Explicit Geometric Chain-of-Thought for Vision-Language-Action in Autonomous Driving

Xingtai Gui[1], **Yucheng Zhou**[1], **Dongqian Guo**[1], **Jiahao Gong**[2], **Feiyang Tan**[2], **Jianbing Shen**[1*]

[1] SKL-IOTSC, Department of AI, University of Macau
[2] Afari Intelligent Drive

Abstract

Vision-language-action (VLA) models have emerged as a promising paradigm for autonomous driving. However, existing VLA models still suffer from a fundamental mismatch: driving actions require precise 3D geometric cues, while visual-language understanding and reasoning are largely conducted in a 2D semantic space. In this paper, we propose GeoCoTDrive, an explicit geometric chain-of-thought framework that grounds geometry in a planning-oriented manner. GeoCoTDrive follows a "*think with 2D first, drive with dedicated 3D priors*" paradigm. It first grounds 2D regions corresponding to decision-critical cues, and then retrieves localized 3D priors by sampling features from a geometric foundation model within the grounded regions. These localized geometric features are interleaved into the autoregressive context to support the trajectory generation. To supervise this process, we introduce planning-relevant grounding, a new region-level grounding task that focuses on local spatial cues directly affecting ego planning decisions, and construct the PlanningGrounding dataset to endow VLAs with planning-oriented grounding capability. Experiments across multiple end-to-end autonomous driving benchmarks show that GeoCoTDrive consistently improves safety-critical planning performance, demonstrating the effectiveness of the explicit geometric chain-of-thought process for VLA-based planning. Code is available at `https://github.com/TabGuigui/GeoCoTDrive`.

1 Introduction

Vision-language models (VLMs) have been increasingly adopted in autonomous driving for question answering [67, 19, 46, 49, 44], scene understanding [24, 60, 8, 53, 10], and end-to-end planning [56, 11, 30, 15, 31, 75]. These efforts collectively demonstrate that natural language can provide an effective interface for structured supervision, interpretable decision making, and planning-oriented reasoning. However, a fundamental mismatch remains in VLM-based end-to-end autonomous driving: the target action space is inherently geometric, whereas the intermediate reasoning space of VLMs is largely semantic and lacks explicit geometric cues.

This mismatch becomes especially problematic in scenarios that require fine-grained local spatial reasoning, where planning decisions are governed by localized spatial constraints. Prior works have explored several strategies to mitigate this gap. Perception-based methods[56, 11, 73] introduce 3D structural representations [51, 25, 4] or BEV-style query features [21, 36, 13] to strengthen perception and planning relying on 3D perception supervision such as object detection or online

*Corresponding author. This work was supported by the Science and Technology Development Fund of Macau SAR (FDCT) under grants 0134/2025/RIA2 and 0102/2023/RIA2, and the Jiangyin Hi-tech Industrial Development Zone under the Taihu Innovation Scheme (EF2025-00003-SKL-IOTSC).

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

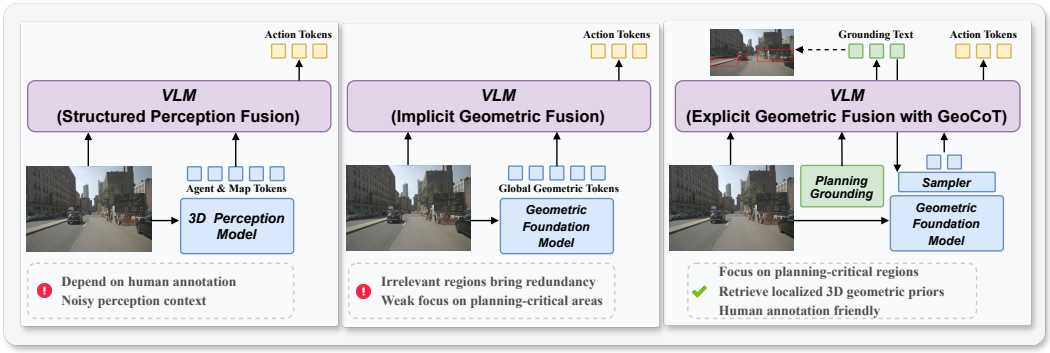

Figure 1: **Comparison of geometry integration paradigms for VLA**. (a) Structure-perception fusion injects agent and map tokens. (b) Geometric fusion incorporates global geometric tokens. (c) GeoCoTDrive introduces an explicit geometric chain-of-thought pipeline: VLM first grounds planning-relevant regions, then retrieves localized 3D geometric priors, and finally interleaves the grounding text with geometry tokens to condition trajectory generation.

mapping. Another line, motivated by recent advances in spatial intelligence [70, 27, 20], incorporates geometric priors into VLA models, either through explicit geometric inputs [40, 30] or multimodal representation alignment [55, 41]. However, these approaches primarily introduce geometry as global scene-level representations, rather than explicitly local and decision-critical spatial cues that directly govern planning.

Unlike prior methods that inject global geometric representations through implicit feature fusion, we propose *GeoCoTDrive*, an explicit geometric chain-of-thought framework that grounds geometry in a planning-oriented manner. As shown in Fig. 1, GeoCoTDrive follows a "*think with 2D first, drive with dedicated 3D priors*" paradigm. It first leverages the reasoning capability of VLMs to identify sparse 2D regions corresponding to local decision-critical cues, then retrieves dedicated 3D geometric priors by sampling features from a geometric foundation model within the grounded regions. These geometry tokens are interleaved into the autoregressive context to support the trajectory generation.

To enable explicit planning-oriented geometry grounding, we introduce *planning-relevant grounding*, a new task for end-to-end planning. Unlike conventional object-level grounding, which is typically built upon human-annotated objects with explicit semantic categories, planning-relevant grounding aims to identify region-level cues that directly affect the ego vehicle's future behavior. These cues go beyond foreground objects and include lane boundaries, conflict regions, occluded areas, and other local spatial constraints that govern planning. By shifting the grounding target from semantically defined objects to decision-critical regions, this task encourages the model to focus on the regions that are relevant to ego planning. Based on this formulation, we construct a semi-automatic annotation pipeline, yielding *PlanningGrounding*, a VQA-style grounding dataset containing 146K planning-relevant grounding annotations.

Given a front-view image, a driving command, and the ego status, GeoCoTDrive first autoregressively generates planning-relevant grounding results, localizing 2D regions that capture decision-critical cues. Meanwhile, the same image is encoded by a frozen geometric foundation model to produce dense geometric features. The generated grounding coordinates are then used to sample region-level geometric features, which are projected into the language-token space and interleaved back into the autoregressive context as localized geometry tokens. Conditioned on both the 2D reasoning trace and the retrieved 3D priors, GeoCoTDrive continues autoregressive decoding to output the future ego trajectory. During training, the grounding stage is supervised by PlanningGrounding annotations, while the planning stage is supervised by expert trajectories. During inference, both stages are executed sequentially in the same autoregressive pipeline.

Extensive experiments show that GeoCoTDrive consistently improves performance across multiple end-to-end autonomous driving benchmarks. By explicitly grounding planning-relevant regions and retrieving dedicated local 3D features, GeoCoTDrive enhances both trajectory accuracy and safety-critical planning metrics.

Our main contributions are summarized as follows:

- We propose GeoCoTDrive, an explicit geometric chain-of-thought framework for VLA-based end-to-end autonomous driving. It follows a "think with 2D first, drive with dedicated 3D priors" paradigm to bridge semantic reasoning and planning.
- We formulate planning-relevant grounding, a new region-level grounding task that focuses on local decision-critical cues directly affecting ego planning, and construct PlanningGrounding, a VQA-style grounding dataset with 146K annotations.
- We introduce an autoregressive geometry retrieval mechanism that samples localized 3D features from a geometric foundation model according to predicted grounding regions and interleaves them into the generation context for trajectory prediction.

2 Related Work

2.1 VLAs for Autonomous Driving

Recent progress in VLMs [38, 1, 6, 63] for autonomous driving has been largely driven by language-annotated driving datasets and dedicated model designs [67, 74, 8, 19]. Early efforts formulate driving scene understanding as visual question answering or instruction following [46, 53, 49, 44, 24, 10, 56, 43, 72, 62, 42]. Building on these language-supervised tasks, recent works extend VLMs toward vision-language-action modeling. Some methods directly formulate planning as autoregressive language generation, where the model predicts high-level decisions or textual future trajectories from multimodal driving inputs [15, 56, 48, 22, 75]. Other methods avoid using language models to generate precise trajectories directly, and leverage VLMs to provide high-level reasoning contexts for dedicated planning modules or action decoders [24, 31, 66, 11, 5, 29].

2.2 Spatial-aware VLAs for Autonomous Driving

Spatial awareness is increasingly recognized as a critical capability for VLMs/VLAs when downstream tasks require precise spatial decisions [3, 64, 39, 47]. Recent studies enhance spatial reasoning by incorporating geometric foundation models or geometry-aware representations through explicit geometric inputs [16, 61, 50, 26] or representation alignment [68, 27, 20, 70, 28]. These works validate the effectiveness of geometric priors in improving spatial understanding and action generation.

