RCVLA: 4D RADAR-GROUNDED SEMANTIC REASONING AND TRAJECTORY ARBITRATION FOR AUTONOMOUS DRIVING

Anonymous authors
Paper under double-blind review

ABSTRACT

4D radar provides geometric and motion cues that complement visual semantics, but integrating it into vision-language-action (VLA) models requires both radar–language alignment for semantic reasoning and explicit use of radar measurements for trajectory refinement and selection. To support these capabilities, we construct Cap4DR with 86,016 radar-image-text samples for alignment pretraining and OmniHD-QA with 520,161 question-answer pairs for instruction tuning across scene description, key-object reasoning, occupancy understanding, and trajectory planning. Building on these datasets, we propose RCVLA, a radar-camera VLA framework consisting of a radar-grounded semantic reasoning stage (RCVLA-Sem) and a trajectory arbitration stage (RCVLA-Phys). RCVLA-Sem performs gated bidirectional interaction between camera and radar tokens for driving question answering and reference trajectory generation, while auxiliary heads provide object and occupancy queries. RCVLA-Phys refines reference-guided trajectory candidates through truncated diffusion conditioned on these queries and cluster-level radar measurements, then calibrates candidate scores using radar-derived time-to-collision risk. On OmniHD-QA, RCVLA-Sem improves CIDEr by 9.92 points and reduces key-object velocity error by 21.9% relative to OmniDrive. RCVLA-Phys further reduces average L2 error from 0.348 to $0.259\,\mathrm{m}$ and average open-loop collision rate from 0.576% to 0.175% relative to RCVLA-Sem. Ablation studies further show that language-aligned radar tokens improve semantic reasoning, while cluster-level radar measurements and risk calibration improve trajectory arbitration. Code will be released.

1 INTRODUCTION

Recent vision-language-action (VLA) models for autonomous driving couple language-based scene reasoning with driving decisions and motion planning (Sima et al., 2024; Wang et al., 2025; Jiang et al., 2026). However, camera-centric VLA models infer geometry and motion primarily from visual observations, which can become unreliable under degraded visibility (Bijelic et al., 2020). 4D radar supplies spatial measurements and Doppler-derived radial velocity that complement visual semantics (Palffy et al., 2022; Zhang et al., 2023; Choi et al., 2023).

Integrating 4D radar into VLA models raises two challenges. First, standard image–text pretraining does not provide 4D radar–language alignment, motivating dedicated alignment supervision and driving instruction data. Second, semantic reasoning and trajectory planning require different 4D radar representations. Compact language-aligned tokens provide an interface for scene reasoning (Li et al., 2023), while trajectory refinement and selection rely on local geometric and motion measurements. We therefore retain cluster-level radar measurements to guide candidate refinement and risk-aware selection, which together constitute *trajectory arbitration*.

To address these challenges, we construct two datasets and develop **RCVLA**. **Cap4DR** contains 86,016 4D radar-image-text samples for radar–language alignment pretraining, with 4D radar relevant entities verified through multi-model cross checking and selective manual review. **OmniHD-QA** contains 520,161 question-answer pairs derived from OmniHD-Scenes (Zheng et al., 2026b), covering scene description, key-object reasoning, occupancy understanding, and trajectory planning

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.

