SpaceDrive: Infusing Spatial Awareness into VLM-based Autonomous Driving

Peizheng Li* [1,2], Zhenghao Zhang* [1,4], David Holtz[1], Hang Yu[1,5], Yutong Yang[1,6],
Yuzhi Lai[2], Rui Song[7], Andreas Geiger[2,3], Andreas Zell[2]
[1]Mercedes-Benz AG, [2]University of Tübingen, [3]Tübingen AI Center,
[4]TU Munich, [5]Karlsruhe Institute of Technology, [6]University of Stuttgart, [7]UCLA
https://zhenghao2519.github.io/SpaceDrive_Page/

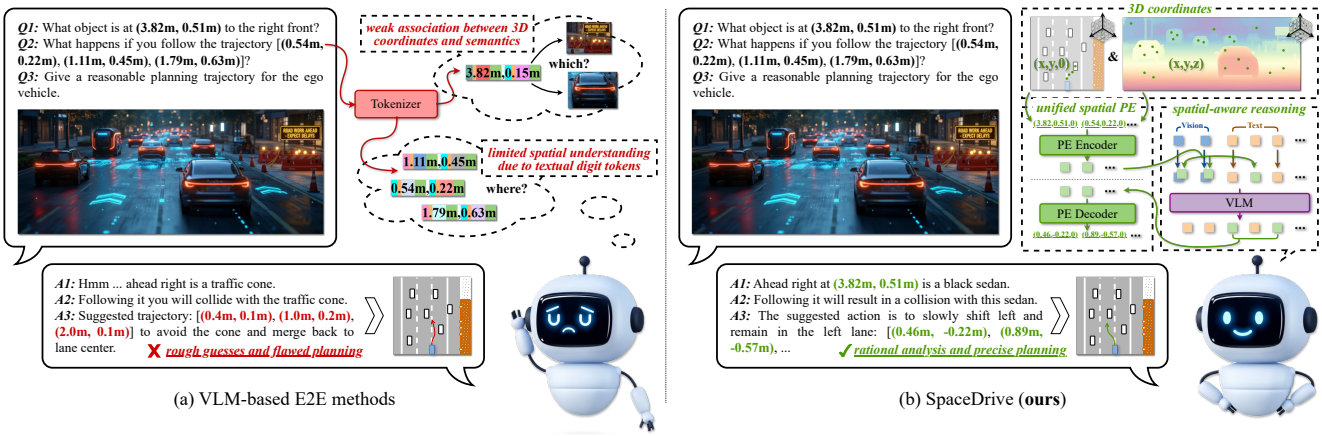


Figure 1. **Spatial awareness in VLM-based end-to-end autonomous driving.** (a) Constrained by insufficient 3D pre-training and discrete token-wise encoding, existing end-to-end planners based on the VLM struggle to precisely ground, associate, and predict 3D spatial positions, limiting their planning capabilities. (b) Our proposed SpaceDrive planner introduces a unified 3D coordinate encoding to replace the original VLM's textual digit tokens and augment visual features, achieving explicit association with 2D perspective semantics to enhance joint spatial reasoning for E2E planning. Compared to current VLM-based methods, it achieves state-of-the-art driving capability in the nuScenes open-loop evaluation and the second-best driving performance in the Bench2Drive closed-loop simulation.

Abstract

End-to-end autonomous driving methods built on vision language models (VLMs) have undergone rapid development driven by their universal visual understanding and strong reasoning capabilities obtained from the large-scale pre-training. However, we find that current VLMs struggle to understand fine-grained 3D spatial relationships which is a fundamental requirement for systems interacting with the physical world. To address this issue, we propose SpaceDrive, a spatial-aware VLM-based driving framework that treats spatial information as explicit positional encodings (PEs) instead of textual digit tokens, enabling joint reasoning over semantic and spatial representations. SpaceDrive employs a universal positional encoder to all 3D coordinates derived from multi-view depth estimation, historical ego-states, and text prompts. These 3D PEs are first superimposed to augment the corresponding 2D visual tokens. Meanwhile, they serve as a task-agnostic coordinate representation, replacing the digit-wise numerical tokens as both inputs and outputs for the VLM. This mechanism enables the model to better index specific visual semantics in spatial reasoning and directly regress trajectory coordinates rather than generating digit-by-digit, thereby enhancing planning accuracy. Extensive experiments validate that SpaceDrive achieves state-of-the-art open-loop performance on the nuScenes dataset and the second-best Driving Score of 78.02 on the Bench2Drive closed-loop benchmark over existing VLM-based methods. Code is available at: https://github.com/zhenghao2519/SpaceDrive.

1. Introduction

Large-scale pre-trained VLMs are known for their vast knowledge bases and strong reasoning capabilities. Lever-

* Equal contribution, names are sorted alphabetically. Correspondence to: {peizheng.li, zhenghao.zhang}@mercedes-benz.com.

aging VLMs to assist [29, 49, 60] or replace [13, 55, 62] traditional end-to-end (E2E) autonomous driving (AD) systems has therefore emerged as a prominent trend recently. These systems typically reformulate AD functions into natural language, and flexibly perform scene understanding, motion prediction and trajectory planning based on semantic information extracted from images. Compared to fixed modular designs [20, 28], VLM-based E2E models promise to achieve superior generalization, addressing increasingly complex and dynamic driving scenarios.

However, current VLMs demonstrate clear limitations in 3D tasks such as geometric measurement and distance estimation [5, 66, 71], which are critical for autonomous driving. This issue stems mainly from two primary factors, as illustrated in Fig. 1.a. First, the absence of 3D-data-based pre-training forces models to rely on inference from existing 2D knowledge. When dealing with 3D coordinates, VLMs struggle to associate them with the corresponding objects and their 2D semantics, leading to ambiguous or even incorrect scene descriptions [82]. Second, language models inherently treat numerical processing as digit-by-digit classification. This classification overlooks the inherent inter-digit proximity between numerical tokens and incorrectly averages the importance of different token positions [12].

In autonomous driving, existing VLM-based planners either introduce task-specific embeddings tailored to individual downstream tasks [13, 51] or represent waypoints as sequences of numeric tokens directly generated by the language model [55, 62]. The former relies on specialized 3D fine-tuning, tying embeddings to particular tasks and domains and thus hindering a transferable, universal spatial representation that preserves VLM generalization. The latter suffers from the aforementioned limitations in the numerical modeling ability of language models, results in inaccurate waypoint predictions. However, an important but underemphasized aspect is that the Transformer architecture is **inherently capable of processing positional relationships between tokens**, which can be conceptualized as **spatial relationships between semantic features** [30]. Therefore, extending this capability to 3D spatial awareness becomes a natural and logical idea.

Inspired by this, we propose *SpaceDrive*, a spatial-aware VLM-based AD framework illustrated in Fig. 1.b, which incorporates a universal encoding for 3D positions to enhance spatial understanding and reasoning in VLMs. Specifically, we first encode 3D coordinates derived from depth estimation and add them onto corresponding 2D visual tokens, establishing an explicit association between semantic features and 3D spatial locations. Meanwhile, this 3D PE serves as a general coordinate representation, replacing either conventional coordinates in natural language or task-specific embeddings as the input and output of VLM. Furthermore, for the output PE, we replace the original classification-based design with regression-based decoder and loss to address the numerical prediction deficiencies in language models. Our framework also exhibits strong adaptability to various VLM base models and reasoning strategies, further underscoring its potential as a universal paradigm.

To directly validate the trajectory planning accuracy, we first conducted an open-loop evaluation. Experiments on the nuScenes dataset [4] demonstrate that SpaceDrive achieves state-of-the-art performance among all VLM-based methods. However, similarity-based open-loop planning evaluation is highly susceptible to dataset overfitting, offering only limited insight into the model's actual driving competence. Therefore, we further validate our method on the closed-loop Bench2Drive [26] benchmark where we achieve a Driving Score of 78.02 (second-best in VLM-based planners), further confirming its capability to perform reasonable planning in dynamic and complex scenarios.

The contributions of this paper are as follows:

- We identify fundamental limitations of current VLMs in 3D spatial reasoning and waypoint prediction, and propose *SpaceDrive*, a spatial-aware VLM-based AD framework with a universal 3D positional encoding that explicitly associates image semantics with 3D coordinates.
- *SpaceDrive* employs a shared 3D PE as a general coordinate representation to augment visual tokens and serve as the coordinate interface for language models, along with a regression-based decoder to enhance the end-to-end trajectory planning.
- Our framework achieves state-of-the-art performance in open-loop planning on nuScenes, while exhibits strong closed-loop planning capabilities under complex driving scenarios on the Bench2Drive benchmark.

2. Related Work

End-to-End Autonomous Driving Over the past years, end-to-end autonomous driving has evolved from traditional modular stacks [9, 34, 35, 38, 46, 63, 67, 75, 80, 83] to fully differentiable, planning-oriented designs. After early methods like ST-P3 [19] achieved joint optimization of perception and planning, UniAD [20] unified the entire stack into a query-based framework, using planning supervision to regularize upstream tasks. Building on this paradigm, follow-up studies [6, 25, 28, 54, 65, 86] achieved further improvements in planning efficiency and decision quality. A key inflection came from AD-MLP [78] and BEV-Planner [39], which exposed open-loop brittleness: simple ego-state priors can rival sophisticated stacks. This finding shifted attention toward closed-loop fidelity and benchmarks that align with driving quality, *e.g.* Bench2Drive [26] and DriveE2E [76], and stimulated numerous subsequent methods [21, 25, 36, 40, 41, 44, 52, 79, 90] based on them. Despite their strong performance, conventional E2E frameworks lack generalized scene understanding, thus struggling