In autonomous driving, spatial representation is fundamental to scene understanding [10, 49], perception [57, 34, 58], prediction [69, 12, 71], and planning [21, 4, 65]. Recent VLA-based driving methods introduce human-annotated 3D supervision to learn structured scene representations [56, 11, 73, 32]. These 3D-structured representations provide important contextual cues for VLA reasoning and trajectory generation. Meanwhile, emerging methods further integrate geometry priors from pretrained foundation models [17, 54, 45, 37] into VLA-based driving systems. VGGDrive [55] injects cross-view VGGT [54] features into VLM hidden states through a hierarchical adaptive enabler. SpaceDrive [30] converts UniDepth-derived 3D coordinates [45] into universal positional encodings for spatially aware visual and coordinate tokens. LaST-VLA [41] distills geometric priors from VGGT into continuous latent CoT states for physically grounded planning. Despite their effectiveness, these methods primarily introduce global geometry priors without 2D reasoning derivation. In contrast, GeoCoTDrive explicitly constructs a planning-oriented reasoning path from identifying 2D decision-critical regions to retrieving localized 3D geometry.

3 Method

3.1 Preliminaries

Vision-Language-Action Model. In vision-language-action for autonomous driving, the core objective is to predict future ego trajectories from multimodal driving contexts based on pretrained VLMs. Given visual observations I and language contexts X including driving command, ego status, and textual prompts, the model predicts a future ego trajectory $\tau = \{(x_t, y_t, h_t)\}_{t=1}^{T}$ where $t \in \{1, \ldots, T\}$ denotes the prediction step within the future horizon, and $\tau_t = (x_t, y_t, h_t)$ denotes

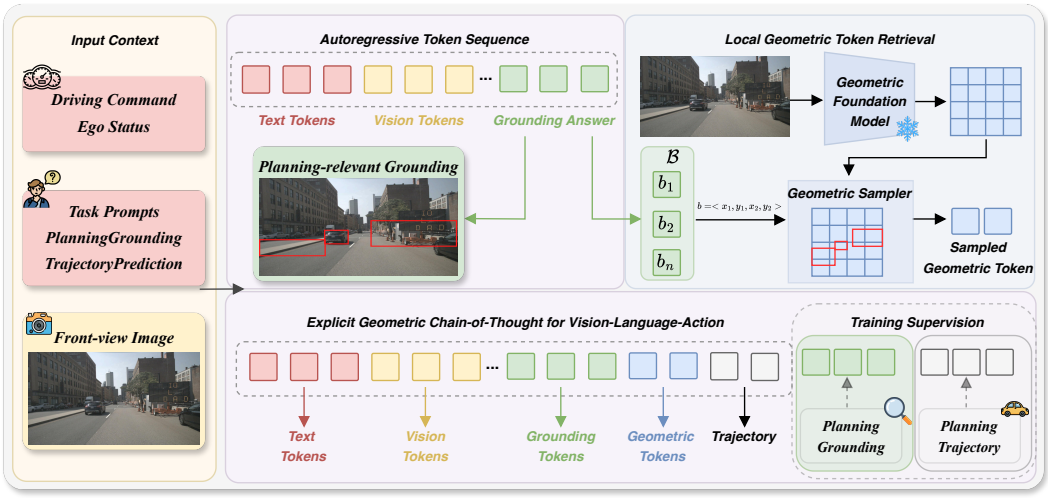

Figure 2: **Overview of GeoCoTDrive.** The model first grounds planning-critical 2D regions, retrieves localized geometry tokens from a geometric foundation model, and then interleaves them into the autoregressive context for trajectory generation.

the trajectory state at step t. A driving VLA models the planning distribution as:

$$P_\theta(\tau \mid X, I) = \prod_{t=1}^{T} P_\theta(\tau_t \mid \tau_{<t}, X, I), \tag{1}$$

Geometric Foundation Model. Geometric foundation models provide dense spatial representations learned from large-scale geometry-aware pretraining, offering complementary geometric priors. We instantiate the geometric feature extractor f_{geo} with two foundation models: Depth Anything V3 [37] and VGGT [54]. DepthAnythingV3 provides dense monocular geometry cues with strong robustness across diverse visual conditions, while VGGT offers 3D-aware latent representations learned from large-scale geometric pretraining. We use the intermediate backbone features from these frozen geometry models as geometric priors for geometric chain-of-thought. Specifically, we sample region-level geometric tokens from the feature map $G = f_{\text{geo}}(I)$.

3.2 GeoCoTDrive

GeoCoTDrive models VLA as an explicit geometric chain-of-thought process. Given the textual prompt X and the front-view image $I \in \mathbb{R}^{H\times W\times 3}$, the model first leverages the reasoning ability inherited from pretrained VLMs to generate a set of planning-relevant regions:

$$\mathcal{B} = \{b_i\}_{i=1}^{N} \sim P_\theta(\mathcal{B} \mid X, I), \tag{2}$$

where each region b_i corresponds to a local decision-critical cue. The generated regions provide an explicit intermediate reasoning trace that specifies where geometry should be retrieved. GeoCoTDrive then samples a set of localized geometric tokens from the dense geometric feature map:

$$\mathcal{Z}_{\mathcal{B}} = \text{Sample}(G, \mathcal{B}), \tag{3}$$

where G denotes the geometric feature map extracted from the same image. After the grounding output, the geometric tokens are interleaved into the autoregressive context, enabling GeoCoTDrive to continue decoding the trajectory conditioned on both 2D grounding and retrieved 3D priors:

$$P_\theta(\tau \mid X, I, \mathcal{B}, \mathcal{Z}_{\mathcal{B}}) = \prod_{t=1}^{T} P_\theta(\tau_t \mid \tau_{<t}, X, I, \mathcal{B}, \mathcal{Z}_{\mathcal{B}}). \tag{4}$$

Concretely, the grounding output is a set of 2D bounding boxes, where each box is represented by normalized coordinates:

$$b_i = (x_i^{\min}, y_i^{\min}, x_i^{\max}, y_i^{\max}), \qquad i = 1, \ldots, N. \tag{5}$$

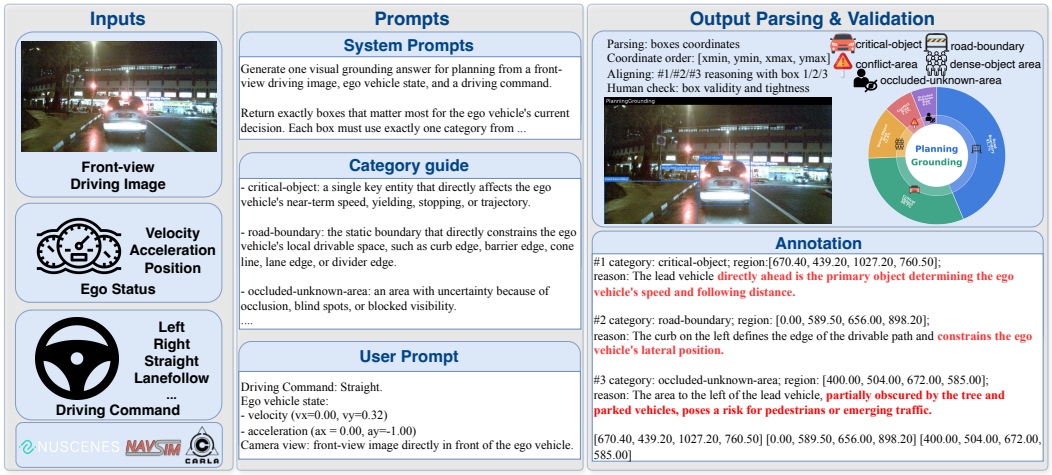

Figure 3: **Overview of the PlanningGrounding data construction pipeline.** The pipeline generates planning-oriented region annotations from driving scenes by identifying decision-critical visual cues. An annotation example and the overall category distribution of the dataset are provided.

Given the dense geometric feature map $G \in \mathbb{R}^{H_g \times W_g \times C_g}$ extracted by the geometric foundation model, GeoCoTDrive samples geometric features from each grounding box. A regular grid with K sampling points is constructed inside each box, and bilinear interpolation is applied over G:

$$\tilde{\mathbf{Z}}_i = S(G, b_i) \in \mathbb{R}^{K \times C_g}, \tag{6}$$

where $S(\cdot)$ denotes the region-based sampler. The resulting $\tilde{\mathbf{Z}}_i$ contains K localized geometric features from the i-th planning-relevant region.

The sampled features are then projected into the LLM space through a lightweight alignment head:

$$\mathbf{Z}_i = f_{\text{align}}(\tilde{\mathbf{Z}}_i) \in \mathbb{R}^{K \times d}, \tag{7}$$

where f_{align} is implemented as a MLP projection head and d is the hidden dimension of the pretrained VLM. The geometry tokens from all grounded regions are concatenated as:

$$\mathcal{Z}_{\mathcal{B}} = [\mathbf{Z}_1; \mathbf{Z}_2; \cdots; \mathbf{Z}_N] \in \mathbb{R}^{(N \times K) \times d}. \tag{8}$$

The concatenated geometry tokens $\mathcal{Z}_{\mathcal{B}}$ are interleaved into the autoregressive context after the grounding output. This enables the model to continue generation with explicit access to localized 3D evidence corresponding to the planning-critical regions it has just identified.

