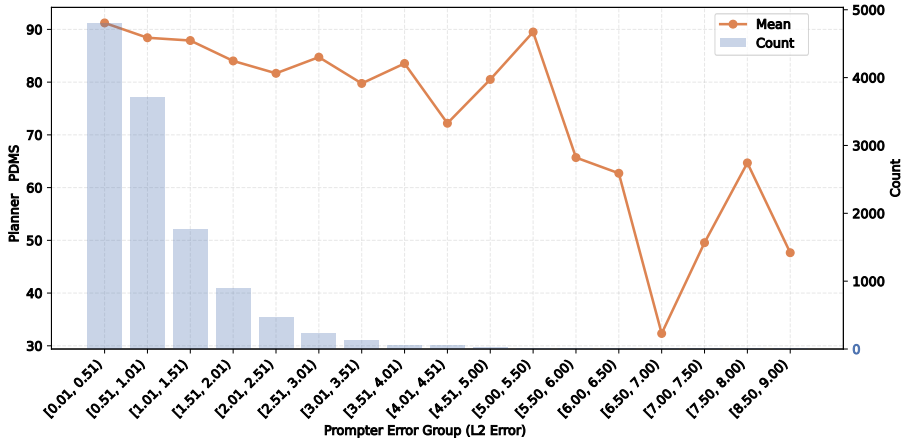

Fig. 5: PDMS by predicted-vs-GT 2D-TGP L2 error. PDMS remains stable across a broad range of trajectory prediction errors, indicating that the policy is robust to moderate inaccuracies in 2D-TGP prediction.

Train-Test Gap Analysis of the Predicted 2D-TGP. The teacher-forcing strategy uses ground-truth 2D-TGP during training but predicts $\hat{P}_I$ during inference. To quantify this train–test gap, we compare the standard inference pipeline against an oracle setting that leaks ground-truth 2D-TGP at test time. As shown in Table 9, leaking ground-truth 2D-TGP only increases PDMS by 0.4 (90.4 $\rightarrow$ 90.8), confirming that the TGP-Prompter produces reliable spatial guidance and the teacher-forcing gap is not a performance bottleneck.

Furthermore, we analyze the effect of 2D-TGP on TGP-Planner's prediction quality. Following Table 9, we linearly bin samples by predicted $\hat{\mathcal{T}}_I$-vs-GT $\mathcal{T}_I$'s L2 error and report PDMS in Figure 5. The small oracle gap (Table 9: 90.4/90.8) is expected: most predicted TGPs are accurate. And when the error grows, PDMS drops gradually, showing robustness rather than TGP irrelevance.

Robustness of DVD. DVD depends heavily on Grounding DINO detection quality. If Grounding DINO misses critical objects in novel scenes or domain-shifted data (*e.g.*, unusual vehicles, occluded pedestrians), the distilled visual priors may be noisy or misleading. So Table 7 adds random bbox drop and jitter

bbox noise on the annotated bboxes and tests TGP-Prompter's PDM-Score. The noise strength denotes the dropped-box ratio or the corner perturbation ratio w.r.t. box width/height. DVD shows robustness and remains helpful under moderate noise (20% drop/jitter: 86.8/87.1 vs. w/o DVD 86.2), while severe 40% noise can mislead guidance and hurt.

Efficiency and Dual Model Design Ablation. Although DriveTeach-VLA achieves SoTA performance, the dual-model architecture increases runtime latency. As shown in Table 8, Prompter-Planner design increases 1x runtime (about 1.14s on H100) during inference. To assess its necessity, we replace it with a multi-turn QA variant. As shown in Table 8, without GRPO, the Prompter-Planner design improves PDMS over the Prompter-only baseline (88.2 vs. 87.3), while the multi-turn QA variant performs worse (87.0 vs. 87.3). This suggests that jointly learning two instruction-following behaviors in one model remains challenging.

Despite the longer runtime, we also found something surprised. We compare DriveTeach-VLA with its most similar baseline, AutoVLA. Table 10 reports the runtime results, measured on one L20 GPU. AutoVLA reports results under two settings: predicting text waypoint and action token trajectories. Although action tokens require fewer tokens and are therefore more efficient, AutoVLA relies on long CoT reasoning, which introduces substantial token overhead. Consequently, even its action-token variant remains slower than DriveTeach-VLA. This indicates that DriveTeach-VLA uses fewer CoT reasoning tokens while still outperforming AutoVLA, highlighting the efficiency advantage of our vision-grounded spatial enhancement.