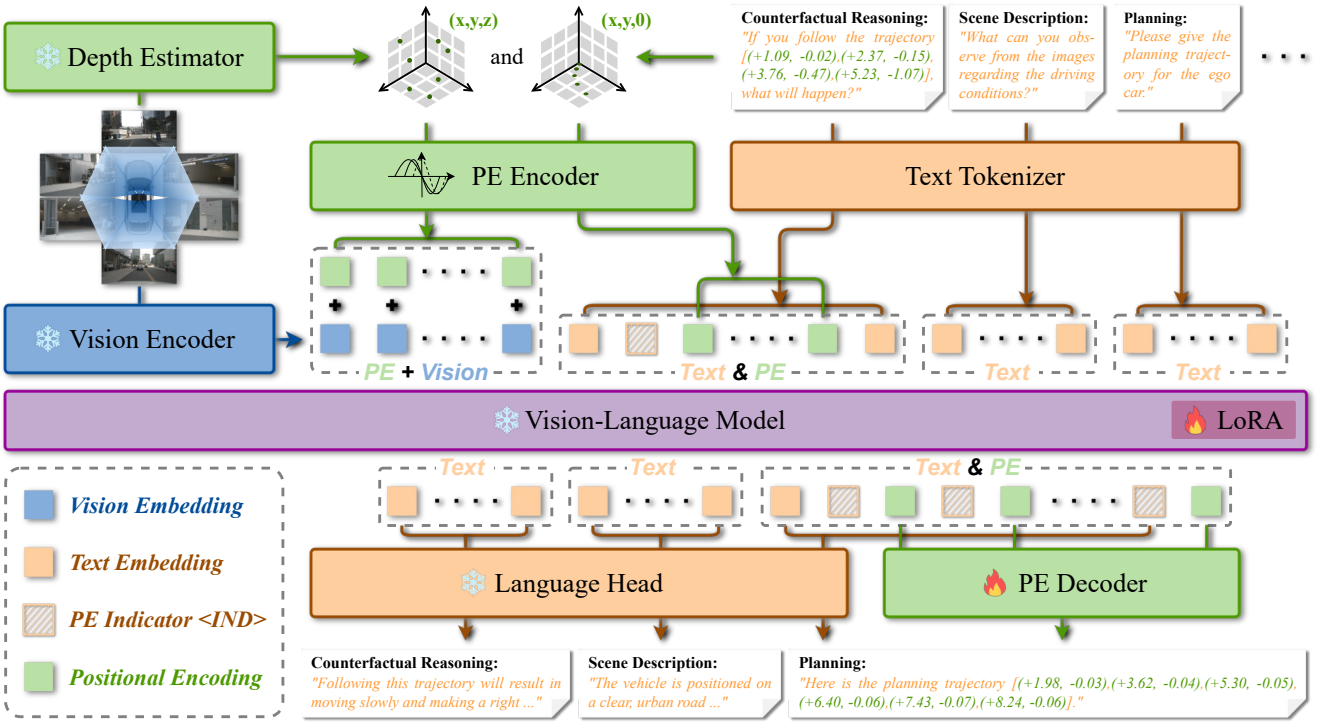


Figure 2. **SpaceDrive framework.** Beyond the base VLM, a frozen depth estimator predicts dense metric depths from surround-view images, which are projected into 3D coordinates and encoded by a universal PE encoder to augment visual tokens with spatial cues. BEV coordinates in text prompts are encoded by the same PE encoder, replacing the original coordinate tokens and preceded by the PE indicator ⟨IND⟩. At the output stage, the recognized PE is passed through a PE decoder to obtain the final coordinates for trajectory planning.

to handle complex and dynamic driving scenarios.

Spatial Intelligence of VLMs Recently, spatial intelligence in VLMs has progressed from 2D relational heuristics to explicit 3D-aware reasoning [5, 11, 14, 32, 33, 66, 69, 84, 85, 89]. This trend was initiated by SpatialVLM [5], which synthesized large-scale spatial Visual Question Answering (VQA) data to support both qualitative and quantitative spatial reasoning from 2D images. Subsequent works injected 3D structure more directly into the modeling pipeline. From integration of 3D features and positional embeddings in Scene-LLM [14] and LLaVA-3D [89], to dynamic and region-prompted spatial reasoning in Video-3D LLM [85], Spatial-MLLM [66], and SR-3D [8], these works collectively advance language-guided 3D understanding, grounding, and planning. Besides, dedicated benchmarks have standardized the evaluation. VSI-Bench [71] probes egocentric video-based visual–spatial intelligence with more than 5,000 QA pairs, while STI-Bench [37] stresses precise spatial–temporal estimation (pose, displacement, motion) across various scene setting. These studies demonstrate the immense potential of VLMs in spatial-aware tasks and suggest clear benefits for the perception, prediction, and planning in autonomous driving.

VLMs-Based Driving Agents Vision-language and multimodal LLMs have reshaped E2E driving by injecting priors, interactivity, and explicit reasoning into perception-prediction-planning. Early work such as DriveGPT4 [70] formulated driving as a language-conditioned sequence modeling, pairing video inputs with textual rationales to produce interpretable low-level controls. VLP [49] and DriveVLM [60] extended this direction by leveraging large vision-language models for scene understanding and trajectory generation, while DriveLM [55] further strengthened structured reasoning via graph-structured VQA over driving scenes. Recent methods [77, 87, 88] achieve further enhancements in areas such as reinforcement learning, symbolic reasoning, and precise control. For example, OmniDrive [62] pursues holistic 3D grounding with counterfactual supervision, while ORION [13] aligns reasoning and action spaces via a long-horizon QT-Former, an LLM reasoner, and a generative planner for strong closed-loop scores. Concomitant with the methodological developments, corresponding benchmarks [1, 48, 53, 55, 62] have also arisen, primarily targeting on open-world reasoning and regulation compliance. Nevertheless, existing VLM-based autonomous driving systems suffer from an inadequate treatment of 3D spatial awareness, a critical deficiency that forms the core focus of this paper.

3. Method

As illustrated in Fig. 2, we propose SpaceDrive, a spatial-aware framework that enhances end-to-end planning through

explicit injection of 3D information into the VLM architecture. Specifically, the surrounding images are first encoded by a visual encoder, and then aligned to the language model's semantic space via a projector. Meanwhile, these images are processed by a depth estimator to obtain absolute depths, which are converted into 3D positional encodings through a universal PE encoder. The visual tokens and their 3D PEs are then added element-wise, yielding spatially-aware visual tokens that serve as inputs to the VLM. Besides, text prompts for various reasoning tasks are also fed into the VLM as text token inputs. Notably, Bird's-Eye-View (BEV) or 3D coordinates within these prompts are processed separately by the same PE encoder to generate universal PEs, replacing the corresponding original text tokens. To avoid semantic confusion with other tokens, a predefined PE indicator is placed before each PE input and output. During reasoning, these PEs leverage their intrinsic similarity for direct interaction and indexing of the spatially-aware visual tokens. At the output stage, general textual outputs are decoded by the language head, while coordinate-related outputs are recognized and decoded by a dedicated PE decoder to produce accurate 3D coordinates for precise trajectory planning.

3.1. Spatial Awareness in Perception

A prerequisite for spatial intelligence is reliable 3D scene understanding, *i.e*. establishing dense correlations between 2D perspective visual features and their 3D geometry.

Vision Encoding A pretrained vision encoder $f_{vis.}$ first converts the K multi-view images $\{I_k\}_{k=1}^{K}$ into N patch tokens:

$$X_v = f_{vis.}(\{I_k\}) = \{x_p\}_{p=1}^{N}. \tag{1}$$

Given that our primary goal is the explicit infusion of spatial awareness, the sparse and highly abstract features within Q-Former-style architectures [62] are fundamentally limited in directly associating with concrete 3D spatial locations. Furthermore, the efficacy of the Q-Former typically requires additional large-scale pre-training for vision-language alignment, largely reducing the adaptabilty of our framework. Therefore, we keep using a simple MLP g to densely align the visual and language feature spaces, consistent with general-purpose VLMs [2, 42]:

$$H_v = g(X_v) = \{h_p\}_{p=1}^{N}. \tag{2}$$

Spatial Encoding To obtain 3D scene information, a pretrained depth estimator $f_{dep.}$ produces dense per-view absolute depth maps $D_k = f_{dep.}(I_k)$. To prioritize the foreground, for each patch p with image-plane support $\mathcal{R}_p$ we assign the minimum depth $d_p = \min_{(u,v)\in\mathcal{R}_p} D_k(u,v)$ as its corresponding depth. With the per-camera calibration matrix $\mathcal{P}_k$, we project the patch center (u_p, v_p) to 3D as $\mathbf{c}_p = \mathcal{P}_k^{\dagger}[u_p, v_p, d_p, 1]z\top$ to obtain explicit metric coordinates. Each $\mathbf{c}_p = (x_p^{3D}, y_p^{3D}, z_p^{3D})$ is then encoded into a universal 3D positional encoding via a PE encoder. To minimize confusion with the existing RoPE [57] used in the VLM, we opt for a 3D sine-cosine positional encoding extending the standard 1D formulation dimension-wise:

$$\begin{aligned}
&\phi(\mathbf{c}_p) = \left[\phi_x(x_p^{3D}), \phi_y(y_p^{3D}), \phi_z(z_p^{3D})\right] \in \mathbb{R}^{dim}, \text{with} \\
&\phi_a(p_a) = \begin{cases} \sin(\frac{p_a}{20000^{2i/d_a}}), \\ \cos(\frac{p_a}{20000^{2i/d_a}}), \end{cases} \quad i = 0, \ldots, \lfloor \tfrac{d_a}{2} \rfloor - 1, \\
&d_x = d_y = \lceil \tfrac{dim}{3} \rceil, d_z = dim - d_x - d_y.
\end{aligned} \tag{3}$$

for spatial dimension $a \in \{x, y, z\}$ and total PE width dim.

Spatial Token Injection Prior works [13, 62] inject learnable 3D cues within or before the vision-language projector, yielding only implicit geometry. In contrast, we explicitly add metric 3D coordinates information $\phi(\mathbf{c}_p)$ on top of modality-aligned visual tokens h_p after the MLP g. This design enables later reuse of the same PE $\phi(\cdot)$ for coordinates from text prompts, allowing the model to directly index spatially grounded visual features and strengthening downstream spatial reasoning, as further discussed in Sec. 3.2.

It is worth noting that direct additive injection of $\phi(\mathbf{c}_p)$ shifts the token norm distribution away from the pretrained VLM regime. To mitigate this, we introduce a learnable normalization factor α_{PE} shared across all 3D PEs, simply

$$\tilde{H}_v = \{\tilde{h_p}\}_{p=1}^{N}, \; \tilde{h}_p = h_p + \alpha_{PE}\, \phi(\mathbf{c}_p). \tag{4}$$

3.2. Spatial Awareness in Reasoning

Existing VLMs exhibit strong general 2D multimodal reasoning yet remain deficient in explicit 3D spatial inference:

1. Insufficient pretraining on metric 3D data and spatial reasoning tasks confines current VLMs mainly to abstract 2D reasoning [89], yielding poor estimation of inter-object spatial relations, physical extent, and distances.
2. The classification-based numerical prediction in existing language models often prioritizes fitting data distributions while neglecting the inherent affinity between numerical symbols and their sequential order [12], thereby degrading precision in continuous waypoint predictions.

Alternatively, existing methods introduce task-specific queries and decode explicit 3D coordinates from them using MLPs [51], generative modules [13] or attention layers [89]. Although partially mitigating the above limitations, the resulting tokens lack unified spatial semantics and thus transfer poorly across tasks. In contrast, we reuse the previously defined 3D PE $\phi(\mathbf{c})$ as a universal spatial representation. This choice enforces representational consistency between perception and reasoning, improving accuracy of coordinate handling and estimation within the VLM.

Encoding of Coordinates in Text Prompts During tokenization of input text prompts, we scan the text sequence $\{t_i\}_{i=1}^{L}$ for substrings $\mathcal{S}$ expressing spatial coordinates. For each detected coordinate expression we extract its numeric

values as a vector $\mathbf{c}_r = (x_r, y_r, z_r)$. The same 3D positional encoder $\phi(\cdot)$ as in Sec. 3.1 is then applied to obtain a corresponding spatial token $\phi(\mathbf{c}_r) \in \mathbb{R}^{dim}$, which replaces the original sequence of numeric tokens corresponding to that coordinate. Each input PE is preceded by a specifically defined token, ⟨IND⟩, serving as the PE identifier (for simplicity, ⟨IND⟩ will be omitted in subsequent descriptions and formulations). The adjusted text token inputs are as follows:

$$\tilde{H}_t = \{\tilde{h}_i\}_{i=1}^{L},\ \tilde{h}_i = \begin{cases} \phi(\mathbf{c}_r) & i \in \mathcal{S}_r \\ \text{Tokenizer}(t_i) & \text{otherwise} \end{cases}. \quad (5)$$

A special case arises for BEV coordinates (*e.g.* trajectory waypoints), where we set all z-axis components in the PE $\phi(\mathbf{c}_r)$ to 0 so that they do not contribute to subsequent attention calculations.