3.3 Training Objective

GeoCoTDrive is optimized with a standard autoregressive language modeling objective under teacher forcing. For each training sample, we construct a multimodal sequence by placing the grounding response, inserted geometry tokens, and trajectory response within the same autoregressive context. Let $\mathbf{s} = \{s_l\}_{l=1}^{L}$ denote the resulting sequence, the training objective is defined as:

$$\mathcal{L}_{\text{AR}} = -\frac{1}{|\mathcal{M}|} \sum_{l \in \mathcal{M}} \log P_\theta(s_l \mid s_{<l}), \tag{9}$$

where $\mathcal{M}$ denotes the set of supervised token positions, including both grounding and trajectory responses. Tokens corresponding to user prompts, image placeholders, and inserted geometry features are masked in the loss function.

4 Planning-oriented Grounding Dataset

End-to-end driving requires models to identify not only semantic objects, but also local regions that directly affect ego planning. Conventional object-level grounding is category-driven and may overlook planning-critical spatial cues such as conflict areas, occluded regions, and dense agent

Table 1: **Planning results on nuScenes valset.** Best and second-best results among VLM/VLA-based methods are highlighted. The result follows the evaluation protocol of OmniDrive [56].

Method	Ego Status		L2 (m) ↓				Collision (%) ↓				Intersection (%) ↓			
	BEV	Planner	1s	2s	3s	Avg.	1s	2s	3s	Avg.	1s	2s	3s	Avg.
Traditional / modular paradigm														
ST-P3[18]	–	–	1.59	2.64	3.73	2.65	0.69	3.62	8.39	4.23	2.53	8.17	14.40	8.37
UniAD[21]	✓	✓	0.20	0.42	0.75	0.46	0.02	0.25	0.84	0.37	0.20	1.33	3.24	1.59
VAD-Base[25]	✓	✓	0.17	0.34	0.60	0.37	0.04	0.27	0.67	0.33	0.21	2.13	5.06	2.47
AD-MLP[35]	–	✓	0.15	0.32	0.59	0.35	0.00	0.27	0.85	0.37	0.27	2.52	6.60	2.93
BEV-Planner++[35]	✓	✓	0.16	0.32	0.57	0.35	0.00	0.29	0.73	0.34	0.35	2.62	6.51	3.16
UAD[14]	✓	✓	0.13	0.28	0.48	0.30	0.00	0.12	0.55	0.22	0.10	0.80	2.38	1.13
VLM / VLA-based paradigm														
RDA-Driver[22]	✓	✓	0.23	0.73	1.54	0.80	**0.00**	0.13	0.83	0.32	–	–	–	–
OmniDrive[56]	✓	✓	**0.14**	0.29	0.55	0.33	**0.00**	0.13	0.78	0.30	0.56	2.48	5.96	3.00
ORION[11]	✓	–	0.17	0.31	0.55	0.34	0.05	0.25	0.80	0.37	–	–	–	–
VGGDrive[55]	–	✓	**0.14**	**0.28**	**0.51**	**0.31**	0.02	0.10	0.55	0.22	0.63	2.27	4.02	2.31
LaST-VLA[41]	–	✓	0.17	0.33	0.64	0.38	**0.00**	0.11	0.42	0.18	–	–	–	–
SpaceDrive[30]	–	✓	0.15	0.29	**0.51**	0.32	0.04	0.18	0.49	0.23	**0.22**	**0.80**	**2.79**	**1.27**
GeoCoTDrive	–	✓	**0.14**	0.29	0.53	0.32	**0.00**	**0.07**	**0.27**	**0.11**	0.37	1.44	4.25	2.02

clusters. To address this limitation, we formulate planning-relevant grounding, which aims to localize a compact set of decision-critical regions conditioned on the driving scene.

As shown in Fig. 3, each annotation sample is constructed from a front-view driving image, ego status, and a high-level driving command. A planning-oriented prompt is used to guide Gemini-3.1 [52] to identify the regions most relevant to the ego decision and assign each region to a planning-related category. The generated outputs are parsed into bounding boxes and category-reasoning pairs, followed by automatic format validation and human verification to ensure box validity and planning relevance. To cover diverse driving scenarios and benchmark settings, the raw driving data is collected from nuScenes [2], NAVSIM [9], and Bench2Drive [23].

Following this semi-automatic annotation pipeline, we construct PlanningGrounding, a VQA-style grounding dataset with 146K annotations. The annotations cover five planning-related categories: *critical-object*, *road-boundary*, *conflict-area*, *occluded-unknown-area*, and *dense-object-area*. The first two categories are closely related to object- and map-centric annotations commonly used in existing driving VLM/VLA datasets [56, 11], and therefore constitute a large proportion of the dataset. Beyond these conventional cues, PlanningGrounding further provides region-level annotations for planning-critical structures, including conflict-area, dense-object-area, and occluded-unknown-area. In addition, 0.6K manually refined samples are provided as the test set of the PlanningGrounding benchmark. For more information about this dataset, please refer to the Appendix.

5 Experiments

5.1 Experimental setup

Evaluation metrics. We evaluate the end-to-end planning performance on three autonomous driving benchmarks: nuScenes [2], NAVSIM [9] and Bench2Drive [23]. For nuScenes, we report open-loop planning performance using L2 distance error, collision rate, and intersection rate. For NAVSIM, we adopt the Predictive Driver Model Score (PDMS) as the closed-loop planning metric, which consists of five sub-scores: no-at-fault collisions (NC), drivable area compliance (DAC), time-to-collision (TTC), comfort (Comf.), and ego progress (EP). For Bench2Drive, we follow the official closed-loop evaluation protocol and report driving score (DS), success rate (SR). Together, these metrics provide a comprehensive assessment of planning accuracy, safety, comfort, and progress.

For planning-relevant grounding, we further evaluate GeoCoTDrive on the manually refined PlanningGrounding test set, which contains 0.6K samples and measures the model's ability to localize decision-critical regions before trajectory generation. The grounding performance is evaluated using Recall@N, average precision (AP), and mean intersection-over-union (mIoU).

Training details. GeoCoTDrive is instantiated with different base VLMs across benchmarks to follow the standard configurations of prior baseline methods. For nuScenes and Bench2Drive, we

Table 2: **Planning results on NAVSIM navtest split**. The result follows the evaluation protocol of official NAVSIMv1[9] with non-reactive simulation.

Method	Base Model	Sensor	Closed-loop Metrics (↑)					
			NC	DAC	EP	TTC	Comf.	PDMS
Traditional / modular paradigm								
TransFuser [7]	-	Image+LiDAR	97.8	92.6	78.9	92.0	99.9	83.8
PARA-Drive [59]	-	Image	97.9	92.4	79.3	93.0	99.8	84.0
Hydra-MDP [33]	-	Image+LiDAR	98.3	96.0	78.7	94.6	100.0	86.5
DiffusionDrive [36]	-	Image+LiDAR	98.2	96.2	82.2	94.7	100.0	88.1
VLM / VLA-based paradigm								
Baseline (Qwen)	Qwen2.5-VL-3B	Image	98.0	94.0	80.4	94.4	99.9	85.9
AutoVLA [75]	Qwen2.5-VL-3B	Image	96.9	92.4	75.8	88.1	99.9	80.5
ReCogDrive [31]	InternVL3-8B	Image	98.3	95.1	81.1	94.3	100.0	86.8
LaST-VLA [41]	InternVL3-8B	Image	**98.7**	95.4	80.5	**95.6**	100.0	87.3
GeoCoTDrive	Qwen2.5-VL-3B	Image	98.4	**95.6**	**81.6**	95.0	100.0	**87.6**

Table 3: **Closed-loop Planning results on Bench2Drive.** Driving Score (DS) and Success Rate (SR) are reported.

Method	Closed-loop Metrics (↑)	
	DS	SR
ORION [11]	77.74	54.62
SpaceDrive [30]	78.02	55.11
GeoCoTDrive	**78.68**	**55.42**

Table 4: **Grounding results on the PlanningGrounding test set.** We report Recall at different IoU thresholds, AP, and mIoU.

Method	R@0.3↑	R@0.5↑	R@0.7↑	AP↑	mIoU↑
LLaVA1.5-7B	41.39	24.49	9.25	25.04	27.98
Qwen2.5-VL-3B	46.27	33.86	9.84	30.61	30.46
OmniDrive	52.56	41.03	10.90	34.83	32.30
GeoCoTDrive	**56.82**	**42.73**	**21.04**	**39.89**	**38.93**

adopt LLaVA-1.5-7B as the base VLM, following OmniDrive [56] and ORION [11]. For NAVSIM, we use Qwen2.5-VL as the base VLM to align with recent VLA-based planning methods [31, 41]. For the trajectory decoder, nuScenes and NAVSIM use autoregressive text decoding, where future trajectory coordinates are generated as language action tokens. On Bench2Drive, we follow the diffusion planner design in ORION [11] to produce executable trajectories. This heterogeneous setup demonstrates that GeoCoTDrive can be integrated with different VLM backbones and trajectory-generation heads, highlighting its generality across diverse autonomous driving benchmarks.

