RECON4DRIVE: SELECTIVE AND EXPLICIT 4D PRIORS FOR VLA DRIVING

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision-language-action (VLA) models reason effectively about driving scenes, but their image-centric representations do not explicitly capture the metric geometry and temporal dynamics required for planning. Pretrained 4D reconstruction models encode such information, yet their dense representations are designed to describe the entire scene. For planning, however, only a small portion of this information is relevant, and the current decision mainly depends on a few ego-centric relations, such as the distance to a lane boundary or the gap to a nearby vehicle, which remain implicit in dense features. To this end, we present RECON4DRIVE, a VLA framework that converts dense 4D reconstruction features into compact and explicit inputs for planning. The Semantic-Guided 4D Retriever (SG4R) uses image semantics to retrieve planning-relevant geometry and motion cues from the 4D representation, keeping the number of 4D tokens small regardless of its density. Planning-Critical Structured Scene Encoding (PSE) further assigns supervised queries to key relations with lanes and surrounding agents, including their future motion, making these relations directly accessible to reasoning and planning. On the closed-loop Bench2Drive benchmark, RECON4DRIVE achieves a Success Rate of 62.27% and a Driving Score of 85.45 with camera-only inference, the best results among methods trained on the official Think2Drive-collected data. The code will be made publicly available upon publication.

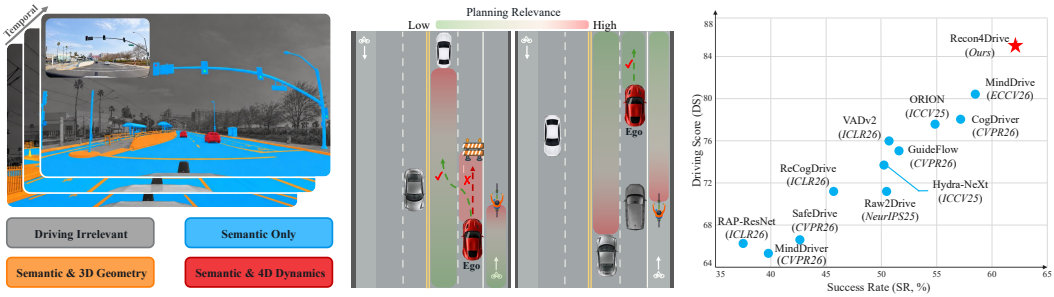
1 INTRODUCTION

Vision-language-action (VLA) models have emerged as a promising paradigm for end-to-end autonomous driving, as they inherit the semantic understanding and reasoning abilities of pretrained vision-language models (VLMs) (Li et al., 2026b; Liu et al., 2026b; Li et al., 2026c; Zhang et al., 2026; Peng et al., 2026). However, driving takes place in a dynamic 3D world (i.e., 4D world), where planning relies not only on semantics but also on metric geometry, such as the distances to lane boundaries and obstacles, and on temporal dynamics, such as the motion of surrounding agents. In interactive scenarios, these cues must be considered jointly: overtaking requires estimating the gap to the leading vehicle, free space in the adjacent lane, and the speed of vehicles approaching from behind, while emergency braking depends on the distance to an obstacle and its closing speed. In contrast, image-centric VLM representations are rich in semantics but do not explicitly capture such 4D (spatiotemporal) scene information, so distances and motion must be inferred from 2D appearance.

Pretrained 4D scene reconstruction models (Yang et al., 2025a; Lu et al., 2025) provide such information by jointly encoding metric scene geometry and temporal dynamics. This distinguishes them from other visual foundation models: 3D geometry models like VGGT (Wang et al., 2025) recover accurate structure but treat observations as static, while video world models like Cosmos (Agarwal et al., 2025; 2026) capture scene evolution for visual generation, without explicit metric geometry. Recent studies also show the value of spatial and spatiotemporal representations for autonomous driving (Lu et al., 2025; Strong et al., 2026; Liu et al., 2026c). However, transferring dense reconstruction features to VLA models is non-trivial, as reconstruction and planning require different information.

Reconstruction aims to represent the full dynamic scene, so its dense 4D representation contains much more information than planning requires. As shown in Fig. 1(a), different scene elements require different cues: many regions are irrelevant to driving, traffic signals mainly require semantics,

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.



(a) Selective Information for Driving **(b) Planning-Critical Relations for Driving** **(c) Closed-loop Performance Comparison**

Figure 1: Not all 4D information matters for planning. (a) Selective information: driving irrelevant (gray), semantic only (blue), semantic & 3D geometry (orange), and semantic & 4D dynamics (red). (b) Planning-critical relations: only nearby lanes and a few agents have high planning relevance. (c) RECON4DRIVE achieves the highest SR and DS among methods trained on the official data.

road boundaries and static obstacles additionally require metric geometry, and moving agents further require temporal dynamics. Processing the entire dense representation is therefore unnecessary and also costly, since the self-attention cost of a VLM grows quadratically with the input length.

Moreover, even if the dense 4D representation contains all planning-relevant cues, the model may fail to extract and use them, as redundant signals can interfere. Compression removes redundancy but risks discarding key geometric or dynamic cues. As illustrated in Fig. 1(b), even among relevant elements, only a few ego-centric relations directly affect the current decision, such as those to the current lane boundaries and the nearest agents in the ego and adjacent lanes. Such relations remain implicit in both dense and compressed 4D tokens. The core challenge is therefore twofold: 1) the dense 4D representation should be compressed without losing planning-relevant information, and 2) these key relations should be made explicitly accessible to downstream reasoning and planning.

To address these challenges, we propose **RECON4DRIVE**, which selects planning-relevant 4D cues from a pretrained 4D reconstruction model (e.g., SLARM (Qiu et al., 2026)) and explicitly encodes key ego-centric relations for VLA planning. Specifically, first, we design the Semantic-Guided 4D Retriever (SG4R), which uses VLM visual tokens as queries to retrieve geometry and motion cues from the dense 4D representation. As a result, the VLM receives only a compact set of 4D tokens, regardless of how dense the original representation is. Second, as the retrieved cues remain implicit, we introduce Planning-Critical Structured Scene Encoding (PSE), which uses a small set of supervised relation queries to encode ego-centric relations to the current lane boundaries and nearby agents, including their future motion. Instead of reconstructing the complete scene, these queries model only decision-relevant relations, and their VLM outputs, termed relation states, expose these relations to downstream reasoning and planning. Benefiting from this design, RECON4DRIVE exploits the metric geometry and temporal dynamics of 4D reconstruction at a small extra cost while keeping decision-critical relations directly accessible, leading to more reliable planning.

Closed-loop evaluations on Bench2Drive (Jia et al., 2024), a widely used autonomous driving benchmark, demonstrate the effectiveness of our method. In particular, with camera-only inference, RECON4DRIVE achieves a Success Rate (SR) of 62.27% and a Driving Score (DS) of 85.45, significantly outperforming prior methods trained on the official Think2Drive-collected data, as shown in Fig. 1(c). Moreover, compared with its image-only counterpart, RECON4DRIVE substantially improves planning while adding only a small number of 4D tokens to the VLM.

We summarize our contributions as follows:

- We present RECON4DRIVE, a VLA framework that selectively transforms the dense representation of a pretrained 4D reconstruction model into compact, planning-relevant information, instead of concatenating it densely. It achieves state-of-the-art performance on the representative closed-loop benchmark, substantially outperforming prior methods using official Think2Drive-collected data.

- We introduce the Semantic-Guided 4D Retriever (SG4R), which retrieves planning-relevant 4D cues with VLM visual tokens, significantly reducing 4D tokens and VLM computation while largely preserving performance. As the retrieved cues remain implicit, we further introduce

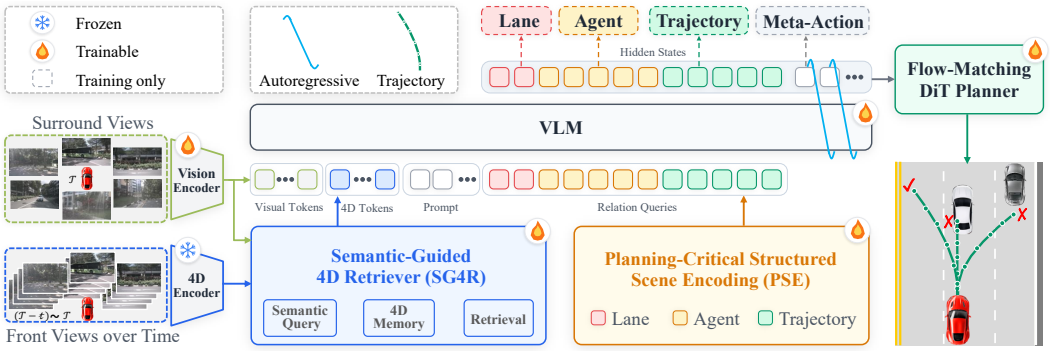


Figure 2: Overview of RECON4DRIVE. A frozen 4D encoder builds a 4D world memory from the front views over time. SG4R uses visual tokens as queries to retrieve compact 4D tokens from it. In PSE, supervised relation queries encode lane, agent, and trajectory relations. The VLM generates a Meta-Action and feeds hidden states to a flow-matching DiT planner for trajectory generation.

