GeoVLDrive: Geometry-Grounded Vision-Language Reasoning for Autonomous Driving

Anonymous authors
Paper under double-blind review

Abstract

Autonomous driving involves continuous interaction between the ego vehicle and surrounding agents in a dynamic 3D environment, requiring geometrically grounded reasoning about current spatial relationships and future scene evolution for safe and efficient planning. Despite advances in geometric grounding and future prediction, reasoning about the implications of evolving 3D spatial relationships for ego planning remains a fundamental challenge. We introduce **GeoVLDrive**, a geometry-grounded vision-language-action (VLA) framework for 3D spatiotemporal reasoning and planning. Multi-layer geometric features from a 3D foundation model complement the semantic representations of a vision-language backbone. A causal latent chain of thought (CoT) reasons from future scene geometry through key-agent motion to meta action. An action expert grounded in geometric reasoning jointly generates and scores trajectory proposals conditioned on these latent states and multi-depth scene features. We further introduce **NAVSIM 3DQA** to support training and evaluation of metric 3D spatial understanding and agent motion reasoning. GeoVLDrive outperforms the compared supervised VLA baselines on NAVSIM v1 with 91.8 PDMS, while attaining 90.9 EPDMS on NAVSIM v2. On NAVSIM 3DQA, it achieves an aggregate score of 89.2, outperforming a matched fine-tuned baseline by 4.6 points. It also achieves 68.6 on LingoQA.

1 Introduction

In recent years, end-to-end autonomous driving has achieved significant progress through joint optimization of perception, prediction, and planning (Hu et al., 2022; 2023; Jiang et al., 2023; Chen et al., 2024; Chitta et al., 2023). Building on the extensive world knowledge and strong reasoning capabilities of pretrained vision-language models, vision-language-action (VLA) frameworks further generate driving trajectories directly from multi-view observations and language instructions (Fu et al., 2025; Hwang et al., 2024; Tian et al., 2024; Zhou et al., 2026; 2025). Applying these capabilities to driving requires reasoning about traffic participants, road geometry, and future scene evolution (Hu et al., 2023; Chen et al., 2024; Chitta et al., 2023; Dauner et al., 2024). This reasoning must be grounded in 3D geometry and scene semantics, linking current spatial relationships and their future evolution to ego planning.

Representative approaches differ in how they incorporate semantic reasoning and geometric representations into trajectory planning (Fig. 1). VLA policies (Fig. 1(a)) build on pretrained vision-language models for scene understanding and reasoning to guide trajectory generation (Fu et al., 2025; Hwang et al., 2024; Zhou et al., 2026). Geometry world action models (Fig. 1(b)) model current scene geometry and predict its future evolution, conditioning planning on geometric representations of the observed and anticipated scene (Zhang et al., 2026). Geometry-grounded VLAs (Fig. 1(c)) use 3D geometric features or reconstruction supervision to support spatial understanding and trajectory planning (Wang et al., 2026; Yao et al., 2026a).

While these methods have achieved impressive results, they still suffer from key limitations (Fig. 1): (1) *Lack of geometric grounding:* VLA policies reason in language over 2D observations, lacking cross-view geometric modeling (Wang et al., 2026) and the continuous spatial precision that safe driving demands (Yao et al., 2026a). (2) *Lack of semantic reasoning over the future:* geometry world action models connect predicted geometry to action directly, never stating which agents matter or

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.

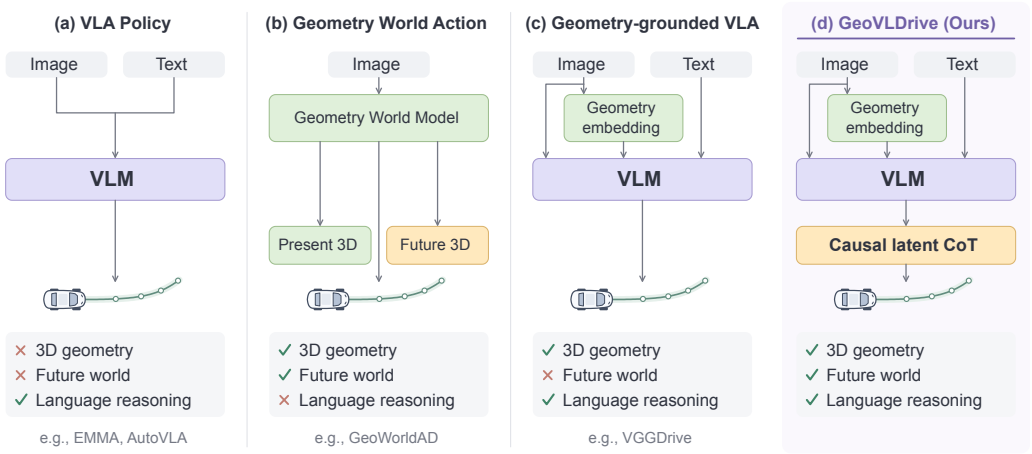

Figure 1: **Existing driving policy paradigms vs. our GeoVLDrive.** (a) VLA policies reason in language over images, without metric 3D structure or a forecast of scene evolution (Hwang et al., 2024; Zhou et al., 2025). (b) Geometry world action models predict present and future 3D and map them to a trajectory directly, with no semantic account of why the action follows (Zhang et al., 2026). (c) Geometry-grounded VLAs ground language in 3D foundation features, but reason only over the current frame (Wang et al., 2026). (d) GeoVLDrive grounds a vision-language model in metric 3D structure and reasons over the predicted future through a *causal latent chain of thought* that conditions trajectory generation on spatio-temporal reasoning.

why their predicted change warrants a maneuver; moreover, the forecast covers the entire scene, whereas planning depends on the few key agents that influence ego motion. (3) *Reasoning confined to the present:* geometry-grounded VLAs ground language in current-frame 3D structure, but do not reason over how that structure will evolve.

To address these limitations, we propose **GeoVLDrive**, a geometry-grounded vision-language-action framework for 3D spatiotemporal reasoning and planning (Fig. 1(d)). Multi-layer geometric features augment the vision-language backbone with 3D spatial information. A causal latent chain-of-thought (CoT) reasons from future scene geometry through key-agent motion to meta action, grounding ego planning in the predicted evolution of 3D spatial relationships. A structured attention mask encodes the reasoning order and temporal dependencies of the chain. An action expert jointly generates and scores trajectory proposals conditioned on the latent CoT and multi-depth geometric and semantic features. Dedicated score tokens incorporate scene and reasoning context and exchange information across proposals, allowing each candidate to be evaluated in relation to the alternatives. We further introduce **NAVSIM 3DQA** to support training and evaluation of reasoning about metric 3D structure and agent motion.

Our main contributions are summarized as follows:

- We propose **GeoVLDrive**, a geometry-grounded VLA framework that connects future scene geometry, key-agent motion, and meta action through a supervised causal latent CoT, providing structured 3D spatiotemporal reasoning for ego planning.
- We develop an **action expert grounded in geometric reasoning** that jointly refines latent reasoning, action, and score states under causal attention. Dedicated score tokens incorporate scene context and exchange information across proposals to support trajectory evaluation.
- We introduce **NAVSIM 3DQA**, a benchmark covering object categories, metric positions, dimensions, and agent motion. With a 2B backbone and a single front camera, GeoVLDrive achieves 91.8 PDMS on NAVSIM v1 and 90.9 EPDMS on NAVSIM v2 (Dauner et al., 2024; Cao et al., 2025), alongside 89.2 on NAVSIM 3DQA and 68.6 on LingoQA.

2 Related Work

2.1 Vision-Language-Action Models for Driving

Vision-language-action models combine pretrained semantic knowledge with trajectory planning. DriveLM (Sima et al., 2024) formulates driving understanding as graph-structured question answering, while EMMA (Hwang et al., 2024), AutoVLA (Zhou et al., 2025), and AdaThinkDrive (Luo et al., 2025) incorporate textual reasoning into action generation. SGDrive (Li et al., 2026) further organizes reasoning through a scene-to-goal hierarchy. Although textual chains (Wei et al., 2022) expose intermediate reasoning, autoregressive decoding introduces inference overhead. Geometry-grounded VLAs complement semantic representations with 3D spatial information. VG-GDrive (Wang et al., 2026) injects features from a frozen geometry backbone through multi-layer cross-attention, while VLGA (Yao et al., 2026a) introduces a geometry expert supervised by dense pointmap reconstruction. Related vision-geometry-action models, such as DVGT-2 (Zuo et al., 2026), jointly reconstruct current geometry and predict trajectories. GeoVLDrive builds on geometric grounding to support structured reasoning about future scene evolution and its implications for ego planning.

2.2 Latent Reasoning and World Modeling

Latent CoT methods replace explicit reasoning traces with continuous intermediate states. Coconut (Hao et al., 2025) recurrently propagates hidden states, CCoT (Cheng & Durme, 2024) learns compressed traces, and CODI (Shen et al., 2025) uses hidden-state self-distillation. SIM-CoT (Wei et al., 2026) adds step-level supervision through an auxiliary decoder.

Driving approaches transfer geometric and dynamic priors into latent representations, as in LaST-VLA (Luo et al., 2026) and CoWorld-VLA (Huang et al., 2026), while PerceptDrive (Liu et al., 2026a) routes distilled perception priors into world-action modeling. OneVL (Lu et al., 2026) supervises shared latents through language and future-frame decoders. DynVLA (Shang et al., 2026) predicts compact dynamics tokens, LCDrive (Tan et al., 2026) interleaves action proposals with predicted outcomes, and ColaVLA (Peng et al., 2026) derives meta-action representations for parallel planning. Related approaches support planning through video prediction (Gao et al., 2024; Wang et al., 2024; Chen et al., 2025), latent forecasting (Li et al., 2025b), occupancy and flow modeling (Zheng et al., 2024; Liu et al., 2026b), and future geometry prediction (Zhang et al., 2026). GeoVLDrive organizes future geometry, key-agent motion, and meta action into a supervised causal latent CoT. Future depth, agent tracks, and maneuver labels supervise its intermediate states, while a structured attention mask encodes their directed dependencies. These states directly condition trajectory generation and scoring.

3 Method

GeoVLDrive grounds a vision-language backbone in 3D geometry and introduces a causal latent CoT connecting future scene geometry, key-agent motion, and meta action (Fig. 2). An action expert jointly generates and scores trajectory proposals conditioned on the latent CoT and multi-depth geometric and semantic features.

3.1 Geometry-Grounded Vision-Language Backbone

We combine a pretrained Qwen3-VL-2B-Instruct backbone (Bai et al., 2025) with a frozen StreamVGGT geometry encoder (Zhuo et al., 2025; Zhang et al., 2026). The VLM encodes the visual input, driving prompt, and current ego state, while StreamVGGT aggregates four front-camera frames to extract current-frame geometric features $\{G^\ell\}_\ell$ at multiple depths.