For geometric feature extraction, we evaluate DepthAnythingV3 [37] and VGGT [54], which provide dense depth-aware and 3D-aware feature maps for localized geometry retrieval. The geometric foundation model is kept frozen. GeoCoTDrive is trained in two stages. In Stage I, we update the vision encoder and LLM with full QA training to adapt the VLM to autonomous-driving scenarios. In Stage II, we freeze the vision encoder and optimize the LLM together with the geometric alignment head, enabling localized geometric features to be aligned with the language space and interleaved into the planning context. More training details are provided in the Appendix.

5.2 Main results

Planning. Table 1 reports the open-loop planning results on nuScenes. GeoCoTDrive achieves competitive performance among VLA-based methods and shows favorable safety-related performance. Table 2 further evaluates GeoCoTDrive on the NAVSIM navtest set. Compared with the Qwen2.5-VL baseline, GeoCoTDrive achieves a modest PDMS improvement and improves safety-related sub-metrics. Among recent VLA-based planners, GeoCoTDrive remains competitive despite using a smaller VLM backbone. Table 3 shows that GeoCoTDrive improves over both the perception-centric and the global geometry fusion method with the Bench2Drive closed-loop evaluation.

Grounding. We evaluate grounding performance on the PlanningGrounding test set shown in Table 4. Driving-specialized models outperform general VLM baselines, suggesting that autonomous-driving instruction tuning helps localize planning-relevant cues. GeoCoTDrive achieves the best results across all metrics, with particularly clear gains under stricter IoU thresholds. These improvements indicate that GeoCoTDrive benefits from the PlanningGrounding dataset to localize decision-critical regions more precisely, providing more reliable spatial anchors for localized geometry retrieval and trajectory generation.

Table 5: **Ablation study on geometry integration strategies.** We compare the vanilla baseline, global geometry fusion, and the proposed GeoCoT.

Method	NAVSIM					nuScenes		
	NC	DAC	TTC	EP	PDMS	L2	CR	Intersection
Baseline	98.0	94.0	94.4	80.4	85.9	0.16 / 0.32 / 0.58	0.04 / 0.11 / 0.52	0.64 / 2.46 / 5.31
Global	97.9	94.1	94.2	80.2	85.8	0.16 / 0.31 / 0.57	0.02 / 0.17 / 0.56	0.48 / 2.34 / 4.90
GeoCoT	**98.4**	**95.6**	**95.0**	**81.6**	**87.6**	**0.14 / 0.29 / 0.53**	**0.00 / 0.07 / 0.27**	**0.37 / 1.44 / 4.25**

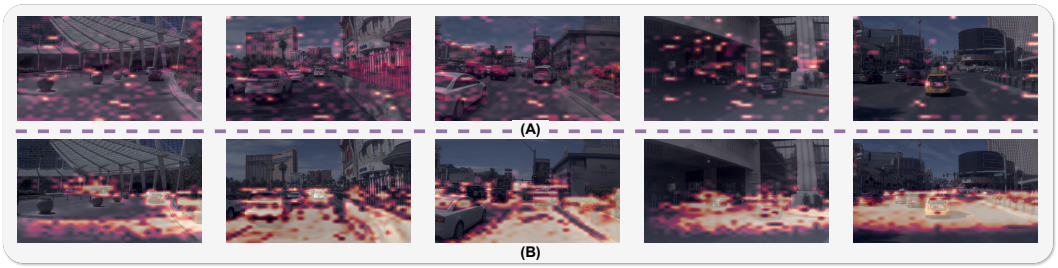

Figure 4: **Visualization of geometry token response with different fusion strategies.** (A) Global geometry tokens interleaved (B) GeoCoT retrieved geometric tokens interleaved.

5.3 Ablation study

Ablation on geometry integration. Table 5 compares different strategies for incorporating geometric information. Directly interleaving global geometry tokens yields limited gains, suggesting that indiscriminate geometry injection may introduce redundant or weakly relevant spatial cues. In contrast, GeoCoTDrive achieves more consistent improvements by retrieving localized geometric priors from planning-relevant grounded regions. These results indicate that geometry is more effective when conditioned on decision-critical regions, rather than injected as a global auxiliary stream.

We provide a qualitative comparison of geometric responses under different geometry insertion strategies. We calculated the attention output of the action token and geometric token. As shown in Fig. 4 (A), directly inserting global geometry tokens fails to focus on regions that are truly relevant to the driving decision. In contrast, Fig. 4 shows that GeoCoTDrive retrieves geometry tokens from planning-grounded regions and yields responses that are more aligned with decision-critical cues. This visualization further supports the advantage of planning-oriented geometry retrieval over indiscriminate global geometry injection.

Effect of grounding annotation source. Table 6 compares different grounding annotation sources for GeoCoTDrive. Perception labels provide comprehensive human-annotated scene elements, but they are primarily designed for general perception and are not necessarily aligned with planning. In contrast, PlanningGrounding focuses on regions that are directly relevant to driving decisions. The consistent improvements, especially on safety-related metrics, indicate that GeoCoTDrive benefits from localized geometry retrieval guided by planning-relevant grounding.

Effect of grounding quality on downstream planning. We perturb the generated grounding results during GeoCoT to analyze their impact on planning. As shown in Fig. 5, higher grounding mIoU consistently improves safety-related metrics, indicating that accurate planning-relevant grounding retrieves more reliable localized geometry. Although OmniDrive achieves comparable grounding mIoU to an intermediate GeoCoTDrive setting, its downstream safety metrics remain worse, suggesting that the gains come from explicitly coupling 2D grounding evidence with localized 3D priors.

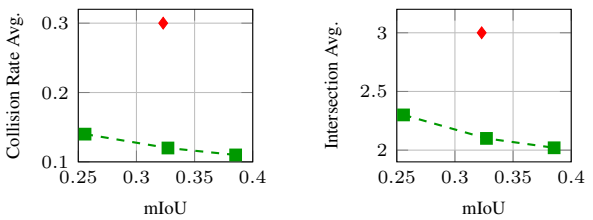

Figure 5: **Effect of grounding quality on downstream driving behavior.** Lower values indicate better safety performance. ▬■▬ denotes GeoCoTDrive with perturbed grounding quality, and ♦ denotes OmniDrive.

Table 6: **Comparison of different grounding annotation sources for GeoCoTDrive.** "Perception label" uses the human-annotated perception boxes.

Method	L2	CR	Intersection
GeoCoTDrive w/ Perception Label	0.17 / 0.30 / 0.55	0.02 / 0.12 / 0.39	0.50 / 2.20 / 4.98
GeoCoTDrive w/ PlanningGrounding	**0.14 / 0.29 / 0.53**	**0.00 / 0.07 / 0.27**	**0.37 / 1.44 / 4.25**

Table 7: **Effect of geometric foundation model.** Models are loaded from official checkpoints.

Method	Geometric Model	NC	DAC	TTC	EP	PDMS
Baseline	–	98.0	94.0	94.4	80.4	85.9
GeoCoTDrive	VGGT	98.3	95.3	94.6	**81.7**	87.2
GeoCoTDrive	DA3-LARGE	**98.4**	**95.6**	**95.0**	81.6	**87.6**

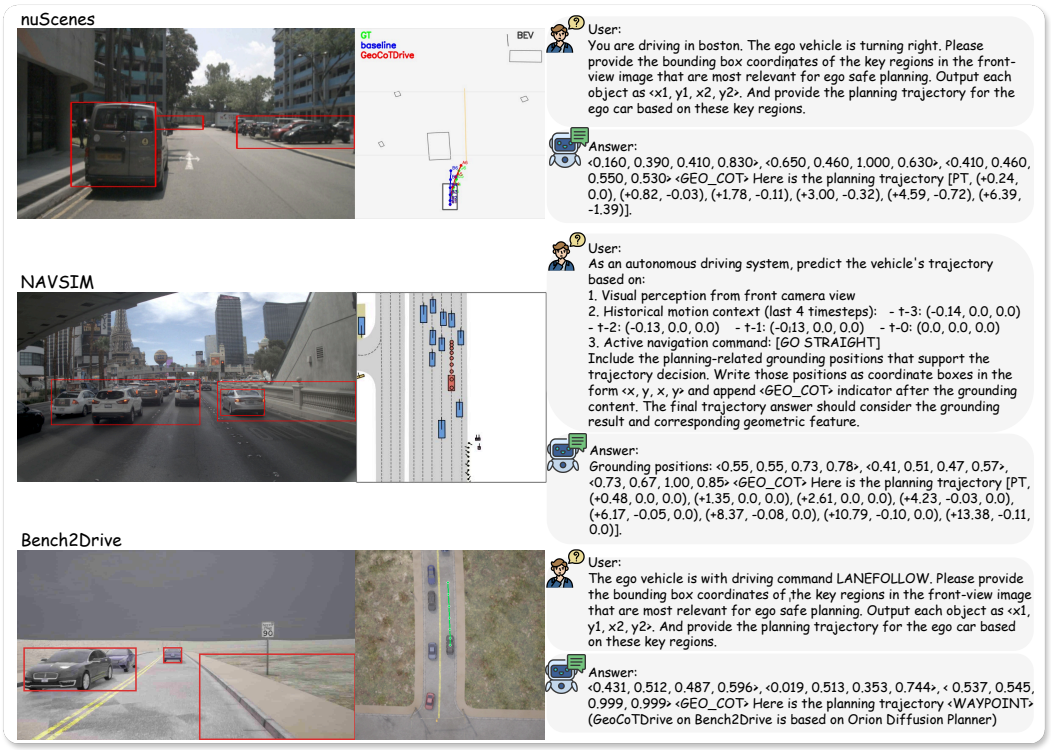

Figure 6: **Qualitative planning results of GeoCoTDrive on nuScenes, NAVSIM and Bench2Drive.**

