ONEDRIVE: UNIFIED MULTI-PARADIGM DRIVING WITH VISION-LANGUAGE-ACTION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision-Language Models (VLMs) excel at autoregressive text generation, yet end-to-end autonomous driving requires multi-task learning with structured outputs and heterogeneous decoding behaviors, such as autoregressive language generation, parallel object detection and trajectory regression. To accommodate these differences, existing systems typically introduce separate or cascaded decoders, resulting in architectural fragmentation and limited backbone reuse. In this work, we present a unified autonomous driving framework built upon a pretrained VLM, where heterogeneous decoding behaviors are reconciled within a single transformer decoder. We demonstrate that pretrained VLM attention exhibits strong transferability beyond pure language modeling. By organizing visual and structured query tokens within a single causal decoder, structured queries can naturally condition on visual context through the original attention mechanism. Textual and structured outputs share a common attention backbone, enabling *stable joint optimization across heterogeneous tasks*. Trajectory planning is realized within the same causal LLM decoder by introducing structured trajectory queries. This unified formulation enables planning to share the pretrained attention backbone with images and perception tokens. Extensive experiments on end-to-end autonomous driving benchmarks demonstrate state-of-the-art performance, including 0.28 L2 and 0.18 collision rate on nuScenes open-loop evaluation and competitive results (87.3 PDMS) on NAVSIM closed-loop evaluation. The full model preserves multi-modal generation capability, while an efficient inference mode achieves approximately 40% lower latency.

1 INTRODUCTION

Recent breakthroughs in Vision Language Models (VLMs) Wang et al. (2024a); Bai et al. (2025); Chen et al. (2024d); Zhu et al. (2025) highlight their extraordinary multimodal reasoning capabilities. This success naturally inspires the pursuit of Vision Language Action (VLA) models for autonomous driving Zhou et al. (2025); Li et al. (2025). Yet integrating these foundational models into driving systems typically demands intricate 3D structural modifications like Bird Eye View (BEV) modeling Jiang et al. (2024); Han et al. (2025). Such severe architectural deviations from native VLMs hinder the possibility of joint training with massive general domain data, fundamentally bottlenecking model scalability. Conversely, attempting to preserve the original architecture exposes a critical dilemma. When structural changes are kept minimal, the model falls significantly behind leading methods on diverse heterogeneous driving tasks. Although previous works manage to mask this deficiency by relying heavily on point cloud inputs, their performance degrades sharply under pure vision settings. This degradation is highly undesirable since pure vision settings naturally align with the original VLM architecture Han et al. (2025). Furthermore, given that these models are inherently constrained to output information in **textual form**, bridging the massive gap between pure text generation and complex driving outputs remains a formidable challenge. Then,

> *Can a single pretrained multimodal model simultaneously handle diverse E2E autonomous driving tasks with various forms of output, while maintaining coherent textual generation and generalizing to generative modeling paradigms?*

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Tackling this question requires confronting the inherent complexity of autonomous driving, where interdependent tasks demand distinct input and output formats. Perception and planning typically employ parallel decoders with specific attention mechanisms for simultaneous predictions Carion et al. (2020); Zhu et al. (2020); Zhang et al. (2026); Li et al. (2022); Yang et al. (2023); Liao et al. (2023a;b); Zhang et al. (2024); Jiang et al. (2023b); Chen et al. (2024b); Weng et al. (2024a); Chitta et al. (2022). Conversely, textual reasoning relies on sequential autoregressive decoding Sima et al. (2023); Chen et al. (2025); Wang et al. (2024b); Jiang et al. (2024) (see Fig. 2 (a)). This profound structural gap forces traditional approaches to use isolated Weng et al. (2024a) or cascaded Hu et al. (2023b); Jiang et al. (2023b); Jia et al. (2025); Wang et al. (2024b); Chen et al. (2025); Tian et al. (2024a) decoders. Such fragmented designs strictly prevent the unified application of pretrained weights and restrict information flow across tasks. Consequently, engineering a singular architecture that harmonizes these diverse modalities while preserving the native VLM structure remains a formidable open challenge.

Table 1: Diagnostic study on transferring pretrained LLM weights to a parallel decoder. ✗ means random initialize.

VLM	Attn	FFN	NDS↑
InternVL3-1B	✓	✓	31.95
	✓	✗	**32.05**
	✗	✓	29.90
	✗	✗	31.48
Qwen2.5-VL-3B	✓	✓	27.14
	✓	✗	**31.37**
	✗	✓	27.95
	✗	✗	30.15

To determine whether native pretrained VLM models can actually overcome this challenge and support diverse driving tasks, we conduct a preliminary diagnostic study. We adapt pretrained VLM weights into a parallel decoder using the first six layers of the language model, inspired by DETR Carion et al. (2020) and StreamPETR Wang et al. (2023b). Specifically, we freeze the visual backbone and map each pretrained self attention module to the corresponding cross attention in the parallel decoder. For the feedforward networks, we compare *pretrained initialization* against *random initialization*, alongside a fully random baseline for all modules to ensure completeness. As Tab. 1 demonstrates, attention weights transfer effectively across tasks. Conversely, reusing feedforward weights provides limited benefits and occasionally causes severe performance degradation alongside training instability (31.37 $\rightarrow$ 27.14). These observations reveal that feedforward networks optimized strictly for text generation lack the flexibility required for heterogeneous downstream tasks. However, attention modules pretrained to align visual and textual tokens retain remarkable transferability. This success likely stems from their ability to capture fundamental correspondence patterns between queries and visual features. Inspired by this crucial insight, we investigate whether such transferable attention mechanisms can model a broader range of relationships, ultimately enabling the construction of a unified multitask decoder.

OneDrive (see Fig. 3) is our definitive answer. It is a unified multitask framework elegantly leverages pretrained attention to support heterogeneous decoding behaviors within a single transformer decoder. Specifically, we organize visual tokens and structured query tokens into a unified sequence. This arrangement allows query tokens to deeply condition on visual context through the pretrained causal attention of the language model. To enable structured prediction, we augment the shallow layers with dedicated attention among perception query tokens. This modification enhances comprehensive perception capabilities while keeping the foundational backbone attention strictly unchanged. Furthermore, we insert task-specific feedforward networks to execute necessary feature transformations. This elegant design allows autoregressive textual generation alongside parallel perception and planning to share a common attention backbone. Consequently, our design enables stable joint optimiza-

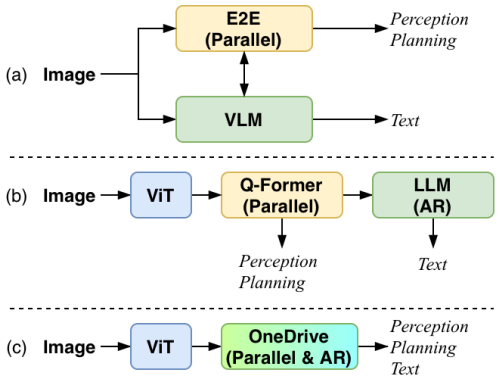

Figure 1: (a) dual-system design with separate decoders; (b) Q-Former–style cascaded decoding; (c) our unified single-decoder framework handling both within one transformer.

tion across diverse tasks without introducing isolated transformer branches. Moreover, the unified sequence and shared attention structure inherently facilitate seamless interaction across all driving tasks.