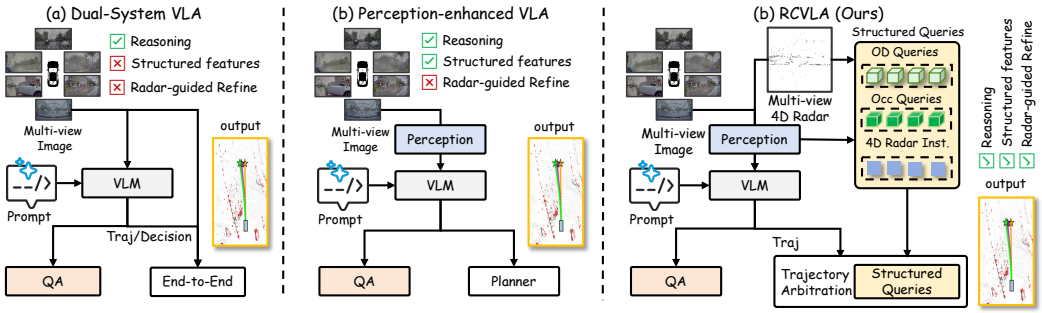


Figure 1: **Architectural comparison.** (a) Dual-system VLA links language reasoning to a separate end-to-end model. (b) Perception-enhanced VLA incorporates structured scene features. (c) RCVLA combines a semantic reference, structured scene queries, and 4D radar measurements for trajectory refinement and risk-aware selection.

for instruction tuning and evaluation. Building on these datasets, RCVLA integrates radar-grounded semantic reasoning with trajectory arbitration. As illustrated in Figure 1, we compare three VLA designs. Dual-system VLA couples language reasoning with a separate end-to-end model, while perception-enhanced VLA further incorporates structured scene features. RCVLA instead combines a reference trajectory from semantic reasoning, structured scene queries, and direct 4D radar measurements for trajectory refinement and risk-aware selection.

Specifically, **RCVLA-Sem** performs gated bidirectional interaction between camera and 4D radar tokens for driving question answering and reference trajectory generation, while auxiliary object and occupancy heads provide structured scene queries. **RCVLA-Phys** constructs a compact candidate set from the reference trajectory and anchors, then refines it through truncated diffusion conditioned on these queries and cluster-level radar measurements. The retained measurements further yield time-to-collision (TTC)-based risk estimates that calibrate candidate scores for risk-aware selection. Our contributions are summarized below.

- We construct **Cap4DR** and **OmniHD-QA**, providing complementary supervision for radar–language alignment pretraining, multitask driving instruction tuning, and evaluation.
- We propose **RCVLA**, a radar-camera VLA framework that uses 4D radar in complementary roles for semantic reasoning and trajectory arbitration. RCVLA-Sem integrates camera and radar tokens for driving question answering and reference trajectory generation, while RCVLA-Phys refines trajectory candidates using structured scene queries and cluster-level radar measurements, with radar-derived TTC risk guiding candidate selection.
- Experiments on **OmniHD-QA** demonstrate improvements in semantic reasoning and open-loop planning over the evaluated baselines. Ablations further validate the complementary roles of language-aligned radar tokens in scene reasoning and preserved physical measurements in trajectory refinement and risk-aware selection.

2 Related Work

2.1 Planning-Oriented End-to-End Driving

End-to-end driving predicts driving actions or future trajectories directly from sensor observations within a unified framework. ST-P3 integrates perception, prediction, and planning with a shared spatio-temporal representation (Hu et al., 2022b), while UniAD coordinates multiple driving tasks through a unified query interface (Hu et al., 2023). BEV-Planner studies the role of ego-state information in open-loop planning (Li et al., 2024), while DiffusionDrive uses trajectory anchors and truncated diffusion to generate diverse future trajectories (Liao et al., 2025). These methods connect structured scene perception with motion planning, but are mainly designed for predefined driving tasks and offer limited scene-level reasoning.

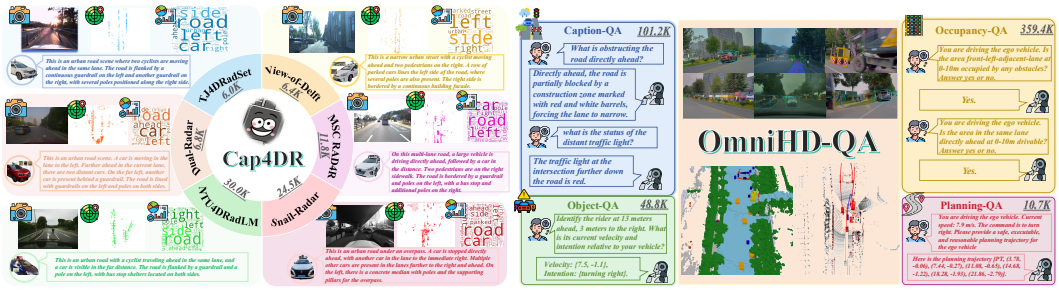


Figure 2: **Dataset overview.** Left: Cap4DR combines samples from six radar datasets with radar-oriented descriptions for alignment pretraining. Right: OmniHD-QA covers scene description, key-object reasoning, 3D occupancy, and trajectory planning for instruction tuning and evaluation.