Encoding of the Ego Status It has been verified that ego state inputs are highly effective for trajectory planning [39, 78]. Existing approaches typically encode all state variables (*e.g.* pose, velocity, acceleration) simply into a single vector embedding $\mathbf{e}_{\text{ego}} \in \mathbb{R}^{dim}$, mostly also augmented with BEV features to obscure explicit metric structure. Thanks to our unified spatial representation, we instead encode the historical ego waypoints via the same $\phi(\cdot)$ employed before, *i.e.* $\{\phi(\mathbf{c}_\tau^{ego})\}_{\tau=-T}^{1}$. It will then be fed into the language model together with $\mathbf{e}_{\text{ego}}$ as explicit spatial-temporal conditioning for accuracy trajectory planning.

Decoding of Text with Coordinates At the output stage, the VLM produces a sequence of embeddings $\{\mathbf{e}_j\}_{j=1}^{J}$. A standard language head W_{lang} maps each $\mathbf{e}_j$ to a distribution over the textual vocabulary $\mathcal{V}$ for ordinary decoding. Additionally, we utilize the previously defined ⟨IND⟩ to signal a forthcoming coordinate emission, extending the original W_{lang} to W'_{lang}, *i.e.*

$$y_j = \arg\max_{y \in \mathcal{V}'} \left(W'_{\text{lang}} \mathbf{e}_j\right)_y,\ \mathcal{V}' = \mathcal{V} \cup \{\langle \text{IND} \rangle\}. \quad (6)$$

If $y_j \neq \langle \text{IND} \rangle$ the text token is emitted normally. When $y_j = \langle \text{IND} \rangle$, $\mathbf{e}_j$ remains in the language context and the subsequent output state $\mathbf{e}_{j+1}$ is routed to a PE decoder $\psi(\cdot)$ to produce metric coordinates:

$$\hat{\mathbf{c}} = \psi(\mathbf{e}_{j+1}),\ \hat{\mathbf{c}} \in \mathbb{R}^3 \quad (7)$$

This mechanism yields precise BEV trajectory waypoints (omitting the z-coordinate) while preserving autoregressive continuity for surrounding text. Because the composite sinusoidal encoding $\phi(\mathbf{c})$ is not analytically invertible (phase and frequency aliasing across dimensions), $\psi(\cdot)$ is set as a fully learnable MLP and trained to regress ground-truth coordinates. The shared use of $\phi(\cdot)$ in both perception and reasoning ensures that $\psi(\cdot)$ operates over embeddings already aligned with unified spatial PEs, improving coordinate fidelity and trajectory planning accuracy.

3.3. Loss Function

A typical training objective combines language modeling $\mathcal{L}_{\text{LM}}$ (applied to all text outputs) with coordinate regression $\mathcal{L}_{\text{reg.}}$ (applied to all coordinate outputs, such as waypoints):

$$\mathcal{L} = \mathcal{L}_{\text{LM}} + \mathcal{L}_{\text{reg.}}(\hat{\mathbf{c}}, \mathbf{c}), \quad (8)$$

where $\mathcal{L}_{\text{reg.}}$ may vary with the type of the decoder $\psi(\cdot)$ and the adopted trajectory generation strategy. For the basic MLP decoder, we adopt the Huber loss for coordinate regression.

4. Experiments

4.1. Experimental Settings

Dataset and Metrics The nuScenes dataset [4] comprises 1,000 urban driving scenes (train/val/test: 700/150/150) with full-stack 360° sensing (6 cameras, 1 LiDAR, 5 radars). For open-loop planning we predict 6 waypoints within a 3 s horizon and evaluate (i) waypoint displacement (L2) error, (ii) Collision rate (fraction of future timestamps overlapping with any dynamic agent), and (iii) Intersection rate (fraction of timestamps intruding into non-drivable map regions).

Bench2Drive [26] is a closed-loop planning benchmark emphasizing interactive scenarios (merging, overtaking, yielding, emergency negotiation) in a deterministic CARLA V2 [10] simulator. Our closed-loop evaluation adopts the official protocol of 220 short routes, covering 44 interactive scenarios, with 5 distinct routes defined for each scenario. Closed-loop metrics include Driving Score (route progress penalized by safety infractions) and Success Rate (percentage of scenarios completed without terminal violation). All reported results adopt identical horizon, temporal sampling, footprint inflation, and map definitions for fair comparison.

Implementation Details Our model adopts Qwen2.5-VL-7B [2] as the base VLM. We finetune the core LLM using LoRA [18] with the rank of 16, while keeping the original vision encoder and vision-language projector frozen. Unidepthv2-ViT-L [50] is chosen as our default depth estimation module without additional finetuning. For the open-loop evaluation on nuScenes, the model is trained for 6 epochs on 8×A100 80GB GPUs with a batch size of 8. The learning rate is set to 1e-4 and cosine annealing is used to ensure stable training. The input resolution is resized to 640×640. For the closed-loop evaluation, the model is trained for 12 epochs using the same training setup as the open-loop evaluation. For VQA training and evaluation, we adopt the data and settings utilized in OmniDrive [62]. Further details are provided in the supplementary materials.

4.2. Quantitative Results

Open-loop Planning To directly validate the impact of spatial awareness on VLM's coordinate regression, we first conducted the open-loop planning evaluation on the nuScenes

Table 1. **Open-loop planning results on nuScenes [4].** SpaceDrive+ denotes the adoption of the ego planner input. Methods categorized as Hybrid Paradigm simultaneously stack traditional and VLM-based approaches, and are thus incomparable. Results are highlighted in **bold** and underline for the best and second-best performance among VLM-based methods, respectively. All results in this table follow the evaluation protocol of OmniDrive [62] and ORION [13]. More methods based on other metrics are provided in the supplementary materials.

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 [19]	-	-	1.33	2.11	2.90	2.11	0.23	0.62	1.27	0.71	2.53	8.17	14.40	8.37
UniAD [20]	✓	✓	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 [28]	✓	✓	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 [78]	-	✓	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++ [39]	✓	✓	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 [15]	✓	✓	0.13	0.28	0.48	0.30	0.00	0.19	0.61	0.27	0.13	1.08	2.89	1.37
Drive-WM [64]	✓	✓	0.43	0.77	1.20	0.80	0.10	0.21	0.48	0.26	-	-	-	-
VLM-based Paradigm														
EMMA [23]	-	-	**0.14**	**0.29**	0.54	**0.32**	-	-	-	-	-	-	-	-
RDA-Driver [22]	✓	✓	0.23	0.73	1.54	0.80	**0.00**	**0.13**	0.83	0.32	-	-	-	-
DriveVLM [60]	-	✓	0.18	0.34	0.68	0.40	0.10	0.22	**0.45**	0.27	-	-	-	-
ORION [13]	✓	-	0.17	0.31	0.55	0.34	0.05	0.25	0.80	0.37	-	-	-	-
OmniDrive-Q [62]	-	-	1.15	1.96	2.84	1.98	0.80	3.12	7.46	3.79	1.66	3.86	8.26	4.59
OmniDrive-Q++ [62]	✓	✓	**0.14**	**0.29**	0.55	0.33	**0.00**	**0.13**	0.78	0.30	0.56	2.48	5.96	3.00
SpaceDrive (**ours**)	-	-	1.06	1.79	2.55	1.80	0.35	1.33	3.97	1.88	0.96	3.38	8.28	4.21
SpaceDrive+ (**ours**)	-	✓	0.15	**0.29**	**0.51**	**0.32**	0.04	0.18	0.49	**0.23**	**0.22**	**0.80**	**2.79**	**1.27**
Hybrid Paradigm														
VLP [49]	✓	-	0.30	0.53	0.84	0.55	0.01	0.07	0.38	0.15	-	-	-	-
ReAL-AD [47]	✓	-	0.30	0.48	0.67	0.48	0.07	0.10	0.28	0.15	-	-	-	-
DriveVLM-Dual [60]	✓	-	0.15	0.29	0.48	0.31	0.05	0.08	0.17	0.10	-	-	-	-
SOLVE-VLM [7]	✓	-	0.13	0.25	0.47	0.28	0.00	0.16	0.43	0.20	-	-	-	-
Senna [29]	✓	-	0.11	0.21	0.35	0.22	0.04	0.08	0.13	0.08	-	-	-	-

Table 2. **Closed-loop planning results on Bench2Drive [26].** Results are highlighted in **bold** and underline for the best and second-best performance among VLM-based methods, respectively.

Method	Closed-loop Metric	
	Driving Score ↑	Success Rate(%) ↑
Traditional Modular Paradigm		
AD-MLP [78]	18.05	0.00
UniAD-Base [20]	45.81	16.36
VAD-Base [28]	42.35	15.00
MomAD [56]	44.54	16.71
GenAD [86]	44.81	15.90
SparseDrive [58]	47.38	17.72
UAD [15]	49.22	20.45
WoTE [36]	61.71	31.36
ThinkTwice [25]	62.44	37.17
DriveTransformer-L [27]	63.46	38.60
DriveAdapter [24]	64.22	42.08
HiP-AD [59]	86.77	69.09
AlignDrive [68]	89.07	73.18
VLM-based Paradigm		
ReAL-AD [47]	41.17	11.36
Dual-AEB [81]	45.23	10.00
VDRive [16]	66.25	50.51
StuckSolver [3]	70.89	50.01
DriveMoE [73]	74.22	48.64
ETA [17]	74.33	48.33
VLR-Drive [31]	75.01	50.00
ORION [13]	77.74	54.62
SimLingo [51]	**85.07**	**67.27**
SpaceDrive+ (**ours**)	78.02	55.11

dataset. As shown in Tab. 1, SpaceDrive+ achieves the SOTA performance across all reported metrics, consistently surpassing existing VLM-based methods. The lowest L2 error (0.32) indicates that coordinate-level regression allows closer adherence to expert driving trajectories. Simultaneously, the markedly reduced Collision (0.23%) and Intersection (1.27%) rates further show that SpaceDrive+ excels not only at fitting ground truth but also enhances autonomous driving safety comprehensively through superior spatial understanding and reasoning.

Notably, our method does not include the BEV features widely adopted in existing pipelines. This provides evidence that a unified positional encoding is sufficient for 3D spatial modeling within VLM-oriented autonomous driving, obviating dense BEV representations. Considering the sensitivity of open-loop metrics to the integration of ego status [39], we further report the variant without ego status inputs, *i.e.* SpaceDrive. In this setting, our method also surpasses its codebase (OmniDrive [62]) across all dimensions (L2: -0.18, Collision: -1.91%, Intersection: -0.38%), validating the effectiveness of explicitly injecting 3D spatial information.