Extensive experiments on end-to-end autonomous driving benchmarks demonstrate that our framework achieves state-of-the-art performance, including 0.28 L_2 error and 0.18 collision rate on nuScenes open-loop evaluation and competitive results on NAVSIM closed-loop evaluation. The full model preserves multi-modal generation capability, while a truncated inference mode that forwards only the early layers reduces latency by approximately 40% (263 $\rightarrow$ 157 ms) for planning-focused deployment.

The main contributions of this work are: **(i)** We show that pretrained VLM causal attention, originally trained to capture text–image or text–text relations, can be adapted to model relations between structured queries and visual tokens, whereas pretrained text FFNs transfer less effectively. **(ii)** We propose OneDrive, a unified multitask architecture in which a single transformer decoder supports autoregressive text generation together with parallel perception and planning. **(iii)** Extensive experiments on nuScenes and NAVSIM demonstrate the effectiveness and efficiency of the proposed approach.

2 Related Work

Unified Architectures for End-to-end Autonomous Driving. Autonomous driving requires the integration of multiple interdependent tasks, including perception, prediction, and planning, such as 3D object detection Huang et al. (2021); Li et al. (2022); Yang et al. (2023); Liu et al. (2023), lane detection Liao et al. (2023a;b); Chen et al. (2024a), BEV segmentation Liu et al. (2023); Zhang et al. (2024), occupancy prediction Zheng et al. (2024b;a), and trajectory planning. Early systems decomposed these tasks into separate modules with predefined interfaces, which limited information flow and hindered global optimization. Recent unified frameworks Hu et al. (2023b); Jiang et al. (2023b); Chen et al. (2024b); Weng et al. (2024a); Jia et al. (2025), often referred to as conventional end-to-end driving models Yang et al. (2025), advocate for the joint optimization of perception, mapping, motion forecasting, and planning within a shared backbone. These approaches demonstrate that multi-task learning effectively enhances cross-task consistency and overall computational efficiency. However, these systems do not incorporate language modeling into the unified architecture. Notably, the autoregressive decoding paradigm of language models (Fig. 2(a)) differs fundamentally from the parallel structured decoders (Fig. 2(b)) commonly adopted in end-to-end driving frameworks.

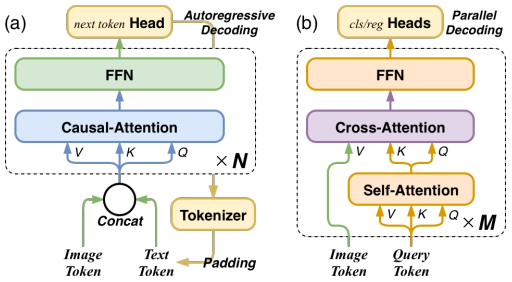

Figure 2: Two representative decoding paradigms: (a) an autoregressive decoder, (b) a parallel decoder. Existing end-to-end multi-task autonomous driving models typically organize heterogeneous decoders either in a cascaded manner or in parallel.

Vision-Language Models in Autonomous Driving. Recent studies have explored integrating vision-language models into autonomous driving to enhance scene understanding, instruction following, and interpretability. Dual-system designs (see Fig. 1(a)), such as Senna Jiang et al. (2024) and DriveVLM Tian et al. (2024a), employ separate decoders for structured prediction and language generation. Other works, including OmniDrive Wang et al. (2024b), Orion Fu et al. (2025) and SOLVE Chen et al. (2025), adopt cascaded architectures that combine structured decoders with autoregressive LLM decoders (see Fig. 1(b)). In contrast, some approaches Hwang et al. (2024); Xing et al. (2024) directly formulate all tasks as text generation using a single autoregressive decoder. Despite these efforts, existing methods typically organize heterogeneous decoders either in parallel or in cascaded forms. Conventional end-to-end driving models rely primarily on parallel decoding, while large language models operate in an autoregressive manner. The structural discrepancy between autoregressive token generation and parallel structured prediction remains largely unresolved.

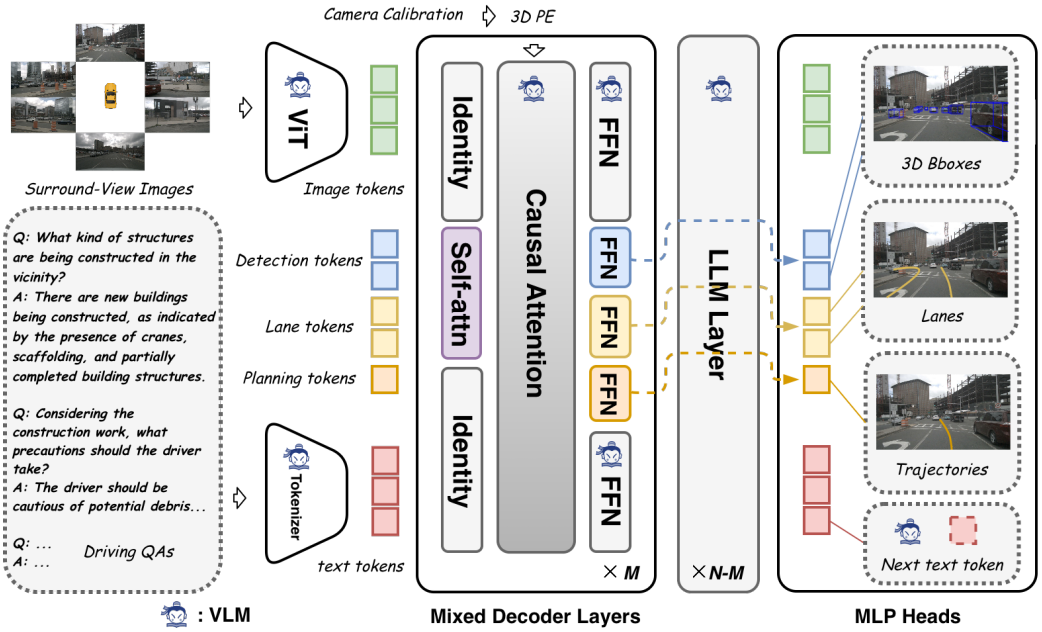

Figure 3: OneDrive architecture. A ViT encodes surround-view images into tokens, concatenated with detection, lane, planning, and text tokens. Mixed decoder layers retain pretrained LLM causal attention and add self-attention for perception queries and task-specific FFNs. MLP heads predict 3D boxes, lanes, and trajectories in parallel, alongside autoregressive text generation, within a single multi-task decoder.