Effect of geometric foundation model. Table 7 studies the impact of different geometric foundation models. Both VGGT and DA3-LARGE improve over the baseline, suggesting that localized geometry is generally beneficial for planning. DA3-LARGE achieves slightly better overall performance, while VGGT obtains a marginally higher EP. These results indicate that GeoCoTDrive is not tied to a specific geometry backbone and can benefit from diverse pretrained geometric priors.

5.4 Qualitative Results

Fig. 6 visualizes the qualitative planning results of GeoCoTDrive on nuScenes, NAVSIM, and Bench2Drive. Compared with the baseline, GeoCoTDrive shows stronger geometry-aware safety behavior. The visualization results across nuScenes, NAVSIM, and Bench2Drive further show that the proposed GeoCoT pipeline is consistently effective under evaluation protocols and driving scenarios. Additional qualitative results are provided in the Appendix.

6 Conclusion

We present GeoCoTDrive, an explicit geometric chain-of-thought framework for VLA-based end-to-end autonomous driving. GeoCoTDrive first grounds planning-relevant 2D regions, retrieves localized 3D geometric priors from a frozen geometric foundation model, and interleaves these geometry tokens into the autoregressive context for trajectory generation. We further introduce PlanningGrounding, a VQA-style grounding dataset that provides region-level supervision for decision-critical spatial cues. Experiments on nuScenes, NAVSIM, and Bench2Drive demonstrate that GeoCoTDrive improves planning performance, highlighting the importance of coupling planning-oriented 2D reasoning with localized 3D geometric priors for safer autonomous driving.

7 Acknowledgments and Disclosure of Funding

This work was supported by the Science and Technology Development Fund of Macau SAR (FDCT) under grants 0134/2025/RIA2 and 0102/2023/RIA2, and the Jiangyin Hi-tech Industrial Development Zone under the Taihu Innovation Scheme (EF2025-00003-SKL-IOTSC).

References

[1] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.

[2] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020.

[3] Boyuan Chen, Zhuo Xu, Sean Kirmani, Brain Ichter, Dorsa Sadigh, Leonidas Guibas, and Fei Xia. Spatialvlm: Endowing vision-language models with spatial reasoning capabilities. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14455–14465, 2024.

[4] Shaoyu Chen, Bo Jiang, Hao Gao, Bencheng Liao, Qing Xu, Qian Zhang, Chang Huang, Wenyu Liu, and Xinggang Wang. Vadv2: End-to-end vectorized autonomous driving via probabilistic planning. *arXiv preprint arXiv:2402.13243*, 2024.

[5] Xuesong Chen, Linjiang Huang, Tao Ma, Rongyao Fang, Shaoshuai Shi, and Hongsheng Li. Solve: Synergy of language-vision and end-to-end networks for autonomous driving. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 12068–12077, 2025.

[6] Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, 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*, pages 24185–24198, 2024.

[7] Kashyap Chitta, Aditya Prakash, Bernhard Jaeger, Zehao Yu, Katrin Renz, and Andreas Geiger. Transfuser: Imitation with transformer-based sensor fusion for autonomous driving. *IEEE transactions on pattern analysis and machine intelligence*, 45(11):12878–12895, 2022.

[8] Yixin Cui, Haotian Lin, Shuo Yang, Yixiao Wang, Yanjun Huang, and Hong Chen. Chain-of-thought for autonomous driving: A comprehensive survey and future prospects. *arXiv preprint arXiv:2505.20223*, 2025.

[9] Daniel Dauner, Marcel Hallgarten, Tianyu Li, Xinshuo Weng, Zhiyu Huang, Zetong Yang, Hongyang Li, Igor Gilitschenski, Boris Ivanovic, Marco Pavone, et al. Navsim: Data-driven non-reactive autonomous vehicle simulation and benchmarking. *Advances in Neural Information Processing Systems*, 37:28706–28719, 2024.

[10] Xinpeng Ding, Jianhua Han, Hang Xu, Xiaodan Liang, Wei Zhang, and Xiaomeng Li. Holistic autonomous driving understanding by bird's-eye-view injected multi-modal large models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13668–13677, 2024.

[11] Haoyu Fu, Diankun Zhang, Zongchuang Zhao, Jianfeng Cui, Dingkang Liang, Chong Zhang, Dingyuan Zhang, Hongwei Xie, Bing Wang, and Xiang Bai. Orion: A holistic end-to-end autonomous driving framework by vision-language instructed action generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 24823–24834, 2025.

[12] Xingtai Gui, Tengteng Huang, Haonan Shao, Haotian Yao, and Chi Zhang. Fiptr: A simple yet effective transformer framework for future instance prediction in autonomous driving. In *European conference on computer vision*, pages 19–35. Springer, 2024.

[13] Xingtai Gui, Jianbo Zhao, Wencheng Han, Jikai Wang, Jiahao Gong, Feiyang Tan, Cheng-zhong Xu, and Jianbing Shen. Trajdiff: End-to-end autonomous driving without perception annotation. *arXiv preprint arXiv:2512.00723*, 2025.

[14] Mingzhe Guo, Zhipeng Zhang, Yuan He, Ke Wang, Liping Jing, and Haibin Ling. End-to-end autonomous driving without costly modularization and 3d manual annotation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

[15] Wencheng Han, Dongqian Guo, Cheng-Zhong Xu, and Jianbing Shen. Dme-driver: Integrating human decision logic and 3d scene perception in autonomous driving. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 3347–3355, 2025.

[16] Yining Hong, Haoyu Zhen, Peihao Chen, Shuhong Zheng, Yilun Du, Zhenfang Chen, and Chuang Gan. 3d-llm: Injecting the 3d world into large language models. *Advances in Neural Information Processing Systems*, 36:20482–20494, 2023.

[17] Mu Hu, Wei Yin, Chi Zhang, Zhipeng Cai, Xiaoxiao Long, Hao Chen, Kaixuan Wang, Gang Yu, Chunhua Shen, and Shaojie Shen. Metric3d v2: A versatile monocular geometric foundation model for zero-shot metric depth and surface normal estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):10579–10596, 2024.

[18] Shengchao Hu, Li Chen, Penghao Wu, Hongyang Li, Junchi Yan, and Dacheng Tao. St-p3: End-to-end vision-based autonomous driving via spatial-temporal feature learning. In *European Conference on Computer Vision*, pages 533–549. Springer, 2022.

[19] Tianshuai Hu, Xiaolu Liu, Song Wang, Yiyao Zhu, Ao Liang, Lingdong Kong, Guoyang Zhao, Zeying Gong, Jun Cen, Zhiyu Huang, et al. Vision-language-action models for autonomous driving: Past, present, and future. *arXiv preprint arXiv:2512.16760*, 2025.

[20] Wenbo Hu, Jingli Lin, Yilin Long, Yunlong Ran, Lihan Jiang, Yifan Wang, Chenming Zhu, Runsen Xu, Tai Wang, and Jiangmiao Pang. G^2vlm: Geometry grounded vision language model with unified 3d reconstruction and spatial reasoning. *arXiv preprint arXiv:2511.21688*, 2025.

[21] Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, et al. Planning-oriented autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 17853–17862, 2023.

[22] Zhijian Huang, Tao Tang, Shaoxiang Chen, Sihao Lin, Zequn Jie, Lin Ma, Guangrun Wang, and Xiaodan Liang. Making large language models better planners with reasoning-decision alignment. In *European Conference on Computer Vision*, pages 73–90. Springer, 2024.

[23] Xiaosong Jia, Zhenjie Yang, Qifeng Li, Zhiyuan Zhang, and Junchi Yan. Bench2drive: Towards multi-ability benchmarking of closed-loop end-to-end autonomous driving. *Advances in Neural Information Processing Systems*, 37:819–844, 2024.

[24] Bo Jiang, Shaoyu Chen, Bencheng Liao, Xingyu Zhang, Wei Yin, Qian Zhang, Chang Huang, Wenyu Liu, and Xinggang Wang. Senna: Bridging large vision-language models and end-to-end autonomous driving. *arXiv preprint arXiv:2410.22313*, 2024.