Layer-wise gated cross-attention for geometry injection. We inject geometry features at multiple VLM depths (Wang et al., 2026), pairing StreamVGGT decoder layers $[4, 11, 17, 23]$ with Qwen3-VL decoder layers $[6, 13, 20, 26]$ (Fig. 2(b)). For each paired depth, the decoder hidden states query the projected geometry tokens through a gated cross-attention residual (Alayrac et al.,

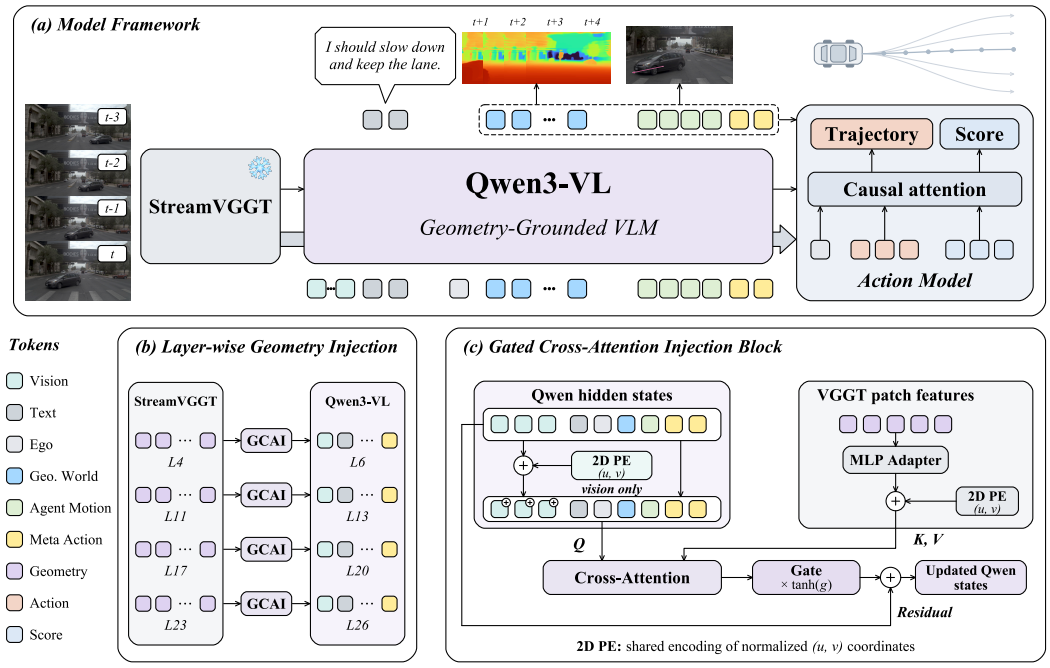

Figure 2: **Overview of GeoVLDrive. (a) Model framework.** The geometry-grounded VLM encodes future geometry, key-agent motion, and driving intent in a causal latent chain that conditions the action expert for trajectory planning. **(b) Layer-wise geometry injection.** Features from multiple VGGT layers are injected at selected VLM decoder depths. **(c) Gated cross-attention injection block.** Gated cross-attention injects geometry into VLM states, with shared 2D positional encoding for spatial alignment.

2022):

$$X'_m = X_m + \tanh(g_m)\ \mathrm{Attn}\big(\mathrm{LN}(X_m),\ \phi_m(G^\ell),\ \phi_m(G^\ell)\big)\,, \tag{1}$$

where ϕ_m projects the geometry features to the decoder width and g_m is a learnable scalar gate that controls the amount of geometric information added at each depth. Because the queries are formed from the decoder hidden states, visual, textual, and latent reasoning tokens can all access the geometry stream.

Spatial alignment of semantic and geometric features. To spatially align the VLM's visual representations with geometry features, we encode normalized image coordinates (u, v) using a shared 2D sinusoidal positional encoding. The resulting encodings are added to the visual queries and the geometry keys and values (Fig. 2(c)). This provides a common image-plane reference across feature grids of different resolutions, helping cross-attention associate features at corresponding locations. Together, geometry injection and spatial encoding improve the backbone's 3D scene understanding and metric localization, as shown by the NAVSIM 3DQA ablations in Table 5.

3.2 Causal Latent Chain-of-Thought

We formulate 3D spatiotemporal reasoning as a causal latent chain connecting future scene geometry, key-agent motion, and meta action. We predict dense scene geometry over $2\,\mathrm{s}$ and extend key-agent motion prediction to $4\,\mathrm{s}$, providing detailed geometric context while focusing longer-horizon prediction on agent dynamics. Both branches model future changes relative to the current state through residual representations. At inference, their hidden states are computed in a single backbone forward pass and directly condition the action expert (Sec. 3.3).

Future geometric world. We retain the current multi-layer geometric features $\{G^\ell\}_\ell$ as a dense spatial reference, allowing compact latent tokens to focus on temporal changes. For each prediction

horizon $t_k = k\delta$, $k = 1, \ldots, K$, a time-conditioned residual decoder updates these features using 32 latent states $\mathbf{F}_k$:

$$\Delta G_k^\ell = \mathcal{R}(G^\ell, \mathbf{F}_k, t_k), \qquad (\hat{\mathbf{z}}_k, \mathbf{c}_k) = \mathcal{D}_{\text{geo}}\left(\{G^\ell + \Delta G_k^\ell\}_\ell\right), \tag{2}$$

where $\mathcal{R}$ denotes the decoder's net feature update and $\mathcal{D}_{\text{geo}}$ produces the future depth and confidence maps $\hat{\mathbf{z}}_k$ and $\mathbf{c}_k$. Valid pixels in recorded future-depth observations provide supervision adapted from VGGT and GeoWorldAD (Wang et al., 2025a; Zhang et al., 2026):

$$\mathcal{L}_{\text{world}} = \sum_{k=1}^{K} \left(\mathcal{L}_{\text{depth}}^{(k)} + \mathcal{L}_{\text{conf}}^{(k)} + \mathcal{L}_{\text{grad}}^{(k)}\right), \tag{3}$$

combining depth regression, confidence-weighted regression, and multi-scale gradient consistency.

Key-agent motion. We represent a planning-relevant agent with latent states $\mathbf{D} = [\mathbf{d}_c; \mathbf{d}_0; \mathbf{d}_{1:J}]$, encoding its category, current position, and future position offsets. Category and current-position states attend to the observed context, while future-motion states additionally attend to future-geometry states. Prediction extends to $J > K$ steps on the same temporal grid, $t_j = j\delta$, with future positions reconstructed relative to the current position:

$$\hat{\mathbf{p}}_0 = g_0(\mathbf{d}_0), \qquad \hat{\mathbf{p}}_j = \hat{\mathbf{p}}_0 + g_\Delta(\mathbf{d}_j), \quad j = 1, \ldots, J, \tag{4}$$

where g_0 estimates the current position and g_Δ predicts offsets from it. Category and track annotations supervise the predictions, including a "none" category for scenes without a key agent:

$$\mathcal{L}_{\text{agent}} = \lambda_{\text{cls}}\mathcal{L}_{\text{CE}} + \lambda_{\text{pos}}\mathcal{L}_{\text{SmoothL1}} + \lambda_{\text{dir}}\mathcal{L}_{\cos}. \tag{5}$$

Smooth L1 regression averages losses on the current position, future offsets, and reconstructed positions. The cosine loss aligns consecutive-step displacement directions, with reduced weights for small ground-truth displacements. Position and direction losses apply only to valid agent tracks.

Meta action. Two meta-action tokens integrate the observed context, future-geometry states, and key-agent motion states to represent longitudinal and lateral driving intent. Their maneuver distributions $\boldsymbol{\pi}_d$ are supervised by the corresponding labels y_d:

$$\mathcal{L}_{\text{meta}} = \sum_{d \in \{\text{lon}, \text{lat}\}} \text{CE}(\boldsymbol{\pi}_d, y_d). \tag{6}$$

Structured causal masking. We impose a structured attention mask to encode the reasoning order and temporal dependencies of the latent chain (Fig. 3). Future-geometry states condition key-agent motion reasoning, while meta-action tokens integrate both geometry and motion states; reverse cross-group connections are masked. Geometry tokens attend to their own and earlier prediction horizons. Future-motion tokens interact bidirectionally, with direct attention from motion queries at t_j to geometry states at t_k restricted to $t_k \le t_j$.

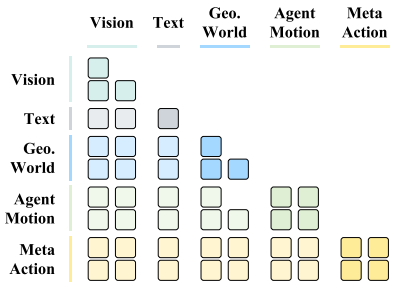

Figure 3: **Causal latent CoT mask.**

3.3 Geometry-Reasoning-Grounded Action Expert

Our action expert jointly generates and scores trajectory proposals conditioned on the latent chain and multi-depth geometric and semantic features (Fig. 4). Causal joint attention directs information from reasoning to actions and from actions to scores. Dedicated score tokens integrate scene context and proposal-specific action states, while exchanging information across candidates to support trajectory selection.

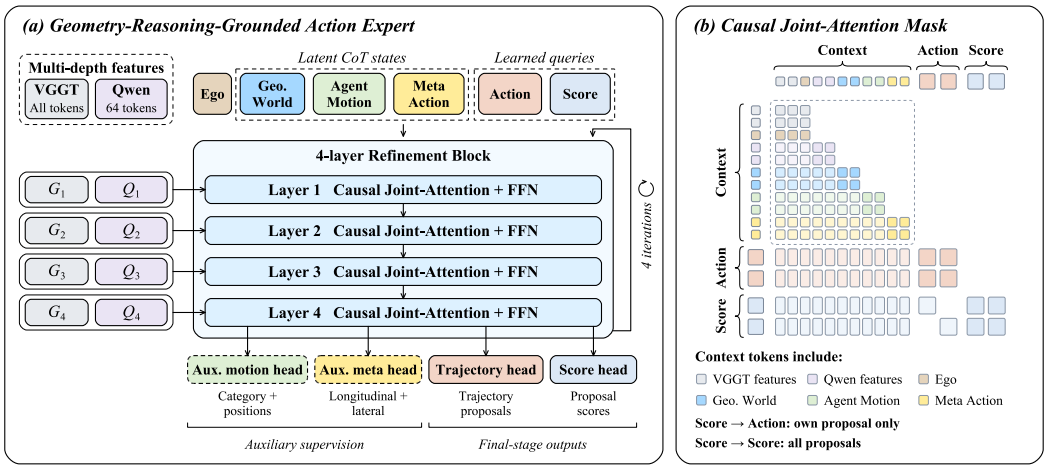

Figure 4: **Geometry-reasoning-grounded action expert.** (a) Overall architecture for joint trajectory generation and scoring, integrating multi-depth geometric (G_i) and semantic (Q_i) features with the latent reasoning chain. (b) Attention mask used in the causal joint-attention layers of (a).