Closed-loop Planning In Tab. 2 we conduct closed-loop evaluation to establish a comprehensive and reliable assessment of planning performance. While our codebase (OmniDrive) attains competitive open-loop metrics, its text-only planning paradigm fails drastically in closed-loop simulation (Under 10% Success Rate). Empirically, its predicted trajectories collapse into near-linear paths with unstable heading oscillations. This substantiates our hypothesis that pure natural-language trajectory generation primarily fits data priors rather than learning a controllable driving pattern. More comparisons are provided in the supplementary materials.

Table 3. **Ablation of positional encoding.** Here, $\phi(\mathbf{c}_r)$, $\phi(\mathbf{c}_p)$ and $\phi(\mathbf{c}_t^{ego})$ denote the usage of spatial positional encodings for textual coordinate inputs, 3D coordinates corresponding to vision tokens and past ego locations, respectively.

Exp.	$\phi(\mathbf{c}_p)$	$\phi(\mathbf{c}_r)$	$\phi(\mathbf{c}_\tau^{ego})$	Avg. L2 ↓	Avg. Col. ↓	Avg. Int. ↓
SpaceDrive						
1				2.51	4.53	6.77
2	✓			1.88	2.45	2.36
3		✓		2.42	5.06	8.94
4	✓	✓		1.80	1.88	4.21
SpaceDrive+						
5				0.41	0.60	4.40
6	✓	✓		0.33	0.23	1.32
7	✓	✓	✓	0.32	0.23	1.27

By introducing explicit spatial tokens, SpaceDrive+ achieves 78.02 Driving Score and 55.11% Success Rate, ranking as the second-best VLM-based method (notably, SimLingo [51] employs extensive data augmentation via Action Dreaming). These gains indicate that injecting structured 3D positional information is sufficient to unlock strong closed-loop planning within a VLM-oriented framework.

4.3. Qualitative Results

Figure 3 illustrates a representative Bench2Drive scenario in which the ego vehicle is required to avoid collision with two cyclists ahead. The planner first accelerates to attempt overtaking and lane changing. After observing that the adjacent vehicle does not yield, our spatial-aware SpaceDrive+ detects sufficient rearward clearance in the target lane and opts to decelerate to create a safe insertion gap. Once the opening emerges, it executes a decisive lateral maneuver. As the lane change nears completion, the model infers from its ego state and surrounding vehicle positions that rapid heading re-alignment is necessary to avoid drifting out of lane boundaries. This case exemplifies that the injected 3D spatial encoding enables SpaceDrive+ to adapt its strategy to evolving scene geometry and generate safety-aware plans.

4.4. Ablation Studies

Our approach focuses on endowing models with spatial awareness rather than tailoring them for open- or closed-loop planning. Therefore, considering that closed-loop performance may be influenced by training strategies, PID controller tuning, and other pipeline heuristics, we conducted our ablations under open-loop settings (excluding the ego status to prevent overfitting) to ensure a fair comparisons.

Positional Encoding We compare the effect of injecting explicit positional encodings into different modules of the VLM-based planner in Tab. 3. First, adding spatial encoding to vision tokens (Exp. 2 vs. 1) yields substantial improvements on all metrics, *i.e.*-0.63 L2, -2.08% Collision and -4.14% Intersection. This is largely attributable to the enhanced spatial understanding achieved by supplying 3D geometric context alongside 2D image features. Meanwhile, the gains from replacing the textual coordinate with PE, as demonstrated in Exp. 3, are relatively smaller, likely because the PE lacks the bridge to associate with 2D semantic space in pretrained VLMs. However, when a unified positional encoding is applied to both vision and textual coordinate streams (Exp. 4 vs. 1; Exp. 6 vs. 5), planning performance improves regardless of the use of ego status, underscoring the value of a shared spatial representation. Finally, with ego status enabled, injecting past ego positions using the same $\phi(\cdot)$ (Exp. 7 vs. 6) further reduces L2 error and Intersection rates. This indicaties that the benefits coming from consistent spatial tokens are stable and reliable, facilitating spatial understanding and reasoning in VLMs.

Table 4. **Ablation of PE encoder & decoder.** Gray indicates that only 4929 out of 5119 output samples are semantically reasonable.

Exp.	Encoder $\phi(\cdot)$	Decoder $\psi(\cdot)$	Avg. L2 ↓	Avg. Col. ↓	Avg. Int. ↓
4	Sine-Cosine	Coordinate-wise	1.80	1.88	4.21
8	MLP	Coordinate-wise	1.96	3.17	6.76
9	RoPE	Coordinate-wise	1.93	3.71	11.40
10	Sine-Cosine	Sine-Cosine	1.87	2.62	9.20
11	Sine-Cosine	Task-specific	1.93	2.41	5.58

Table 5. **Ablation of PE normalization.** Gray indicates that only 2421 out of 5119 output samples are semantically reasonable.

Init. α_{PE}	Learnable	Avg. L2 ↓	Avg. Collision ↓	Avg. Intersection ↓
1		2.34	3.63	8.46
0.1		2.43	3.79	9.42
0.02		2.22	2.71	10.17
1	✓	1.82	2.04	4.62
0.1	✓	1.80	1.88	4.21
0.02	✓	1.86	2.03	5.42

PE Encoder & Decoder Table 4 compares different encoders and decoders of PE. Using a Sine–Cosine encoder yields a translation-invariant encodability. It assists the attention layers in recovering inter-token spatial relations, giving clear gains over a fully learnable MLP encoder (Exp. 4 vs. 8). Although RoPE shares the same property as the additive Sine–Cosine encoding, it leads to a large performance degradation and instability due to the confusion with existing RoPE used in the base VLM (Exp. 4 vs. 9). On the decoder side, numerically inverting Sine–Cosine encodings to precise coordinates is ill-posed (only coarse interpolation is possible), and the output embedding space of a large VLM is typically not fully aligned with its input space. These factors make a learnable coordinate-wise MLP decoder preferable, as reflected in its lower L2 error (1.80 in Exp. 4 vs 1.87 in Exp. 10). A common paradigm in VLM planners is to use a task-specific embedding and decode an entire trajectory from it. This limits reuse across tasks and forces retraining when objectives change. The comparison between experiments 4 and 11 shows that jointly decoding multiple waypoints from a single embedding underperforms the coordinate-wise strategy in all metrics, which predicts each waypoint conditioned on shared spatial tokens.

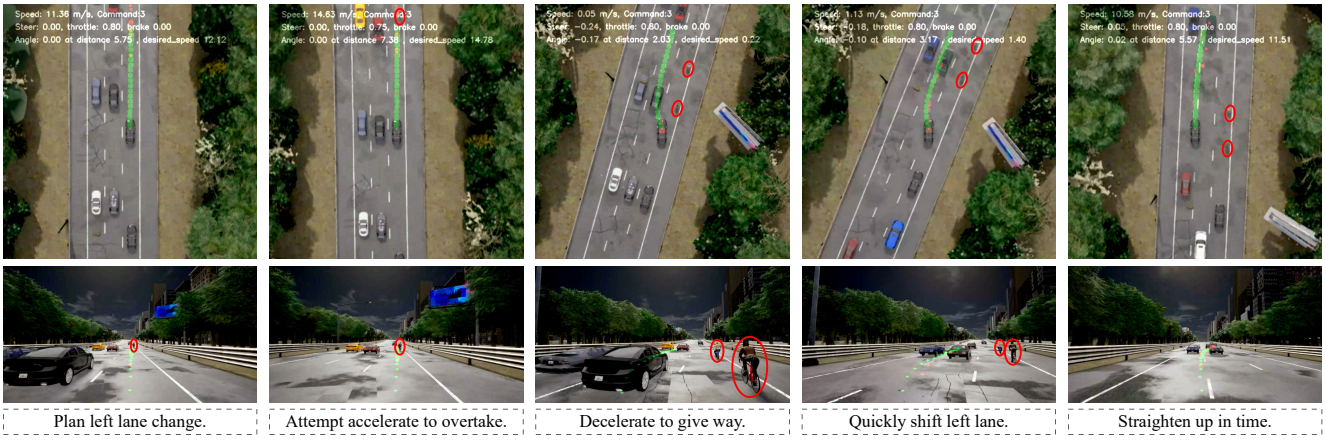


Figure 3. **Qualitative results of closed-loop evaluation on Bench2Drive [26].** Green and pink dots represent path and speed waypoints, respectively. Red circles indicate cyclists ahead that the vehicle needs to avoid. Parameters such as speed and steering wheel angle can be found in the figures.

Table 6. **SpaceDrive based on different VLM foundations.**

Method	Base VLM	Avg. L2 ↓	Avg. Collision ↓	Avg. Intersection ↓
SpaceDrive	LLaVA	1.82	2.44	4.08
	Qwen-VL	1.80	1.88	4.21
SpaceDrive+	LLaVA	0.31	0.23	1.42
	Qwen-VL	0.32	0.23	1.27

PE Normalization In transformer-based VLMs, the embedding norm directly modulates its relative importance in the attention operations. Consequently, the norm of the PE can largely affect training stability and planning accuracy. In Tab. 5 we compare performance under different fixed initialization scales α_{PE}. For Qwen-VL, smaller static α_{PE} values lead to consistent degradation across all open-loop metrics and even semantic instability, implying that excessively small PE norms result in negligible attention scores and hinder convergence. Since the optimal PE scale differs between foundation models and may shift over training, we promote α_{PE} to a learnable parameter. This adaptive normalization, while mitigating semantic instability, produces a about -0.5 m reduction in Avg. L2 together with marked moderation in Collision and Intersection rates. This outcome validates the importance of a learnable normalization coefficient for the unified 3D PE.

More ablations and experiments regarding VQA, depth estimator, etc., are provided in the supplementary materials.

4.5. Adaptability

Our proposed 3D spatial representation also demonstrates excellent adaptability. First, it attains comparable performance on both Qwen-VL and LLaVA (See Tab. 6), indicating that the strong performance arise from unified spatial reasoning rather than backbone-specific biases. Injecting spatial awareness also preserves compatibility with inference-time reasoning enhancements. For example, without ego state inputs we observe distributional degeneration of predicted trajectories, *i.e.* mode collapse. Augmenting SpaceDrive with lightweight chain-of-thought prompting (CoT) stabilizes waypoint diversity and reduces collapse without retraining. Furthermore, we can also adapt the PE decoder into a VAE-based generative model, which has also been proven effective in improving closed-loop robustness [13]. More comparisons are provided in the supplementary materials. All these results collectively show that our spatial encoding is a general booster for spatial reasoning and E2E planning across VLM foundations and inference paradigms.