In contrast to prior VLM-based driving works that treat language as an auxiliary modality and unified driving frameworks that integrate tasks through shared representations, our work focuses on unifying heterogeneous driving outputs at the decoding level. We adopt a token-centric formulation based on a pretrained VLM, enabling a single Transformer decoder to generate structured perception, planning trajectories, and textual responses within a unified causal attention framework.

3 ONEDRIVE

Table 1 shows that pretrained VLM attention transfers well to heterogeneous driving tasks, whereas feedforward layers often degrade performance, motivating a tailored multitask decoder. OneDrive (Fig. 3) unifies perception, planning, and language generation in one Transformer decoder. It processes concatenated visual, structured query, and text tokens using pretrained causal attention, adding self-attention among perception queries and task-specific FFNs to accommodate heterogeneous predictions.

3.1 PROBLEM FORMULATION

We formulate end-to-end driving as multi-task prediction from multi-view images $\mathcal{I}$ and optional text: **(i)** structured perception, including 3D boxes and lanes; **(ii)** continuous planning trajectories; and **(iii)** optional textual descriptions. These outputs require both parallel query-based prediction and autoregressive generation.

3.2 UNIFIED TOKEN REPRESENTATION

We represent detection and lanes with fixed sets of learnable queries, planning with dedicated tokens, and text with autoregressive tokens. A single Transformer decoder processes their concatenation with image tokens:

$$\mathbf{Z} = [\mathbf{X}_{img}, \mathbf{Q}_{det}, \mathbf{Q}_{lane}, \mathbf{Q}_{plan}, \mathbf{X}_{text}],$$

where $\mathbf{X}_{img}$, $\mathbf{Q}.$, and $\mathbf{X}_{text}$ denote image, task-specific query, and text tokens, respectively. Following SOLVE Chen et al. (2025) and OmniDrive Wang et al. (2024b), we construct detection and lane queries using StreamPETR Wang et al. (2023b). For planning, we assign one query per future timestep, initialized with a VAD-derived anchor trajectory Jiang et al. (2023b), and prepend an ego-status embedding to form $\mathbf{Q}_{plan}$.

3.3 Mixed Decoder Layers

Although all tokens share a unified representation space, heterogeneous tasks entail distinct interaction patterns and feature transformation requirements. To preserve the transferable relational modeling capacity of pretrained VLMs while enabling parallel structured prediction, mixed decoder layers are introduced in the shallow stages, retaining the original causal attention while incorporating minimal task-specific adaptations. The query outputs for structured prediction are directly taken from these shallow mixed layers, whereas the deeper layers remain unchanged from the pretrained decoder.

Causal Attention as Conditional Interface. We retain pretrained LLM self-attention with a shared causal mask over image, perception, planning, and text tokens. Following StreamPETR, we add 3D positional embeddings Liu et al. (2022) after RoPE Su et al. (2024) for image and structured query tokens to enhance spatial modeling:

$$\mathbf{Q} = \text{RoPE}(\mathbf{X}W_q) + e^{3D}, \quad \mathbf{K} = \text{RoPE}(\mathbf{X}W_k) + e^{3D},$$

where $\mathbf{X} \in \{\mathbf{X}_{img}, \mathbf{Q}_{det}, \mathbf{Q}_{lane}, \mathbf{Q}_{plan}\}$ and e^{3D} is the corresponding 3D embedding. Text tokens retain the original RoPE without 3D encoding. For query tokens, removing the causal-attention residual connection stabilizes adaptation from pretrained VLM weights. This shared attention backbone serves as a unified conditional modeling interface across modalities and tasks. Since query tokens are placed after image tokens, they naturally condition on visual context via causal attention without requiring explicit cross-attention modules. Planning tokens are further appended after perception queries, enabling implicit conditioning on perception features through the same attention mechanism.

Query Interaction and Task-Specific Transformation. Causal attention imposes sequential dependencies that are suboptimal for parallel prediction. For end-to-end perception Carion et al. (2020); Sun et al. (2021), we add self-attention among structured perception queries:

$$\mathbf{Q}_{\mathbf{perception}} = \text{SelfAttn}_q(\mathbf{Q}_{\mathbf{perception}}) = \text{SelfAttn}_q([\mathbf{Q}_{\mathbf{det}}, \mathbf{Q}_{\mathbf{lane}}]),$$

This enables bidirectional interaction within each query group without changing backbone causal attention. Structured queries also use task-specific FFNs in place of pretrained language-modeling FFNs:

$$\mathbf{Q}' = \text{FFN}_t(\tilde{\mathbf{Q}}),$$

where $t \in \{\text{det}, \text{lane}, \text{plan}\}$. Text tokens retain pretrained FFNs, preserving transferable attention while adapting feature transformations to each task.

3.4 Multi-Stage Training

We adopt a three-stage training strategy for stable adaptation from pretrained VLMs to multi-task driving. In the first stage (**Perception–language pretraining**), we freeze the ViT encoder and train the mixed decoder with perception and text-generation objectives. The causal attention is fully fine-tuned, the LLM decoder is adapted via LoRA, and the perception-specific self-attention, perception FFNs, and MLP heads are randomly initialized and optimized. Gradients are driven by perception and language modeling losses, denoted as $\mathcal{L}_{\text{pretrain}} = \lambda_{perc}\mathcal{L}_{\text{perc}} + \mathcal{L}_{\text{text}}$. In the second stage (**Planning adaptation**), planning tokens are introduced. We optimize planning queries, the planning FFN, and the planning MLP head in an end-to-end manner, while continuing LoRA adaptation of the LLM decoder. Perception-specific modules remain fixed. Gradients are provided by planning and language modeling loss $\mathcal{L}_{\text{adaptation}} = \lambda_{plan}\mathcal{L}_{\text{plan}} + \mathcal{L}_{\text{text}}$. In the final stage (**Joint finetuning**), all modules, including the ViT encoder, are jointly fine-tuned under the combined perception, planning,

and text objectives, with the overall training loss $\mathcal{L}_{\text{joint}} = \lambda_{perc}\mathcal{L}_{\text{perc}} + \lambda_{plan}\mathcal{L}_{\text{plan}} + \mathcal{L}_{\text{text}}$, enabling full end-to-end optimization of the framework. For object detection Zhu et al. (2020); Zhang et al. (2026); Wang et al. (2023b); Li et al. (2022), deep supervision Lee et al. (2015) is often applied to propagate gradients at every layer, where each decoder layer computes its own loss. Similarly, we employ deep supervision for the planning task.

4 EXPERIMENTS

Table 2: Performance comparison of different methods on the nuScenes dataset for open-loop planning. Methods are categorized into text-based driving models (top) and action-based driving models (bottom). † indicates that the model does not use ego-vehicle status as input.