Multi-depth causal joint attention. The expert is a dedicated transformer operating on latent reasoning states, ego-state tokens, $R \times J$ action tokens, and R score tokens. Geometric and semantic features from multiple depths are concatenated with these states at corresponding layers. Qwen features are compressed to 64 tokens at each selected depth. All VLM-derived inputs are detached before projection, preventing planning gradients from updating the VLM. Masked self-attention and an FFN jointly update the resulting sequence under the group-causal mask in Fig. 4(b). The refined reasoning, action, and score states propagate across S refinement stages, with stage s decoding R trajectory proposals $\boldsymbol{\tau}_r^{(s)}$, each comprising J poses. Trajectory supervision follows the winner-takes-all objective used by Guo et al. (2025a), accumulated across stages:

$$\mathcal{L}_{\text{traj}} = \sum_{s=1}^{S} \gamma^{S-s} \min_{r=1,\dots,R} \frac{1}{J} \sum_{j=1}^{J} \left\| \boldsymbol{\tau}_{r,j}^{(s)} - \hat{\boldsymbol{\tau}}_j \right\|_1, \tag{7}$$

where $\hat{\boldsymbol{\tau}}$ denotes the logged trajectory and γ discounts earlier stages. The minimum selects the closest proposal at each stage for trajectory regression. Auxiliary heads apply the agent and meta-action objectives to the expert's final motion and meta-action states, yielding $\mathcal{L}_{\text{agent}}^{\text{aux}}$ and $\mathcal{L}_{\text{meta}}^{\text{aux}}$, respectively.

Scene-grounded cross-proposal scoring. Dedicated score tokens jointly evaluate trajectory proposals using shared scene and reasoning context. Direct score-to-action attention is restricted to the corresponding proposal, while attention among score tokens enables information exchange across candidates (Fig. 4(b)). An MLP maps the final score states to scalar scores, supervised by binary cross-entropy $\mathcal{L}_{\text{score}}$ against simulated trajectory scores (Dauner et al., 2024). At inference, the highest-scoring proposal is selected.

3.4 Training Objectives

We combine latent-chain and textual CoT supervision with trajectory regression and proposal scoring. Auxiliary losses supervise the action expert's refined motion and meta-action states. The overall objective is

$$\mathcal{L} = \mathcal{L}_{\text{traj}} + \sum_{q \in \mathcal{T}} \lambda_q \mathcal{L}_q + \sum_{q \in \{\text{agent},\text{meta}\}} \lambda_q^{\text{aux}} \mathcal{L}_q^{\text{aux}}, \tag{8}$$

where $\mathcal{T} = \{\text{world}, \text{agent}, \text{meta}, \text{cot}, \text{score}\}$, and the λ coefficients balance the individual objectives. VLM representations are detached at the expert input, preventing expert losses from updating the VLM.

Table 1: **Comparison with state-of-the-art methods on NAVSIM v1 (PDMS).** C and L denote camera and LiDAR; numbers indicate camera counts. Higher scores are better; boldface marks the best displayed values, including ties. Metric abbreviations are defined in Table 11.

Method	Backbone / family	Sensors	NC↑	DAC↑	TTC↑	Comf.↑	EP↑	PDMS↑
VADv2 (Chen et al., 2024)	E2E	1C	97.2	89.1	91.6	**100.0**	76.0	80.9
UniAD (Hu et al., 2023)		6C	97.8	91.9	92.9	**100.0**	78.8	83.4
Hydra-MDP (Li et al., 2024)		3C+L	98.3	96.0	94.6	**100.0**	78.7	86.5
DiffusionDrive (Liao et al., 2025)		3C+L	98.2	96.2	94.7	**100.0**	82.2	88.1
FSDrive (Zeng et al., 2025)	World model	C	98.2	93.8	93.3	99.9	80.1	85.1
PWM (Zhao et al., 2025)		C	98.6	95.9	95.4	**100.0**	81.8	88.1
WoTE (Li et al., 2025d)		3C+L	98.5	96.8	94.9	99.9	81.9	88.3
WorldDrive (Gui et al., 2026)		C	98.4	96.8	95.2	**100.0**	83.3	89.0
DriveVLA-W0 (Li et al., 2025c)	Qwen2.5-VL-7B	1C	98.4	95.3	95.4	**100.0**	80.9	87.2
LaST-VLA (Luo et al., 2026)	InternVL3-8B	1C	98.7	95.4	95.6	**100.0**	80.5	87.3
SGDrive (Li et al., 2026)	InternVL3-2B	1C	98.6	95.1	95.4	**100.0**	81.2	87.4
VGGDrive (Wang et al., 2026)	Qwen2.5-VL-7B	1C	98.6	96.3	95.6	**100.0**	82.9	88.8
CoWorld-VLA (Huang et al., 2026)	Qwen3-VL-2B	1C	**99.1**	97.0	**96.5**	**100.0**	84.0	90.0
GeoVLDrive	Qwen3-VL-2B	1C	98.7	**98.0**	95.3	**100.0**	**88.1**	**91.8**

4 Experiments

4.1 Implementation Details

Datasets. We use LingoQA (Marcu et al., 2024) and NAVSIM 3DQA for visual question answering. NAVSIM-WM-CoT provides chain-of-thought annotations and supervision for future geometry and key-agent motion. For planning, we train on *navtrain* and evaluate on the 12,146 scenes in *navtest* (Dauner et al., 2024; Cao et al., 2025). At inference, the planner uses front-camera images and ego-state inputs, without LiDAR.

NAVSIM 3DQA. We construct NAVSIM 3DQA to supervise and evaluate spatiotemporal 3D reasoning. Each sample comprises four consecutive front-camera frames and a question about an object identified by a normalized 2D bounding box. Questions cover four categories: coarse object category, ground-plane position in metres, vehicle dimensions, and temporal motion. The dataset contains approximately 1.28M QA pairs from 95K anchor frames, including 1.12M training pairs and 165K validation pairs. The training and validation sets follow NAVSIM's official partition. Dataset construction is detailed in Appendix A.1.

Metrics. We report the Predictive Driver Model Score (PDMS) on NAVSIM v1 and the Extended PDMS (EPDMS) on NAVSIM v2 for planning evaluation. Both aggregate safety, comfort, and progress; v2 includes additional evaluation criteria. NAVSIM 3DQA scores four per-task rates: coarse category and motion behavior by semantic match, and ground-plane position and vehicle dimensions by whether the regressed metres fall within the benchmark's tolerance. The reported Overall score is their equal-weight mean. On LingoQA we report Lingo-Judge accuracy.

Training details. We use Qwen3-VL-2B-Instruct (Bai et al., 2025) as the language backbone and a frozen StreamVGGT (Zhuo et al., 2025) as the geometry encoder, trained across four phases. In the first phase we align semantics and geometry on the question-answering mixture for 3 epochs. In the second we supervise the prediction of the future, that is future geometry and key-agent dynamics, for 5 epochs. In the third we train the causal latent chain of thought for 10 epochs. Finally we train the chain and the action expert jointly for 30 epochs. Training uses 64 NVIDIA H20 GPUs. Additional details and hyperparameters are provided in Appendix A.2.

4.2 Evaluation Results

NAVSIM benchmark. Using a 2B VLM backbone and a single front camera, GeoVLDrive achieves 91.8 PDMS on NAVSIM v1 and 90.9 EPDMS on NAVSIM v2 under supervised training without reinforcement learning (Tables 1 and 2). VLM baselines in Table 1 also use SFT. On NAVSIM v1, it exceeds the SFT results of SGDrive (Li et al., 2026) and CoWorld-VLA (Huang et al., 2026) by 4.4 and 1.8 PDMS points, respectively; both baselines also use 2B VLM backbones. GeoVLDrive attains 88.1 ego progress and 98.0 drivable-area compliance on v1. Under the extended v2 metrics, ego progress and lane keeping reach 91.4 and 97.6, respectively.

Table 2: **Comparison with state-of-the-art methods on NAVSIM v2 (EPDMS).**

Method	NC↑	DAC↑	DDC↑	TLC↑	EP↑	TTC↑	LK↑	HC↑	EC↑	EPDMS↑
Ego Status (Cao et al., 2025)	93.1	77.9	92.7	99.6	86.0	91.5	89.4	**98.3**	85.4	64.0
TransFuser (Chitta et al., 2023)	96.9	89.9	97.8	99.7	87.1	95.4	92.7	**98.3**	87.2	76.7
HydraMDP++ (Li et al., 2025a)	97.2	97.5	99.4	99.6	83.1	96.5	94.4	98.2	70.9	81.4
DriveSuprim (Yao et al., 2026b)	97.5	96.5	99.4	99.6	88.4	96.6	95.5	**98.3**	77.0	83.1
ARTEMIS (Feng et al., 2025)	98.3	95.1	98.6	**99.8**	81.5	97.4	96.5	**98.3**	–	83.1
DiffusionDrive (Liao et al., 2025)	98.2	95.9	99.4	**99.8**	87.5	97.3	96.8	**98.3**	**87.7**	84.5
GeoWorldAD (Zhang et al., 2026)	**99.0**	97.8	**99.6**	99.7	89.1	**98.6**	**97.6**	98.0	82.2	90.4
GeoVLDrive	98.7	**98.0**	99.5	99.6	**91.4**	98.2	**97.6**	97.9	82.4	**90.9**

LingoQA and NAVSIM 3DQA benchmarks. Compared with Qwen3-VL-2B-Instruct fine-tuned on the same data and schedule, GeoVLDrive improves Lingo-Judge accuracy on LingoQA from 67.0 to 68.6 (Table 3). On NAVSIM 3DQA, the overall score increases from 84.6 to 89.2, with the largest gains in position accuracy within 1 m (66.8 → 76.7) and motion accuracy (85.8 → 90.9).

Table 3: **Results on LingoQA and NAVSIM 3DQA.** Overall averages the four benchmark-defined task scores (Appendix A.1.3); the displayed Position and Dimensions metrics are reported separately. The upper block is evaluated without task-specific fine-tuning; $*$ denotes our fine-tuned baseline. Baselines use Qwen3-VL (Bai et al., 2025) and InternVL3.5 (Wang et al., 2025b).

	LingoQA (%)	**NAVSIM 3DQA (%)**				
Method	Lingo-Judge↑	Overall↑	Category↑	Motion↑	Position↑ $L_2 \leq 1$ m	Dimensions↑ All ≤ 0.5 m
Qwen3-VL-2B-Instruct	44.0	32.2	80.3	38.5	6.1	10.1
Qwen3-VL-8B-Instruct	54.8	54.4	82.3	50.7	9.4	68.7
InternVL3.5-2B	53.8	29.9	66.9	27.3	5.0	8.4
InternVL3.5-8B	51.4	45.1	67.0	33.5	7.4	62.5
Qwen3-VL-2B-Instruct*	67.0	84.6	97.9	85.8	66.8	87.8
GeoVLDrive	**68.6**	**89.2**	**98.0**	**90.9**	**76.7**	**89.1**

4.3 Ablation Study

We evaluate latent reasoning and proposal scoring on NAVSIM v1, and geometry injection and spatial alignment on LingoQA and NAVSIM 3DQA.

Latent reasoning and geometry injection. Adding future geometry, key-agent motion, and meta-action states increases PDMS from 90.1 to 91.0 (Table 4). With geometry injection, the full model achieves 91.8 PDMS, improving ego progress from 86.0 to 88.1 and drivable-area compliance from 97.8 to 98.0, while TTC decreases from 95.7 to 95.3.