5. Conclusion

In this paper, we presented SpaceDrive, a spatial-aware VLM-based end-to-end autonomous driving framework, that unifies 3D spatial awareness and multimodal reasoning through a universal positional encoding interface. The approach infuses metric 3D positional encodings into visual tokens, textual coordinate mentions, and historical ego states, and decodes coordinates with a regression head instead of digit-wise language generation. In this way, the model achieves superior trajectory planning performance compared to methods relying solely on natural language fitting. Besides, this design reduces reliance on task-specific embeddings, unleashing the generalization capabilities of pre-trained VLMs for space-relevant reasoning. Extensive experiments show state-of-the-art open-loop planning results on nuScenes across displacement, collision, and map compliance metrics, as well as strong closed-loop performance on Bench2Drive, substantially narrowing the gap between VLM planners and specialized end-to-end baselines. Ablations confirm the benefit of unified spatial tokens, coordinate-wise decoding, and learnable normalization of positional encodings. Meanwhile, SpaceDrive also achieves good transferability across VLM backbones and remains adaptable to language reasoning strategies. Limitations in-

clude the absence of explicit uncertainty handling, and no exploitation of multi-frame temporal memory mechanisms, which also represent potential directions for future research. Nevertheless, we believe the proposed unified spatial representation offers a principled path toward more reliable, generalizable spatial-aware VLM-driven autonomy.

Acknowledgement

This work is a result of the joint research project STADT:up (Förderkennzeichen 19A22006O). The project is supported by the German Federal Ministry for Economic Affairs and Climate Action (BMWK), based on a decision of the German Bundestag. The author is solely responsible for the content of this publication.

References

[1] Hidehisa Arai, Keita Miwa, Kento Sasaki, Kohei Watanabe, Yu Yamaguchi, Shunsuke Aoki, and Issei Yamamoto. Covla: Comprehensive vision-language-action dataset for autonomous driving. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2025. 3

[2] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025. 4, 5, 1

[3] Zhipeng Bao and Qianwen Li. Large language model-assisted autonomous vehicle recovery from immobilization. *arXiv preprint arXiv:2510.26023*, 2025. 6

[4] 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*, 2020. 2, 5, 6, 3, 4

[5] 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*, 2024. 2, 3

[6] 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. 2

[7] 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*, 2025. 6, 4

[8] An-Chieh Cheng, Yang Fu, Yukang Chen, Zhijian Liu, Xiaolong Li, Subhashree Radhakrishnan, Song Han, Yao Lu, Jan Kautz, Pavlo Molchanov, et al. 3d aware region prompted vision language model. *arXiv preprint arXiv:2509.13317*, 2025. 3

[9] Shuxiao Ding, Yutong Yang, Julian Wiederer, Markus Braun, Peizheng Li, Juergen Gall, and Bin Yang. Tqd-track: Temporal query denoising for 3d multi-object tracking. *arXiv preprint arXiv:2504.03258*, 2025. 2

[10] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. In *Conference on robot learning*, 2017. 5

[11] Songcheng Du, Yang Zou, Zixu Wang, Xingyuan Li, Ying Li, Changjing Shang, and Qiang Shen. Unsupervised hyperspectral image super-resolution via self-supervised modality decoupling. *International Journal of Computer Vision*, 134: 152, 2026. 3

[12] Xiang Fei, Jinghui Lu, Qi Sun, Hao Feng, Yanjie Wang, Wei Shi, An-Lan Wang, Jingqun Tang, and Can Huang. Advancing sequential numerical prediction in autoregressive models. *arXiv preprint arXiv:2505.13077*, 2025. 2, 4

[13] 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. *arXiv preprint arXiv:2503.19755*, 2025. 2, 3, 4, 6, 8, 1

[14] Rao Fu, Jingyu Liu, Xilun Chen, Yixin Nie, and Wenhan Xiong. Scene-llm: Extending language model for 3d visual understanding and reasoning. *arXiv preprint arXiv:2403.11401*, 2024. 3

[15] 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. 6, 4

[16] Ziang Guo and Zufeng Zhang. Vdrive: Leveraging reinforced vla and diffusion policy for end-to-end autonomous driving. *arXiv preprint arXiv:2510.15446*, 2025. 6

[17] Shadi Hamdan, Chonghao Sima, Zetong Yang, Hongyang Li, and Fatma Guney. Eta: Efficiency through thinking ahead, a dual approach to self-driving with large models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025. 6

[18] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 2022. 5, 2

[19] 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*, 2022. 2, 6, 3, 4

[20] 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*, 2023. 2, 6, 3, 4

[21] Yi Huang, Lihui Jiang, Bingbing Liu, Hongbo Zhang, et al. Prioritizing perception-guided self-supervision: A new paradigm for causal modeling in end-to-end autonomous driving. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. 2, 6

[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*, 2024. 6, 4

[23] Jyh-Jing Hwang, Runsheng Xu, Hubert Lin, Wei-Chih Hung, Jingwei Ji, Kristy Choi, Di Huang, Tong He, Paul Covington, Benjamin Sapp, et al. Emma: End-to-end multimodal model for autonomous driving. *arXiv preprint arXiv:2410.23262*, 2024. 6, 4

[24] Xiaosong Jia, Yulu Gao, Li Chen, Junchi Yan, Patrick Langechuan Liu, and Hongyang Li. Driveadapter: Breaking the coupling barrier of perception and planning in end-to-end autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023. 6

[25] Xiaosong Jia, Penghao Wu, Li Chen, Jiangwei Xie, Conghui He, Junchi Yan, and Hongyang Li. Think twice before driving: Towards scalable decoders for end-to-end autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 2, 6

[26] 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*, 2024. 2, 5, 6, 8, 4

[27] Xiaosong Jia, Junqi You, Zhiyuan Zhang, and Junchi Yan. Drivetransformer: Unified transformer for scalable end-to-end autonomous driving. In *International Conference on Learning Representations (ICLR)*, 2025. 6

[28] 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*, 2023. 2, 6, 4

[29] 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. 2, 6, 4

[30] Guolin Ke, Di He, and Tie-Yan Liu. Rethinking positional encoding in language pre-training. In *International Conference on Learning Representations*, 2021. 2

[31] Fanjie Kong, Yitong Li, Weihuang Chen, Chen Min, Yizhe Li, Zhiqiang Gao, Haoyang Li, Zhongyu Guo, and Hongbin Sun. Vlr-driver: Large vision-language-reasoning models for embodied autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025. 6

[32] Yuzhi Lai, Shenghai Yuan, Peizheng Li, Jun Lou, and Andreas Zell. Seer-var: Semantic egocentric environment reasoner for vehicle augmented reality. *arXiv preprint arXiv:2508.17255*, 2025. 3

[33] Yuzhi Lai, Shenghai Yuan, Boya Zhang, Benjamin Kiefer, Peizheng Li, Tianchen Deng, and Andreas Zell. Famhri: Foundation-model assisted multi-modal human-robot interaction combining gaze and speech. *arXiv preprint arXiv:2503.16492*, 2025. 3

[34] Peizheng Li, Shuxiao Ding, Xieyuanli Chen, Niklas Hanselmann, Marius Cordts, and Juergen Gall. Powerbev: A powerful yet lightweight framework for instance prediction in bird's-eye view. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23*, 2023. 2

[35] Peizheng Li, Shuxiao Ding, You Zhou, Qingwen Zhang, Onat Inak, Larissa Triess, Niklas Hanselmann, Marius Cordts, and Andreas Zell. Ago: Adaptive grounding for open world 3d occupancy prediction. In *Proceedings of the IEEE/CVF international conference on computer vision*, 2025. 2

[36] Yingyan Li, Yuqi Wang, Yang Liu, Jiawei He, Lue Fan, and Zhaoxiang Zhang. End-to-end driving with online trajectory evaluation via bev world model. *arXiv preprint arXiv:2504.01941*, 2025. 2, 6

[37] Yun Li, Yiming Zhang, Tao Lin, XiangRui Liu, Wenxiao Cai, Zheng Liu, and Bo Zhao. Sti-bench: Are mllms ready for precise spatial-temporal world understanding? *arXiv preprint arXiv:2503.23765*, 2025. 3

[38] 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*, 2024. 2

[39] 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*, 2024. 2, 5, 6, 1

[40] Zhenxin Li, Shihao Wang, Shiyi Lan, Zhiding Yu, Zuxuan Wu, and Jose M. Alvarez. Hydra-next: Robust closed-loop driving with open-loop training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025. 2, 6

[41] 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*, 2025. 2, 4, 6

[42] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 2023. 4, 1

[43] Haochen Liu, Tianyu Li, Haohan Yang, Li Chen, Caojun Wang, Ke Guo, Haochen Tian, Hongchen Li, Hongyang Li, and Chen Lv. Reinforced refinement with self-aware expansion for end-to-end autonomous driving. *arXiv preprint arXiv:2506.09800*, 2025. 6

[44] Shuai Liu, Quanmin Liang, Zefeng Li, Boyang Li, and Kai Huang. Gaussianfusion: Gaussian-based multi-sensor fusion for end-to-end autonomous driving. *arXiv preprint arXiv:2506.00034*, 2025. 2, 6

[45] Wei Liu, Jiyuan Zhang, Binxiong Zheng, Yufeng Hu, Yingzhan Lin, and Zengfeng Zeng. X-driver: Explainable autonomous driving with vision-language models. *arXiv preprint arXiv:2505.05098*, 2025. 6

[46] Yingfei Liu, Tiancai Wang, Xiangyu Zhang, and Jian Sun. Petr: Position embedding transformation for multi-view 3d

object detection. In *European conference on computer vision*. Springer, 2022. 2

[47] Yuhang Lu, Jiadong Tu, Yuexin Ma, and Xinge Zhu. Realad: Towards human-like reasoning in end-to-end autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025. 6, 4

[48] Ming Nie, Renyuan Peng, Chunwei Wang, Xinyue Cai, Jianhua Han, Hang Xu, and Li Zhang. Reason2drive: Towards interpretable and chain-based reasoning for autonomous driving. In *European Conference on Computer Vision*. Springer, 2024. 3

[49] Chenbin Pan, Burhaneddin Yaman, Tommaso Nesti, Abhirup Mallik, Alessandro G Allievi, Senem Velipasalar, and Liu Ren. Vlp: Vision language planning for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. 2, 3, 6, 4

[50] 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. *arXiv preprint arXiv:2502.20110*, 2025. 5, 2

[51] Katrin Renz, Long Chen, Elahe Arani, and Oleg Sinavski. Simlingo: Vision-only closed-loop autonomous driving with language-action alignment. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025. 2, 4, 6, 7, 1

[52] Shuyao Shang, Yuntao Chen, Yuqi Wang, Yingyan Li, and Zhaoxiang Zhang. Drivedpo: Policy learning via safety dpo for end-to-end autonomous driving. *arXiv preprint arXiv:2509.17940*, 2025. 2, 6

[53] 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*, 2024. 3

[54] Yinzhe Shen, Omer Sahin Tas, Kaiwen Wang, Royden Wagner, and Christoph Stiller. Divide and merge: Motion and semantic learning in end-to-end autonomous driving. *arXiv preprint arXiv:2502.07631*, 2025. 2

[55] 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*, 2024. 2, 3

[56] Ziying Song, Caiyan Jia, Lin Liu, Hongyu Pan, Yongchang Zhang, Junming Wang, Xingyu Zhang, Shaoqing Xu, Lei Yang, and Yadan Luo. Don't shake the wheel: Momentum-aware planning in end-to-end autonomous driving. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025. 6, 4