Method	Reference	L2 (m) ↓				Col. Rate (%) ↓			
		$1s$	$2s$	$3s$	Avg.	$1s$	$2s$	$3s$	Avg.
Text-Based Driving Models									
DriveVLM Tian et al. (2024b)	CoRL 2024	0.18	0.34	0.68	0.40	–	–	–	–
DriveVLM-Dual Tian et al. (2024b)	CoRL 2024	0.15	0.29	0.48	0.31	–	–	–	–
OmniDrive Wang et al. (2024b)	CVPR 2025	0.14	0.29	0.55	0.33	**0.00**	0.13	0.78	0.30
EMMA Hwang et al. (2024)	TMLR	0.14	0.29	0.54	0.32	–	–	–	–
EMMA+ Hwang et al. (2024)	TMLR	**0.13**	0.27	0.48	0.29	–	–	–	–
ImpromptuVLA Chi et al. (2025)	NeurIPS 2025	**0.13**	0.27	0.53	0.30	–	–	–	–
SOLVE-VLM Chen et al. (2025)	CVPR 2025	**0.13**	**0.25**	**0.47**	**0.28**	**0.00**	0.16	**0.43**	**0.20**
VGGDrive Wang et al. (2026)	CVPR 2026	0.14	0.28	0.51	0.31	0.02	**0.10**	0.55	0.22
Action-Based Driving Models									
UniAD† Hu et al. (2023a)	CVPR 2023	0.59	1.01	1.48	1.03	0.16	0.51	1.64	0.77
VAD-Base† Jiang et al. (2023a)	ICCV 2023	0.69	1.22	1.83	1.25	0.06	0.68	2.52	1.09
BEV-Planner† Li et al. (2024b)	CVPR 2024	0.30	0.52	0.83	0.55	0.10	0.37	1.30	0.59
UniAD Hu et al. (2023a)	CVPR 2023	0.20	0.42	0.75	0.46	0.02	0.25	0.84	0.37
VAD-Base Jiang et al. (2023a)	ICCV 2023	0.17	0.34	0.60	0.37	0.04	0.27	0.67	0.33
AD-MLP Zhai et al. (2023)	ArXiv 2023	0.15	0.32	0.59	0.35	**0.00**	0.27	0.85	0.37
BEV-Planner Li et al. (2024b)	CVPR 2024	0.16	0.32	0.57	0.35	**0.00**	0.29	0.73	0.34
SOLVE-E2E Chen et al. (2025)	CVPR 2025	0.14	0.28	0.50	0.31	0.04	0.17	0.68	0.30
ColaVLA Peng et al. (2026)	CVPR 2026	0.14	0.27	0.50	0.30	0.04	0.17	0.47	0.23
OneDrive	–	**0.13**	**0.25**	**0.46**	**0.28**	**0.00**	**0.12**	**0.43**	**0.18**

4.1 IMPLEMENT DETAILS

Datasets and Evaluation. We conduct experiments on two autonomous driving benchmarks: **nuScenes** and **NAVSIM**.

nuScenes Caesar et al. (2020b) contains 1,000 urban scenes of approximately 20 seconds, captured by six surround-view cameras and LiDAR, with 3D boxes for 10 categories and HD semantic maps. We use the standard 700/150/150 train/validation/test split, 3D detection and OpenLaneV2-derived lane annotations Wang et al. (2023a) for perception, and OmniDrive's perception, prediction, and planning QA annotations Wang et al. (2024b) for language and reasoning. Open-loop planning is evaluated using L2 displacement error for accuracy and collision rate for safety.

NAVSIM Dauner et al. (2025) builds on OpenScene Contributors (2023), a redistribution of nuPlan Caesar et al. (2021), with 1,192 navtrain and 136 navtest scenes. It emphasizes interactive, safety-critical planning more than nuScenes. Average displacement error, collision rate, and the official PDMS jointly assess safety, rule compliance, and driving efficiency.

Training. We use InternVL3-1B Zhu et al. (2025) on both benchmarks. On nuScenes, we follow the three-stage strategy in Sec. 3.4, training each stage for 20 epochs with an initial learning rate of 1×10^{-4} and a batch size of 64 on $64\times$ NVIDIA H20 GPUs. On NAVSIM, we also use InternVL3-1B, whose InternViT-300M encoder has the same architecture and size as that in ReCogDrive's InternVL3-2B backbone Li et al. (2025); we align the input images and visual preprocessing with ReCogDrive.

4.2 MAIN RESULTS

Open-loop evaluation in nuScenes. In Table 2, OneDrive achieves the lowest average L2 error (0.28 m) and collision rate (0.18%) among action-based planners, outperforming SOLVE-E2E (0.31 m; 0.30%) and ColaVLA (0.30 m; 0.23%), with 23% fewer collisions than ColaVLA. It remains competitive with text-based VLM planners without autoregressive trajectory generation or explicit chain-of-thought decoding. Planning uses only shallow LLM layers and standard causal attention compatible with FlashAttention Dao et al. (2022), keeping surround-view inference manageable (Table 7).

Table 3: Performance on NAVSIM navtest. OneDrive uses a ViT of the same size as ReCogDrive, retains six LLM layers, and uses no additional autonomous-driving data for pretraining. The OneDrive variants differ only in detection queries. SFT: supervised fine-tuning. † denotes LiDAR input.

Method	Reference	NC↑	DAC↑	TTC↑	Comf.↑	EP↑	PDMS↑
Ego Status MLP	-	93.0	77.3	83.6	100	62.8	65.6
VADv2 Chen et al. (2024b)	ArXiv 2024	97.2	89.1	91.6	100	76.0	65.6
DrivingGPT Chen et al. (2024c)	ICCV 2025	98.9	90.7	94.9	95.6	79.7	82.4
UniAD Hu et al. (2023b)	CVPR 2023	97.8	91.9	92.9	100	78.8	83.4
TransFuser† Chitta et al. (2022)	PAMI 2022	97.7	92.8	92.8	100	79.2	84.0
PARA-Drive Weng et al. (2024b)	CVPR 2024	97.9	92.4	93.0	99.8	79.3	84.0
DRAMA† Yuan et al. (2024)	ArXiv 2024	98.0	93.1	94.8	100	80.1	85.5
Hydra-MDP† Li et al. (2024a)	ArXiv 2024	98.3	96.0	94.6	100	78.7	86.5
DiffusionDrive† Liao et al. (2024)	CVPR 2025	98.2	96.2	94.7	100	82.2	88.1
Qwen2.5-VL-8B Bai et al. (2025)	-	97.8	92.1	92.8	100	78.3	83.3
InternVL3-8B Zhu et al. (2025)	-	97.0	92.1	91.8	100	78.9	83.3
AutoVLA(SFT) Zhou et al. (2025)	NeurIPS 2025	96.9	94.4	88.1	99.9	75.8	80.5
ReCogDrive Q-Dec.^{SFT} Li et al. (2025)	ICLR 2026	–	–	–	–	–	85.0
ReCogDrive DiT^{SFT} Li et al. (2025)	ICLR 2026	98.1	94.7	94.2	100	80.9	86.5
OneDrive w/o Det.	–	98.2	95.2	94.1	100	81.1	86.7
OneDrive w/ Det.	–	98.3	95.6	94.5	100	81.5	**87.3**