Geometry injection and spatial alignment. Geometry injection improves the NAVSIM 3DQA score from 84.6 to 87.5, and normalized (u, v) alignment further raises it to 89.2 (Table 5). Position

Table 4: **Ablation of latent reasoning and geometry injection.** Starting from a semantically pre-aligned backbone, we progressively add latent reasoning groups and geometry injection (VGGT). Checkmarks indicate enabled components.

World	Motion	Meta	VGGT	NC↑	DAC↑	TTC↑	Comf.↑	EP↑	PDMS↑
				98.2	97.3	93.6	99.9	87.0	90.1
✓				98.4	97.7	93.8	**100.0**	87.0	90.5
✓	✓			98.8	97.5	95.3	99.9	86.1	90.7
✓	✓	✓		**99.0**	97.8	**95.7**	99.9	86.0	91.0
✓	✓	✓	✓	98.7	**98.0**	95.3	**100.0**	**88.1**	**91.8**

Table 5: **Ablation of geometry injection and spatial alignment.** VGGT denotes geometry injection, and UVPos denotes normalized (u, v) positional encoding. Metrics follow Table 3.

	LingoQA (%)	**NAVSIM 3DQA (%)**				
Configuration	Lingo-Judge↑	Overall↑	Category↑	Motion↑	Position↑ $L_2 \leq 1\,\mathrm{m}$	Dimensions↑ All $\leq 0.5\,\mathrm{m}$
Qwen3-VL-2B-Instruct	67.0	84.6	97.9	85.8	66.8	87.8
+ VGGT	67.6	87.5	97.9	90.1	72.2	88.4
+ VGGT + UVPos	**68.6**	**89.2**	**98.0**	**90.9**	**76.7**	**89.1**

accuracy increases from 66.8 to 72.2 and 76.7, respectively. LingoQA accuracy improves from 67.0 to 67.6 with geometry injection and reaches 68.6 after alignment.

Proposal scoring. With all other settings fixed, Table 6 compares an iPad-style MLP readout of max-pooled action states (Guo et al., 2025a) with three score-token variants refined within causal joint attention. Action-only score tokens achieve 90.4 PDMS, compared with 90.2 for the MLP readout. Direct scene attention increases PDMS to 90.8, while enabling cross-proposal score attention further raises it to 91.8.

Table 6: **Effect of trajectory scoring design on planning performance.** We compare an iPad-style MLP readout of max-pooled action states (Guo et al., 2025a) with dedicated score tokens within our framework on NAVSIM v1. Scene attention provides direct access to scene context, while cross-proposal attention enables information exchange among score tokens. Checkmarks indicate enabled attention paths; blank cells indicate disabled paths, and – means not applicable.

Scoring mechanism	Direct scene attention	Cross-proposal score attention	NC↑	DAC↑	TTC↑	Comf.↑	EP↑	PDMS↑
Max pooling + MLP	–	–	98.4	97.4	94.3	99.9	86.4	90.2
Score tokens			98.5	97.4	93.9	99.9	87.0	90.4
Score tokens	✓		98.6	97.7	94.2	**100.0**	87.2	90.8
Score tokens	✓	✓	**98.7**	**98.0**	**95.3**	**100.0**	**88.1**	**91.8**

5 Conclusion

GeoVLDrive grounds vision-language planning in 3D geometry and future dynamics through a latent reasoning chain. Experiments on NAVSIM v1/v2, NAVSIM 3DQA, and LingoQA show improvements in planning and scene understanding. The current focus on one key agent and supervision from a single recorded future motivate extensions to multi-agent interactions and multiple plausible futures.

Reproducibility Statement

We describe the architecture and training objectives in Section 3. Appendix A.1 details NAVSIM 3DQA construction and evaluation. Appendix A.2 provides implementation settings, loss definitions, and the training schedule.

AI Use Statement

We used generative AI tools for manuscript drafting, language editing, LaTeX formatting, and training data cleaning. The authors take responsibility for the final manuscript and associated research artifacts.

References

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, et al. Flamingo: A visual language model for few-shot learning. In *Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2022.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Wei Cao, Marcel Hallgarten, Tianyu Li, Daniel Dauner, Xunjiang Gu, Caojun Wang, Yakov Miron, Marco Aiello, Hongyang Li, Igor Gilitschenski, Boris Ivanovic, Marco Pavone, Andreas Geiger, and Kashyap Chitta. Pseudo-simulation for autonomous driving. *arXiv preprint arXiv:2506.04218*, 2025.

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.

Yuntao Chen, Yuqi Wang, and Zhaoxiang Zhang. DrivingGPT: Unifying driving world modeling and planning with multi-modal autoregressive transformers. In *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2025.

Jeffrey Cheng and Benjamin Van Durme. Compressed chain of thought: Efficient reasoning through dense representations. *arXiv preprint arXiv:2412.13171*, 2024.

Kashyap Chitta, Aditya Prakash, Bernhard Jaeger, Zehao Yu, Katrin Renz, and Andreas Geiger. TransFuser: Imitation with transformer-based sensor fusion for autonomous driving. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2023.

Daniel Dauner, Marcel Hallgarten, Tianyu Li, Xinshuo Weng, Zhiyu Huang, Zetong Yang, Hongyang Li, Igor Gilitschenski, Boris Ivanovic, Marco Pavone, Andreas Geiger, and Kashyap Chitta. NAVSIM: Data-driven non-reactive autonomous vehicle simulation and benchmarking. In *Adv. Neural Inf. Process. Syst. (NeurIPS) Datasets and Benchmarks*, 2024.

Renju Feng, Ning Xi, Duanfeng Chu, Rukang Wang, Zejian Deng, Anzheng Wang, Liping Lu, Jinxiang Wang, and Yanjun Huang. ARTEMIS: Autoregressive end-to-end trajectory planning with mixture of experts for autonomous driving. *arXiv preprint arXiv:2504.19580*, 2025.

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.

Shenyuan Gao, Jiazhi Yang, Li Chen, Kashyap Chitta, Yihang Qiu, Andreas Geiger, Jun Zhang, and Hongyang Li. Vista: A generalizable driving world model with high fidelity and versatile controllability. In *Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2024.

Xingtai Gui, Meijie Zhang, Tianyi Yan, Wencheng Han, Jiahao Gong, Feiyang Tan, Chengzhong Xu, and Jianbing Shen. Bridging scene generation and planning: Driving with world model via unifying vision and motion representation. *arXiv preprint arXiv:2603.14948*, 2026.

Ke Guo, Haochen Liu, Xiaojun Wu, Jia Pan, and Chen Lv. iPad: Iterative proposal-centric end-to-end autonomous driving. *arXiv preprint arXiv:2505.15111*, 2025a.

Xianda Guo, Ruijun Zhang, Yiqun Duan, Yuhang He, Dujun Nie, Wenke Huang, Chenming Zhang, Shuai Liu, Hao Zhao, and Long Chen. SURDS: Benchmarking spatial understanding and reasoning in driving scenarios with vision language models. In *Adv. Neural Inf. Process. Syst. (NeurIPS) Datasets and Benchmarks Track*, 2025b.

Shibo Hao, Sainbayar Sukhbaatar, DiJia Su, Xian Li, Zhiting Hu, Jason Weston, and Yuandong Tian. Training large language models to reason in a continuous latent space. In *Proc. Conf. Lang. Model. (COLM)*, 2025.

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 *Proc. Eur. Conf. Comput. Vis. (ECCV)*, pp. 533–549, 2022.

Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, Lewei Lu, Xiaosong Jia, Qiang Liu, Jifeng Dai, Yu Qiao, and Hongyang Li. Planning-oriented autonomous driving. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 17853–17862, 2023.

Minqing Huang, Yujiao Xiang, Zihan Liang, Jiajie Huang, Jingqi Wang, Yuheng Zhou, Zhi Xu, Feiyang Tan, Hangning Zhou, Mu Yang, and Gong Che. CoWorld-VLA: Thinking in a multi-expert world model for autonomous driving. *arXiv preprint arXiv:2605.10426*, 2026.

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.

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 *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, pp. 8340–8350, 2023.

Jingyu Li, Junjie Wu, Dongnan Hu, Xiangkai Huang, Bin Sun, Zhihui Hao, Xianpeng Lang, Xiatian Zhu, and Li Zhang. SGDrive: Scene-to-goal hierarchical world cognition for autonomous driving. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2026.

Kailin Li, Zhenxin Li, Shiyi Lan, Yuan Xie, Zhizhong Zhang, Jiayi Liu, Zuxuan Wu, Zhiding Yu, and Jose M. Alvarez. Hydra-MDP++: Advancing end-to-end driving via expert-guided hydra-distillation. *arXiv preprint arXiv:2503.12820*, 2025a.

Yingyan Li, Lue Fan, Jiawei He, Yuqi Wang, Yuntao Chen, Zhaoxiang Zhang, and Tieniu Tan. Enhancing end-to-end autonomous driving with latent world model. In *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2025b.

Yingyan Li, Shuyao Shang, Weisong Liu, Bing Zhan, Haochen Wang, Yuqi Wang, Yuntao Chen, Xiaoman Wang, Yasong An, Chufeng Tang, et al. DriveVLA-W0: World models amplify data scaling law in autonomous driving. *arXiv preprint arXiv:2510.12796*, 2025c.

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. In *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, pp. 27137–27146, 2025d.

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

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 *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2025.

Yushan Liu, Tianxiong Lv, Bohua Wang, Hangqi Fan, Chenxu Zhao, He Zheng, Xuchang Zhong, Yifan Xie, Congyang Zhao, Zhihao Liao, Leigang Luo, Yang Cai, Xiao-Ping Zhang, and Wenbo Ding. PerceptDrive: Perception prior world-action modeling with adaptive expert routing for end-to-end autonomous driving. *arXiv preprint arXiv:2607.20175*, 2026a.

Zhe Liu, Runhui Huang, Rui Yang, Siming Yan, Zining Wang, Lu Hou, Di Lin, Xiang Bai, and Hengshuang Zhao. DrivePI: Spatial-aware 4D MLLM for unified autonomous driving understanding, perception, prediction and planning. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2026b.

Jinghui Lu, Jiayi Guan, Zhijian Huang, Jinlong Li, Guang Li, Lingdong Kong, Yingyan Li, Han Wang, Shaoqing Xu, Yuechen Luo, et al. OneVL: One-step latent reasoning and planning with vision-language explanation. *arXiv preprint arXiv:2604.18486*, 2026.

Yuechen Luo, Fang Li, Shaoqing Xu, Zhiyi Lai, Lei Yang, Qimao Chen, Ziang Luo, Zixun Xie, Shengyin Jiang, Jiaxin Liu, Long Chen, Bing Wang, and Zhi-Xin Yang. AdaThinkDrive: Adaptive thinking via reinforcement learning for autonomous driving. *arXiv preprint arXiv:2509.13769*, 2025.

Yuechen Luo, Fang Li, Shaoqing Xu, Yang Ji, Zehan Zhang, Bing Wang, Yuannan Shen, Jianwei Cui, Long Chen, Guang Chen, Hangjun Ye, Zhi-Xin Yang, and Fuxi Wen. LaST-VLA: Thinking in latent spatio-temporal space for vision-language-action in autonomous driving. *arXiv preprint arXiv:2603.01928*, 2026.