Planning-Critical Structured Scene Encoding (PSE) to make key ego-centric lane and agent relations explicit via supervised relation queries.

2 RELATED WORK

2.1 VLA MODELS FOR END-TO-END DRIVING

Driving VLAs mainly explore how to reason over scenes and map high-level understanding to continuous actions, including language-guided closed-loop driving, hierarchical VLM planning, and unified vision-language-action generation (Xu et al., 2024; Shao et al., 2024; Sima et al., 2025; Tian et al., 2025; Zhao et al., 2026a; Zhou et al., 2026c; Wang et al., 2026b), as well as specialized action generation, latent action representations, and driving foundation models (Yang et al., 2026; Xie et al., 2026a; Zhou et al., 2026b), alongside advances in general VLA models (Fang et al., 2026; Xie et al., 2026b). Recent methods couple VLM reasoning with generative or diffusion planners to bridge language decisions and trajectories (Fu et al., 2025; Li et al., 2026c), refine decisions with reinforcement learning (Fu et al., 2026; Zhou et al., 2025b), or improve temporal consistency and efficiency via temporal memory and latent reasoning (Liu et al., 2026b; Peng et al., 2026). These works mainly improve how driving VLAs reason and act. We instead study a complementary question: what scene information should be exposed to the VLA for planning. Specifically, we selectively compress pretrained 4D representations and explicitly expose planning-critical scene relations.

2.2 GEOMETRIC AND SPATIOTEMPORAL REPRESENTATIONS FOR VLM-BASED DRIVING

Another line injects explicit geometric and spatiotemporal representations into driving VLAs, supported by geometric foundation models such as DUSt3R, MASt3R, and VGGT that learn transferable 3D representations directly from images (Wang et al., 2024; Leroy et al., 2025; Wang et al., 2025) and efficient adaptation of pretrained 3D models (Liang et al., 2025). These methods encode 3D coordinates as continuous positional encodings instead of text tokens (Li et al., 2026b), inject cross-view geometric features from a frozen VGGT into the VLM (Wang et al., 2026a), jointly model occupancy, occupancy flow, language understanding, and planning (Liu et al., 2026c), or learn 4D representations from driving videos through reconstruction or teacher supervision and transfer them to downstream planning (Lu et al., 2025; Strong et al., 2026). Occupancy- and reconstruction-based world models capture explicit scene evolution with occupancy or Gaussian representations (Zuo et al., 2025; Diehl et al., 2025; Zhao et al., 2025; He et al., 2026). Other works unify 3D scene understanding with future scene generation (Zhou et al., 2025a; 2026a), jointly model future appearance and geometry (Liang et al., 2026), or transfer video-generation dynamics priors to action prediction (Zhao et al., 2026b). Despite the value of such cues, efficiently compressing dense pretrained 4D features while explicitly exposing compact planning-critical scene relations remains underexplored.

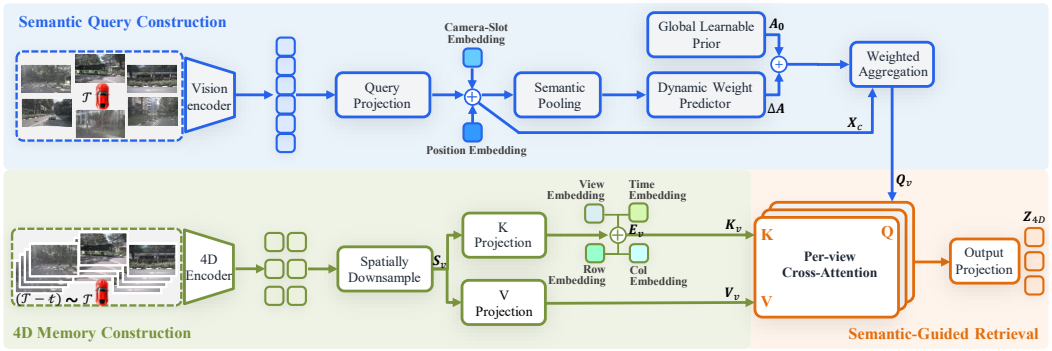


Figure 3: Semantic-Guided 4D Retriever (SG4R). Visual tokens from all cameras are aggregated into view-specific semantic queries. Downsampled 4D encoder features are projected into key and value memories with positional embeddings. Per-view cross-attention then retrieves compact 4D tokens.

3 METHOD

RECON4DRIVE augments a VLM with a frozen 4D encoder, which serves as a 4D world memory for the VLM. We instantiate this encoder with SLARM (Qiu et al., 2026), a representative feed-forward 4D reconstruction model for driving scenes, whose features encode both the metric geometry and temporal dynamics required for planning. As shown in Fig. 2, two key modules connect this world memory to planning. The Semantic-Guided 4D Retriever (SG4R) extracts a compact set of 4D tokens from the world memory, and Planning-Critical Structured Scene Encoding (PSE) exposes ego-centric lane and agent relations via supervised relation queries. A flow-matching DiT planner then decodes the trajectory from the last-layer VLM hidden states. At each time step $\mathcal{T}$, the model takes multi-view images with ego and navigation information and predicts a Meta-Action and future waypoints.

3.1 SEMANTIC-GUIDED 4D RETRIEVER

To reduce the dense 4D representation without losing planning-relevant cues, SG4R retrieves a compact set of 4D tokens guided by image semantics. As shown in Fig. 3, it consists of three steps: semantic query construction from VLM visual tokens, 4D world memory construction from a frozen 4D encoder, and semantic-guided retrieval from the world memory with the queries.

Semantic Query Construction. We take the N_q visual tokens of each camera from the Qwen3-VL vision encoder (Bai et al., 2025). They are projected to an attention dimension d and augmented with camera-slot and token-position embeddings, giving $\mathbf{X}_c \in \mathbb{R}^{N_q \times d}$ for camera c. The semantic queries $\mathbf{Q}_v$ of each of the N_v 4D views then aggregate all N_c cameras with scene-dependent weights:

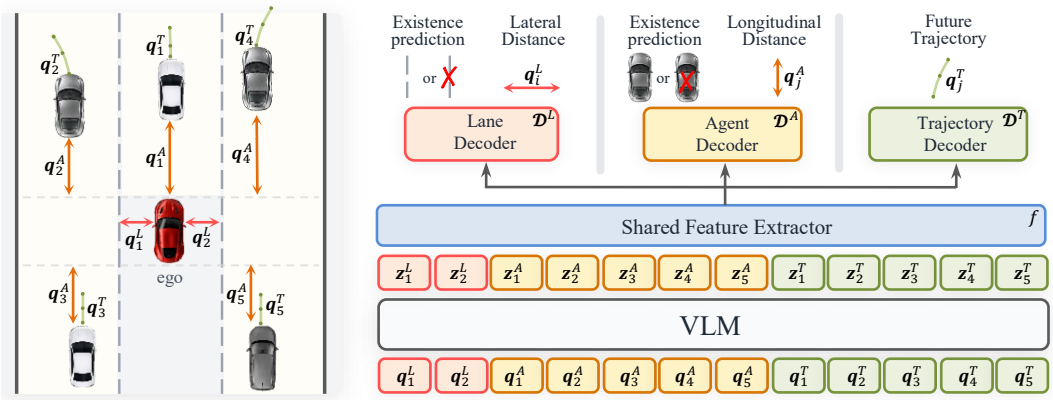
$$\mathbf{Q}_v = \sum_{c=1}^{N_c} \frac{\exp\big(a_{v,c} + \phi_v(\mathbf{X}_c)\big)}{\sum_{c'=1}^{N_c} \exp\big(a_{v,c'} + \phi_v(\mathbf{X}_{c'})\big)} \mathbf{X}_c, \quad v = 1, \ldots, N_v, \tag{1}$$

where the fraction is the camera weight of camera c for view v. The logits $a_{v,c}$ form the global learnable prior $\mathbf{A}_0$, which encodes the default view-to-camera assignment. The residual $\phi_v(\mathbf{X}_c)$, which forms $\Delta\mathbf{A}$, adapts this assignment to the scene: ϕ applies semantic pooling to the tokens of camera c and maps the result to N_v logits with the dynamic weight predictor.