[57] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 2024. 4

[58] 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)*, 2025. 6, 4

[59] Yingqi Tang, Zhuoran Xu, Zhaotie Meng, and Erkang Cheng. Hip-ad: Hierarchical and multi-granularity planning with deformable attention for autonomous driving in a single decoder. *arXiv preprint arXiv:2503.08612*, 2025. 6

[60] Xiaoyu Tian, Junru Gu, Bailin Li, Yicheng Liu, Yang Wang, Zhiyong Zhao, Kun Zhan, Peng Jia, XianPeng Lang, and Hang Zhao. Drivevlm: The convergence of autonomous driving and large vision-language models. In *Conference on Robot Learning*, 2025. 2, 3, 6, 4

[61] Chi Wan, Yixin Cui, Jiatong Du, Shuo Yang, Yulong Bai, Peng Yi, Nan Li, and Yanjun Huang. Geminus: Dual-aware global and scene-adaptive mixture-of-experts for end-to-end autonomous driving. *arXiv preprint arXiv:2507.14456*, 2025. 6

[62] 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*, 2025. 2, 3, 4, 5, 6, 1

[63] 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*, 2022. 2

[64] Yuqi Wang, Jiawei He, Lue Fan, Hongxin Li, Yuntao Chen, and Zhaoxiang Zhang. Driving into the future: Multiview visual forecasting and planning with world model for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. 6, 4

[65] 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*, 2024. 2

[66] Diankun Wu, Fangfu Liu, Yi-Hsin Hung, and Yueqi Duan. Spatial-mllm: Boosting mllm capabilities in visual-based spatial intelligence. *arXiv preprint arXiv:2505.23747*, 2025. 2, 3

[67] Xian Wu, Yitao Wu, Xiaoyu Li, Zijia Li, Lijun Zhao, and Lining Sun. Fusion-poly: A polyhedral framework based on spatial-temporal fusion for 3d multi-object tracking. *arXiv preprint arXiv:2603.08199*, 2026. 2

[68] Yanhao Wu, Haoyang Zhang, Fei He, Rui Wu, Yanhu Shan, Congpei Qiu, Liang Gao, Wei Ke, and Tong Zhang. Aligndrive: Aligned lateral-longitudinal planning for end-to-end autonomous driving. *arXiv preprint arXiv:2601.01762*, 2026. 6

[69] Zhengri Wu, Yiran Wang, Yu Wen, Zeyu Zhang, Biao Wu, and Hao Tang. Stereoadapter: Adapting stereo depth estimation to underwater scenes. *arXiv preprint arXiv:2509.16415*, 2025. 3

[70] 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*, 2024. 3

[71] 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*, 2025. 2, 3

[72] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *Advances in Neural Information Processing Systems*, 2024. 2

[73] 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. 6

[74] Zhenjie Yang, Xiaosong Jia, Qifeng Li, Xue Yang, Maoqing Yao, and Junchi Yan. Raw2drive: Reinforcement learning with aligned world models for end-to-end autonomous driving (in carla v2). *arXiv preprint arXiv:2505.16394*, 2025. 6

[75] Hang Yu, Julian Jordan, Julian Schmidt, Silvan Lindner, Alessandro Canevaro, and Wilhelm Stork. Hype: Hybrid planning with ego proposal-conditioned predictions. *arXiv preprint arXiv:2510.12733*, 2025. 2

[76] Haibao Yu, Wenxian Yang, Ruiyang Hao, Chuanye Wang, Jiaru Zhong, Ping Luo, and Zaiqing Nie. Drivee2e: Closed-loop benchmark for end-to-end autonomous driving through real-to-simulation. *arXiv preprint arXiv:2509.23922*, 2025. 2

[77] Jianhao Yuan, Shuyang Sun, Daniel Omeiza, Bo Zhao, Paul Newman, Lars Kunze, and Matthew Gadd. Rag-driver: Generalisable driving explanations with retrieval-augmented in-context multi-modal large language model learning. In *Robotics: Science and Systems*, 2024. 3

[78] Jiang-Tian Zhai, Ze Feng, Jinhao Du, Yongqiang Mao, Jiang-Jiang Liu, Zichang Tan, Yifu Zhang, Xiaoqing Ye, and Jingdong Wang. Rethinking the open-loop evaluation of end-to-end autonomous driving in nuscenes. *arXiv preprint arXiv:2305.10430*, 2023. 2, 5, 6, 1

[79] Bozhou Zhang, Nan Song, Xiatian Zhu, Jiankang Deng, Li Zhang, et al. Future-aware end-to-end driving: Bidirectional modeling of trajectory planning and scene evolution. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. 2, 6

[80] Qingwen Zhang, Yi Yang, Peizheng Li, Olov Andersson, and Patric Jensfelt. Seflow: A self-supervised scene flow method in autonomous driving. In *European Conference on Computer Vision*. Springer, 2024. 2

[81] Wei Zhang, Pengfei Li, Junli Wang, Bingchuan Sun, Qihao Jin, Guangjun Bao, Shibo Rui, Yang Yu, Wenchao Ding, Peng Li, et al. Dual-aeb: Synergizing rule-based and multimodal large language models for effective emergency braking. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, 2025. 6

[82] Zhiyuan Zhang, Xiaofan Li, Zhihao Xu, Wenjie Peng, Zijian Zhou, Miaojing Shi, and Shuangping Huang. Mpdrive: Improving spatial understanding with marker-based prompt learning for autonomous driving. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025. 2

[83] Zhenjun Zhao. Balf: Simple and efficient blur aware local feature detector. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3362–3372, 2024. 2

[84] Zhenjun Zhao, Heng Yang, Bangyan Liao, Yingping Zeng, Shaocheng Yan, Yingdong Gu, Peidong Liu, Yi Zhou, Haoang Li, and Javier Civera. Advances in global solvers for 3d vision. *arXiv preprint arXiv:2602.14662*, 2026. 3

[85] Duo Zheng, Shijia Huang, and Liwei Wang. Video-3d llm: Learning position-aware video representation for 3d scene understanding. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025. 3

[86] Wenzhao Zheng, Ruiqi Song, Xianda Guo, Chenming Zhang, and Long Chen. Genad: Generative end-to-end autonomous driving. In *European Conference on Computer Vision*, 2024. 2, 6, 4

[87] 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. 3

[88] 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. 3

[89] Chenming Zhu, Tai Wang, Wenwei Zhang, Jiangmiao Pang, and Xihui Liu. Llava-3d: A simple yet effective pathway to empowering lmms with 3d-awareness. *arXiv preprint arXiv:2409.18125*, 2024. 3, 4

[90] Julian Zimmerlin, Jens Beißwenger, Bernhard Jaeger, Andreas Geiger, and Kashyap Chitta. Hidden biases of end-to-end driving datasets. *arXiv preprint arXiv:2412.09602*, 2024. 2, 6

SpaceDrive: Infusing Spatial Awareness into VLM-based Autonomous Driving

Supplementary Material

The main content of this supplementary material is organized as follows:

- Section A: More implementation details of our method;
- Section B: Additional experiments and ablation studies;
- Section C: Additional visualization comparisons and qualitative analysis.

A. Additional Implementation Details

A.1. SpaceDrive Framework

To ensure seamless adaptability, our method avoids any model-specific customization and fully preserves the original image preprocessing, patchification strategy, text tokenization, and chat template used by each base VLM model. Given the shape of the preprocessed visual patches, the depth map is resized accordingly using min-pooling (*e.g.* patch shapes of $6 \times 24 \times 24$ for LLaVA-1.5-7B [42] and $6 \times 23 \times 23$ for Qwen2.5-VL-7B [2] in our configuration). For the coordinate parsing of the text inputs, We employ a strict regex rule matching the format "(*X*, *Y*)" or "(*X*, *Y*, *Z*)". Substrings matching this pattern (*e.g.* "(7.5, -3.2)") are extracted and extended as the 3D coordinates (*e.g.* $(7.5, -3.2, 0)$) and converted to 3D PE. Any numeric string failing this match (*e.g.* "3") is treated as a standard textual token. This strict separation prevents ambiguity between spatial coordinates and general numerical values, ensuring that only spatial data triggers the PE en/decoding. Newly introduced tokens, such as ⟨IND⟩, are set as learnable and appended to the frozen input embedding layer and output language-model head. The PE decoder for coordinates is implemented as a standard two-layer MLP with the same hidden dimensionality as the base VLM. In all experiments, we set the seed to 888. The LoRA configurations are listed in Tab. A.

A.2. Training Details

Open-loop Planning For open-loop planning, we follow prior works and use 6 future trajectory points sampled at 2 Hz over a 3-second horizon as ground truth supervision. As emphasized in previous studies [39, 78], strong open-loop planning performance can be achieved using only ego-status. To rigorously validate the effectiveness of our framework, the standard SpaceDrive variant intentionally excludes motion dynamics and high-level driving commands (*e.g.* "go straight", "turn right") from its inputs. In this configuration, the model performs trajectory planning exclusively from image observations, enabling a clean evaluation of the spatial reasoning capability brought by our design. The variant SpaceDrive+ includes the current commands and ego dynamics of past 2 frames that are widely used in other works [13, 62].

Table A. **LoRA configurations for VLM fine-tuning.**

Setting	Rank (r)	Alpha (α)	Dropout	Target Modules
Value	16	16	0.05	q_proj, k_proj, v_proj, o_proj

Table B. **Counterfactual reasoning comparison in the open-loop planning (without ego status).** P and R here stand for Precision and Recall. Results are highlighted in **bold** and underline for the best and the second-best performance.

Method	Safe		Red Light		Collision		Drivable Area	
	P	R	P	R	P	R	P	R
BEV-MLP	70.2	17.3	48.7	53.6	31.1	70.4	32.4	56.6
Omni-L [62]	**72.1**	58.0	59.2	63.3	34.3	71.3	49.1	**59.2**
Omni-Q [62]	70.7	49.0	57.6	58.3	32.3	**72.6**	48.5	58.6
SpaceDrive (**ours**)	65.7	**63.6**	**70.3**	**72.7**	**37.5**	66.4	**55.0**	37.0

For VQA training and evaluation, we adopt the dataset provided by OmniDrive [62], which includes scene description, attention, counterfactual reasoning, planning, as well as other general conversations. Consistent with the implementation of OmniDrive, the other VQA tasks are appended subsequent to the trajectory planning task to ensure semantic stability.

Closed-loop Planning Inspired by SimLingo [51], we augment the supervision of 6 trajectory points with 20 additional path waypoints, uniformly spaced at 1-meter intervals. In this setup, the trajectory points serve two purposes: estimating the target speed and identifying the appropriate waypoint for the target direction. This leads to generally more stable steering regardless of whether the ego vehicle is moving or not. Two PID contollers are applied to determine acceleration and steering, respectively. During training, we use a subset of SimLingo routes containing 3600 episodes with PDM-lite as the expert driver.

A.3. Learnable Parameter Convergence

We report that the learnable norm factor α_{PE} (initialized at 0.1) generally converges to the range of $[0.085, 0.11]$ across different backbones and settings. For instance, in the default open-loop QwenVL-based setting, it stabilizes at 0.087.