[25] Bo Jiang, Shaoyu Chen, Qing Xu, Bencheng Liao, Jiajie Chen, Helong Zhou, Qian Zhang, Wenyu Liu, Chang Huang, and Xinggang Wang. Vad: Vectorized scene representation for efficient autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8340–8350, 2023.

[26] Chengmeng Li, Junjie Wen, Yaxin Peng, Yan Peng, and Yichen Zhu. Pointvla: Injecting the 3d world into vision-language-action models. *IEEE Robotics and Automation Letters*, 11(3):2506–2513, 2026.

[27] Fuhao Li, Wenxuan Song, Han Zhao, Jingbo Wang, Pengxiang Ding, Donglin Wang, Long Zeng, and Haoang Li. Spatial forcing: Implicit spatial representation alignment for vision-language-action model. *arXiv preprint arXiv:2510.12276*, 2025.

[28] Haoyuan Li, Qihang Cao, Tao Tang, Kun Xiang, Zihan Guo, Jianhua Han, Hang Xu, and Xiaodan Liang. Thinking with geometry: Active geometry integration for spatial reasoning. *arXiv preprint arXiv:2602.06037*, 2026.

[29] Jingyu Li, Junjie Wu, Dongnan Hu, Xiangkai Huang, Bin Sun, Zhihui Hao, Xianpeng Lang, Xiatian Zhu, and Li Zhang. Sgdrive: Scene-to-goal hierarchical world cognition for autonomous driving. *arXiv preprint arXiv:2601.05640*, 2026.

[30] Peizheng Li, Zhenghao Zhang, David Holtz, Hang Yu, Yutong Yang, Yuzhi Lai, Rui Song, Andreas Geiger, and Andreas Zell. Spacedrive: Infusing spatial awareness into vlm-based autonomous driving. *arXiv preprint arXiv:2512.10719*, 2, 2025.

[31] Yongkang Li, Kaixin Xiong, Xiangyu Guo, Fang Li, Sixu Yan, Gangwei Xu, Lijun Zhou, Long Chen, Haiyang Sun, Bing Wang, et al. Recogdrive: A reinforced cognitive framework for end-to-end autonomous driving. *arXiv preprint arXiv:2506.08052*, 2025.

[32] Yongkang Li, Lijun Zhou, Sixu Yan, Bencheng Liao, Tianyi Yan, Kaixin Xiong, Long Chen, Hongwei Xie, Bing Wang, Guang Chen, et al. Unidrivevla: Unifying understanding, perception, and action planning for autonomous driving. *arXiv preprint arXiv:2604.02190*, 2026.

[33] Zhenxin Li, Kailin Li, Shihao Wang, Shiyi Lan, Zhiding Yu, Yishen Ji, Zhiqi Li, Ziyue Zhu, Jan Kautz, Zuxuan Wu, et al. Hydra-mdp: End-to-end multimodal planning with multi-target hydra-distillation. *arXiv preprint arXiv:2406.06978*, 2024.

[34] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Qiao Yu, and Jifeng Dai. Bevformer: learning bird's-eye-view representation from lidar-camera via spatiotemporal transformers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(3):2020–2036, 2024.

[35] Zhiqi Li, Zhiding Yu, Shiyi Lan, Jiahan Li, Jan Kautz, Tong Lu, and Jose M Alvarez. 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*, pages 14864–14873, 2024.

[36] Bencheng Liao, Shaoyu Chen, Haoran Yin, Bo Jiang, Cheng Wang, Sixu Yan, Xinbang Zhang, Xiangyu Li, Ying Zhang, Qian Zhang, et al. Diffusiondrive: Truncated diffusion model for end-to-end autonomous driving. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 12037–12047, 2025.

[37] Haotong Lin, Sili Chen, Junhao Liew, Donny Y Chen, Zhenyu Li, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth anything 3: Recovering the visual space from any views. *arXiv preprint arXiv:2511.10647*, 2025.

[38] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023.

[39] Weichen Liu, Qiyao Xue, Haoming Wang, Xiangyu Yin, Boyuan Yang, and Wei Gao. Spatial reasoning in multimodal large language models: A survey of tasks, benchmarks and methods. *arXiv preprint arXiv:2511.15722*, 2025.

[40] Zhe Liu, Runhui Huang, Rui Yang, Siming Yan, Zining Wang, Lu Hou, Di Lin, Xiang Bai, and Hengshuang Zhao. Drivepi: Spatial-aware 4d mllm for unified autonomous driving understanding, perception, prediction and planning. *arXiv preprint arXiv:2512.12799*, 2025.

[41] Yuechen Luo, Fang Li, Shaoqing Xu, Yang Ji, Zehan Zhang, Bing Wang, Yuannan Shen, Jianwei Cui, Long Chen, Guang Chen, et al. Last-vla: Thinking in latent spatio-temporal space for vision-language-action in autonomous driving. *arXiv preprint arXiv:2603.01928*, 2026.

[42] Yuechen Luo, Fang Li, Shaoqing Xu, Zhiyi Lai, Lei Yang, Qimao Chen, Ziang Luo, Zixun Xie, Shengyin Jiang, Jiaxin Liu, et al. Adathinkdrive: Adaptive thinking via reinforcement learning for autonomous driving. *arXiv preprint arXiv:2509.13769*, 2025.

[43] Yingzi Ma, Yulong Cao, Jiachen Sun, Marco Pavone, and Chaowei Xiao. Dolphins: Multimodal language model for driving. In *European Conference on Computer Vision*, pages 403–420. Springer, 2024.

[44] Ana-Maria Marcu, Long Chen, Jan Hünermann, Alice Karnsund, Benoit Hanotte, Prajwal Chidananda, Saurabh Nair, Vijay Badrinarayanan, Alex Kendall, Jamie Shotton, et al. Lingoqa: Visual question answering for autonomous driving. In *European Conference on Computer Vision*, pages 252–269. Springer, 2024.

[45] Luigi Piccinelli, Christos Sakaridis, Yung-Hsu Yang, Mattia Segu, Siyuan Li, Wim Abbeloos, and Luc Van Gool. Unidepthv2: Universal monocular metric depth estimation made simpler. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

[46] Tianwen Qian, Jingjing Chen, Linhai Zhuo, Yang Jiao, and Yu-Gang Jiang. Nuscenes-qa: A multi-modal visual question answering benchmark for autonomous driving scenario. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 4542–4550, 2024.

[47] Delin Qu, Haoming Song, Qizhi Chen, Yuanqi Yao, Xinyi Ye, Yan Ding, Zhigang Wang, JiaYuan Gu, Bin Zhao, Dong Wang, et al. Spatialvla: Exploring spatial representations for visual-language-action model. *arXiv preprint arXiv:2501.15830*, 2025.

[48] Hao Shao, Yuxuan Hu, Letian Wang, Guanglu Song, Steven L Waslander, Yu Liu, and Hongsheng Li. Lmdrive: Closed-loop end-to-end driving with large language models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15120–15130, 2024.

[49] Chonghao Sima, Katrin Renz, Kashyap Chitta, Li Chen, Hanxue Zhang, Chengen Xie, Jens Beißwenger, Ping Luo, Andreas Geiger, and Hongyang Li. Drivelm: Driving with graph visual question answering. In *European conference on computer vision*, pages 256–274. Springer, 2024.

[50] Lin Sun, Bin Xie, Yingfei Liu, Hao Shi, Tiancai Wang, and Jiale Cao. Geovla: Empowering 3d representations in vision-language-action models. *arXiv preprint arXiv:2508.09071*, 2025.

[51] Wenchao Sun, Xuewu Lin, Yining Shi, Chuang Zhang, Haoran Wu, and Sifa Zheng. Sparsedrive: End-to-end autonomous driving via sparse scene representation. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8795–8801. IEEE, 2025.

[52] Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

[53] Kexin Tian, Jingrui Mao, Yunlong Zhang, Jiwan Jiang, Yang Zhou, and Zhengzhong Tu. Nuscenes-spatialqa: A spatial understanding and reasoning benchmark for vision-language models in autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4567–4576, 2025.

[54] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 5294–5306, 2025.

[55] Jie Wang, Guang Li, Zhijian Huang, Chenxu Dang, Hangjun Ye, Yahong Han, and Long Chen. Vggdrive: Empowering vision-language models with cross-view geometric grounding for autonomous driving. *arXiv preprint arXiv:2602.20794*, 2026.

[56] Shihao Wang, Zhiding Yu, Xiaohui Jiang, Shiyi Lan, Min Shi, Nadine Chang, Jan Kautz, Ying Li, and Jose M Alvarez. Omnidrive: A holistic vision-language dataset for autonomous driving with counterfactual reasoning. In *Proceedings of the computer vision and pattern recognition conference*, pages 22442–22452, 2025.

[57] Yue Wang, Vitor Campagnolo Guizilini, Tianyuan Zhang, Yilun Wang, Hang Zhao, and Justin Solomon. Detr3d: 3d object detection from multi-view images via 3d-to-2d queries. In *Conference on robot learning*, pages 180–191. PMLR, 2022.