4D World Memory Construction. While the VLM receives all surround-view cameras at $\mathcal{T}$, the 4D branch encodes only the N_v front-view cameras over $[\mathcal{T} - t, \mathcal{T}]$. For each view and frame, the frozen 4D encoder produces a dense feature grid, which we spatially downsample. Let $\mathbf{S}_v \in \mathbb{R}^{N_s \times C}$ stack the downsampled features of view v over its N_t frames and grid positions, one row per memory slot. Two learnable linear projections then map $\mathbf{S}_v$ to its key and value memories:

$$\mathbf{K}_v = \mathrm{LN}(\mathbf{S}_v)\,\mathbf{W}_K + \mathbf{E}_v, \quad \mathbf{V}_v = \mathrm{LN}(\mathbf{S}_v)\,\mathbf{W}_V, \tag{2}$$

where $\mathbf{W}_K, \mathbf{W}_V \in \mathbb{R}^{C \times d}$ are learnable projection matrices, LN is layer normalization, and $\mathbf{E}_v \in \mathbb{R}^{N_s \times d}$ sums learnable view, time, row, and column embeddings. Keys thus carry both content and position, whereas values carry only content.



(a) Structured Query Construction (b) Query Decoding and Supervision

Figure 4: Planning-critical structured scene encoding. (a) lane, agent, and trajectory queries for the current lane boundaries, the closest agent in each surrounding region, and their future trajectories. (b) a shared feature extractor and three decoders map their VLM outputs to existence predictions, distances, and future trajectories, which are supervised by simulator annotations during training.

Semantic-Guided Retrieval. The semantic queries $\mathbf{Q}_v$ of each view attend to its key and value memories via per-view cross-attention:

$$\mathbf{Z}_{4\mathrm{D}} = \left[\mathrm{MHA}(\mathbf{Q}_1, \mathbf{K}_1, \mathbf{V}_1);\ \ldots;\ \mathrm{MHA}(\mathbf{Q}_{N_v}, \mathbf{K}_{N_v}, \mathbf{V}_{N_v})\right] \in \mathbb{R}^{N_v N_q \times D}, \tag{3}$$

where MHA denotes multi-head cross-attention followed by an output projection from dimension d to the VLM hidden dimension D, and $[\cdot;\cdot]$ concatenates along the token dimension. Each query yields one 4D token, so the 4D tokens of each view match the visual tokens of a camera in number and dimension. The compact 4D tokens $\mathbf{Z}_{4\mathrm{D}}$ then enter the VLM input at placeholder positions alongside the visual tokens.

SG4R reads the frozen 4D world memory with image semantics, so the VLM receives only the geometry and motion cues selected by the image. Since the 4D token count is set by the queries rather than the world memory, the dense 4D representation is compressed into compact tokens.

3.2 Planning-Critical Structured Scene Encoding

Only a few ego-centric relations, such as those to lane boundaries and nearby agents, directly affect the current decision, yet they are only implicit in visual and 4D tokens. As shown in Fig. 4, PSE makes them explicit by assigning each relation a dedicated supervised query.

Planning-Critical Relation Construction. We construct relation targets from simulator annotations in the ego frame at $\mathcal{T}$, with the x-axis along the ego heading. We consider M lane relations, one for each boundary of the current lane, and M' agent relations, one for each surrounding region, such as ahead in the ego lane or behind in an adjacent lane. Each relation i has a candidate set $\mathcal{O}_i$, a distance function δ_i, and a range r_i. For a lane relation, $\mathcal{O}_i$ contains the boundary segments on one side of the current lane, and δ_i is the perpendicular distance from the ego center. For an agent relation, $\mathcal{O}_i$ contains the agents in the region, and $\delta_i(o) = |x_o|$ is the longitudinal offset of agent o along the ego heading. The target of relation i is its closest candidate, which is valid only within range:

$$o_i^\star = \arg\min_{o \in \mathcal{O}_i} \delta_i(o), \qquad d_i = \delta_i(o_i^\star), \qquad e_i = \mathbb{I}\big[d_i \le r_i\big], \tag{4}$$

where $\mathbb{I}[\cdot]$ is the indicator, and d_i and e_i are the distance and existence labels. For an agent relation, the trajectory label $\mathbf{U}_i$ contains the per-step displacements of agent $o_i^\star$ over H future steps. We write $e_i^{\mathrm{L}}, d_i^{\mathrm{L}}$ for lane and $e_i^{\mathrm{A}}, d_i^{\mathrm{A}}$ for agent relations; d_i and $\mathbf{U}_i$ are supervised only when $e_i = 1$.

Relation Queries. We append a set of learnable relation queries to the VLM input: M lane queries $\mathbf{q}_i^{\mathrm{L}}$, M' agent queries $\mathbf{q}_i^{\mathrm{A}}$, and M' trajectory queries $\mathbf{q}_i^{\mathrm{T}}$. Each query is bound to a fixed relation, so no matching is needed. Their VLM outputs $\mathbf{z}_i^{\mathrm{L}}$, $\mathbf{z}_i^{\mathrm{A}}$, and $\mathbf{z}_i^{\mathrm{T}}$ form the relation states $\mathbf{Z}_{\mathrm{rel}} \in \mathbb{R}^{(M+2M')\times D}$. These positions are excluded from the language-modeling loss, and the planner cross-attends to $\mathbf{Z}_{\mathrm{rel}}$ along with other VLM hidden states.

Relation Supervision. During training, a shared feature extractor f and a decoder $\mathcal{D}^k$ for each relation type k read the relation states:

$$\hat{\mathbf{y}}_i^k = \mathcal{D}^k\big(f(\mathbf{z}_i^k)\big), \quad k \in \{\mathrm{L}, \mathrm{A}, \mathrm{T}\}, \tag{5}$$

where $\hat{\mathbf{y}}_i^{\mathrm{L}} = (\hat{e}_i^{\mathrm{L}}, \hat{d}_i^{\mathrm{L}})$, $\hat{\mathbf{y}}_i^{\mathrm{A}} = (\hat{e}_i^{\mathrm{A}}, \hat{d}_i^{\mathrm{A}})$, and $\hat{\mathbf{y}}_i^{\mathrm{T}} = \hat{\mathbf{U}}_i$ are the existence predictions, distances, and future trajectories. Each decoder is shared by all queries of its type, so relations of the same type are read out in the same way. The training loss is

$$\mathcal{L}_{\mathrm{scene}} = \sum_{k\in\{\mathrm{L},\mathrm{A}\}} \Big(\operatorname*{mean}_i \mathrm{BCE}\big(\hat{e}_i^k, e_i^k\big) + \operatorname*{mean}_{i:\, e_i^k=1} \frac{|\hat{d}_i^k - d_i^k|}{r_i} \Big) + \operatorname*{mean}_{i:\, e_i^{\mathrm{A}}=1} \big\|\hat{\mathbf{U}}_i - \mathbf{U}_i\big\|_1, \tag{6}$$

where BCE is the binary cross-entropy, and the distance is normalized by the relation range.

PSE turns the planning-critical relations that are implicit in the visual and compact 4D tokens into explicit relation states. It adds only $M{+}2M'$ queries and models a few decision-relevant relations instead of the full scene, keeping the extra cost small.

3.3 Training and Inference

The VLM takes the visual tokens, compact 4D tokens, relation queries, and a prompt containing the current speed and the navigation command, and autoregressively generates a Meta-Action.

An adaLN-Zero DiT planner (Peebles & Xie, 2023) conditioned on the VLM hidden states and ego inputs then generates the trajectory with flow matching (Lipman et al., 2023), trained with an L_1 velocity loss $\mathcal{L}_{\mathrm{flow}}$. Following VAD (Jiang et al., 2023), a boundary term $\mathcal{L}_{\mathrm{bound}}$ penalizes waypoints too close to non-crossable road boundaries. All trainable modules are optimized jointly with

$$\mathcal{L} = \lambda_{\mathrm{text}}\,\mathcal{L}_{\mathrm{text}} + \lambda_{\mathrm{flow}}\,\mathcal{L}_{\mathrm{flow}} + \lambda_{\mathrm{scene}}\,\mathcal{L}_{\mathrm{scene}} + \lambda_{\mathrm{bound}}\,\mathcal{L}_{\mathrm{bound}}, \tag{7}$$

where $\mathcal{L}_{\mathrm{text}}$ is the cross-entropy over the Meta-Action tokens and λ are loss weights.