B. Additional Experiments and Analyses

B.1. VQA for Counterfactual Reasoning

As aforementioned, to validate the spatial reasoning capabilities of SpaceDrive, we conduct counterfactual reasoning experiments following the setting in OmniDrive [62], as presented in Tab. B. In this evaluation, keywords such as

Table C. **Ablation of depth estimator.**

$f_{dep.}$	Avg. L2 ↓	Avg. Collision ↓	Avg. Intersection ↓
DepthAnythingV2 [72]	1.76	1.95	3.96
UniDepthV2 [50]	1.80	1.88	4.21

Table D. **Ablation of pooling strategy.**

Pooling Strategy	Avg. L2 ↓	Avg. Collision ↓	Avg. Intersection ↓
Min	1.80	1.88	4.21
Average	1.83	1.88	4.08
Median	1.83	2.17	4.64

"safety", "collision", "running a red light", and "out of the drivable area" are extracted from the VQA outputs and compared against ground truth keywords to compute Precision and Recall. The results demonstrate that our framework achieves superior performance across the majority of metrics, *e.g*. a Recall of 63.6% in the safety task. It is particularly noteworthy that, without any specific prompt engineering for the dialogue, the mere incorporation of the unified 3D spatial representation enables significantly higher Precision in tasks demanding rigorous spatial understanding, such as Collision (37.5%) and Drivable Area (55.0%). This further confirms our SpaceDrive possesses strong spatial reasoning capabilities.

B.2. More Ablation Studies

Depth Estimator In Tab. C, we compare the influence of different pre-trained depth estimator on the planning performance. DepthAnythingV2 [72] and UniDepthV2 [50] are selected as representative examples of relative and metric depth estimation models, respectively. We observe that both variants perform similarly on the L2 error metric and Collision rate, which are the most reliable indicator of planning performance. This suggests that the effectiveness of our SpaceDrive is independent of a specific pre-trained depth model, implicitly demonstrating the adaptability of our framework. Notably, LiDAR-based depth ground truth (GT) is inherently sparse and lacks valid depth values in regions such as the sky, necessitating manual definition. Together with factors like camera distortion and projection error, GT-based comparisons are unreliable and thus excluded from the comparison.

Depth Pooling Strategy Min pooling in our default setting may be sensitive to outliers, yet the compact patch resolution renders such artifacts empirically rare. In fact, every pooling strategy has its inherent limitations: average pooling blurs object boundaries, while median pooling induces intra-object depth discontinuities of adjacent patches. In contrast, min-depth is relatively reliable and safety-critical as it preserves the closest obstacles. Tab. D validates that min pooling yields the lowest L2 error and collision rate.

Table E. **Ablation of LoRA rank.** Learn. Par. is the abbreviation for the number of LoRA parameters when selecting Qwen2.5-VL-7B as the base VLM.

Rank	Learn. Par.	Avg. L2 ↓	Avg. Collision ↓	Avg. Intersection ↓
16	10.09M	1.80	1.88	4.21
64	40.37M	1.88	2.13	4.08
128	80.74M	1.82	2.25	4.68

Table F. **Ablation of PE frequency.**

Frequency	Avg. L2 ↓	Avg. Collision ↓	Avg. Intersection ↓
$1000^{-2i/d_a}$	1.78	2.01	3.93
$10000^{-2i/d_a}$	1.83	1.83	3.18
$20000^{-2i/d_a}$	1.80	1.88	4.21

LoRA Rank Table E presents a comparison of different LoRA [18] ranks in the VLM during fine-tuning. Benefiting from our universal spatial positional encoding, the coordinate regression process in the language model is simplified. Utilizing only low-rank fine-tuning (rank 16) achieves the optimal overall result (L2 error of 1.80, Collision rate of 1.88%, and Intersection rate of 4.21%). While increasing the rank to 128 substantially raises the number of learnable parameters from 10.09M to 80.74M, it fails to improve the planning accuracy and, instead, leads to a degradation in Collision and Intersection rates. We attribute this to the excessive degrees of training freedom in the high-rank adapter, which hinders the convergence. The above comparison further demonstrates that our method not only offers stronger planning reliability but also maintains parameter efficiency.

PE Frequency Table F investigates the base frequency of the Sin-Cos PE, which impacts both encoding resolution and smoothness. Utilizing a smaller frequency base (corresponding to a higher frequency) introduces a larger phase shift between adjacent positions but leads to positional aliasing at long distances, thereby inhibiting the representation of far-field positions. As shown in the comparison, setting the base to 1000 enhances local resolution and achieves the lowest L2 error of 1.78. However, distant coordinates exhibit near-random phase characteristics, which compromises overall safety (leading to worse Collision and Intersection rates). Conversely, an excessively large base (*e.g*. 20000) generates smoother, more stable encodings over long distances but diminishes local discriminative capability. Compared to the original base of 10000, the resulting L2 error reduction is less pronounced, at only -0.03, but the collision rate increase is negligible. Overall, comparing all variants reveals that the influence of different PE frequencies is relatively limited and non-decisive. We finally adopt 20000 as our PE frequency base.

Regression Loss In Tab. G, we compares different regression losses for trajectory prediction. MAE provides robustness to

Table G. **Ablation of regression loss.**

$\mathcal{L}_{\text{reg.}}$	Avg. L2 ↓	Avg. Collision ↓	Avg. Intersection ↓
MAE	1.86	1.82	5.73
MSE	1.82	2.14	6.22
Huber Loss	1.80	1.88	4.21

Table H. **Robustness to Depth Estimation Errors.**

Depth Noise		Avg. L2 ↓	Avg. Collision ↓	Avg. Intersection ↓
Training & Inference	w. -5% Global Shift	1.86	1.80	5.22
	w. 5% Global Shift	1.86	1.93	4.41
	w. 5% Random Noise	1.83	2.16	4.44
Inference only	w. -2.5% Global Shift	1.80	1.89	4.22
	w. 2.5% Global Shift	1.80	1.86	4.24
	w. 2.5% Random Noise	1.80	1.89	4.17

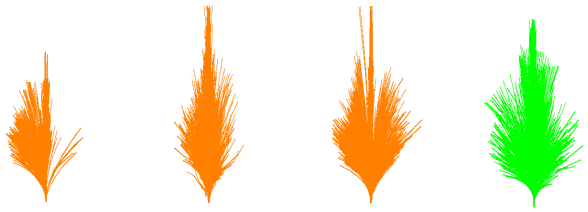
outliers but yields the worst L2 and intersection metrics, suggesting insufficient pressure on medium-scale errors. MSE reduces L2 compared to MAE, but its quadratic growth on large residuals makes optimization more sensitive to outliers, leading to noticeably higher collision and intersection rates. Huber loss strikes a balance between them and achieves the best L2 error together with markedly improved safety metrics. So we adopt Huber loss as our final regression objective.

B.3. Robustness to Depth Estimation Errors

We present experiments when injecting relative depth noise ($\pm$ 5% Global Shift or 5% Random Noise) in inference or training & inference. As shown in Tab. H, injecting noise *during inference* results in negligible performance drop (*e.g.* Avg. L2 remains 1.80). This confirms that SpaceDrive is robust to depth inaccuracies. It relies on *3D spatial structure* rather than precise metric depth, allowing VLM's semantic understanding to compensate for noise. Conversely, noise injection *during training* slightly degrades performance. This verifies that depth information is actively utilized in training, but the learned policy generalizes well against test-time perturbations. In summary, SpaceDrive treats depth as a geometric guide for attention, not strong geometric priors.

B.4. Comprehensive Benchmark

Constrained by the limited space in the main paper, we list only the primary relevant works in the benchmark comparisons. Therefore, we provide more comprehensive benchmark comparisons for open-loop and closed-loop planning in Tab. I and Tab. J, respectively. It is worth noting that existing nuScenes [4] open-loop evaluations utilize differing sets of metrics in different studies. While the main paper employs the OmniDrive [62] version commonly used by VLM-based frameworks, Table I provides results derived using the evaluation metrics from ST-P3 [19] and UniAD [20].



(a) OmniDrive-L (b) SpaceDrive (c) SpaceDrive-CoT (d) Ground Truth

Figure A. **Trajectory distribution of open-loop planning for different frameworks and ground truth.**

C. Additional Visualization

C.1. More Adaptability Analysis

Figure A illustrates the distribution of planned trajectories across all scenarios under the open-loop setting of nuScenes [4]. We first analyze the output trajectory distribution of OmniDrive-L [62], a typical scheme utilizing textual digit tokens for waypoint coordinates, shown in Fig. A.a. Due to the VLM's limitations in numerical processing, as discussed in Section 1, OmniDrive-L exhibits clear mode collapse for right-turn cases. In sharp contrast, our SpaceDrive, which is based on the universal 3D PE representation, significantly mitigates this issue, as shown in Fig. A.b. Furthermore, when adopting inference techniques such as Chain-of-Thought during inference, the output trajectory planning demonstrates enhanced robustness (Fig. A.c) and closer alignment with the ground truth distribution (Fig. A.d). This result further supports the strong adaptability of our method to language model inference techniques.

C.2. Failure Analysis of Textual Coordinate Output

A further quantitative analysis is conducted to assess the driving capability of conventional VLM-based models that output trajectory coordinates as textual digit tokens in closed-loop simulation, as shown in Fig. B. We the exact same scenario as in Fig. 3 and employ OmniDrive-L [62], a framework structurally analogous to SpaceDrive, utilizing the same closed-loop training configuration as in Sec. A.2. This figure clearly illustrates that in the closed-loop setting, the planned trajectories generated by OmniDrive-L collapse into an approximately straight line, and the directional control exhibits random oscillation. This phenomenon aligns with the mode collapse previously observed during open-loop evaluation (See Sec. C.1). Critically, this oscillation is amplified over time, leading to vehicle instability and ultimately making the vehicle veer off the road and collide with the guardrail. This result provides strong empirical support for our analysis in Sec. 4.2: purely text-based trajectory coordinate output from VLMs is inadequate for reliable closed-loop driving.

Table I. **Open-loop planning results on nuScenes [4].** SpaceDrive+ denotes the adoption of the ego planner input. ‡: The model is trained using only the trajectory prediction task for open-loop planning, without utilizing our generated OmniDrive Q&A data. Methods marked as Hybrid Paradigm here stack traditional and VLM-based approaches, and are thus incomparable. Results are highlighted in **bold** and underline for the best and the second-best performance among VLM-based methods.