Closed-loop evaluation in NAVSIM. On navtest (Table 3), OneDrive uses the same ViT size as ReCogDrive Li et al. (2025), only six LLM layers, and no additional driving pretraining data. Without detection queries, it reaches 86.7 PDMS, exceeding ReCogDrive's SFT Query Decoder (85.0) and SFT DiT (86.5). Detection queries raise PDMS to 87.3, improving no-at-fault collisions (98.17→98.29), drivable-area compliance (95.20→95.55), ego progress (81.06→81.46), time-to-collision within bound (94.09→94.55), and driving-direction compliance (98.13→98.28), with unchanged comfort (99.97). These gains support the compact decoder and detection supervision for planning. Decoding from shallow mixed layers without repeated passes through the full decoder also reduces latency by approximately 40% (Table 7).

4.3 ABLATION STUDY

Text Evaluation. To verify that the proposed framework preserves the language generation capability of the underlying VLM, we compare it with OmniDrive Wang et al. (2024b) under the same training data setting. Specifically, we train our model using the same dataset as OmniDrive and evaluate the text-conditioned trajectory prediction accuracy. As shown in Table 4, our method achieves comparable or better L2 errors across the prediction horizons, yielding a lower average error (0.32 vs. 0.33). These results indicate that integrating the planning decoder and perception modules does not degrade the language capability of the model, while maintaining competitive performance in text-conditioned driving tasks.

Table 4: Comparison with OmniDrive under the same training data.

Method	**L2** $(m)\downarrow$			
	$1s$	$2s$	$3s$	Avg.
OmniDrive-7B	0.14	0.29	0.55	0.33
Ours-Text-1B	0.15	0.29	0.51	0.32

Disentangling the Role of Text Supervision. As shown in Table 5, we analyze the effect of retaining text supervision when using a unified causal attention to jointly model perception (3D object

Table 5: Impact of Text Supervision on Perception and Planning.

Task	Text	NDS/mAP	L2 (m) ↓				Col. Rate (%) ↓			
			$1s$	$2s$	$3s$	Avg.	$1s$	$2s$	$3s$	Avg.
3D Det		32.31/22.47	–	–	–	–	–	–	–	–
3D Det	✓	**33.94/24.39**	–	–	–	–	–	–	–	–
E2E		–	**0.13**	**0.27**	**0.51**	**0.31**	0.04	**0.19**	0.98	0.40
E2E	✓	–	**0.13**	**0.27**	**0.51**	**0.31**	**0.02**	0.20	**0.85**	**0.36**

detection) or planning. Enabling the text loss leads to small but consistent improvements across tasks. For 3D detection, the NDS/mAP increases slightly after introducing text supervision. For end-to-end planning, the L2 displacement error remains largely unchanged, while the collision rate is reduced (from 0.40 to 0.36), indicating improved driving safety. We attribute this behavior to the optimization characteristics of the pretrained causal attention. The causal attention module is originally pretrained under autoregressive text generation objectives, where its parameter space is shaped to capture sequential semantic dependencies. When adapting it to heterogeneous structured prediction tasks such as object detection and trajectory regression, the optimization process may gradually drift from the regime favored during pretraining. Maintaining a text loss during training helps preserve partial alignment with the original objective, which can provide a mild regularization effect on the shared attention module. This regularization stabilizes the unified causal attention to some extent, benefiting both perception and planning.

Table 6: Ablation on hyperparameter λ_{plan}.We vary the weight of the planning loss in the third stage (joint training) to study its effect on planning performance.

λ_{plan}	NDS/mAP	L2 (m) ↓				Col. Rate (%) ↓			
		$1s$	$2s$	$3s$	Avg.	$1s$	$2s$	$3s$	Avg.
0.25	44.82/35.10	0.127	0.258	0.473	0.287	**0.000**	0.078	0.449	**0.176**
0.5	45.45/35.58	**0.126**	0.255	0.467	0.283	**0.000**	**0.039**	0.488	**0.176**
1.0	**45.75/35.95**	**0.126**	**0.252**	**0.461**	**0.280**	**0.000**	0.117	**0.430**	0.182
2.0	44.72/34.01	0.127	0.258	0.472	0.286	**0.000**	0.098	0.762	0.287

Key Hyperparameter λ_{plan}. We conduct an ablation study on the planning loss weight λ_{plan}, as shown in Table 6. Increasing λ_{plan} from 0.25 to 1.0 consistently improves both perception and planning metrics, with the best overall performance achieved at $\lambda_{plan} = 1.0$, yielding an NDS/mAP of 45.75/35.95, the lowest average L2 error of 0.280 m, and a competitive collision rate. Further increasing λ_{plan} to 2.0 leads to a degradation in both NDS/mAP and collision rate, suggesting that over-emphasizing the planning objective can hurt model stability. These results indicate that a balanced weighting ($\lambda_{plan} = 1.0$) effectively aligns perception and planning tasks for optimal closed-loop performance.

Table 7: End-to-end latency (ms/frame) on one NVIDIA H20 GPU at the same batch size. Upper: NAVSIM; lower: nuScenes. †: without FlashAttention, using the ColaVLA implementation; OneDrive uses FlashAttention. Attention implementations are not matched.

Method	AR	Latency(ms)
ReCogDrive (SFT)		263
Ours w/o Det.		130
Ours w/ Det.		**157**
OmniDrive†	✓	3727
SOLVE-VLM†	✓	3719
ColaVLA†		727
Ours		513

Inference Latency. Table 7 reports end-to-end latency on a single NVIDIA H20 GPU with the same batch size, comparing OneDrive with ReCogDrive Li et al. (2025), OmniDrive Wang et al. (2024b), SOLVE-VLM Chen et al. (2025), and ColaVLA Peng et al. (2026). On NAVSIM, the complete model with detection queries takes 157 ms per frame, compared with 263 ms for ReCogDrive. Removing detection queries reduces latency to 130 ms; the complete 157 ms model is our primary configuration. On nuScenes, only measurements without FlashAttention are available for OmniDrive, SOLVE-VLM, and ColaVLA, whereas OneDrive uses FlashAttention. These timings therefore reflect different implementation settings and do not isolate architectural efficiency under matched attention kernels. OneDrive takes 513 ms per frame while feeding image tokens from all

surround-view cameras directly into the LLM decoder. Despite the resulting long token sequence, its shallow decoder and compatibility with standard attention acceleration keep inference latency at an acceptable level for this multi-camera VLM setting.