4 Experiments

4.1 Benchmark and Evaluation Metrics

Bench2Drive (Jia et al., 2024) is a CARLA-based benchmark for end-to-end autonomous driving across diverse interactive scenarios, towns, and weather conditions. Unlike purely open-loop benchmarks, it supports systematic closed-loop evaluation with sequential environment feedback, which better reflects actual driving capability in interactive settings. Following prior work (Li et al., 2025; Fu et al., 2025; 2026), we train our models on original Think2Drive-collected data released with Bench2Drive, including a 1,000-clip base set and a 13,638-clip full set. For consistency with prior methods in Tab. 1, we follow Bench2Drive protocol used before v0.0.4 update. For open-loop evaluation, we use the official 50 validation clips. For closed-loop evaluation, we use the corresponding 220 short routes covering 44 interactive scenarios, with five routes per scenario. We report Success Rate (SR), Driving Score (DS), Efficiency, and Comfortness for closed-loop evaluation, Multi-Ability for scenario-level driving capability, and average L2 error for open-loop trajectory prediction.

4.2 Implementation Details

We use Qwen3-VL-4B (Bai et al., 2025) as VLM backbone with a hidden dimension of $D{=}2560$. The VLM takes $N_c{=}6$ surround-view images at a resolution of 192×320. For the 4D branch, we keep the pretrained SLARM encoder frozen and process $N_v{=}3$ front-view cameras over $N_t{=}4$ frames sampled at 2 fps, with each image resized to 192×280. SG4R spatially downsamples SLARM representation from 10,080 to 2,592 tokens and further compresses it into 180 tokens using 60 visual queries per front-view camera, with an attention dimension of $d{=}512$ and 8 heads. PSE uses $M{=}2$ lane relations and $M'{=}5$ agent relations, giving $M{+}2M'{=}12$ relation queries. The eight-block DiT planner predicts 6 waypoints over a 3 s horizon with five sampling steps. The loss weights in Eq. 7 are set to $\lambda_{\mathrm{text}}{=}\lambda_{\mathrm{flow}}{=}1$, $\lambda_{\mathrm{scene}}{=}0.1$, and $\lambda_{\mathrm{bound}}{=}0.2$. Final model is trained on official Full split for 100 epochs using 256 NPUs. Base learning rate is 1×10^{-5}, with a vision-tower learning rate of 2×10^{-6} and learning rates of 1×10^{-3} for planner and relation decoders. Per-device batch size is 6. The main closed-loop results use full 4,000-step setting. To reduce the cost of ablations, ablation models are trained on official Base split for 120 epochs with 128 NPUs and evaluated with 800 steps.

Table 1: Closed-loop, Multi-Ability, and Open-loop comparison on Bench2Drive. "–" denotes results not reported in the original paper, and bold indicates the best result for each metric. † denotes training with both camera and LiDAR inputs; all other methods use camera-only training.

Method	Venue	Closed-loop				Multi-Ability	Open-loop
		SR (%) ↑	DS ↑	Efficiency ↑	Comfortness ↑	Mean (%) ↑	Avg. L2 (m) ↓
End-to-End Driving Methods							
Hydra-NeXt (Li et al., 2025)	ICCV25	50.00	73.86	197.76	20.68	53.22	0.92
Raw2Drive (Yang et al., 2025b)	NeurIPS25	50.24	71.36	**214.17**	22.42	53.34	–
RAP-ResNet (Feng et al., 2026)	ICLR26	37.27	66.42	165.47	23.63	–	–
DriveMamba (Su et al., 2026)	ICLR26	37.73	66.82	152.91	18.77	–	0.70
VADv2 (Jiang et al., 2026)	ICLR26	50.46	76.15	178.24	**37.81**	–	–
CausalVAD (Tang et al., 2026)	CVPR26	19.42	49.83	–	–	–	–
SafeDrive† (Kim et al., 2026)	CVPR26	42.40	66.77	–	–	–	–
GuideFlow (Liu et al., 2026a)	CVPR26	51.36	75.21	–	–	–	–
VLA-based Driving Methods							
ORION (Fu et al., 2025)	ICCV25	54.62	77.74	151.48	17.38	54.72	0.68
ReCogDrive (Li et al., 2026c)	ICLR26	45.45	71.36	138.18	17.45	42.03	–
MindDriver (Zhang et al., 2026)	CVPR26	39.55	65.48	143.21	34.63	–	–
SGDrive (Li et al., 2026a)	CVPR26	51.36	75.47	–	–	–	–
CogDriver (Liu et al., 2026b)	CVPR26	56.93	78.21	169.52	20.50	–	0.63
MindDrive (Fu et al., 2026)	ECCV26	58.26	80.59	–	–	60.81	0.66
Recon4Drive (Ours)	–	**62.27**	**85.45**	179.64	16.14	**63.68**	**0.58**

4.3 Main Results

For a fair comparison, Tab. 1 primarily includes recent state-of-the-art methods trained on the official Think2Drive-collected data. We exclude methods trained on data collected by different privileged experts, such as PDM-Lite (Renz et al., 2025) and LEAD (Nguyen et al., 2026), since differences in expert policies and data-collection pipelines may introduce additional factors beyond the driving model itself. SR and DS are the most consistently reported closed-loop metrics across existing methods, and we therefore treat them as the primary indicators of driving performance. For completeness, we also report all available Efficiency, Comfortness, Multi-Ability, and Open-loop trajectory metrics from the original papers. Efficiency and Comfortness follow the official Bench2Drive evaluation protocol. Entries marked with "–" indicate metrics that are not reported.

As shown in Tab. 1, Recon4Drive achieves the strongest closed-loop performance among all compared methods, reaching an SR of 62.27% and a DS of 85.45. This corresponds to absolute gains of 4.01 points in SR and 4.86 points in DS over MindDrive, the strongest prior VLA-based method in this setting. The advantage is even larger over end-to-end methods, with gains of 10.91 points in SR and 9.30 points in DS over their best reported results. Since both metrics improve consistently, the gain extends beyond route success to the overall quality of closed-loop driving. Beyond closed-loop evaluation, Recon4Drive achieves the highest Multi-Ability mean score, with the largest gain in overtaking, which relies on accurate geometry and motion understanding (see Tab. 6 in the appendix for details). In open-loop evaluation, Recon4Drive also achieves the lowest average L2 error of 0.58 m in Tab. 1. Taken together, these results show consistent gains across all three evaluation dimensions, supporting the effectiveness of compact 4D representations and planning-critical relations for VLA driving.

4.4 Ablation Studies and Analysis

Component Ablation. Tab. 2 presents controlled ablations of the pretrained 4D representation, SG4R, and PSE. Starting from the image-only baseline, introducing pretrained 4D features improves SR from 45.00% to 50.91% and DS from 70.89 to 74.43, confirming the benefit of explicit geometry and motion. However, directly injecting the dense 4D representation, even after spatial downsampling, increases VLM latency from 157 ms to 268 ms. SG4R largely removes this overhead, reducing the latency to 161 ms, close to the image-only baseline, while preserving most of the closed-loop gains from 4D features. PSE further helps the VLM make use of 4D information. Without SG4R, adding PSE improves SR from 50.91% to 52.27% and Multi-Ability from 53.84 to 59.37. The main gains appear in interaction-intensive abilities such as overtaking and give-way, which rely heavily on

Table 2: Ablation of the 4D representation, SG4R, and PSE.

ID	Inputs		Modules		Closed-loop		VLM latency	Ability (%) ↑					
	Image feature	4D feature	SG4R	PSE	SR (%) ↑	DS ↑	↓ (ms)	**Mean**	Over taking	Merging	Emerg. Brake	Give way	Traffic Sign
1	✓	–	–	–	45.00	70.89	157	52.03	55.56	36.25	48.33	70.00	50.00
2	✓	✓	–	–	50.91	74.43	268 (+71%)	53.84	55.56	38.75	63.33	60.00	51.58
3	✓	✓	✓	–	50.45	72.16	161 (+3%)	54.62	64.44	33.75	63.33	60.00	51.58
4	✓	✓	–	✓	52.27	74.86	268 (+71%)	59.37	68.89	33.75	60.00	80.00	54.21
5	✓	✓	✓	✓	**53.64**	**75.40**	162 (+3%)	**63.13**	64.44	40.00	63.33	90.00	57.89

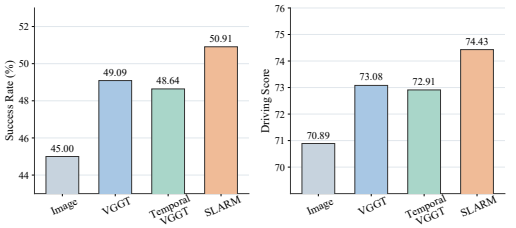


Figure 5: Comparison of scene representations on closed-loop driving performance.

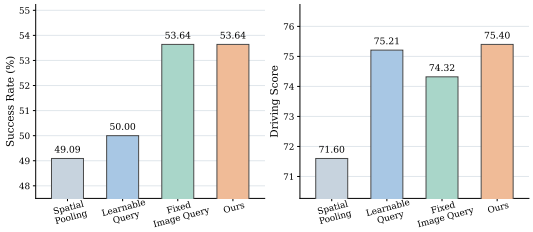


Figure 6: Comparison of 4D token compression strategies on closed-loop driving performance.