5 Conclusion

This work identifies a key limitation of existing VLA paradigms: their heavy reliance on text-centric pretraining, which leads to trajectory decoding without explicit semantic guidance. To address this issue, we propose DriveTeach-VLA, a VLA framework that decouples spatial guidance from trajectory planning through a dual-module architecture consisting of a TGP-Prompter and a TGP-Planner. Extensive experiments on NAVSIM and nuScenes demonstrate the effectiveness of the proposed approach, supported by comprehensive ablation studies. Furthermore, despite the dual-model design increases runtime, our efficiency analysis shows that visual spatial enhancement reduces the reliance on reasoning tokens. These results highlight the importance of vision-grounded spatial guidance for improving performance in autonomous driving systems.

6 Acknowledge

DriveTeach-VLA is funded by the National Key Research and Development Program of China, 2024YFE0217600, Xiongan AI Institute, DiDi and Wuxi Research Institute of Applied Technologies, Tsinghua University under Grant 20242001120.

References

1. Ang, S., Chen, Y., Haiyan, L., Mao, X., Bao, J., Xuliang, Sun, B., Wang, Y.: Asscg: Just-right gating over chattering for fast-slow llm planning in autonomous driving (2026), https://arxiv.org/abs/2606.25509
2. Bai, S., Chen, K., Liu, X., Wang, J., Ge, W., Song, S., Dang, K., Wang, P., Wang, S., Tang, J., et al.: Qwen2. 5-vl technical report. arXiv preprint arXiv:2502.13923 (2025)
3. Broekman, N.: Toward safe and scalable autonomy: A comprehensive review of technologies, deployments, and challenges in autonomous driving. Transactions on Computational and Scientific Methods **5**(4) (2025)
4. Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging properties in self-supervised vision transformers. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 9650–9660 (2021)
5. Chen, C., Yang, Y., Tan, Z., Wang, Y., Zhan, R., Liu, H., Mao, X., Bao, J., Tang, X., Yang, L., et al.: Devil is in narrow policy: Unleashing exploration in driving vla models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1062–1072 (2026)
6. Chen, Y., Ding, Z.h., Wang, Z., Wang, Y., Zhang, L., Liu, S.: Asynchronous large language model enhanced planner for autonomous driving. In: European Conference on Computer Vision. pp. 22–38. Springer (2024)
7. Chen, Z., Wu, J., Wang, W., Su, W., Chen, G., Xing, S., Zhong, M., Zhang, Q., Zhu, X., Lu, L., et al.: Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 24185–24198 (2024)
8. Chi, H., Gao, H.a., Liu, Z., Liu, J., Liu, C., Li, J., Yang, K., Yu, Y., Wang, Z., Li, W., et al.: Impromptu vla: Open weights and open data for driving vision-language-action models. arXiv preprint arXiv:2505.23757 (2025)
9. 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)
10. Dauner, D., Hallgarten, M., Li, T., Weng, X., Huang, Z., Yang, Z., Li, H., Gilitschenski, I., Ivanovic, B., Pavone, M., et al.: Navsim: Data-driven non-reactive autonomous vehicle simulation and benchmarking. Advances in Neural Information Processing Systems **37**, 28706–28719 (2024)
11. 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. arXiv preprint arXiv:2504.19580 (2025)
12. Fu, H., Zhang, D., Zhao, Z., Cui, J., Liang, D., Zhang, C., Zhang, D., Xie, H., Wang, B., Bai, X.: Orion: A holistic end-to-end autonomous driving framework by vision-language instructed action generation. arXiv preprint arXiv:2503.19755 (2025)
13. Guo, D., Yang, D., Zhang, H., Song, J., Wang, P., Zhu, Q., Xu, R., Zhang, R., Ma, S., Bi, X., et al.: Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. Nature **645**(8081), 633–638 (2025)
14. Heyden, A., Pollefeys, M.: Multiple view geometry. Emerging topics in computer vision **3**, 45–108 (2005)
15. 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)