Table 8: Perception and Planning Performance at Each Stage. We report perception performance (NDS/mAP) and planning metrics at different training stages, including direct E2E adaptation, perception pretraining with planning adaptation, and final joint training.

Stage	**NDS/mAP**	**L2** $(m)\downarrow$				**Col. Rate (%)** $\downarrow$			
		$1s$	$2s$	$3s$	Avg.	$1s$	$2s$	$3s$	Avg.
Only Planning Adapation	–	0.14	0.29	0.54	0.32	0.02	0.23	1.01	0.42
Perception Pretrain	33.18/22.66	–	–	–	–	–	–	–	–
Planning Adaption	33.18/22.66	0.13	0.26	0.49	0.29	0.02	0.12	0.66	0.29
Joint Training	**45.75/35.95**	**0.13**	**0.25**	**0.46**	**0.28**	0.00	**0.12**	**0.43**	**0.18**

Multi-Stage Training. We conduct an ablation study on the proposed multi-stage training strategy, as summarized in Table 8. Directly adapting the model for end-to-end planning already produces reasonable planning performance. Introducing perception pretraining before planning adaptation further improves the planning metrics, reducing the average L2 error from 0.32 m to 0.29 m and the collision rate from 0.42% to 0.29%. Finally, performing joint training on both perception and planning tasks significantly improves overall performance, boosting the perception accuracy to 45.75/35.95 NDS/mAP while further reducing the planning error to 0.28 m and the collision rate to 0.18%. These results suggest that perception pretraining provides a strong initialization for planning, while joint optimization of perception and planning leads to the best overall performance. The staged procedure first adapts perception and language, then introduces planning while keeping perception-specific modules fixed. The final stage jointly updates all modules, including the visual encoder, allowing the perception and planning objectives to refine the shared representation together.

Table 9: Ablation on Token Sequence. We compare different ordering of perception tokens (lane and detection) before the E2E planning tokens, under both sequential adaptation and joint training.

Token Sequence	**L2** $(m)\downarrow$				**Col. Rate (%)** $\downarrow$			
	$1s$	$2s$	$3s$	Avg.	$1s$	$2s$	$3s$	Avg.
Lane→Det→Planning (Adaptation)	0.14	0.28	0.52	0.31	0.02	0.18	0.82	0.34
Lane→Det→Planning (Joint Training)	0.13	0.27	0.50	0.30	0.02	0.21	0.64	0.29
Det→Lane→Planning (Adaptation)	0.13	0.26	0.49	0.29	0.00	**0.12**	0.66	0.27
Det→Lane→Planning (Joint Training)	**0.13**	**0.25**	**0.46**	**0.28**	0.00	**0.12**	**0.43**	**0.18**

Task Sequences. Det→Lane→Planning outperforms Lane→Det→Planning in L2 error and collision rate (Table 9). Joint training further reduces average L2 from 0.31 to 0.28 m and collision rate from 0.34% to 0.18%, supporting detection-first ordering and end-to-end optimization for accurate, safe planning.

5 Conclusion

In this paper, we presented OneDrive, a unified end-to-end autonomous driving framework that reconciles heterogeneous decoding behaviors within a single pretrained VLM decoder. By retaining the original causal attention as a shared backbone and organizing visual and structured query tokens within a unified sequence, the framework enables stable joint optimization across perception and planning while preserving multi-modal generation capability. Extensive experiments on nuScenes and NAVSIM demonstrate state-of-the-art performance and improved inference efficiency. We hope that OneDrive can contribute to the advancement of both vision–language action modeling and end-to-end autonomous driving research.

AI USE STATEMENT

AI-based tools were used to support writing and language refinement, preparation of figures and LaTeX source files, and discussion of how the work could be presented more clearly. Their outputs were treated only as editorial and technical suggestions and were not used as independent scientific evidence. The authors made all research and methodological decisions, implemented the proposed approach, and verified the experimental results. The authors remain fully accountable for the accuracy, originality, and integrity of this manuscript.

REFERENCES

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

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 *CVPR*, 2020a.

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 *CVPR*, 2020b.

Holger Caesar, Juraj Kabzan, Kok Seang Tan, Whye Kit Fong, Eric Wolff, Alex Lang, Luke Fletcher, Oscar Beijbom, and Sammy Omari. nuplan: A closed-loop ml-based planning benchmark for autonomous vehicles. *arXiv preprint arXiv:2106.11810*, 2021.

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020.

Jiacheng Chen, Yuefan Wu, Jiaqi Tan, Hang Ma, and Yasutaka Furukawa. Maptracker: Tracking with strided memory fusion for consistent vector hd mapping. In *ECCV*, 2024a.

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*, 2024b.

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 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12068–12077, June 2025.

Yuntao Chen, Yuqi Wang, and Zhaoxiang Zhang. Drivinggpt: Unifying driving world modeling and planning with multi-modal autoregressive transformers. *arXiv preprint arXiv:2412.18607*, 2024c.

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 *CVPR*, 2024d.

Haohan Chi, Huan-ang Gao, Ziming Liu, Jianing Liu, Chenyu Liu, Jinwei Li, Kaisen Yang, Yangcheng Yu, Zeda Wang, Wenyi Li, et al. Impromptu vla: Open weights and open data for driving vision-language-action models. *arXiv preprint arXiv:2505.23757*, 2025.

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.

OpenScene Contributors. Openscene: The largest up-to-date 3d occupancy prediction benchmark in autonomous driving. `https://github.com/OpenDriveLab/OpenScene`, 2023.

Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in neural information processing systems*, 35:16344–16359, 2022.

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.

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, 2025.

Shangzhe Di, Zhonghua Zhai, and Weidi Xie. Revisiting multi-task visual representation learning. *arXiv preprint arXiv:2601.13886*, 2026.

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.

Jianhua Han, Meng Tian, Jiangtong Zhu, Fan He, Huixin Zhang, Sitong Guo, Dechang Zhu, Hao Tang, Pei Xu, Yuze Guo, et al. Percept-wam: Perception-enhanced world-awareness-action model for robust end-to-end autonomous driving. *arXiv preprint arXiv:2511.19221*, 2025.

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 *CVPR*, pp. 17853–17862, 2023a.

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 *CVPR*, 2023b.

Junjie Huang, Guan Huang, Zheng Zhu, and Dalong Du. Bevdet: High-performance multi-camera 3d object detection in bird-eye-view. *arXiv preprint arXiv:2112.11790*, 2021.

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.

Xiaosong Jia, Junqi You, Zhiyuan Zhang, and Junchi Yan. Drivetransformer: Unified transformer for scalable end-to-end autonomous driving. *arXiv preprint arXiv:2503.07656*, 2025.

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 *ICCV*, pp. 8340–8350, 2023a.

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 *ICCV*, 2023b.

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.