[58] Yi Wei, Linqing Zhao, Wenzhao Zheng, Zheng Zhu, Jie Zhou, and Jiwen Lu. Surroundocc: Multi-camera 3d occupancy prediction for autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 21729–21740, 2023.

[59] Xinshuo Weng, Boris Ivanovic, Yan Wang, Yue Wang, and Marco Pavone. Para-drive: Parallelized architecture for real-time autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15449–15458, 2024.

[60] Shaoyuan Xie, Lingdong Kong, Yuhao Dong, Chonghao Sima, Wenwei Zhang, Qi Alfred Chen, Ziwei Liu, and Liang Pan. Are vlms ready for autonomous driving? an empirical study from the reliability, data and metric perspectives. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6585–6597, 2025.

[61] Runsen Xu, Xiaolong Wang, Tai Wang, Yilun Chen, Jiangmiao Pang, and Dahua Lin. Pointllm: Empowering large language models to understand point clouds. In *European Conference on Computer Vision*, pages 131–147. Springer, 2024.

[62] Zhenhua Xu, Yujia Zhang, Enze Xie, Zhen Zhao, Yong Guo, Kwan-Yee K Wong, Zhenguo Li, and Hengshuang Zhao. Drivegpt4: Interpretable end-to-end autonomous driving via large language model. *IEEE Robotics and Automation Letters*, 9(10):8186–8193, 2024.

[63] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

[64] Jihan Yang, Shusheng Yang, Anjali W Gupta, Rilyn Han, Li Fei-Fei, and Saining Xie. Thinking in space: How multimodal large language models see, remember, and recall spaces. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 10632–10643, 2025.

[65] Yu Yang, Jianbiao Mei, Yukai Ma, Siliang Du, Wenqing Chen, Yijie Qian, Yuxiang Feng, and Yong Liu. Driving in the occupancy world: Vision-centric 4d occupancy forecasting and planning via world models for autonomous driving. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 9327–9335, 2025.

[66] Zhenjie Yang, Yilin Chai, Xiaosong Jia, Qifeng Li, Yuqian Shao, Xuekai Zhu, Haisheng Su, and Junchi Yan. Drivemoe: Mixture-of-experts for vision-language-action model in end-to-end autonomous driving. *arXiv preprint arXiv:2505.16278*, 2025.

[67] Zhenjie Yang, Xiaosong Jia, Hongyang Li, and Junchi Yan. Llm4drive: A survey of large language models for autonomous driving. *arXiv preprint arXiv:2311.01043*, 2023.

[68] Tianyuan Yuan, Yicheng Liu, Chenhao Lu, Zhuoguang Chen, Tao Jiang, and Hang Zhao. Depthvla: Enhancing vision-language-action models with depth-aware spatial reasoning. *arXiv preprint arXiv:2510.13375*, 2025.

[69] Yunpeng Zhang, Zheng Zhu, Wenzhao Zheng, Junjie Huang, Guan Huang, Jie Zhou, and Jiwen Lu. Beverse: Unified perception and prediction in birds-eye-view for vision-centric autonomous driving. *arXiv preprint arXiv:2205.09743*, 2022.

[70] Duo Zheng, Shijia Huang, Yanyang Li, and Liwei Wang. Learning from videos for 3d world: Enhancing mllms with 3d vision geometry priors. *arXiv preprint arXiv:2505.24625*, 2025.

[71] Wenzhao Zheng, Weiliang Chen, Yuanhui Huang, Borui Zhang, Yueqi Duan, and Jiwen Lu. Occworld: Learning a 3d occupancy world model for autonomous driving. In *European conference on computer vision*, pages 55–72. Springer, 2024.

[72] Xin Zhou, Dingkang Liang, Sifan Tu, Xiwu Chen, Yikang Ding, Dingyuan Zhang, Feiyang Tan, Hengshuang Zhao, and Xiang Bai. Hermes: A unified self-driving world model for simultaneous 3d scene understanding and generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 27817–27827, 2025.

[73] Xingcheng Zhou, Xuyuan Han, Feng Yang, Yunpu Ma, and Alois C Knoll. Opendrivevla: Towards end-to-end autonomous driving with large vision language action model. *arXiv preprint arXiv:2503.23463*, 2025.

[74] Xingcheng Zhou, Mingyu Liu, Ekim Yurtsever, Bare Luka Zagar, Walter Zimmer, Hu Cao, and Alois C Knoll. Vision language models in autonomous driving: A survey and outlook. *IEEE Transactions on Intelligent Vehicles*, 2024.

[75] Zewei Zhou, Tianhui Cai, Seth Z Zhao, Yun Zhang, Zhiyu Huang, Bolei Zhou, and Jiaqi Ma. 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.

Appendix

The Appendix is organized as follows:

- Sec. A: comparison with existing driving VQA datasets on grounding supervision.
- Sec. B: detailed analysis of the proposed PlanningGrounding dataset.
- Sec. C: detailed training and inference implementation of GeoCoTDrive.
- Sec. D: more ablation studies.
- Sec. E: more qualitative results.
- Sec. F: limitations of GeoCoTDrive.

A Autonomous Driving VQA

Table 8: **Comparison with existing driving VQA datasets on grounding supervision.**

Dataset	Main supervision	Grounding granularity
NuScenes-QA [46]	Template-based VQA from structured scene graphs.	Object-level attributes and relations derived from 3D detection annotations.
DriveLM [49]	Graph Visual Question Answering for perception, prediction, planning, behavior, and motion.	Object-level graph nodes and interaction-level reasoning.
LingoQA [44]	Video-based driving QA with natural-language answers and explanations.	Mainly language-level supervision without explicit box-level grounding.
OmniDrive [56]	Counterfactual QA using simulated trajectories, expert trajectories, 3D objects, and map elements.	3D object/map-centric detection and trajectory-level consequence reasoning.
PlanningGrounding	Command-conditioned grounding of planning-relevant visual regions.	Region-level 2D boxes covering objects, boundaries, conflict zones, occlusions, and dense object groups.

Recent autonomous-driving VQA datasets have significantly advanced language-based scene understanding, reasoning, and planning. However, the grounding supervision of these datasets is usually designed around object-level perception annotations, rather than explicit region-level cues related to the ego vehicle's planning demand. We summarize the main differences in Table 8.

Unlike conventional object-centric grounding, PlanningGrounding is designed to identify the regions that most directly affect the ego vehicle's planning decision. This distinction is important because the planning decisions are often constrained not only by individual traffic participants, but also by local road boundaries, conflict driving zones, or uncertain regions. Therefore, PlanningGrounding complements existing driving VQA resources by providing explicit, command-aware grounding supervision for planning-oriented reasoning.

B Analysis of PlanningGrounding

PlanningGrounding is constructed to supervise explicit planning-oriented visual grounding. Given a front-view driving image, driving command, and optional ego status, the task is to localize the most decision-critical regions that affect the planning decision. Each grounding region is assigned one of five categories: *critical-object*, *road-boundary*, *conflict-area*, *occluded-unknown-area*, and *dense-object-area*. These categories cover both object-level and region-level spatial cues required for safe trajectory generation.

- **Critical object** denotes a single key entity that directly affects the ego vehicle's near-term speed, yielding behavior, stopping decision, or trajectory selection. Typical examples include a leading vehicle requiring deceleration, a pedestrian near the ego path, a cyclist around the turning area, a traffic cone blocking the lane, or a traffic light/sign that immediately constrains the maneuver. An object is annotated as critical only when its influence is directly tied to the current command and local planning decision.
- **Road boundary** refers to an immediate static boundary that constrains the ego vehicle's local drivable space. This category includes lane edges, curbs, barriers, dividers, cone lines, road edges, and other visible structures that define the lateral feasible region.
- **Conflict area** denotes a compact local region where the ego vehicle's intended path may interact with other traffic flows or potential agent motions, such as a merge point, crossing zone, or turn conflict entrance. Unlike critical object, this category describes a spatial zone rather than a single entity, encouraging the model to reason about where potential interactions may occur.
- **Occluded unknown area** denotes a region with meaningful uncertainty caused by occlusion, blind spots, or blocked visibility. Such regions commonly appear behind parked vehicles, large trucks, roadside buildings, vegetation, or sharp turns. Although no explicit object may be visible, these regions can still require conservative planning, such as slowing down before entering a blind spot or preparing to yield.
- **Dense object area** represents a compact cluster of objects whose collective presence affects planning. Examples include a group of pedestrians, a queue of vehicles, a cluster of cones, or multiple parked vehicles narrowing the drivable corridor. This category is used when the planning-relevant cue is better represented by a region containing multiple nearby entities rather than by a single object.

The above categories are not intended to provide exhaustive perception labels. Instead, they define a sparse set of command-relevant regions that directly influence near-term planning. Unlike generic object detection or visual grounding, PlanningGrounding selects annotation targets according to planning relevance, covering not only salient objects but also road boundaries, conflict zones, occluded areas, and dense object regions that constrain safe trajectory generation.