Method	Ego Status		ST-P3 Metrics								UniAD Metrics							
			L2 (m) ↓				Collision (%) ↓				L2 (m) ↓				Collision (%) ↓			
	BEV	Planner	1s	2s	3s	Avg.	1s	2s	3s	Avg.	1s	2s	3s	Avg.	1s	2s	3s	Avg.
Traditional Modular Paradigm																		
ST-P3 [19]	-	-	1.33	2.11	2.90	2.11	0.23	0.62	1.27	0.71	1.72	3.26	4.86	3.28	0.44	1.08	3.01	1.51
UniAD [20]	-	-	0.44	0.67	0.96	0.69	0.04	0.08	0.23	0.12	0.48	0.96	1.65	1.03	0.05	0.17	0.71	0.31
VAD-Base [28]	-	-	0.41	0.70	1.05	0.72	0.07	0.17	0.41	0.22	0.54	1.15	1.98	1.22	0.10	0.24	0.96	0.43
UAD [15]	-	-	0.28	0.41	0.65	0.45	0.01	0.03	0.14	0.06	0.39	0.80	1.50	0.90	0.01	0.12	0.43	0.19
MomAD [56]	-	-	0.28	0.49	0.78	0.52	0.08	0.14	0.34	0.19	0.36	0.83	1.56	0.91	0.06	0.23	1.00	0.43
GenAD [86]	-	✓	0.31	0.57	0.91	0.60	0.01	0.05	0.22	0.09	0.43	0.88	1.62	0.98	0.06	0.16	0.68	0.30
Drive-WM [64]	✓	✓	0.43	0.77	1.20	0.80	0.10	0.21	0.48	0.26	-	-	-	-	-	-	-	-
SparseDrive [58]	-	✓	0.29	0.55	0.91	0.58	0.01	0.02	0.13	0.06	0.44	0.92	1.69	1.01	0.07	0.19	0.71	0.32
DiffusionDrive [41]	-	✓	0.27	0.54	0.90	0.57	0.03	0.05	0.16	0.08	-	-	-	-	-	-	-	-
VLM-based Paradigm																		
EMMA [23]	-	-	**0.14**	**0.29**	0.54	**0.32**	-	-	-	-	-	-	-	-	-	-	-	-
RDA-Driver [22]	✓	✓	0.17	0.37	0.69	0.40	**0.01**	**0.05**	0.26	**0.10**	0.23	0.73	1.54	0.80	**0.00**	**0.13**	0.83	**0.32**
DriveVLM [60]	-	✓	0.18	0.34	0.68	0.40	-	-	-	-	-	-	-	-	-	-	-	-
ORION [13]	✓	-	0.17	0.31	0.55	0.34	-	-	-	-	-	-	-	-	-	-	-	-
OmniDrive-Q [62]	-	-	1.15	1.96	2.84	1.98	-	-	-	-	-	-	-	-	-	-	-	-
OmniDrive-Q++ [62]	✓	✓	**0.14**	**0.29**	0.55	0.33	-	-	-	-	-	-	-	-	-	-	-	-
OmniDrive-L‡ [62]	-	-	1.47	2.43	3.38	2.43	-	-	-	-	-	-	-	-	-	-	-	-
OmniDrive-L++‡ [62]	-	✓	0.31	0.62	1.06	0.66	-	-	-	-	-	-	-	-	-	-	-	-
OmniDrive-L [62]	-	-	1.43	2.34	3.24	2.34	-	-	-	-	-	-	-	-	-	-	-	-
OmniDrive-L++ [62]	-	✓	0.15	0.36	0.70	0.40	-	-	-	-	-	-	-	-	-	-	-	-
SpaceDrive (**ours**)	-	-	1.06	1.79	2.55	1.80	0.35	0.61	1.31	0.76	1.41	2.88	4.51	2.93	0.59	1.72	4.53	2.28
SpaceDrive+ (**ours**)	-	✓	0.15	**0.29**	**0.51**	**0.32**	0.05	0.08	**0.16**	**0.10**	**0.20**	**0.53**	**1.13**	**0.62**	0.10	0.31	**0.80**	0.40
Hybrid Paradigm																		
VLP [49]	✓	-	0.30	0.53	0.84	0.55	0.01	0.07	0.38	0.15	0.36	0.68	1.19	0.74	0.03	0.12	0.32	0.16
ReAL-AD [47]	✓	-	0.30	0.48	0.67	0.48	0.07	0.10	0.28	0.15	0.40	0.71	1.14	0.77	0.02	0.12	0.37	0.17
DriveVLM-Dual [60]	✓	-	0.15	0.29	0.48	0.31	0.05	0.08	0.17	0.10	-	-	-	-	-	-	-	-
SOLVE-VLM [7]	✓	-	0.13	0.25	0.47	0.28	-	-	-	-	-	-	-	-	-	-	-	-
Senna [29]	✓	-	0.11	0.21	0.35	0.22	0.04	0.08	0.13	0.08	-	-	-	-	-	-	-	-

Figure B. **Qualitative results of OmniDrive-L [62] in closed-loop setting on Bench2Drive [26].** Green and pink dots represent path and speed waypoints, respectively. Parameters such as speed and steering wheel angle can be found in the figures.

C.3. More Qualitative Closed-Loop Results

We present additional closed-loop simulation visualizations for SpaceDrive in Fig. C, covering 3 representative safety-critical scenarios: (a) navigating around a construction zone requiring a brief excursion into the oncoming lane; (b) decelerating and yielding due to a sudden pedestrian crossing during normal driving; and (c) performing an emergency stop and yielding to an ambulance rapidly approaching from

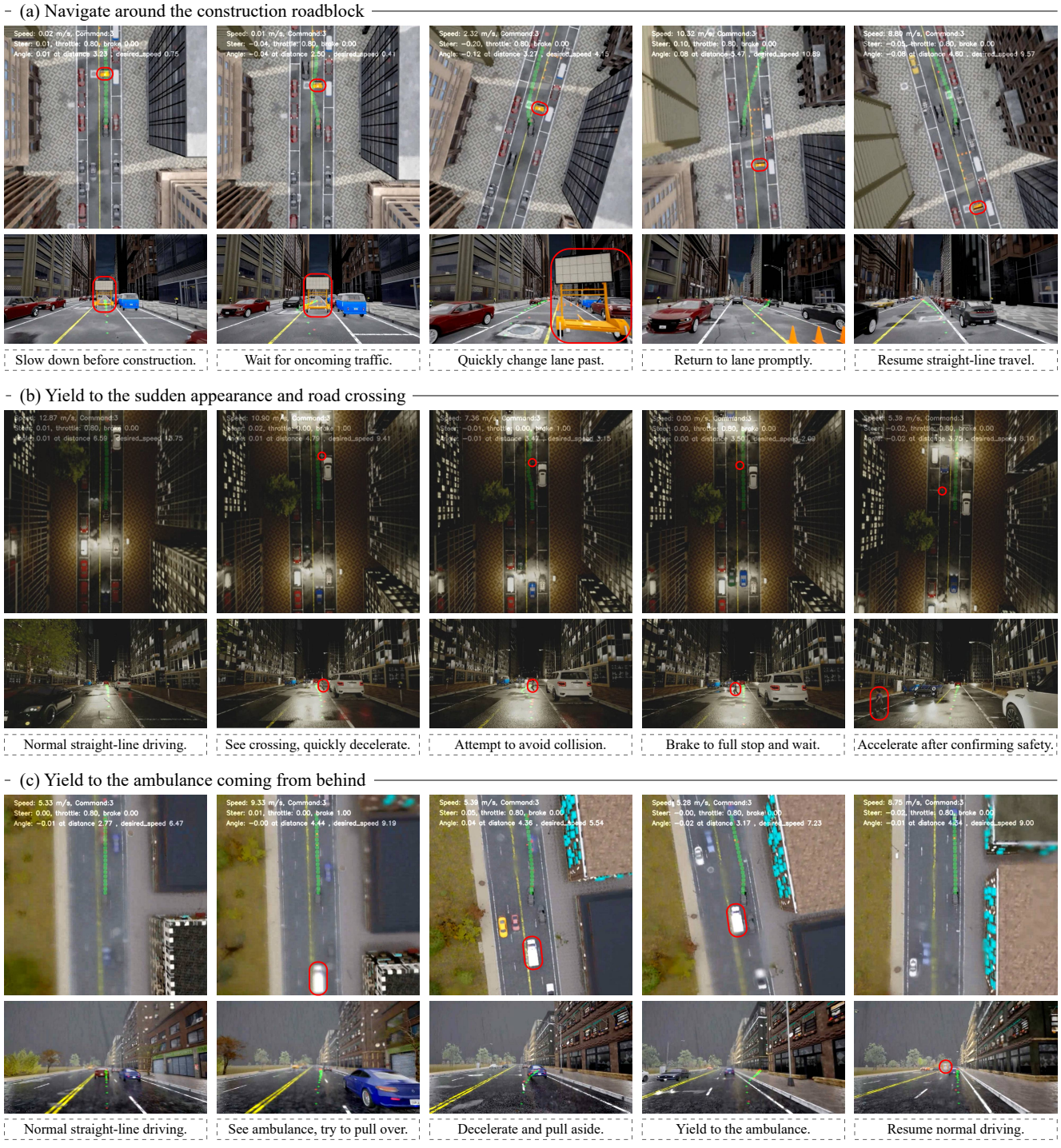


Figure C. **More qualitative results of closed-loop evaluation on Bench2Drive [26].** We include 3 scenarios here to demonstrate the closed-loop planning capability of SpaceDrive: (a) urban road construction; (b) a sudden pedestrian crossing; (c) yielding to an ambulance. Green and pink dots represent path and speed waypoints, respectively. Red circles indicate objects requiring attention in the scenario. Parameters such as speed and steering wheel angle can be found in the figures.

the rear. All these scenarios demand the model to quickly establish a deep understanding of the 3D spatial context and generate a sound trajectory in a minimal timeframe. The visualizations clearly indicate that our proposed framework, by

Table J. **Closed-loop planning results on Bench2Drive [26].** Results are highlighted in **bold** and underline for the best and the second-best performance among VLM-based methods.

Method	Closed-loop Metric	
	Driving Score ↑	Success Rate(%) ↑
Traditional Modular Paradigm		
AD-MLP [78]	18.05	0.00
UniAD-Base [20]	45.81	16.36
VAD-Base [28]	42.35	15.00
MomAD [56]	44.54	16.71
GenAD [86]	44.81	15.90
SparseDrive [58]	47.38	17.72
UAD [15]	49.22	20.45
SeerDrive [79]	58.32	30.17
WoTE [36]	61.71	31.36
DriveDPO [52]	62.02	30.62
ThinkTwice [25]	62.44	37.17
DriveTransformer-L [27]	63.46	38.60
DriveAdapter [24]	64.22	42.08
GEMINUS [61]	65.39	37.73
Raw2Drive [74]	71.36	50.24
Hydra-NeXt [40]	73.86	53.22
DiffusionDrive [41]	77.68	52.72
PGS [21]	78.08	48.64
GaussianFusion [44]	79.10	54.40
TF++ [90]	84.21	64.39
R2SE [43]	86.28	67.76
HiP-AD [59]	86.77	69.09
AlignDrive [68]	89.07	73.18
VLM-based Paradigm		
ReAL-AD [47]	41.17	11.36
Dual-AEB [81]	45.23	10.00
X-Driver [45]	51.70	18.10
VDRive [16]	66.25	50.51
StuckSolver [3]	70.89	50.01
DriveMoE [73]	74.22	48.64
ETA [17]	74.33	48.33
VLR-Drive [31]	75.01	50.00
ORION [13]	77.74	54.62
SimLingo [51]	**85.07**	**67.27**
SpaceDrive+ (**ours**)	78.02	55.11

leveraging its unified 3D representation, effectively manages these critical, unforeseen situations. This further substantiates the efficacy of our proposed SpaceDrive framework.