Chen-Yu Lee, Saining Xie, Patrick Gallagher, Zhengyou Zhang, and Zhuowen Tu. Deeply-supervised nets. In *Artificial intelligence and statistics*, pp. 562–570. Pmlr, 2015.

Tianyu Li, Li Chen, Huijie Wang, Yang Li, Jiazhi Yang, Xiangwei Geng, Shengyin Jiang, Yuting Wang, Hang Xu, Chunjing Xu, et al. Graph-based topology reasoning for driving scenes. *arXiv preprint arXiv:2304.05277*, 2023.

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.

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*, 2024a.

Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Yu Qiao, and Jifeng Dai. Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. In *ECCV*, 2022.

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 *CVPR*, pp. 14864–14873, 2024b.

Bencheng Liao, Shaoyu Chen, Xinggang Wang, Tianheng Cheng, Qian Zhang, Wenyu Liu, and Chang Huang. Maptr: Structured modeling and learning for online vectorized hd map construction. In *ICLR*, 2023a.

Bencheng Liao, Shaoyu Chen, Yunchi Zhang, Bo Jiang, Qian Zhang, Wenyu Liu, Chang Huang, and Xinggang Wang. Maptrv2: An end-to-end framework for online vectorized hd map construction. *arXiv preprint arXiv:2308.05736*, 2023b.

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. *arXiv preprint arXiv:2411.15139*, 2024.

Yingfei Liu, Tiancai Wang, Xiangyu Zhang, and Jian Sun. Petr: Position embedding transformation for multi-view 3d object detection. *arXiv preprint arXiv:2203.05625*, 2022.

Zhijian Liu, Haotian Tang, Alexander Amini, Xinyu Yang, Huizi Mao, Daniela L Rus, and Song Han. Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation. 2023.

Qihang Peng, Xuesong Chen, Chenye Yang, Shaoshuai Shi, and Hongsheng Li. Colavla: Leveraging cognitive latent reasoning for hierarchical parallel trajectory planning in autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17809–17819, June 2026.

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. *arXiv preprint arXiv:2312.14150*, 2023.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

Peize Sun, Yi Jiang, Enze Xie, Wenqi Shao, Zehuan Yuan, Changhu Wang, and Ping Luo. What makes for end-to-end object detection? In *International Conference on Machine Learning*, pp. 9934–9944. PMLR, 2021.

Xiaoyu Tian, Junru Gu, Bailin Li, Yicheng Liu, Chenxu Hu, Yang Wang, Kun Zhan, Peng Jia, Xianpeng Lang, and Hang Zhao. Drivevlm: The convergence of autonomous driving and large vision-language models. 2024a.

Xiaoyu Tian, Junru Gu, Bailin Li, Yicheng Liu, Chenxu Hu, Yang Wang, Kun Zhan, Peng Jia, Xianpeng Lang, and Hang Zhao. Drivevlm: The convergence of autonomous driving and large vision-language models. *arXiv preprint arXiv:2402.12289*, 2024b.

Huijie Wang, Tianyu Li, Yang Li, Li Chen, Chonghao Sima, Zhenbo Liu, Bangjun Wang, Peijin Jia, Yuting Wang, Shengyin Jiang, et al. Openlane-v2: A topology reasoning benchmark for unified 3d hd mapping. *Advances in Neural Information Processing Systems*, 36:18873–18884, 2023a.

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.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024a.

Shihao Wang, Yingfei Liu, Tiancai Wang, Ying Li, and Xiangyu Zhang. Exploring object-centric temporal modeling for efficient multi-view 3d object detection. In *ICCV*, 2023b.

Shihao Wang, Zhiding Yu, Xiaohui Jiang, Shiyi Lan, Min Shi, Nadine Chang, Jan Kautz, Ying Li, and Jose M Alvarez. Omnidrive: A holistic llm-agent framework for autonomous driving with 3d perception, reasoning and planning. *arXiv preprint arXiv:2405.01533*, 2024b.

Xinshuo Weng, Boris Ivanovic, Yan Wang, Yue Wang, and Marco Pavone. Para-drive: Parallelized architecture for real-time autonomous driving. In *CVPR*, 2024a.

Yuxuan Weng, Zhenyu Wu, Yifei Ren, Yifan Zhang, Yifan Xu, Yifan Liu, Yifan Wang, Yifan Chen, Yifan Li, and Yifan Zhao. Para-drive: Parallelized architecture for real-time autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024b.

Shuo Xing, Chengyuan Qian, Yuping Wang, Hongyuan Hua, Kexin Tian, Yang Zhou, and Zhengzhong Tu. Openemma: Open-source multimodal model for end-to-end autonomous driving. *arXiv preprint arXiv:2412.15208*, 2024.

Chenyu Yang, Yuntao Chen, Hao Tian, Chenxin Tao, Xizhou Zhu, Zhaoxiang Zhang, Gao Huang, Hongyang Li, Yu Qiao, Lewei Lu, et al. Bevformer v2: Adapting modern image backbones to bird's-eye-view recognition via perspective supervision. In *CVPR*, 2023.

Yixiang Yang, Chuanrong Han, Runhao Mao, Hanshi Wang, Zhiwen Chen, Yantai Yang, Qianli Ma, Xuesong Chen, Shaoshuai Shi, and Zhipeng Zhang. Survey of general end-to-end autonomous driving: A unified perspective. *Authorea Preprints*, 2025.

Chengran Yuan, Zhanqi Zhang, Jiawei Sun, Shuo Sun, Zefan Huang, Christina Dao Wen Lee, Dongen Li, Yuhang Han, Anthony Wong, Keng Peng Tee, et al. Drama: An efficient end-to-end motion planner for autonomous driving with mamba. *arXiv preprint arXiv:2408.03601*, 2024.

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*, 2023.

Yiwei Zhang, Jin Gao, Fudong Ge, Guan Luo, Bing Li, Zhao-Xiang Zhang, Haibin Ling, and Weiming Hu. Vq-map: Bird's-eye-view map layout estimation in tokenized discrete space via vector quantization. *Advances in Neural Information Processing Systems*, 37:70453–70475, 2024.

Yiwei Zhang, Jin Gao, Hanshi Wang, Fudong Ge, Guan Luo, Weiming Hu, and Zhipeng Zhang. Integrating diverse assignment strategies into detrs. *arXiv preprint arXiv:2601.09247*, 2026.

Wenzhao Zheng, Weiliang Chen, Yuanhui Huang, Borui Zhang, Yueqi Duan, and Jiwen Lu. Occworld: Learning a 3d occupancy world model for autonomous driving. In *ECCV*, 2024a.

Yupeng Zheng, Xiang Li, Pengfei Li, Yuhang Zheng, Bu Jin, Chengliang Zhong, Xiaoxiao Long, Hao Zhao, and Qichao Zhang. Monoocc: Digging into monocular semantic occupancy prediction. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 18398–18405. IEEE, 2024b.

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.

Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Yuchen Duan, Hao Tian, Weijie Su, Jie Shao, et al. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*, 2025.

Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020.

A QUALITATIVE RESULTS



Figure 4: Comparison of trajectories from different methods, with colors corresponding to trajectories: joint training, planning adaption, text output, and ground truth. It can be observed that the joint training version performs best, closely following the ground truth curvature during turns.

B FAILURE CASES



Figure 5: Two failure cases are shown. In the first case, the ground truth is preparing for a lane change, while the algorithm receives a straight-driving command and cannot anticipate the intended turn. The second case is similar: the current command is straight, so the algorithm is uncertain about the future trajectory.

C Additional Implementation Details

For the NuScenes Caesar et al. (2020a) benchmark, we use images from all six surrounding cameras. Each image is resized to 896×448 and represented using a thumbnail tokenization, resulting in 768 tokens per image. With six cameras, the total number of visual tokens per frame is $6 \times 768 = 4608$. For task-specific queries, we use 900 object queries and 900 lane queries, with 300 queries of each type allocated for temporal fusion following Wang et al. (2023b). In addition, we use $1 + 3 \times 6 \times 6 = 109$ planning queries, where the first query encodes the ego-status, the next dimension corresponds to 3 modalities, 6 time steps, and 6 anchor-initialized trajectories Jiang et al. (2023b). For NAVSIM Dauner et al. (2024), we use the same input images and visual preprocessing as ReCogDrive Li et al. (2025) so that the visual inputs are aligned for comparison.

D Additional Experiments

D.1 Lane Evaluation

We evaluate our method on the OpenLane-V2 Wang et al. (2023a) benchmark using the DET_l metric. As shown in Table 10, our approach with perception pretraining achieves competitive performance (19.9), surpassing MapTR Liao et al. (2023a) but falling short of TopoNet Li et al. (2023). When we apply joint training, which optimizes both perception and planning objectives simultaneously, the performance improves substantially to 32.2. These results demonstrate that our framework is capable of completing lane detection effectively, and empirically validate the utility of joint training in complex driving scenarios.

Table 10: Performance comparison on OpenLane-V2 in terms of DET_l. Joint training significantly improves lane detection accuracy over perception pretraining.

Method	DET_l
MapTR Liao et al. (2023a)	8.3
TopoNet Li et al. (2023)	24.3
Ours(Perception Pretrain)	19.9
Ours(Joint Training)	**32.2**

D.2 Component Analysis of the Mixed Decoder

The mixed decoder adds structured-query self-attention and task-specific FFNs to pretrained causal attention. We ablate both components. To avoid confounding parameter sharing with optimization, the controlled FFN comparison makes the text FFN trainable and compares a shared text/detection FFN with a task-specific detection FFN.

Table 11 shows that structured-query self-attention improves NDS by 4.79 and mAP by 4.73. The task-specific FFN also consistently outperforms the shared FFN. Thus, pretrained attention transfers across tasks, while structured outputs benefit from bidirectional query interaction and task-specific transformations.

Table 11: Component analysis of the mixed decoder. † denotes the controlled setting in which the text FFN is trainable.

Variant	NDS↑	mAP↑
w/o query self-attn	29.63	19.59
w/ query self-attn	**34.42**	**24.32**
Shared text/det FFN†	37.23	26.30
Task-specific det FFN†	**37.49**	**26.77**

D.3 Number of Mixed Decoder Layers

We study how the number of mixed decoder layers affects perception, planning, and latency. Table 12 shows that increasing the depth from three to six layers improves both perception and planning. Nine layers further improve NDS and mAP, but slightly degrade trajectory error and collision rate while increasing latency. Six layers therefore provide the best overall trade-off for planning, achieving the lowest L2 error and collision rate with moderate inference cost.

Table 12: Effect of the number of mixed decoder layers. Latency is measured in milliseconds.

Layers	NDS↑	mAP↑	L2↓	Col.↓	Latency↓
3	43.40	33.07	0.284	0.20	**485**
6	45.75	35.95	**0.280**	**0.18**	513
9	**46.45**	**36.40**	0.282	0.27	551

D.4 Detection Query Ablation

Both NAVSIM variants use a ViT of the same size as ReCogDrive, retain six LLM layers, and use no additional autonomous-driving data for pretraining. Without detection queries, OneDrive obtains 86.7 PDMS, exceeding the ReCogDrive SFT Query Decoder baseline (85.0) and SFT DiT planner (86.5). Adding detection queries raises PDMS to 87.3 and improves no-at-fault collisions (98.17 to 98.29), drivable-area compliance (95.20 to 95.55), ego progress (81.06 to 81.46), time-to-collision within bound (94.09 to 94.55), and driving-direction compliance (98.13 to 98.28); comfort remains 99.97.

D.5 3D Detection with the InternVL3-ViT Backbone

Following StreamPETR, we analyze the 3D detection capability of the InternVL3 visual backbone. We re-implement StreamPETR using InternVL3-ViT and align the decoder dimension for a fair comparison.

As shown in Table 13, under the same InternVL3-ViT backbone, our method consistently outperforms the StreamPETR baseline in both frozen and fully fine-tuned settings (33.94 vs. 31.48 NDS with the frozen backbone and 47.73 vs. 45.31 NDS with full fine-tuning). However, the overall detection performance with InternVL3-ViT remains lower than the original StreamPETR baseline reported with standard ViT-Large backbones Wang et al. (2023b). We attribute this gap to two factors. First, VLM pipelines downsample image features before the decoder, reducing the effective token resolution. Second, the language-centric pretraining objective of InternVL3 does not explicitly optimize detection-oriented visual representations Han et al. (2025); Di et al. (2026). While Percept-WAM Han et al. (2025) achieves strong detection performance, it also uses additional point-cloud inputs.

Table 13: 3D detection performance with the InternVL3-ViT backbone on nuScenes.

Method	NDS ↑	mAP ↑
StreamPETR (frozen)	31.48	20.26
Ours (frozen)	33.94	24.39
StreamPETR	45.31	36.11
Ours	47.73	40.03

E Limitations and Future Work

Our evaluation is limited to nuScenes and NAVSIM, and the scaling behavior of OneDrive with larger models and datasets remains unverified. Evaluating larger architectures on substantially larger datasets will help determine whether the current improvements generalize to more demanding settings and clarify the remaining limitations in efficiency and robustness.

Perception performance is constrained by the VLM visual backbone and its reduced token resolution, as discussed in Sec. D.5. Future work could incorporate detection-aware objectives during multimodal pretraining to develop more object-centric visual representations. Higher-resolution or multi-scale tokenization may also reduce the loss of spatial detail caused by feature downsampling, improving detection and other perception tasks.

Finally, the failure cases in Fig. 5 show that ambiguous or mismatched driving commands can limit trajectory prediction. Further evaluation under such conditions is needed to assess generalization and robustness.