2.2 Vision-Language-Action Models for Driving

Vision-language models (VLMs) have shown strong capabilities in visual understanding and reasoning (Liu et al., 2023; Chen et al., 2024). Recent methods extend these capabilities to scene reasoning and motion planning. DriveLM organizes perception, prediction, and planning through graph-based question answering (Sima et al., 2024). OmniDrive combines multi-view 3D scene understanding with trajectory generation (Wang et al., 2025), while Senna uses a large vision-language model to support decision making within an end-to-end driving framework (Jiang et al., 2026). These methods bring language reasoning into autonomous driving, but still rely mainly on visual observations, so geometric and motion information is inferred rather than directly measured..

2.3 4D Radar Perception and Language Grounding

Recent studies connect 4D radar perception with language models. WaterVG and NanoMVG combine language, camera, and 4D radar inputs for grounding in waterway scenes (Guan et al., 2025a;b). Talk2Radar grounds referring expressions in 4D radar point clouds (Guan et al., 2025c), while Text4Radar-V2X introduces language guidance into cooperative 4D radar-based 3D detection (Peng et al., 2026). However, these studies mainly use 4D radar for grounding and detection, while its physical measurements have not been fully exploited for scene understanding and downstream trajectory planning.

3 Datasets

Existing image–text datasets are largely centered on visual inputs, with little consideration of 4D radar observations. We therefore construct Cap4DR for radar–language pretraining and OmniHD-QA for multitask instruction tuning and evaluation.

Cap4DR. Cap4DR integrates synchronized front-view images and 4D radar data from View-of-Delft (Palffy et al., 2022), TJ4DRadSet (Zheng et al., 2022), NTU4DRadLM (Zhang et al., 2023), MSC-RAD4R (Choi et al., 2023), SNAIL Radar (Huai et al., 2025), and Dual Radar (Zhang et al., 2025). We first apply temporal alignment and unified radar preprocessing across datasets. Rather than using generic image captions directly, we retain only objects and spatial relations that can be associated with radar observations, turning visual descriptions into radar-relevant language supervision. The extracted entities are cross-checked by three VLMs, with ambiguous cases manually reviewed. Our Cap4DR dataset contains 86,016 radar–text pairs for radar–language alignment pretraining.

OmniHD-QA. OmniHD-QA is constructed from OmniHD-Scenes (Zheng et al., 2026b) and contains 520,161 question–answer pairs covering scene description, key-object reasoning, occupancy understanding, and trajectory planning. We use Gemini2.5-Pro (Comanici et al., 2025) to generate scene-description QA from multi-view images with available annotations. The remaining tasks are constructed from structured annotations, including object tracks, semantic occupancy, and ego pose. OmniHD-QA therefore provides unified instruction supervision for driving-specific scene rea-