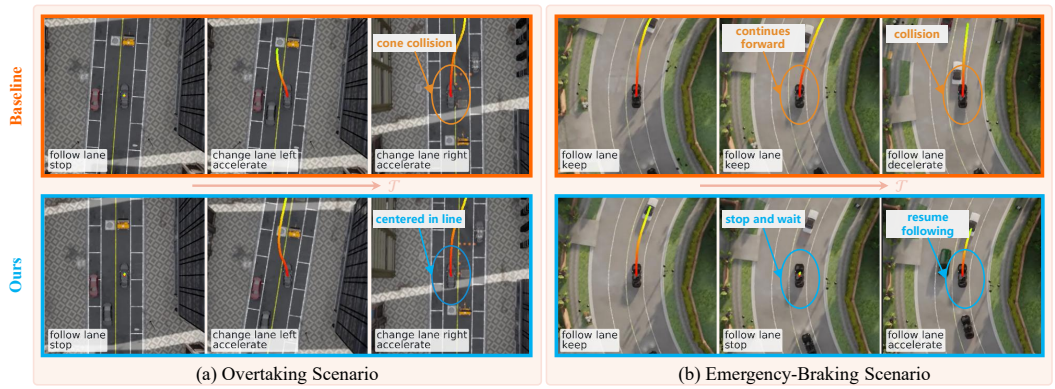
understanding surrounding geometry, motion, and interactions. This indicates that PSE makes planning-critical 4D information easier for the VLM to exploit.

Combining SG4R and PSE yields the best overall performance. Compared with SG4R alone, adding PSE improves SR by 3.19 points, DS by 3.24 points, and Multi-Ability by 8.51 points with only 1 ms additional latency. These results show that SG4R and PSE are mutually reinforcing. SG4R removes redundant 4D information to form a compact representation, while PSE guides SG4R to retain planning-critical geometry and motion cues and exposes them as explicit relations that the VLA can directly use for planning. Together, the two modules substantially outperform either one alone with negligible additional latency.

Choice of Scene Representation. Fig. 5 compares different scene representations without SG4R or PSE. Adding VGGT (Wang et al., 2025) features improves the image-only baseline, showing the benefit of explicit geometric cues. To separate the effect of temporal input from explicit dynamic modeling, we construct Temporal VGGT, which uses exactly the same multi-frame, multi-view input as SLARM and feeds the resulting features, concatenated with the image features, directly into the VLM. Despite the same temporal observations, Temporal VGGT performs slightly worse than single-frame VGGT, whereas SLARM performs best. Simply adding temporal observations is therefore insufficient; the advantage of 4D representations comes from explicitly modeling scene dynamics, which is particularly important for interaction-aware driving.

Ablation of 4D Token Compression Strategies. We compare four compression strategies that all compress the 4D features to the same number of tokens, with PSE enabled for all variants. *Spatial Pooling* directly aggregates the dense 4D features. *Learnable Query* retrieves 4D features using scene-independent learned queries, while *Fixed Image Query* directly uses semantically pooled image features as queries, without the $\mathbf{A}_0$ and $\Delta\mathbf{A}$ components in Fig. 3. *Ours* uses the full SG4R design.

Spatial Pooling performs the worst, showing that uniform aggregation is insufficient to preserve driving-relevant 4D information. Learnable Query substantially improves DS from 71.60 to 75.21, but yields only a modest SR gain from 49.09% to 50.00%. In contrast, Fixed Image Query raises SR to 53.64%, demonstrating the benefit of scene-specific semantic guidance, but its DS decreases to 74.32. These complementary results support the SG4R design, which combines general retrieval patterns with scene-adaptive guidance: $\mathbf{A}_0$ captures shared aggregation patterns, while $\Delta\mathbf{A}$ adapts them to the current driving context. The full design keeps the highest SR of 53.64% while improving DS to 75.40, achieving the best balance between the two closed-loop metrics.



(a) Overtaking Scenario (b) Emergency-Braking Scenario

Figure 7: Qualitative comparison of the baseline (top) and our model (bottom) in (a) overtaking and (b) emergency braking scenarios. The baseline excludes SG4R and PSE, and the lower-left text shows the predicted Meta-Action.

Ablation of Relation Queries. Tab. 3 studies how different relation queries affect closed-loop driving. Adding lane queries increases DS from 72.16 to 74.46, while SR decreases from 50.45% to 48.18%. This may stem from the stronger structural priors of explicit lane relations, which encourage regular lane following but may reduce flexibility in atypical or interaction-intensive situations. Adding agent queries further improves DS to 74.95 and partially recovers SR to 49.09%, suggesting that explicit awareness of surrounding agents alleviates this limitation but static relations alone remain insufficient. Finally, trajectory queries provide dynamic context about how surrounding interactions evolve, substantially increasing SR to 53.64% while further improving DS to 75.40. Static geometric and interaction priors thus improve driving quality, while dynamic motion cues are key to adapting them to evolving situations.

Table 3: Ablation of different relation queries in PSE.

Relation Queries			Closed-loop	
Lane	Agent	Trajectory	SR(%)↑	DS↑
–	–	–	50.45	72.16
✓	–	–	48.18	74.46
✓	✓	–	49.09	74.95
✓	✓	✓	53.64	75.40

4.5 Qualitative Analysis

Fig. 7 shows two representative challenging cases: (a) overtaking and (b) emergency braking. In Fig. 7(a), the ego vehicle must move into the left lane to bypass an obstacle and then return to its original lane while avoiding traffic cones near the right boundary. The baseline stays too close to the right boundary during the return maneuver and collides with the cones, whereas our model remains better centered and completes the maneuver safely. In Fig. 7(b), the lead vehicle suddenly decelerates during lane following. The baseline responds too weakly and collides with it, while our model stops in time and resumes following once the lead vehicle moves again. These examples are consistent with the complementary roles of SG4R and PSE: compact 4D cues provide geometry and motion information, while explicit lane, agent, and trajectory relations make planning-critical information directly accessible to the planner.

5 Conclusion

We presented a VLA driving framework that incorporates pretrained 4D reconstruction representations to complement image features with explicit geometric and dynamic information. To this end, we introduce SG4R to efficiently retrieve useful 4D features and PSE to explicitly represent key lane, agent, and motion cues with relation queries. Experiments on Bench2Drive show that this design consistently improves closed-loop driving performance while substantially reducing the computational overhead introduced by dense 4D features. Moreover, our analysis suggests that the advantage of 4D representations does not simply come from adding temporal observations, but from effectively modeling dynamic information that directly affects driving decisions. Overall, these results suggest that pretrained 4D reconstruction models can serve as reusable geometric and dynamic priors for VLA-based driving, providing a practical way to bridge reconstruction-oriented representations and downstream decision making.

REPRODUCIBILITY STATEMENT

Sec. 3 specifies all components of RECON4DRIVE: the formulation of SG4R (Eqs. 1–3), the construction and supervision of planning-critical relations in PSE (Eqs. 4–6), and the overall training objective (Eq. 7). Sec. 4 and App. A provide the remaining details needed to reproduce our results, including the model inputs and prompt template (App. A.1), the hyperparameters of SG4R, PSE, and the planner (App. A.2–A.4), the baselines and ablation variants (App. A.5), and the training and evaluation protocols (App. A.6). All experiments use the public Bench2Drive benchmark and its official training data (Jia et al., 2024), and build on the pretrained Qwen3-VL (Bai et al., 2025) and SLARM (Qiu et al., 2026) models. We will release the code upon publication.

AI USE STATEMENT

Large language models were used only to polish the writing, such as improving grammar, wording, and readability of text drafted by the authors. They were not used to design the method, run experiments, analyze results, or produce any reported numbers. The authors wrote or verified all technical claims, equations, figures, and results, and take full responsibility for this paper.

REFERENCES

Niket Agarwal, Arslan Ali, Maciej Bala, Yogesh Balaji, Erik Barker, Tiffany Cai, Prithvijit Chattopadhyay, Yongxin Chen, Yin Cui, Yifan Ding, et al. Cosmos world foundation model platform for physical ai. *arXiv preprint arXiv:2501.03575*, 2025.

Niket Agarwal, Arslan Ali, Jon Allen, Martin Antolini, Adeline Aubame, Alisson Azzolini, Junjie Bai, Maciej Bala, Yogesh Balaji, Josh Bapst, et al. Cosmos 3: Omnimodal world models for physical ai. *arXiv preprint arXiv:2606.02800*, 2026.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Christopher Diehl, Quinlan Sykora, Ben Agro, Thomas Gilles, Sergio Casas, and Raquel Urtasun. Dio: Decomposable implicit 4d occupancy-flow world model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 27456–27466, 2025.