Ana-Maria Marcu, Long Chen, Jan Hünermann, Alice Karnsund, Benoit Hanotte, Prajwal Chidananda, Saurabh Nair, Vijay Badrinarayanan, Alex Kendall, Jamie Shotton, Elahe Arani, and Oleg Sinavski. LingoQA: Visual question answering for autonomous driving. In *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2024.

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 *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2026.

Shuyao Shang, Bing Zhan, Yunfei Yan, Yuqi Wang, Yingyan Li, Yasong An, Xiaoman Wang, Jierui Liu, Lu Hou, Lue Fan, Zhaoxiang Zhang, and Tieniu Tan. DynVLA: Learning world dynamics for action reasoning in autonomous driving. *arXiv preprint arXiv:2603.11041*, 2026.

Zhenyi Shen, Hanqi Yan, Linhai Zhang, Zhanghao Hu, Yali Du, and Yulan He. CODI: Compressing chain-of-thought into continuous space via self-distillation. *arXiv preprint arXiv:2502.21074*, 2025.

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 *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2024.

Shuhan Tan, Kashyap Chitta, Yuxiao Chen, Ran Tian, Yurong You, Yan Wang, Wenjie Luo, Yulong Cao, Philipp Krähenbühl, Marco Pavone, and Boris Ivanovic. Latent chain-of-thought world modeling for end-to-end autonomous driving. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2026.

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. *arXiv preprint arXiv:2402.12289*, 2024.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. VGGT: Visual geometry grounded transformer. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 5294–5306, 2025a.

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. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2026.

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, et al. InternVL3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*, 2025b.

Xiaofeng Wang, Zheng Zhu, Guan Huang, Xinze Chen, Jiagang Zhu, and Jiwen Lu. DriveDreamer: Towards real-world-driven world models for autonomous driving. In *Proc. Eur. Conf. Comput. Vis. (ECCV)*, pp. 55–72, 2024.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2022.

Xilin Wei, Xiaoran Liu, Yuhang Zang, Xiaoyi Dong, Yuhang Cao, Jiaqi Wang, Xipeng Qiu, and Dahua Lin. SIM-CoT: Supervised implicit chain-of-thought. In *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2026.

Jin Yao, Dhruva Dixith Kurra, Tom Lampo, Zezhou Cheng, Danhua Guo, and Burhan Yaman. VLGA: Vision-language-geometry-action models for autonomous driving. *arXiv preprint arXiv:2606.12396*, 2026a.

Wenhao Yao, Zhenxin Li, Shiyi Lan, Zi Wang, Xinglong Sun, Jose M. Alvarez, and Zuxuan Wu. DriveSuprim: Towards precise trajectory selection for end-to-end planning. In *Proc. AAAI Conf. Artif. Intell.*, volume 40, pp. 11910–11918, 2026b. doi: 10.1609/aaai.v40i14.38178.

Shuang Zeng, Xinyuan Chang, Mengwei Xie, Xinran Liu, Yifan Bai, Zheng Pan, Mu Xu, Xing Wei, and Ning Guo. FutureSightDrive: Thinking visually with spatio-temporal CoT for autonomous driving. *arXiv preprint arXiv:2505.17685*, 2025.

Songyan Zhang, Jinyuan Tian, Hanbing Li, Daqi Liu, Hao Chen, Wenhui Huang, Fang Li, Guang Chen, Hangjun Ye, Long Chen, Kuiyuan Yang, and Chen Lv. GeoWorldAD: Geometry world action model for autonomous driving. *arXiv preprint arXiv:2607.17521*, 2026.

Zhida Zhao, Talas Fu, Yifan Wang, Lijun Wang, and Huchuan Lu. From forecasting to planning: Policy world model for collaborative state-action prediction. *arXiv preprint arXiv:2510.19654*, 2025.

Wenzhao Zheng, Weiliang Chen, Yuanhui Huang, Borui Zhang, Yueqi Duan, and Jiwen Lu. OccWorld: Learning a 3D occupancy world model for autonomous driving. In *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2024.

Xingcheng Zhou, Xuyuan Han, Feng Yang, Yunpu Ma, Volker Tresp, and Alois Knoll. OpenDriveVLA: Towards end-to-end autonomous driving with large vision language action model. In *Proc. AAAI Conf. Artif. Intell.*, 2026.

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.

Dong Zhuo, Wenzhao Zheng, Jiahe Guo, Yuqi Wu, Jie Zhou, and Jiwen Lu. Streaming 4D visual geometry transformer. *arXiv preprint arXiv:2507.11539*, 2025.

Sicheng Zuo, Zixun Xie, Wenzhao Zheng, Shaoqing Xu, Fang Li, Hanbing Li, Long Chen, Zhi-Xin Yang, and Jiwen Lu. DVGT-2: Vision-geometry-action model for autonomous driving at scale. *arXiv preprint arXiv:2604.00813*, 2026.

A Appendix for GeoVLDrive

A.1 NAVSIM 3DQA

A.1.1 Dataset Construction

Data sources and object grounding. We construct questions from NAVSIM front-camera observations, camera calibration, annotated 3D boxes, instance tracks, and HD maps. Target objects

are selected independently of the latent-chain annotations. Each cuboid is projected by clipping its twelve edges at a 0.1 m near plane, sampling nine points per retained edge, and applying camera distortion and intrinsics. The projected convex hull is clipped to the image, and its axis-aligned box is normalized to $[0, 1000]^2$ to identify the target in the text prompt. Model inputs contain unannotated images; the box overlays in Fig. 5 are used only for visualization.

Visibility filtering. Candidate objects must belong to a supported class, have positive box dimensions, and lie ahead of the ego vehicle within a planar distance $d = \sqrt{c_x^2 + c_y^2} \leq 80\,\text{m}$, where (c_x, c_y) is the box center in the current ego frame. Let P and P' be the projected hull before and after image clipping, with widths w and w'. Horizontal truncation $T_h = 1 - w'/w$ and area truncation $T_a = 1 - |P'|/|P|$ must each be at most 0.5; $|P|$ denotes polygon area. We approximate occlusion using overlaps with objects at least $0.05\,\text{m}$ closer in camera depth:

$$O_i = \frac{\left|\bigcup_{j:\, z_j < z_i - 0.05} (P_i' \cap P_j')\right|}{|P_i'|} \leq 0.5, \tag{9}$$

where z_i is the camera-depth coordinate of object i. This approximation does not account for unannotated occluders such as buildings or vegetation. At a reference resolution of 280×518 (height $\times$ width), the minimum box short sides are 10, 12, and 16 pixels and the minimum areas are 160, 250, and 400 pixels squared for $d \leq 40\,\text{m}$, $40 < d \leq 60\,\text{m}$, and $60 < d \leq 80\,\text{m}$, respectively. Both thresholds must be met.

Motion annotation. Vehicle, pedestrian, and bicycle tracks are transformed to the ego frame at the anchor time. For an object center in homogeneous coordinates,

$$\bar{\mathbf{p}}_i^{(0)}(t) = \mathbf{T}_{\text{ego}(0)\leftarrow\text{global}}\, \mathbf{T}_{\text{global}\leftarrow\text{ego}(t)}\, \bar{\mathbf{p}}_i^{(t)}(t), \tag{10}$$

where $\mathbf{T}$ denotes a rigid coordinate transformation. This removes ego motion, so *stationary* refers to motion relative to the ground. Each motion label requires at least three valid track states, including the anchor, spanning at least one second. We fit a linear trajectory to three states and a quadratic trajectory to four or more, using its derivative at the anchor to estimate velocity. Fits with residuals above $0.8\,\text{m}$ or direction discrepancies above 40° relative to the annotated velocity are marked ambiguous; insufficient evidence is marked unavailable. These observations remain eligible for other question types.

Motion classification combines lane overlap, centerline distance, and local road direction (Table 7). Cut-in and cut-out labels require confirmation from annotations up to $2\,\text{s}$ into the future; other boundary cases, except crossing, may also use this evidence. The logged future ego trajectory is excluded from label construction. Model observations are restricted to the four front views at $[-1.5, -1.0, -0.5, 0]$ s.

Event grouping and object selection. Cut-in and cut-out observations are grouped by scene, track, and label, with temporal gaps above $1.25\,\text{s}$ defining separate events. Each event contributes one motion question, selected by annotation confidence, visibility, transition stage, and salience. Other question types may use adjacent frames. Each frame retains at most five objects, prioritizing up to two event anchors, the two most salient candidates, and the nearest eligible object. Remaining slots promote category and spatial coverage, prioritizing vulnerable road users and traffic-control objects over vehicles, followed by fine-grained category diversity. Spatial bins partition longitudinal position c_x at $[20, 40, 60]$ m and lateral position c_y at $[-6, -2, 2, 6]$ m. The salience score is

$$S_i = \log(1 + A_i) - 0.0125\, d_i - 1.5\, O_i - T_{a,i} - 0.5\, T_{h,i}, \tag{11}$$

where A_i is the box area at the reference resolution. Ties are resolved deterministically using a stable hash. Coverage-based selection excludes candidates dominated by a same-category object that is at least four times larger in projected area, closer, and no more occluded or truncated.

Question generation. Structured annotations are converted into the four tasks in Table 8. Prompts specify coordinates and units, and multiple-choice options are deterministically shuffled using a stable hash. The coarse categories are vehicle, vulnerable road user (pedestrian or bicycle), traffic-control or work-zone object (cone, barrier, or construction-zone sign), and other static object. Category questions identify the target by its box; other tasks also provide its class. Every selected object