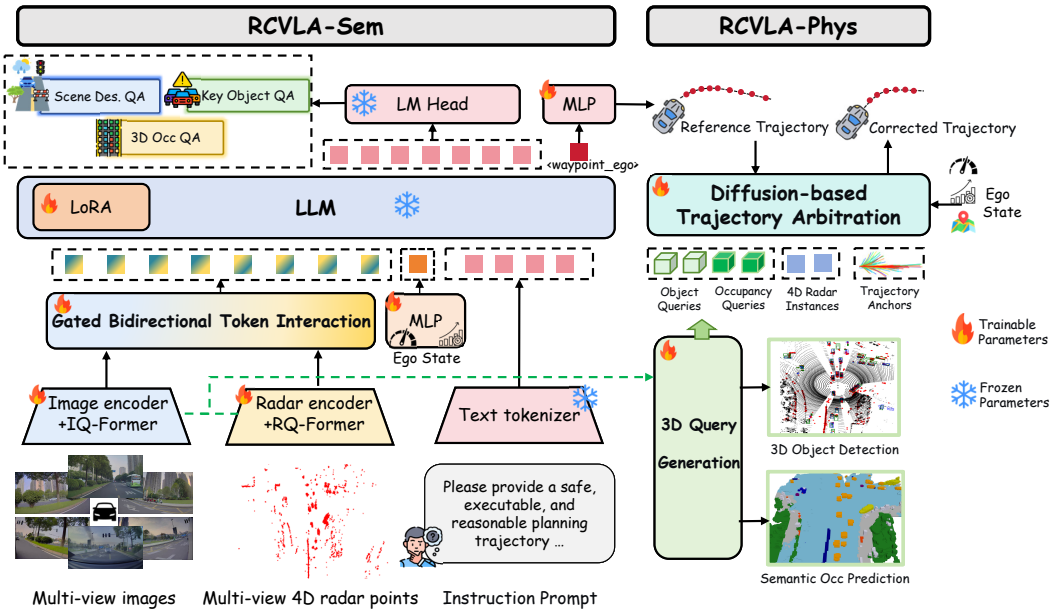


Figure 3: **RCVLA architecture.** RCVLA-Sem integrates camera and radar tokens for driving question answering and long-horizon reference planning, while providing structured object and occupancy queries. RCVLA-Phys combines the reference trajectory, structured queries, and cluster-level radar measurements for trajectory refinement and TTC-aware selection.

soning and planning. Detailed construction procedures and dataset statistics for both Cap4DR and OmniHD-QA are provided in Appendix B.

4 Method

4.1 Overview

As shown in Figure 3, RCVLA consists of two stages: radar-grounded semantic reasoning (**RCVLA-Sem**) and trajectory arbitration (**RCVLA-Phys**). In RCVLA-Sem, multi-view image and 4D radar points are converted into fixed-length tokens and enhanced through gated bidirectional interaction. The enhanced camera and radar tokens are fed into the LLM with the prompt and ego features for driving question answering or long-horizon reference planning, while the encoded sensor features produce object and occupancy queries for subsequent trajectory arbitration. RCVLA-Phys combines the reference trajectory with selected anchors to form candidate priors, refines them through truncated diffusion conditioned on structured scene queries and cluster-level radar measurements, and calibrates their scores with radar-derived time-to-collision (TTC) risk for final selection, as shown in Figure 4.

4.2 Radar-Grounded Semantic Reasoning

Modality tokens. We use EVA-02-L (Fang et al., 2024) to encode multi-view images into features $\mathbf{F}_{\mathrm{I}}$, and RadarPillarNet (Zheng et al., 2023) to extract 4D radar BEV features $\mathbf{F}_{\mathrm{R}}$. The image features are organized into a multi-view sequence with view and camera-aware 3D positional encodings, while the radar BEV features are converted into a compact spatial sequence with 2D positional encodings, yielding $\mathbf{M}_{\mathrm{I}}$ and $\mathbf{M}_{\mathrm{R}}$. Following the query-based interfaces of BLIP-2 and OmniDrive (Li et al., 2023; Wang et al., 2025), independent IQ-Former and RQ-Former modules use learnable queries $\mathbf{Q}_m^0$ to extract fixed-length language-aligned tokens:

$$\mathbf{T}_m = \mathrm{Proj}_m\big(\mathrm{QFormer}_m(\mathbf{Q}_m^0, \mathbf{M}_m)\big), \qquad m \in \{\mathrm{I}, \mathrm{R}\}. \tag{1}$$

where $\mathrm{QFormer}_m$ denotes IQ-Former or RQ-Former, and Proj_m maps its query outputs to the LLM hidden space, yielding $\mathbf{T}_{\mathrm{I}}$ and $\mathbf{T}_{\mathrm{R}}$.