Heng Fang, Shangru Li, Shuhan Wang, Xuanyang Xi, Dingkang Liang, and Xiang Bai. Towards Generalizable Robotic Manipulation in Dynamic Environments. In Paolo Favaro, Zuzana Kukelova, Atsuto Maki, Anna Rohrbach, Konrad Schindler, and Federico Tombari (eds.), *Proceedings of the European Conference on Computer Vision*, pp. 314–333, Cham, 2026. Springer Nature Switzerland. ISBN 978-3-032-37592-6. doi: 10.1007/978-3-032-37592-6_18.

Lan Feng, Yang Gao, Eloi Zablocki, Quanyi Li, Wuyang Li, Sichao Liu, Matthieu Cord, and Alexandre Alahi. Rap: 3d rasterization augmented end-to-end planning. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 37722–37737, 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. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 24823–24834, 2025.

Haoyu Fu, Diankun Zhang, Zongchuang Zhao, Jianfeng Cui, Hongwei Xie, Bing Wang, Guang Chen, Hangjun Ye, Dingkang Liang, and Xiang Bai. MindDrive: A Vision-Language-Action Model for Autonomous Driving via Online Reinforcement Learning. In Paolo Favaro, Zuzana Kukelova, Atsuto Maki, Anna Rohrbach, Konrad Schindler, and Federico Tombari (eds.), *Proceedings of the European Conference on Computer Vision*, pp. 503–522, Cham, 2026. Springer Nature Switzerland. ISBN 978-3-032-37026-6. doi: 10.1007/978-3-032-37026-6_28.

Zhuolin He, Jing Li, Guanghao Li, Xiaolei Chen, Jiacheng Tang, Siyang Zhang, Zhounan Jin, Feipeng Cai, Bin Li, Jian Pu, Jia Cai, and Xiangyang Xue. Dynamicvggt: Learning dynamic point maps for 4d scene reconstruction in autonomous driving. *arXiv preprint arXiv:2603.08254*, 2026.

Xiaosong Jia, Zhenjie Yang, Qifeng Li, Zhiyuan Zhang, and Junchi Yan. Bench2drive: Towards multi-ability benchmarking of closed-loop end-to-end autonomous driving. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 819–844. Curran Associates, Inc., 2024. doi: 10.52202/079017-0025.

Xiaosong Jia, Yuqian Shao, Zhenjie Yang, Qifeng Li, Zhiyuan Zhang, and Junchi Yan. Bench2drive-vl: Benchmarks for closed-loop autonomous driving with vision-language models. *arXiv preprint arXiv:2604.01259*, 2026.

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

Bo Jiang, Shaoyu Chen, Hao Gao, Bencheng Liao, Qian Zhang, Wenyu Liu, and Xinggang Wang. Vadv2: End-to-end vectorized autonomous driving via probabilistic planning. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 68886–68900, 2026.

Jungho Kim, Jiyong Oh, Seunghoon Yu, Hongjae Shin, Donghyuk Kwak, and Jun Won Choi. Safedrive: Fine-grained safety reasoning for end-to-end driving in a sparse world. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24854–24864, 2026.

Vincent Leroy, Yohann Cabon, and Jerome Revaud. Grounding image matching in 3d with mast3r. In *Proceedings of the European Conference on Computer Vision*, pp. 71–91, 2025. doi: 10.1007/978-3-031-73220-1_5.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4032–4042, 2026a.

Peizheng Li, Zhenghao Zhang, David Holtz, Hang Yu, Yutong Yang, Yuzhi Lai, Rui Song, Andreas Geiger, and Andreas Zell. Spacedrive: Infusing spatial awareness into vlm-based autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 40096–40107, 2026b.

Yongkang Li, Kaixin Xiong, Xiangyu Guo, Fang Li, Sixu Yan, Gangwei Xu, Lijun Zhou, Long Chen, Haiyang Sun, Bing Wang, Kun Ma, Guang Chen, Hangjun Ye, Wenyu Liu, and Xinggang Wang. Recogdrive: A reinforced cognitive framework for end-to-end autonomous driving. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 157518–157556, 2026c.

Zhenxin Li, Shihao Wang, Shiyi Lan, Zhiding Yu, Zuxuan Wu, and Jose M. Alvarez. Hydra-next: Robust closed-loop driving with open-loop training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 27305–27314, 2025.

Dingkang Liang, Tianrui Feng, Xin Zhou, Yumeng Zhang, Zhikang Zou, and Xiang Bai. Parameter-efficient fine-tuning in spectral domain for point cloud learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(12):10949–10966, 2025. doi: 10.1109/TPAMI.2025.3594749.

Dingkang Liang, Dingyuan Zhang, Xin Zhou, Sifan Tu, Tianrui Feng, Xiaofan Li, Yumeng Zhang, Mingyang Du, Xiao Tan, and Xiang Bai. Unifuture: A 4d driving world model for future generation and perception. In *IEEE International Conference on Robotics and Automation*, 2026.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023.

Lin Liu, Caiyan Jia, Guanyi Yu, Ziying Song, Junqiao Li, Feiyang Jia, Peiliang Wu, Xiaoshuai Hao, and Yadan Luo. Guideflow: Constraint-guided flow matching for planning in end-to-end autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3719–3728, 2026a.

Pei Liu, Qingtian Ning, Xinyan Lu, Haipeng Liu, Weiliang Ma, Dangen She, Xianpeng Lang, and Jun Ma. Cogdriver: Integrating cognitive inertia for temporally coherent planning in autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18150–18160, 2026b.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3688–3698, 2026c.

Hao Lu, Tianshuo Xu, Wenzhao Zheng, Yunpeng Zhang, Wei Zhan, Dalong Du, Masayoshi Tomizuka, Kurt Keutzer, and Yingcong Chen. Drivingrecon: Large 4d gaussian reconstruction model for autonomous driving. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 167933–167952. Curran Associates, Inc., 2025. doi: 10.52202/085713-5599.

Long Nguyen, Micha Fauth, Bernhard Jaeger, Daniel Dauner, Maximilian Igl, Andreas Geiger, and Kashyap Chitta. Lead: Minimizing learner-expert asymmetry in end-to-end driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 39775–39785, 2026.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4195–4205, 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*, pp. 17809–17819, 2026.

Zhicheng Qiu, Jiarui Meng, Tong-an Luo, Yican Huang, Xuan Feng, Xuanfu Li, and Zhan Xu. Slarm: Streaming and language-aligned reconstruction model for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 29023–29034, June 2026.

Katrin Renz, Long Chen, Elahe Arani, and Oleg Sinavski. Simlingo: Vision-only closed-loop autonomous driving with language-action alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11993–12003, 2025.

Hao Shao, Yuxuan Hu, Letian Wang, Guanglu Song, Steven L. Waslander, Yu Liu, and Hongsheng Li. Lmdrive: Closed-loop end-to-end driving with large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15120–15130, 2024.

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 *Proceedings of the European Conference on Computer Vision*, pp. 256–274, 2025. doi: 10.1007/978-3-031-72943-0_15.

Matthew Strong, Wei-Jer Chang, Quentin Herau, Jiezhi Yang, Yihan Hu, Chensheng Peng, and Wei Zhan. Learning to drive is a free gift: Large-scale label-free autonomy pretraining from unposed in-the-wild videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 32144–32153, 2026.

Haisheng Su, Wei Wu, Feixiang Song, Junjie Zhang, Zhenjie Yang, and Junchi Yan. Drivemamba: Task-centric scalable state space model for efficient end-to-end autonomous driving. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 152680–152698, 2026.

Jiacheng Tang, Zhiyuan Zhou, Zhuolin He, Jia Zhang, Kai Zhang, and Jian Pu. Causalvad: Deconfounding end-to-end autonomous driving via causal intervention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 32124–32133, 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. In *Proceedings of the Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 4698–4726. PMLR, 2025.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5294–5306, 2025.

Jie Wang, Guang Li, Zhijian Huang, Chenxu Dang, Hangjun Ye, Yahong Han, and Long Chen. Vgdrive: Empowering vision-language models with cross-view geometric grounding for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10954–10964, 2026a.

Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20697–20709, 2024.

Xinyang Wang, Qian Liu, Wenjie Ding, Zhao Yang, Wei Li, Chang Liu, Bailin Li, Kun Zhan, Xianpeng Lang, and Wei Chen. Unifying language-action understanding and generation for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 25193–25203, 2026b.

Chengen Xie, Bin Sun, Tianyu Li, Junjie Wu, Zhihui Hao, XianPeng Lang, and Hongyang Li. Latentvla: Efficient vision-language models for autonomous driving via latent action prediction. *arXiv preprint arXiv:2601.05611*, 2026a.