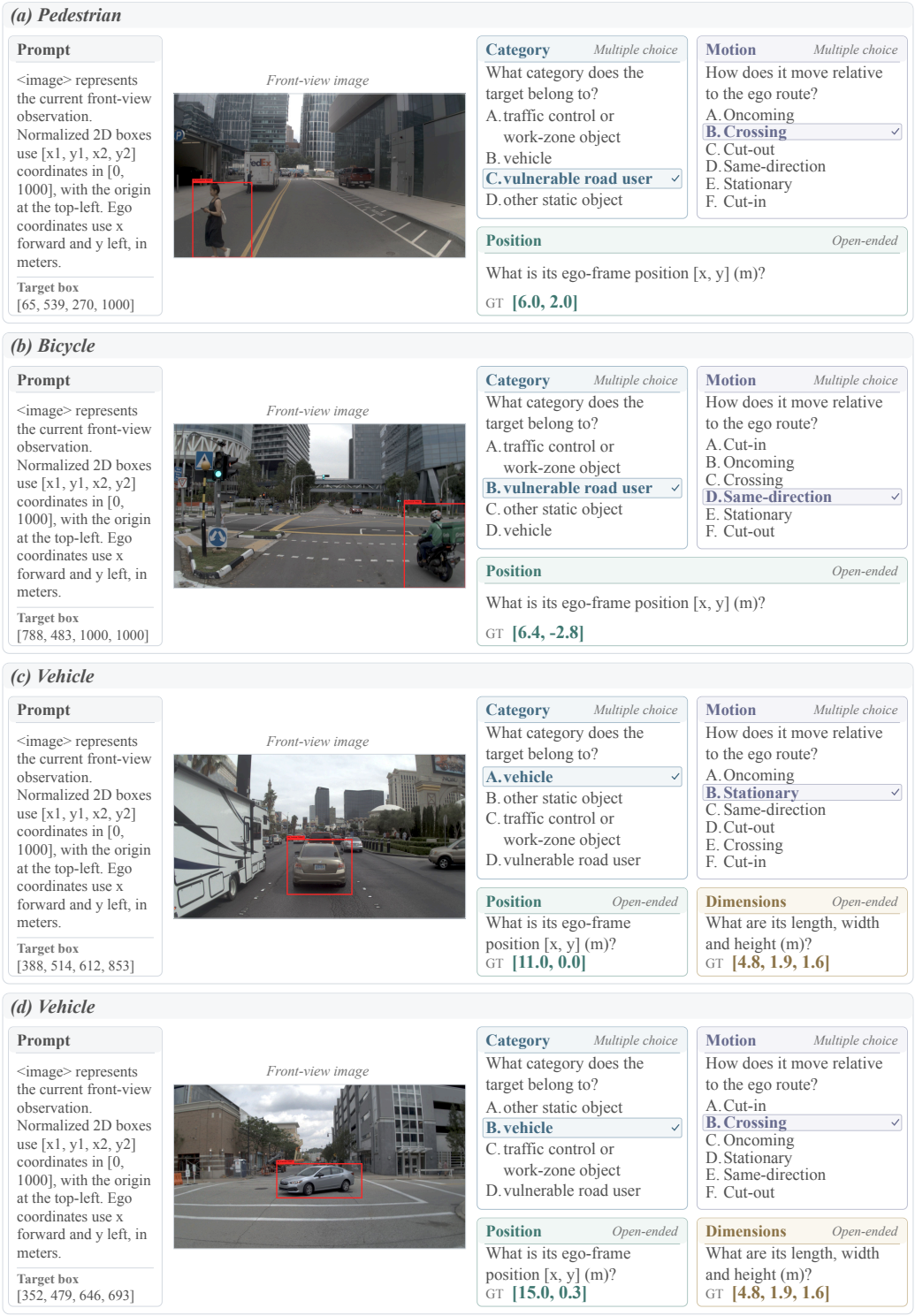

Figure 5: **NAVSIM 3DQA examples.** Each row presents the shared prompt and target box, the annotated front view, and the associated questions. Colors distinguish question types. Multiple-choice options retain their original order, with correct answers outlined and checked. Numerical answers are labeled GT (ground truth). Questions are abbreviated for display; box, coordinate, and unit conventions are specified in the prompt.

Table 7: **Motion categories and annotation criteria.** The angle θ measures the difference between the object's direction of travel and the local road tangent. Cut-in and cut-out additionally require the direction of travel to be within 55° of the ego forward direction.

Class	Annotation criteria
stationary	Low ground speed and small recent displacement; selected low-speed cases additionally use future displacement and terminal speed for confirmation.
same direction	Motion aligned with the local road tangent ($\theta \leq 40^\circ$).
oncoming	Motion opposing the local road tangent ($\theta \geq 140^\circ$).
crossing	Lateral motion at or through the ego road corridor ($55^\circ \leq \theta \leq 125^\circ$). Pedestrian annotation also uses road-surface and crosswalk occupancy.
cut-in	Vehicle or bicycle entering the ego lane over the observed history, confirmed by future lane occupancy and motion direction.
cut-out	Vehicle or bicycle leaving the ego lane over the observed history, with the same future-confirmation criteria.

contributes category and position questions. Vehicles additionally contribute dimension questions, and objects with valid motion labels contribute motion questions, yielding at most twenty questions per frame. Numerical answers are rounded to one decimal place. Evaluation follows Appendix A.1.3.

Table 8: **NAVSIM 3DQA tasks and outputs.** Positions use the current ego frame; all numerical outputs are in meters. The tasks exclude vertical position and object yaw. Accuracy criteria are defined in Appendix A.1.3.

Task	Eligible objects	Output	Evaluation
Coarse category	All selected objects	Category label	Accuracy
Ground-plane position	All selected objects	$[x, y]$	Axis / L_2 accuracy
Vehicle dimensions	Vehicles	$[\ell, w, h]$	Component / joint accuracy
Motion behavior	Vehicles, pedestrians, and bicycles with valid labels	Motion label	Accuracy

A.1.2 Dataset Statistics

NAVSIM 3DQA contains 1,281,855 QA pairs: 1,117,204 for training and 164,651 for validation. Of the 85,070 training and 18,179 validation source anchors, 80,161 and 14,467 yield eligible questions. The two splits contain 333,920 and 48,958 object observations, covering 89,533 and 12,792 distinct tracks, respectively. An object observation denotes one object at one anchor frame. The source split is preserved, with no overlap in scene, frame, track, or log-segment identifiers.

Figure 6 summarizes the dataset composition. Stationary and same-direction labels account for 86.89% of motion questions; cut-in and cut-out contribute 524 (0.19%). Ambiguous observations represent 2.6% of labeled or ambiguous candidates, and unavailable observations represent 8.5% of all dynamic candidates. Automated audits verified that each frame retains its two most salient and nearest eligible objects; the largest projected box is retained in over 99.5% of frames. A stratified visual audit of 40 validation frames found no visibility, projection, or scene-consistency issues. This single-frame audit does not independently validate temporal motion labels.

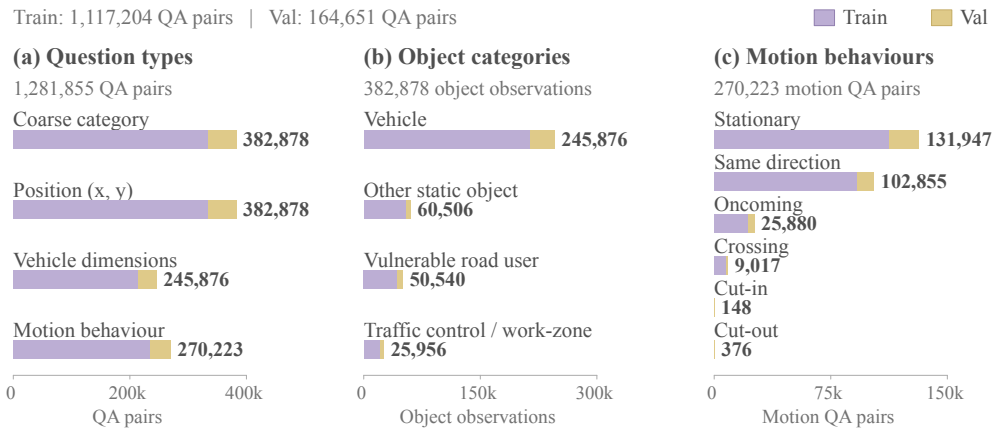

Figure 6: **NAVSIM 3DQA composition.** Distributions of question types, object categories, and motion behaviors. Stacked bars show training (lavender) and validation (pale yellow) counts; labels report their sum. Object counts refer to object–anchor observations. All axes use a linear scale.

A.1.3 Evaluation Protocol

Task scores. Responses are parsed deterministically, with parsing failures counted as incorrect. Category and motion accuracy are computed after mapping responses to semantic labels. A position answer is correct when both coordinate errors are within the distance-dependent tolerance: $0.5\,\mathrm{m}$ for $d \leq 30\,\mathrm{m}$, $1\,\mathrm{m}$ for $30 < d \leq 60\,\mathrm{m}$, and $2\,\mathrm{m}$ for $d > 60\,\mathrm{m}$. Dimension accuracy averages the correctness of length, width, and height, each with an absolute error tolerance of $0.5\,\mathrm{m}$. Each task is averaged over its eligible questions. For task accuracies A_{cat}, A_{motion}, $A_{\mathrm{pos}}^{\mathrm{axis}}$, and $A_{\mathrm{dim}}^{\mathrm{component}}$ in $[0, 1]$, the aggregate is

$$\mathrm{Overall} = \frac{100}{4}\left(A_{\mathrm{cat}} + A_{\mathrm{motion}} + A_{\mathrm{pos}}^{\mathrm{axis}} + A_{\mathrm{dim}}^{\mathrm{component}}\right). \tag{12}$$

Reported position and dimension diagnostics. In Table 3, Position reports the fraction of answers with ground-plane error $\|\hat{\mathbf{p}} - \mathbf{p}\|_2 \leq 1\,\mathrm{m}$, while Dimensions requires all three component errors to be at most $0.5\,\mathrm{m}$. Overall uses the adaptive per-axis position accuracy and per-component dimension accuracy defined above, so it is not the arithmetic mean of the four displayed columns.

A.1.4 Comparison with SURDS

NAVSIM 3DQA complements SURDS (Guo et al., 2025b) by emphasizing metric geometry and temporal motion. SURDS evaluates orientation, depth ranges, pixel localization, and pairwise spatial relations from individual nuScenes images, using categorical correctness and a centerness-based localization score. NAVSIM 3DQA uses four consecutive front views and a normalized 2D box to specify the queried object. It adds numerical estimation of ground-plane position and vehicle dimensions in meters, together with temporal motion classification.

The geometric diagnostics assess physical location and scale. Position accuracy at $L_2 \leq 1\,\mathrm{m}$ jointly constrains longitudinal and lateral errors in the ego frame; joint dimension accuracy requires length, width, and height all to be within $0.5\,\mathrm{m}$ of their targets. These criteria distinguish metric accuracy from correct depth-range or relative-order predictions, which can remain correct despite substantial coordinate errors. Estimating physical dimensions further probes object scale, which image-plane localization alone does not measure. Both tasks require numerical answers without candidate values supplied in the question.

Motion classification extends the evaluation to temporal observations, with labels derived from ego-motion-compensated tracks and local road geometry. This grounds categories such as stationary, crossing, and cut-in in object motion relative to the ground and the road, rather than apparent image motion. Together, the tasks provide complementary diagnostics of metric spatial grounding and

temporal motion understanding relevant to planning; they do not replace SURDS's orientation and multi-object relation tests.

A.2 Implementation and Training Details

A.2.1 Architecture and Latent Interface

Backbone configuration. StreamVGGT features from layers $[4, 11, 17, 23]$ are injected into Qwen3-VL layers $[6, 13, 20, 26]$, respectively. The VLM and geometry streams use image resolutions of 416×672 and 280×518 (height $\times$ width), with a 20×37 patch grid in the geometry stream. Features across the selected geometry layers differ in representation depth while sharing the same spatial resolution.

Latent token interface. The observed visual and language prefix is followed by an ego-state token, future-geometry tokens, key-agent motion tokens, and meta-action tokens. Group-specific embeddings replace the reserved-token embeddings at these positions in the VLM input sequence. With $\delta = 0.5\,\mathrm{s}$, future geometry uses $K = 4$ horizons and 32 tokens per horizon (128 in total). Agent motion uses ten tokens for category, current position, and eight current-relative future offsets. Two tokens encode longitudinal and lateral meta actions. Future-geometry and future-motion tokens receive sinusoidal encodings of their prediction times, with attention governed by Fig. 3. Textual CoT targets follow the latent block during training. At inference, planning uses the latent states without autoregressive text generation.