16. 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)
17. Huang, Z., Liu, H., Lv, C.: Gameformer: Game-theoretic modeling and learning of transformer-based interactive prediction and planning for autonomous driving. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3903–3913 (2023)
18. Huang, Z., Liu, H., Wu, J., Lv, C.: Differentiable integrated motion prediction and planning with learnable cost function for autonomous driving. IEEE transactions on neural networks and learning systems (2023)
19. Huang, Z., Weng, X., Igl, M., Chen, Y., Cao, Y., Ivanovic, B., Pavone, M., Lv, C.: Gen-drive: Enhancing diffusion generative driving policies with reward modeling and reinforcement learning fine-tuning. arXiv preprint arXiv:2410.05582 (2024)
20. Hurst, A., Lerer, A., Goucher, A.P., Perelman, A., Ramesh, A., Clark, A., Ostrow, A., Welihinda, A., Hayes, A., Radford, A., et al.: Gpt-4o system card. arXiv preprint arXiv:2410.21276 (2024)
21. Hwang, J.J., Xu, R., Lin, H., Hung, W.C., Ji, J., Choi, K., Huang, D., He, T., Covington, P., Sapp, B., et al.: Emma: End-to-end multimodal model for autonomous driving. arXiv preprint arXiv:2410.23262 (2024)
22. Jiang, B., Chen, S., Xu, Q., Liao, B., Chen, J., Zhou, H., Zhang, Q., Liu, W., Huang, C., Wang, X.: Vad: Vectorized scene representation for efficient autonomous driving. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 8340–8350 (2023)
23. Jiao, S., Qian, K., Ye, H., Zhong, Y., Luo, Z., Jiang, S., Huang, Z., Fang, Y., Miao, J., Fu, Z., et al.: Evadrive: Evolutionary adversarial policy optimization for end-to-end autonomous driving. arXiv preprint arXiv:2508.09158 (2025)
24. Kirby, E., Boulch, A., Xu, Y., Yin, Y., Puy, G., Zablocki, É., Bursuc, A., Gidaris, S., Marlet, R., Bartoccioni, F., et al.: Driving on registers. arXiv preprint arXiv:2601.05083 (2026)
25. Li, J., Zhang, B., Jin, X., Deng, J., Zhu, X., Zhang, L.: Imagidrive: A unified imagination-and-planning framework for autonomous driving. arXiv preprint arXiv:2508.11428 (2025)
26. Li, K., Li, Z., Lan, S., Xie, Y., Zhang, Z., Liu, J., Wu, Z., Yu, Z., Alvarez, J.M.: Hydra-mdp++: Advancing end-to-end driving via expert-guided hydra-distillation. arXiv preprint arXiv:2503.12820 (2025)
27. Li, Y., Shang, S., Liu, W., Zhan, B., Wang, H., Wang, Y., Chen, Y., Wang, X., An, Y., Tang, C., et al.: Drivevla-w0: World models amplify data scaling law in autonomous driving. arXiv preprint arXiv:2510.12796 (2025)
28. Li, Y., Wang, Y., Liu, Y., He, J., Fan, L., Zhang, Z.: End-to-end driving with online trajectory evaluation via bev world model. arXiv preprint arXiv:2504.01941 (2025)
29. 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)
30. 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)
31. 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)