Hengyi Xie, Chenfei Yao, Xianjin Wu, Xuanyang Xi, Yiping Tang, Di Xu, Yingying Zhu, Dingkang Liang, Xiang Bai, and Han Ding. Turbovla: Real-time vision-language-action model at 32 hz on an RTX 4090 with $<$1 GB VRAM. *arXiv preprint arXiv:2607.27205*, 2026b.

Zhenhua Xu, Yujia Zhang, Enze Xie, Zhen Zhao, Yong Guo, Kwan-Yee K. Wong, Zhenguo Li, and Hengshuang Zhao. Drivegpt4: Interpretable end-to-end autonomous driving via large language model. *IEEE Robotics and Automation Letters*, 9(10):8186–8193, 2024. doi: 10.1109/LRA.2024.3440097.

Jiawei Yang, Jiahui Huang, Boris Ivanovic, Yuxiao Chen, Yan Wang, Boyi Li, Yurong You, Apoorva Sharma, Maximilian Igl, Peter Karkus, Danfei Xu, Yue Wang, and Marco Pavone. Storm: Spatio-temporal reconstruction model for large-scale outdoor scenes. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 50446–50465, 2025a.

Zhenjie Yang, Xiaosong Jia, Qifeng Li, Xue Yang, Maoqing Yao, and Junchi Yan. Raw2drive: Reinforcement learning with aligned world models for end-to-end autonomous driving (in carla v2). In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 134122–134147. Curran Associates, Inc., 2025b. doi: 10.52202/085713-4471.

Zhenjie Yang, Yilin Chai, Xiaosong Jia, Qifeng Li, Yuqian Shao, Xuekai Zhu, Haisheng Su, and Junchi Yan. Drivemoe: Mixture-of-experts for vision-language-action model in end-to-end autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10678–10688, 2026.

Lingjun Zhang, Yujian Yuan, Changjie Wu, Xinyuan Chang, Xin Cai, Shuang Zeng, Linzhe Shi, Sijin Wang, Hang Zhang, and Mu Xu. Minddriver: Introducing progressive multimodal reasoning for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17831–17841, 2026.

Guosheng Zhao, Chaojun Ni, Xiaofeng Wang, Zheng Zhu, Xueyang Zhang, Yida Wang, Guan Huang, Xinze Chen, Boyuan Wang, Youyi Zhang, Wenjun Mei, and Xingang Wang. Drivedreamer4d: World models are effective data machines for 4d driving scene representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12015–12026, 2025.

Zongchuang Zhao, Haoyu Fu, Dingkang Liang, Xin Zhou, Dingyuan Zhang, Hongwei Xie, Bing Wang, and Xiang Bai. Extending large vision–language model for diverse interactive tasks in autonomous driving. *IEEE Transactions on Image Processing*, 35:8558–8573, 2026a. doi: 10.1109/TIP.2026.3719473.

Zongchuang Zhao, Xin Zhou, Tianyang Xu, Zhengyang Sun, Kaixuan Zhou, Yu Wu, Honglin Li, Dingkang Liang, and Xiang Bai. Simwam: A simple world action model for end-to-end autonomous driving. *arXiv preprint arXiv:2608.07468*, 2026b.

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

Xin Zhou, Dingkang Liang, Xiwu Chen, Feiyang Tan, Dingyuan Zhang, Hengshuang Zhao, and Xiang Bai. Hermes++: Toward a unified driving world model for 3d scene understanding and generation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–18, 2026a. doi: 10.1109/TPAMI.2026.3735606.

Xin Zhou, Zongchuang Zhao, Zhibo Yang, Mingsheng Li, Humen Zhong, Shuai Bai, Du Chu, Ruizhe Chen, Zhaohai Li, Jun Tang, Qiuyue Wang, Mingkun Yang, Jiazhao Zhang, Dayiheng Liu, Dingkang Liang, and Xiang Bai. Qwen-drive-1.0: An initial step towards a vision-language foundation model for autonomous driving. *arXiv preprint arXiv:2609.00111*, 2026b.

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 *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 13782–13790, 2026c. doi: 10.1609/aaai.v40i16.38386.

Zewei Zhou, Tianhui Cai, Seth 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. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 27920–27956. Curran Associates, Inc., 2025b. doi: 10.52202/085713-0942.

Sicheng Zuo, Wenzhao Zheng, Yuanhui Huang, Jie Zhou, and Jiwen Lu. Gaussianworld: Gaussian world model for streaming 3d occupancy prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6772–6781, 2025.

RECON4DRIVE: SELECTIVE AND EXPLICIT 4D PRIORS FOR VLA DRIVING — APPENDIX

This appendix complements the main paper with implementation details, additional quantitative and qualitative results, and a discussion of limitations. It is organized as follows.

A – Implementation Details: architecture, training, and evaluation details.
B – Additional Quantitative Results: Multi-Ability comparison, ablation values, and efficiency.
C – Additional Qualitative Results: visualizations of relation states.
D – Limitations.

A IMPLEMENTATION DETAILS

A.1 MODEL INPUTS

4D Representation. The 4D branch uses the front-left, front, and front-right cameras. For each view and frame, SLARM outputs a 24×35 feature grid with C=1,536 channels, giving $3\times4\times24\times35$=10,080 native tokens. The dense baseline, termed *direct concatenation*, reduces each grid to 12×18 by 2×2 merging with padding and projects the 2,592 tokens to dimension D.

Text Prompt. Besides the visual tokens and the compact 4D tokens, the VLM receives a text prompt with the current speed and the navigation command, followed by the relation queries (Tab. 4).

Table 4: Input template of the VLM. `<image>` and `<slarm>` stand for the visual tokens and the compact 4D tokens of each view, `<lane>`, `<agent>`, and `<traj>` stand for the relation queries, and the italic fields are filled in for each sample.

`<FRONT VIEW>: <image> <FRONT LEFT VIEW>: <image> ... <BACK VIEW>: <image>` *(six cameras)* `<SLARM FRONT VIEW>: <slarm> ... <SLARM FRONT RIGHT VIEW>: <slarm>` *(three 4D views)*
`Your current speed is {`*speed*`} m/s, the navigation command is {`*command*`}. Based on the surrounding scene and navigation command, choose a safe driving direction from [follow lane, change lane left, change lane right, go straight, turn left, turn right] and a speed action from [keep, accelerate, decelerate, stop].`
`<lane>×2 <agent>×5 <traj>×5` *(12 relation queries)*

Ego Inputs of the Planner. The planner additionally receives the 7-D ego status, which consists of the speed, longitudinal and lateral accelerations, yaw rate, vehicle length and width, and steering-wheel angle, and the past ego trajectory over the last 1 s at 2 Hz.

A.2 SEMANTIC-GUIDED 4D RETRIEVER

The N_q=60 visual tokens of each camera form a 6×10 grid. The global learnable prior $\mathbf{A}_0$ is initialized to a fixed correspondence that assigns each 4D view the front and back cameras of the matching side (left, center, or right). The output layer of the dynamic weight predictor is zero-initialized, so $\Delta\mathbf{A}=\mathbf{0}$ at the start of training. The 4D world memory is obtained by parameter-free adaptive average pooling from 24×35 to 12×18, which gives N_s=864 slots per view.

A.3 PLANNING-CRITICAL STRUCTURED SCENE ENCODING

The M=2 lane relations are the left and right boundaries of the current lane, and the M'=5 agent regions are ahead in the ego lane and ahead and behind in each adjacent lane. Agents include vehicles, pedestrians, and obstacles. The range r_i is 4 m for each lane boundary, 80 m for each region ahead, and 40 m for each region behind. Each trajectory target has H=6 steps at 0.5 s intervals, and steps at which the agent is no longer annotated are masked. The trajectory loss is evaluated only over valid future steps. The decoders in Eq. 5 are used only for training and are not run at inference.

A.4 Planner and Training Objective

Meta-Action. The Meta-Action combines one of six directions and one of four speed actions in Tab. 4, giving 24 entries. Its labels follow the rule-based B2D-VL annotations (Jia et al., 2026).

Planner. Each of the eight DiT blocks, with a hidden width of 1,024, applies self-attention over the six waypoint tokens and one cross-attention layer for each of the four context groups: the visual tokens, the compact 4D tokens, the Meta-Action tokens, and the relation states. The flow timestep and the ego status modulate each block through adaLN-Zero. Flow matching uses the linear interpolant $\mathbf{x}_\tau = (1-\tau)\,\boldsymbol{\epsilon}+\tau\,\mathbf{x}_1$ between Gaussian noise $\boldsymbol{\epsilon}$ and the normalized ground-truth waypoint increments $\mathbf{x}_1$, with the flow time τ drawn from a shifted $\mathrm{Beta}(1.5, 1.0)$ distribution. At inference, five Euler steps integrate the velocity field, and the cumulative sum of the predicted increments gives the trajectory.