Coordinate frames. Agent positions and ego trajectories are expressed in meters in the ego frame at the observation time, with x forward and y left. Future agent annotations are transformed to this fixed frame before computing the current-relative position offsets $\Delta\mathbf{p}_j = \mathbf{p}_j - \mathbf{p}_0$, where $\mathbf{p}_0$ is the agent's current position.

Action expert. The expert comprises four refinement stages, each containing four independently parameterized attention blocks conditioned on features from different depths. It maintains $R = 64$ proposals, each represented by $J = 8$ action tokens and one score token. Reasoning, action, and score states are propagated between blocks, while scene features are supplied at each depth. Action queries incorporate ego-state and meta-action context. VLM features and latent states are detached before the expert's trainable projections, blocking expert gradients from updating the VLM while allowing refinement within the expert.

A.2.2 Supervision and Loss Definitions

Future geometry. At each supervised horizon k, let V_k denote the valid depth pixels and $\tilde{\mathbf{z}}_k$ and $\hat{\tilde{\mathbf{z}}}_k$ the normalized target and predicted depth maps. For the per-pixel error $e_{k,i} = |\hat{\tilde{z}}_{k,i} - \tilde{z}_{k,i}|$, the regression and confidence terms in Eq. 3 are

$$\mathcal{L}^{(k)}_{\mathrm{depth}} = \frac{1}{|V_k|} \sum_{i \in V_k} e_{k,i}, \qquad \mathcal{L}^{(k)}_{\mathrm{conf}} = \frac{1}{|V_k|} \sum_{i \in V_k} \left(c_{k,i} e_{k,i} - \alpha \log c_{k,i} \right), \tag{13}$$

where $c_{k,i} > 0$ is predicted confidence and α controls its regularization. Gradient consistency is computed from horizontal and vertical finite differences at subsampling strides $[1, 2, 4, 8]$. Only adjacent pairs with valid depths contribute, and each scale is normalized by its number of valid pixels.

Key-agent motion. Category supervision uses cross-entropy, including the *none* class. Let SL1 denote Smooth L1 with mean reduction. Position supervision averages the current-position, offset, and reconstructed-position losses:

$$\mathcal{L}_{\mathrm{SmoothL1}} = \frac{1}{3} \left[\mathrm{SL1}(\hat{\mathbf{p}}_0, \mathbf{p}_0) + \mathrm{SL1}(\widehat{\Delta\mathbf{p}}_{1:J}, \Delta\mathbf{p}_{1:J}) + \mathrm{SL1}(\hat{\mathbf{p}}_{0:J}, \mathbf{p}_{0:J}) \right]. \tag{14}$$

For consecutive-step displacements $\mathbf{v}_j = \mathbf{p}_j - \mathbf{p}_{j-1}$ and $\hat{\mathbf{v}}_j = \hat{\mathbf{p}}_j - \hat{\mathbf{p}}_{j-1}$, the direction term is

$$\mathcal{L}_{\cos} = \frac{1}{J} \sum_{j=1}^{J} w_j \left(1 - \frac{\hat{\mathbf{v}}_j^\top \mathbf{v}_j}{\max(\|\hat{\mathbf{v}}_j\|_2, \epsilon) \max(\|\mathbf{v}_j\|_2, \epsilon)} \right), \qquad w_j = \min\left(\frac{\|\mathbf{v}_j\|_2}{\eta}, 1 \right). \tag{15}$$

Here $\eta = 0.25\,\mathrm{m}$ downweights nearly stationary targets, and $\epsilon = 10^{-6}$ stabilizes normalization. Position and direction losses are averaged over valid annotated tracks; samples labeled *none* contribute only to classification. The direction loss uses consecutive-step displacements, whereas position regression uses current-relative offsets.

Meta actions and textual CoT. Longitudinal and lateral meta-action predictions use the cross-entropy objectives in Eq. 6. Textual CoT is supervised by autoregressive cross-entropy $\mathcal{L}_{\mathrm{cot}}$ on the annotated response tokens. Geometry, motion, and maneuver losses are applied only to samples with the corresponding annotations.

Trajectory and score supervision. Trajectory regression follows Eq. 7. Following iPad (Guo et al., 2025a), we obtain scoring targets by simulating each final-stage proposal in its corresponding NAVSIM scene (Dauner et al., 2024):

$$q_r = \mathrm{NC}_r \,\mathrm{DAC}_r \,\frac{5\,\mathrm{EP}_r + 5\,\mathrm{TTC}_r + 2\,\mathrm{Comf}_r}{12}. \tag{16}$$

Metric definitions are given in Table 11. For final proposal logits a_r, binary cross-entropy uses these simulator-derived scores $q_r \in [0, 1]$ as soft targets:

$$\mathcal{L}_{\mathrm{score}} = -\frac{1}{R}\sum_{r=1}^{R}\left[q_r \log \sigma(a_r) + (1 - q_r)\log(1 - \sigma(a_r))\right], \tag{17}$$

where σ is the sigmoid function.

Auxiliary supervision. The expert's auxiliary agent loss $\mathcal{L}^{\mathrm{aux}}_{\mathrm{agent}}$ combines category cross-entropy and the position terms in Eq. 14, excluding the direction term. Its auxiliary meta-action loss $\mathcal{L}^{\mathrm{aux}}_{\mathrm{meta}}$ uses the same maneuver labels as the backbone classifiers. Both objectives supervise the expert's final refined states.

A.2.3 Training Schedule

Table 9 summarizes the four training phases. Training uses 64 NVIDIA H20 GPUs with the geometry encoder frozen. The VLM continues to receive language and latent-CoT supervision during joint action training.

Table 9: **Four-phase training schedule.** Each phase initializes from the preceding checkpoint.

Phase	Epochs	Training focus
1 Alignment	3	Geometry-language alignment with driving QA
2 Forecasting	5	Future geometry and key-agent motion
3 Chain	10	Causal latent CoT and language supervision
4 Action	30	Joint chain, trajectory, and proposal-score supervision

A.2.4 NAVSIM-WM-CoT

Annotation structure. NAVSIM-WM-CoT associates observed scenes with future depth, key-agent tracks, longitudinal and lateral meta actions, and textual reasoning annotations. Table 10 summarizes the temporal object-QA and chain annotation sets before phase-specific sampling. Future observations are used only to construct supervision targets.

Key-agent annotations. Each scene contains at most one selected key agent, labeled as *vehicle*, *pedestrian*, *cyclist*, or *other*; *none* denotes the absence of a selected agent. A valid motion target comprises the current position and eight future positions sampled at 0.5 s intervals in the current ego frame. Future offsets follow the coordinate convention in Appendix A.2.1. Regression requires a valid track, while category supervision includes *none*. The validation annotations contain 5,087 scenes with a selected agent and 13,092 without one. The *none* label refers only to agent selection and does not exclude other planning constraints.

Table 10: **NAVSIM-WM-CoT annotation sets.** Counts precede phase-specific sampling.

Annotation set	Train	Validation	Supervision
Temporal object QA	177K	25K	Language answers, future geometry, and agent motion
Chain	85,070	18,179	Key agent, track, meta actions, and textual rationale

Meta-action labels. Longitudinal labels are *maintain speed*, *accelerate*, *decelerate*, *stop*, and *reverse*. Lateral labels are *keep lane*, *turn left*, *turn right*, *change left*, *change right*, *nudge left*, and *nudge right*. These labels supervise both the backbone meta-action tokens and the expert's auxiliary meta-action readouts.

A.2.5 Evaluation Settings

NAVSIM planning metrics. Planning is evaluated using PDMS on NAVSIM v1 and EPDMS on NAVSIM v2 (Dauner et al., 2024; Cao et al., 2025). Table 11 defines the metric abbreviations; all values are reported on a 0–100 scale. NAVSIM v1 evaluates four-second ego rollouts with non-reactive log-replay traffic. PDMS combines collision and drivable-area terms with weighted progress, time-to-collision, and comfort scores, as in Eq. 16. NAVSIM v2 adds further compliance and comfort criteria. Aggregate scores follow the benchmark protocol, with per-scene scoring performed before dataset-level aggregation.

Table 11: **NAVSIM metric abbreviations.** Definitions follow the benchmark protocols (Dauner et al., 2024; Cao et al., 2025). Higher values indicate better performance.

Abbreviation	Metric	Reported in
NC	No at-fault collisions	v1, v2
DAC	Drivable-area compliance	v1, v2
TTC	Time-to-collision within bound	v1, v2
EP	Ego progress	v1, v2
Comf.	Comfort	v1
DDC	Driving-direction compliance	v2
TLC	Traffic-light compliance	v2
LK	Lane keeping	v2
HC	History comfort	v2
EC	Extended comfort	v2

LingoQA. We use Lingo-Judge accuracy under the LingoQA evaluation protocol (Marcu et al., 2024). The fine-tuned Qwen3-VL baseline and GeoVLDrive share the training data and schedule specified in the main comparison.

A.3 Reasoning and Planning Visualizations

The following visualizations examine decoded latent-CoT predictions, generated textual CoT, and planned ego trajectories in selected driving scenes.

A.3.1 Latent CoT, Textual CoT, and Action Predictions

Figure 7 presents model inference on six selected scenes from the NAVSIM *val* split. Future depth, key-agent motion, and meta actions are decoded from latent CoT states. Text excerpts are drawn from the generated textual CoT, and ego trajectories are produced by the action expert. Positions and trajectories use the current ego frame.

For visualization, each depth prediction is normalized by its mean magnitude over the displayed valid pixels and rescaled to the mean valid GT depth. The masked row uses GT-valid pixels; the full prediction row uses pixels with finite, positive predicted depth.