Table 9: **Dataset composition of PlanningGrounding.** The table summarizes the number of grounding QA samples from each source dataset and the distribution of five planning-oriented grounding categories. "K" denotes thousands of samples.

Source Dataset	# Samples	Category Distribution (%)				
		Critical Object	Road Boundary	Conflict	Occluded Unknown	Dense Object
nuScenes	28K	30.68	43.65	7.05	6.14	12.45
NAVSIM	102K	34.47	34.05	6.86	3.99	20.60
Bench2Drive	16K	38.31	47.16	2.99	3.48	8.05
Total	146K	34.16	37.33	6.47	4.35	17.66

Table 9 summarizes the source composition and category distribution of PlanningGrounding. Considering the diversity of driving scenarios, data sources, and evaluation protocols, PlanningGrounding is constructed from three widely studied autonomous-driving planning benchmarks: nuScenes, NAVSIM, and Bench2Drive. This combination enables PlanningGrounding to cover a broad range of planning-relevant cues.

In Fig. 8, we show examples from PlanningGrounding-nuScenes and compare the original human-annotated perception labels with our planning-relevant grounding annotations. Fig. 9 further provides examples from PlanningGrounding-NAVSIM and PlanningGrounding-Bench2Drive.

C Implementation Details

VLM backbone. GeoCoTDrive adopts different VLM backbones according to the benchmark setting. For nuScenes and Bench2Drive, LLaVA-1.5-7B is used as the base VLM, while Qwen2.5-

Table 10: **Two-stage training structure of GeoCoTDrive.** Stage-I adapts the vision-language backbone to autonomous-driving QA and planning knowledge, while Stage-II trains GeoCoT planning.

Stage	Trainable Modules	Objective	Epochs	Peak LR
Stage-I	Vision + LLM	Full QA training	3	4×10^{-5}
Stage-II	LLM + Geometric Aligner	GeoCoT	3	2×10^{-5}

Table 11: **Ablation on the sampling grid size for localized geometry retrieval.** The grid size denotes the number of sampled geometric features within each grounded region.

Grid Size	Geo. Tokens	NC	DAC	TTC	PDMS
1×1	1	98.2	94.7	94.5	86.5
2×2	4	98.3	95.1	94.6	87.0
4×4	16	98.4	95.6	95.0	87.6
5×5	25	98.3	95.6	95.0	87.6

VL-3B is used for NAVSIM. Across all settings, the model takes the front-view image, ego status, and high-level navigation command as input, and autoregressively generates planning-relevant grounding results followed by the planning answer.

Geometric foundation model. Two geometric foundation models are used to extract geometric priors: Depth Anything V3 with the METRIC-LARGE configuration and the vanilla version of VGGT. Both models are kept frozen. For each model, the feature map from the last backbone layer is used as the source of geometric tokens. Given a grounding region, localized geometric features are sampled from this feature map and projected into the language embedding space.

Two-stage training. The training process contains two stages. The first stage performs full QA training to adapt the VLM to autonomous-driving knowledge. Specifically, nuScenes uses OmniDrive together with PlanningGrounding, NAVSIM uses NAVSIM-Traj together with PlanningGrounding, and Bench2Drive uses B2D-Chat together with PlanningGrounding. In this stage, both the vision encoder and the LLM are updated, allowing the model to acquire driving-related visual understanding and planning knowledge. The model is trained for 3 epochs with a cosine learning-rate scheduler, where the peak learning rate is set to 4×10^{-5}.

The second stage performs geometry-interleaved planning training. Planning answers and grounding annotations are merged into a single-turn conversation, where the model is required to first generate planning-relevant grounding results and then predict the planning trajectory. A special token `<GEO_COT>` is inserted after the grounding output as a placeholder for geometric reasoning. During training, the ground-truth grounding boxes are used as the sampler input to retrieve localized geometric features. The sampled geometric tokens are then interleaved into the autoregressive sequence at the `<GEO_COT>` position, enabling subsequent trajectory prediction to condition on localized 3D priors. This stage is also trained for 3 epochs with a cosine learning-rate scheduler, where the peak learning rate is set to 2×10^{-5}. For the Bench2Drive benchmark, we follow the planner implementation of ORION. We remove the detection and mapping head, and use the VLM tokens and localized geometric tokens from the front-view image as conditions for the planner. All experiments are conducted with 8 A800 GPUs.

Inference. The inference process follows the same question format as training. The model first autoregressively generates grounding results. Once the `<GEO_COT>` token is generated, the predicted grounding boxes are parsed and used to sample localized geometric features from the geometric foundation model. These resulting geometric tokens are interleaved into the context, after which the model continues autoregressive decoding to generate the final trajectory prediction.

D Additional Ablation Studies

Effect of sampling grid size. Table 11 studies the sampling grid size for localized geometry retrieval. A small grid may miss local structures, while an overly dense grid introduces redundancy. The results show that a moderate grid size achieves the best trade-off between geometric detail and token efficiency.

Table 12: **Ablation on training stages.** Stage-I denotes full QA training, and Stage-II denotes geometry-interleaved planning training.

Stage-I QA	Stage-II GeoCoT	NC	DAC	TTC	PDMS
✓	×	98.0	94.0	94.4	85.9
×	✓	98.3	94.7	94.5	86.6
✓	✓	98.4	95.6	95.0	87.6

Table 13: **Performance on turning scenarios in NAVSIM.** Values in parentheses indicate absolute gains over the baseline.

Scenario	Method	PDMS	NC	DAC	EP
Left turn	Baseline	83.96	98.34	91.04	78.63
	GeoCoTDrive	86.66 (+2.70)	99.04 (+0.70)	93.80 (+2.76)	80.78 (+2.15)
Right turn	Baseline	80.10	97.30	89.84	73.23
	GeoCoTDrive	82.32 (+2.22)	97.61 (+0.31)	91.80 (+1.96)	74.35 (+1.12)



Figure 7: **Qualitative planning results on nuScenes and Bench2Drive.** GeoCoTDrive is compared with the corresponding baseline methods.

Effect of training stages. Table 12 studies the contribution of the two-stage training strategy. Stage-I alone provides basic autonomous-driving QA and planning knowledge, while Stage-II alone enables geometry-interleaved planning. Combining both training stages achieves the best planning performance across all metrics, indicating that driving knowledge adaptation and localized geometry conditioning are complementary for end-to-end planning.

Scenario-wise Analysis. We evaluate GeoCoTDrive on NAVSIM turning scenarios, where planning requires reasoning about road geometry and surrounding agents. As shown in Table 13, GeoCoTDrive improves PDMS for left and right turns, with gains across NC, DAC, and EP. These results support the effectiveness of grounded geometric priors in planning under local spatial constraints.



Figure 8: **PlanningGrounding-nuScenes dataset.** Human annotations are shown for comparison.

E Additional Qualitative Results

Fig. 7 presents additional qualitative planning comparison. Each example pairs a front-view image with BEV visualizations of the road layout, surrounding agents, and predicted trajectories. The examples span curved roads, dynamic intersections, and low-visibility conditions. GeoCoTDrive first grounds regions that influence driving decisions, such as nearby vehicles, pedestrians, and drivable areas. It then incorporates geometric priors from these regions into planning to generate safety-aware trajectories that account for road boundaries and potential conflicts with surrounding agents.

F Limitations

GeoCoTDrive primarily relies on front-view images without explicit multi-view or temporal modeling, limiting its ability to account for side and rear traffic, prolonged occlusions, and interaction histories. Localized geometry retrieval also depends on grounding accuracy. The incorrect or incomplete regions may yield less relevant geometric cues and impair trajectory prediction. Future work will extend GeoCoTDrive to multi-view and temporal settings and improve grounding robustness.



Figure 9: **PlanningGrounding-NAVSIM** and **PlanningGrounding-Bench2Drive** dataset examples.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and introduction clearly outline the contribution and proposed methodological frameworks.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: In the Appendix

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [N/A]

Justification: This work does not include theoretical results.

Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: Detailed implementation details are included in Appendix.

   Guidelines:

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

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

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

Answer: [Yes]

Justification: We attach part of PlanningGrounding data in the supplemental material. We have publicly released the code.

Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: The main details are provided in the main paper and full details are provided in the Appendix.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [No]

   Justification: The paper reports mean performance metrics averaged over a large-scale evaluation set, which is a standard practice in the autonomous driving domain.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: We provide the compute resources in Appendix.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: The research was conducted in full compliance with the NeurIPS Code of Ethics, adhering to established standards for responsible model development and fair experimental practices.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: The work may improve autonomous-driving planning safety by encouraging models to focus on planning-relevant spatial cues. We discuss these limitations and emphasize that the method is a research prototype rather than a deployable autonomous-driving system.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [N/A]

    Justification:

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: All third-party assets, including the datasets and pre-trained model are properly credited through citations and their original licenses have been reviewed and respected.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [Yes]

    Justification: The paper provides a thorough description of the introduced PlanningGrounding dataset, covering the generation pipeline, data distribution, and the proposed evaluation protocol.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification:

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification:

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

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

    Answer: [Yes]

    Justification: We declare to use VLM for building the grounding dataset in the main paper and appendix

    Guidelines:

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