A.5 Baselines and Ablation Variants

All variants follow the ablation protocol and differ only in the components described below.

Component Ablation. In Tab. 2, ID 2 is the direct concatenation baseline (App. A.1), which also serves as the baseline in Fig. 7. The PSE design study in Tab. 3 starts from ID 3 and adds the lane, agent, and trajectory queries in turn.

Scene Representation Study. The VGGT variant uses the patch tokens of the last aggregator layer of a frozen VGGT (Wang et al., 2025) on the three front-view images, 280 tokens per image, and projects them in the same way as the SLARM features in direct concatenation.

Compression Study. All variants keep PSE, read the same 12×18 memory, and output 60 tokens per view. *Spatial pooling* uses no queries. It average-pools each value-projected 12×18 frame to 3×5 and keeps all four frames, giving $4\times3\times5{=}60$ tokens. *Learnable query* replaces the semantic queries with 60 learnable queries per view. *Fixed image query* averages the semantic queries of the front and back cameras of the matching side, without $\mathbf{A}_0$ or $\Delta\mathbf{A}$. *SG4R* corresponds to ID 5 in Tab. 2.

A.6 Training and Evaluation Protocols

Tab. 5 summarizes the two protocols. We use AdamW without weight decay, cosine learning-rate decay with a 3% warmup, gradient clipping at a norm of 1.0, and BF16 precision. SG4R shares the learning rate of the VLM. In closed-loop evaluation, a route that is unfinished after `TickRunTime` simulator steps counts as a failure, so scores under the two protocols are not directly comparable.

Table 5: Training and evaluation protocols of the final model and the ablations.

	Final model (Tab. 1)	**Ablations** (Sec. 4)
Training split	Full (13,638 clips)	Base (1,000 clips)
Epochs	100	120
Number of training NPUs	256	128
Global batch size	1,536	768
Learning rate (VLM, SG4R / vision tower)	1×10^{-5} / 2×10^{-6}	1×10^{-5} / 2×10^{-6}
Learning rate (planner / relation decoders)	1×10^{-3}	1×10^{-3}
Closed-loop evaluation steps	4,000	800

A.7 Metrics

Success Rate (SR) is the percentage of routes completed within the time limit without any infraction. **Driving Score (DS)** averages, over all routes, the route completion multiplied by an infraction penalty that decreases multiplicatively with each infraction. **Efficiency** measures the ego speed relative to

surrounding vehicles, and **Comfortness** measures the smoothness of the executed trajectory. **Multi-Ability** reports the success rate on the route subsets of five abilities and their mean. The open-loop L2 error is averaged over the first 2 s of the predicted waypoints.

B Additional Quantitative Results

B.1 Multi-Ability Comparison

Tab. 6 lists per-ability results of the methods in Tab. 1 that report them. Recon4Drive achieves the highest mean, 2.87 points above MindDrive, with its largest gain in Overtaking (91.11% versus 71.11% for ORION). It trails the best baselines in Merging, Emergency Brake, and Traffic Sign.

Table 6: Multi-Ability results (%) on Bench2Drive for the methods in Tab. 1 that report this metric.

Method	Merging	Overtaking	Emerg. Brake	Give Way	Traffic Sign	**Mean**
Hydra-NeXt (Li et al., 2025)	40.00	64.44	61.67	**50.00**	50.00	53.22
Raw2Drive (Yang et al., 2025b)	**43.35**	51.11	60.00	**50.00**	62.26	53.34
ORION (Fu et al., 2025)	25.00	71.11	**78.33**	30.00	69.15	54.72
ReCogDrive (Li et al., 2026c)	29.73	20.00	69.09	20.00	**71.34**	42.03
MindDrive (Fu et al., 2026)	42.31	66.67	76.67	**50.00**	68.42	60.81
Recon4Drive (Ours)	41.25	**91.11**	75.00	**50.00**	61.05	**63.68**

B.2 Scene Representation and 4D Compression

Tab. 7 and Tab. 8 report two additional studies under the ablation protocol. Tab. 7 compares scene representations that enter the VLM by direct concatenation, without SG4R or PSE (App. A.5). All geometric representations improve over the image-only baseline, and SLARM achieves the highest SR and DS. Tab. 8 compares strategies that compress the 4D memory into the same number of tokens, with PSE kept in all variants. SG4R achieves the highest DS and ties for the highest SR.

Table 7: Scene representations.

Representation	SR (%)	DS
Image only	45.00	70.89
+ VGGT	49.09	73.08
+ Temporal VGGT	48.64	72.91
+ SLARM	**50.91**	**74.43**

Table 8: 4D compression strategies.

Compression	SR (%)	DS
Spatial pooling	49.09	71.60
Learnable query	50.00	75.21
Fixed image query	**53.64**	74.32
SG4R (ours)	**53.64**	**75.40**

B.3 Efficiency and Token Budget

Tab. 9 lists the number of 4D tokens at each stage, and Tab. 10 reports the VLM computation, VLM latency, and peak memory of the two 4D interfaces. SG4R reduces the 4D tokens fed to the VLM by 98.2% relative to the native SLARM token count and by 93.1% relative to the dense reference. All efficiency numbers are measured with a batch size of 1 and averaged over 500 runs. GFLOPs and latency cover only the VLM, while peak memory covers the whole model.

C Additional Qualitative Results

Fig. 8 decodes the relation states of PSE with the training-time decoders in Eq. 5 and shows them in a top-down view for two scenes. The ego vehicle is shown in cyan, and the white bars on its two sides reflect the predicted lateral distances to the left and right boundaries of the current lane. The red dots mark the closest agents predicted by the agent queries, and the orange curves show the future trajectories predicted by the trajectory queries, with the ground-truth trajectories in green. Lane geometry is used to place agents laterally for visualization, while PSE predicts their longitudinal

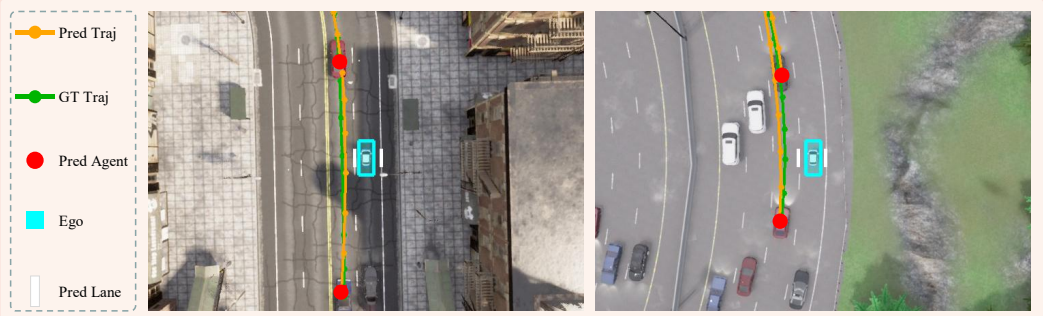
Table 9: 4D token budget. Reductions are relative to the native and dense representations.

Stage	Layout	4D tokens
Native SLARM representation	3×4×24×35	10,080
Dense reference (direct concatenation)	3×4×12×18	2,592
SG4R memory (per view, not fed to the VLM)	4×12×18	864
SG4R output (fed to the VLM)	3×60	**180** (↓98.2% / ↓93.1%)
For reference: visual tokens / relation queries		360 / 12

Table 10: VLM efficiency of the two 4D interfaces. Both start from the same native SLARM representation, and percentages denote reductions relative to direct concatenation.

4D interface	4D tokens	GFLOPs ↓	VLM latency (ms) ↓	Peak memory (MB) ↓
Direct concatenation	2,592	27,466	267.8	13,765
SG4R (ours)	**180** (↓93.06%)	**6,191** (↓77.46%)	**161.6** (↓39.66%)	**11,195** (↓18.67%)

distances. Scene (a) is an urban road, where the agent relations capture the vehicles ahead of and behind the ego vehicle in the adjacent lane. Scene (b) is a curved multi-lane road, where the predicted trajectories follow the road curvature and stay close to the ground truth. These examples illustrate the lane and agent information that the relation states expose to the planner.



(a) Urban Road (b) Curved Multi-Lane Road

Figure 8: Visualization of relation states, where lateral lane-boundary distances and longitudinal agent distances are implicitly reflected by the top-down layout, together with predicted future trajectories.

D LIMITATIONS

RECON4DRIVE relies on a frozen pretrained 4D reconstruction model, SLARM (Qiu et al., 2026), so the quality of its 4D cues is bounded by this encoder, and estimation errors under challenging conditions may carry over to planning. In addition, the relation targets of PSE are built from simulator annotations, and all experiments are conducted in simulation. Evaluating other 4D encoders and extending RECON4DRIVE to real-world driving are natural next steps.