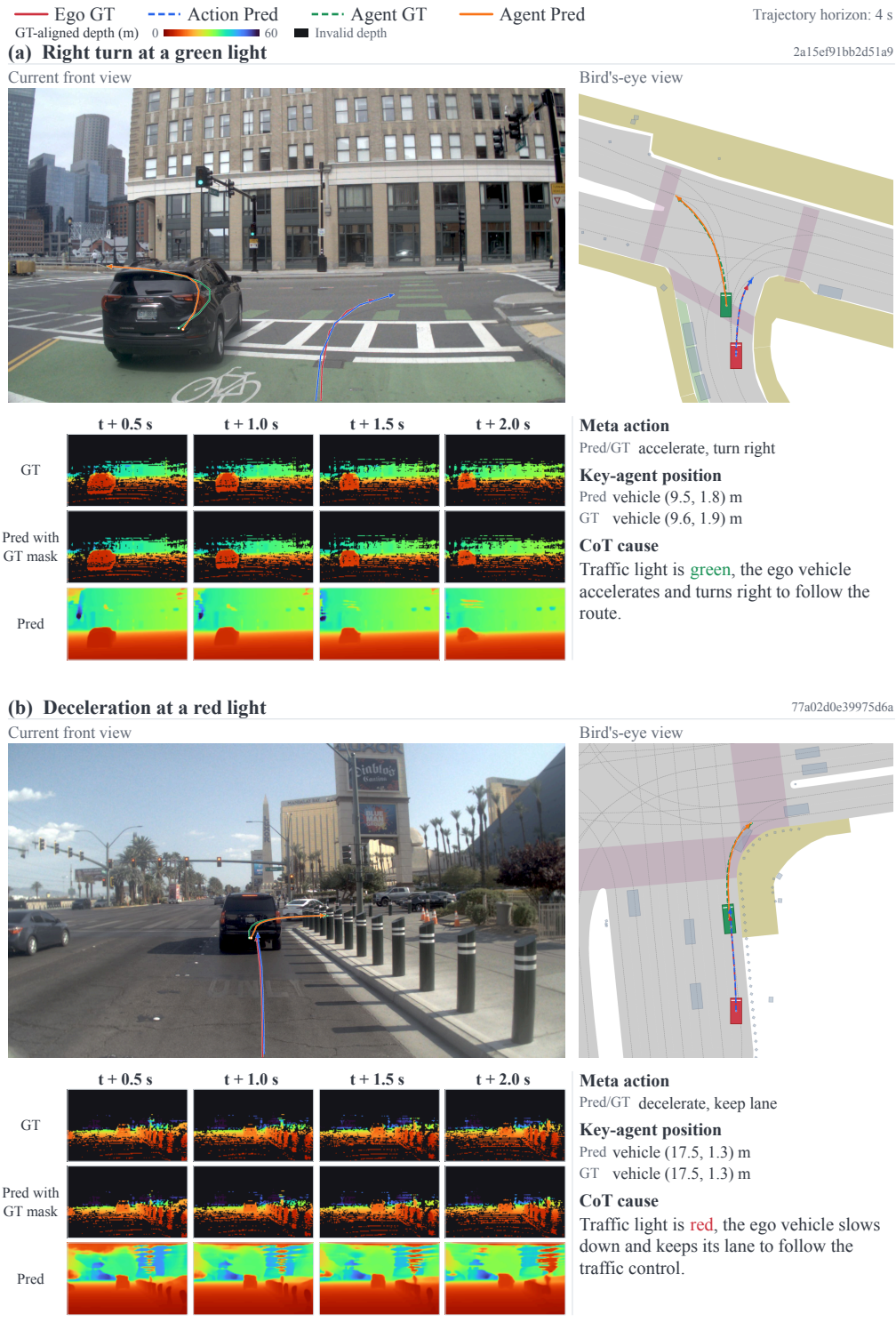

Figure 7: **GeoVLDrive inference on NAVSIM val.** Depth rows show ground truth (GT), predictions at GT-valid pixels, and full predictions. Depth horizons span 0.5–2.0 s; trajectories cover 4 s. Both prediction rows are normalized and rescaled to the mean valid GT depth for display.



Figure 7 (continued): Panels (c)–(d).

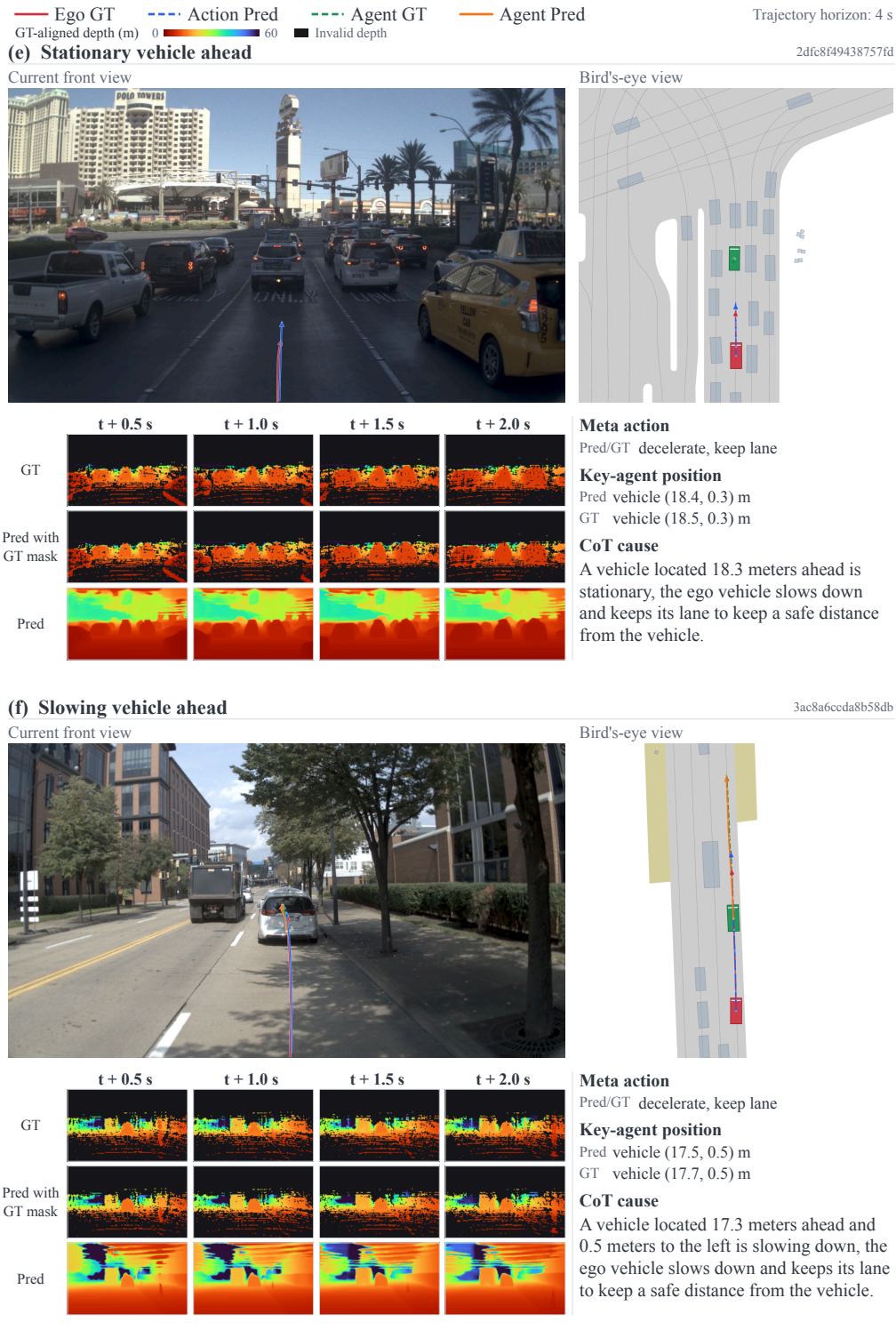

Figure 7 (continued): Panels (e)–(f): responses to a vehicle ahead.

A.3.2 Representative Planning Cases



Figure 8: **Qualitative planning comparisons.** Panels (a)–(c) of six selected examples. Each panel pairs a front-camera observation with a bird's-eye view of baseline (red) and GeoVLDrive (green) trajectories.

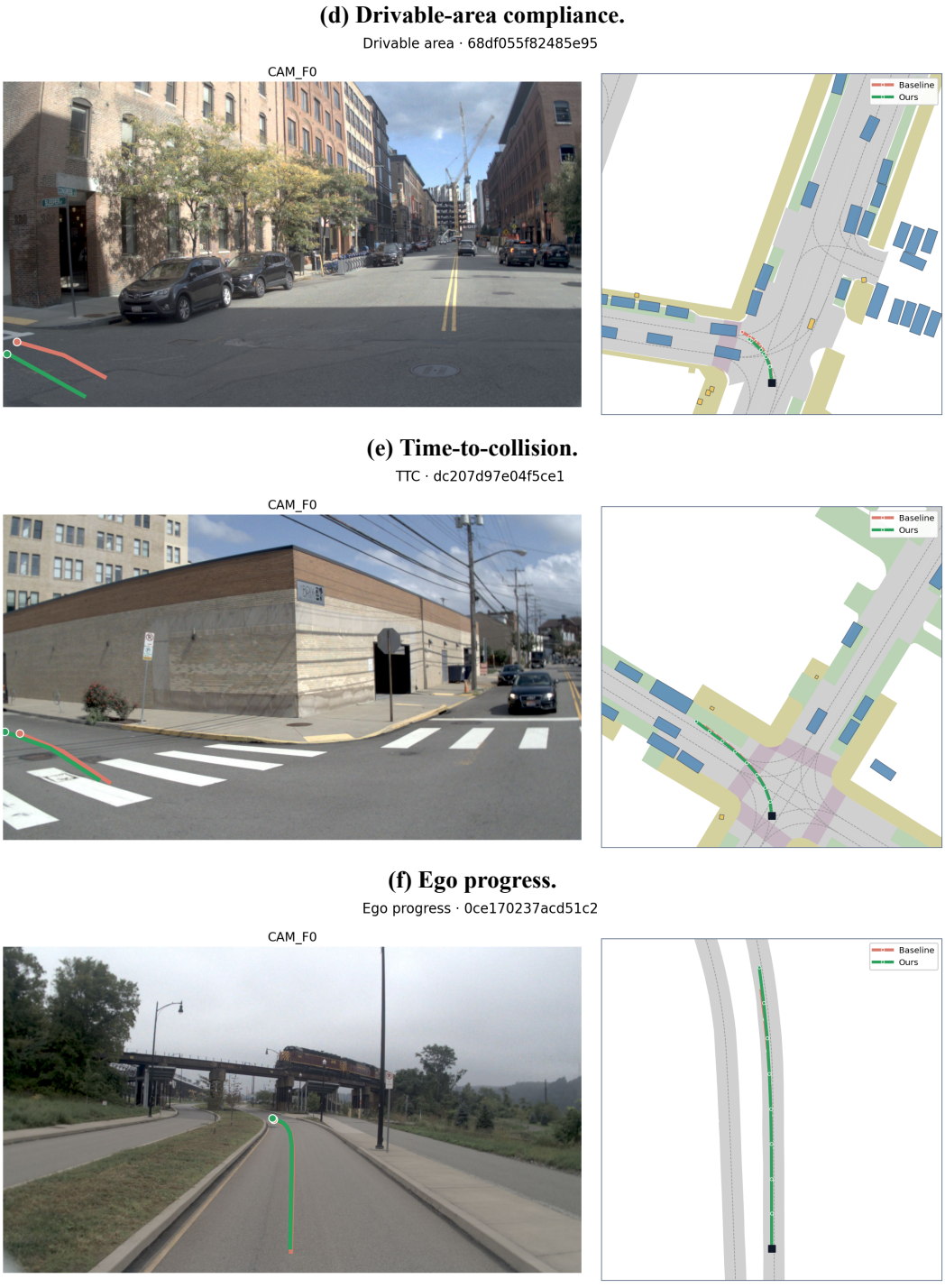

Figure 8 (continued): Panels (d)–(f). The color convention is shared with panels (a)–(c).

Scope and limitations. GeoVLDrive explicitly models one selected key agent within a single front-camera view. Geometry supervision covers 2 s, compared with the 4 s motion and planning horizon. Supervision follows the recorded future and does not cover all alternative scene evolutions. Directed attention specifies representation dependencies without establishing causal relationships in the environment. The selected qualitative examples do not characterize failure frequencies or replace quantitative evaluation.