32. Li, Z., Yu, Z., Lan, S., Li, J., Kautz, J., Lu, T., Alvarez, J.M.: Is ego status all you need for open-loop end-to-end autonomous driving? In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14864–14873 (2024)
33. Liang, T., Xie, H., Yu, K., Xia, Z., Lin, Z., Wang, Y., Tang, T., Wang, B., Tang, Z.: Bevfusion: A simple and robust lidar-camera fusion framework. Advances in Neural Information Processing Systems **35**, 10421–10434 (2022)
34. 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)
35. Liu, C., Yu, S., Yu, M., Wei, B., Li, B., Li, G., Huang, W.: Adaptive smooth l1 loss: A better way to regress scene texts with extreme aspect ratios. In: 2021 IEEE Symposium on Computers and Communications (ISCC). pp. 1–7. IEEE (2021)
36. Liu, H., Huang, Z., Huang, W., Yang, H., Mo, X., Lv, C.: Hybrid-prediction integrated planning for autonomous driving. IEEE Transactions on Pattern Analysis and Machine Intelligence (2025)
37. Liu, H., Li, C., Wu, Q., Lee, Y.J.: Visual instruction tuning. Advances in neural information processing systems **36**, 34892–34916 (2023)
38. Liu, R., Kong, L., Li, D., Zhao, H.: Occvla: Vision-language-action model with implicit 3d occupancy supervision. arXiv preprint arXiv:2509.05578 (2025)
39. Liu, S., Zeng, Z., Ren, T., Li, F., Zhang, H., Yang, J., Jiang, Q., Li, C., Yang, J., Su, H., et al.: Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In: European conference on computer vision. pp. 38–55. Springer (2024)
40. 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)
41. Qian, T., Chen, J., Zhuo, L., Jiao, Y., Jiang, Y.G.: Nuscenes-qa: A multi-modal visual question answering benchmark for autonomous driving scenario. In: Proceedings of the AAAI Conference on Artificial Intelligence. pp. 4542–4550 (2024)
42. Qiao, Z., Li, H., Cao, Z., Liu, H.X.: Lightemma: Lightweight end-to-end multimodal model for autonomous driving. arXiv preprint arXiv:2505.00284 (2025)
43. Renz, K., Chen, L., Arani, E., Sinavski, O.: Simlingo: Vision-only closed-loop autonomous driving with language-action alignment. arXiv preprint arXiv:2503.09594 (2025)
44. Renz, K., Chen, L., Marcu, A.M., Hünermann, J., Hanotte, B., Karnsund, A., Shotton, J., Arani, E., Sinavski, O.: Carllava: Vision language models for camera-only closed-loop driving. arXiv preprint arXiv:2406.10165 (2024)
45. Rowe, L., de Schaetzen, R., Girgis, R., Pal, C., Paull, L.: Poutine: Vision-language-trajectory pre-training and reinforcement learning post-training enable robust end-to-end autonomous driving. arXiv preprint arXiv:2506.11234 (2025)
46. Shi, S., Jiang, L., Dai, D., Schiele, B.: Mtr++: Multi-agent motion prediction with symmetric scene modeling and guided intention querying. IEEE Transactions on Pattern Analysis and Machine Intelligence **46**(5), 3955–3971 (2024)
47. Sima, C., Renz, K., Chitta, K., Chen, L., Zhang, H., Xie, C., Beißwenger, J., Luo, P., Geiger, A., Li, H.: Drivelm: Driving with graph visual question answering. In: European conference on computer vision. pp. 256–274. Springer (2024)
48. 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. arXiv preprint arXiv:2402.12289 (2024)

49. Wang, S., Yu, Z., Jiang, X., Lan, S., Shi, M., Chang, N., Kautz, J., Li, Y., Alvarez, J.M.: Omnidrive: A holistic llm-agent framework for autonomous driving with 3d perception, reasoning and planning. arXiv preprint arXiv:2405.01533 (2024)
50. Wang, Y., Guizilini, V.C., Zhang, T., Wang, Y., Zhao, H., Solomon, J.: Detr3d: 3d object detection from multi-view images via 3d-to-2d queries. In: Conference on Robot Learning. pp. 180–191. PMLR (2022)
51. Wu, Z., Chen, X., Pan, Z., Liu, X., Liu, W., Dai, D., Gao, H., Ma, Y., Wu, C., Wang, B., Xie, Z., Wu, Y., Hu, K., Wang, J., Sun, Y., Li, Y., Piao, Y., Guan, K., Liu, A., Xie, X., You, Y., Dong, K., Yu, X., Zhang, H., Zhao, L., Wang, Y., Ruan, C.: Deepseek-vl2: Mixture-of-experts vision-language models for advanced multimodal understanding. `https://arxiv.org/abs/2412.10302` (2024), accessed: 2025-05-16
52. Xing, S., Qian, C., Wang, Y., Hua, H., Tian, K., Zhou, Y., Tu, Z.: Openemma: Open-source multimodal model for end-to-end autonomous driving. In: Proceedings of the Winter Conference on Applications of Computer Vision. pp. 1001–1009 (2025)
53. Yang, Z., Chai, Y., Jia, X., Li, Q., Shao, Y., Zhu, X., Su, H., Yan, J.: Drivemoe: Mixture-of-experts for vision-language-action model in end-to-end autonomous driving. arXiv preprint arXiv:2505.16278 (2025)
54. Zhou, X., Han, X., Yang, F., Ma, Y., Knoll, A.C.: Opendrivevla: Towards end-to-end autonomous driving with large vision language action model. arXiv preprint arXiv:2503.23463 (2025)
55. Zhou, Z., Cai, T., Zhao, S.Z., 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. arXiv preprint arXiv:2506.13757 (2025)
56. Zhou, Z., Wen, Z., Wang, J., Li, Y.H., Huang, Y.K.: Qcnext: A next-generation framework for joint multi-agent trajectory prediction. arXiv preprint arXiv:2306.10508 (2023)