Gated bidirectional interaction. Direct interaction in the high-dimensional LLM space is computationally expensive, while sparse and noisy radar observations can degrade cross-modal interaction. We therefore project the language-aligned tokens into a compact interaction space and apply token-wise gating to the radar features:

$$\mathbf{X}_m = \mathrm{Down}_m(\mathrm{LN}(\mathbf{T}_m)) \,. \tag{2}$$

$$\mathbf{g}_{\mathrm{R}} = \sigma(\mathrm{MLP}_{\mathrm{g}}(\mathrm{LN}(\mathbf{X}_{\mathrm{R}}))) \,, \qquad \widehat{\mathbf{X}}_{\mathrm{R}} = \mathbf{g}_{\mathrm{R}} \odot \mathbf{X}_{\mathrm{R}}. \tag{3}$$

where $\mathrm{Down}_m(\cdot)$ denotes a modality-specific linear projection, $\sigma(\cdot)$ is the sigmoid function, and $\mathbf{g}_{\mathrm{R}}$ contains a scalar gate for each radar token.

The image branch attends to the gated radar features $\widehat{\mathbf{X}}_{\mathrm{R}}$, while the radar branch attends to the image features $\mathbf{X}_{\mathrm{I}}$. The two branches use separate cross-attention modules, followed by residual and FFN updates, yielding $\widetilde{\mathbf{X}}_{\mathrm{I}}$ and $\widetilde{\mathbf{X}}_{\mathrm{R}}$. We retain the original language-aligned tokens and inject only the residual introduced by cross-modal interaction:

$$\mathbf{T}'_m = \mathbf{T}_m + \alpha_m \,\mathrm{Up}_m \left(\widetilde{\mathbf{X}}_m - \mathbf{X}_m \right). \tag{4}$$

where $\mathrm{Up}_m(\cdot)$ projects the interaction residual back to the LLM hidden space, and α_m is a learnable scalar initialized to a small positive value.

Language, reference trajectory, and structured queries. The enhanced image and radar tokens are combined with the prompt embeddings and ego features and fed into a LoRA-adapted LLM (Hu et al., 2022a). For QA tasks, the model generates language responses autoregressively. For planning prompts, we introduce a special token `<waypoint_ego>` and pass its hidden state through an MLP to directly regress a continuous long-horizon reference trajectory of 12 waypoints over the next 6 s.

In parallel, two lightweight transformer decoders operate on the shared image and radar features to produce object queries $\mathbf{Q}^{\mathrm{OD}}$ and occupancy queries $\mathbf{Q}^{\mathrm{OCC}}$. The object branch uses learnable 3D reference points, while the occupancy branch uses a regular 3D grid. Both aggregate image and radar BEV features through deformable attention and are supervised by 3D detection and semantic occupancy, respectively. The resulting queries provide structured conditions for RCVLA-Phys.

4.3 Trajectory Arbitration

Reference-guided query initialization. We align the long-horizon reference trajectory from RCVLA-Sem to the current ego frame and retain its next 3 s segment as a candidate prior. Additional priors are retrieved from an offline trajectory-anchor bank to preserve both reference consistency and motion diversity. Following truncated diffusion (Liao et al., 2025), each candidate prior $\boldsymbol{\mu}_m$ is perturbed as

$$\mathbf{x}_m^i = \sqrt{\bar{\alpha}_i}\,\boldsymbol{\mu}_m + \sqrt{1-\bar{\alpha}_i}\,\boldsymbol{\epsilon}_m. \tag{5}$$

where $\boldsymbol{\epsilon}_m \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ denotes Gaussian noise and $\bar{\alpha}_i$ is the cumulative diffusion coefficient. Each noisy prior is encoded by an MLP and combined with ego-state and candidate-index embeddings to initialize the trajectory queries, which are then passed to the diffusion decoder for iterative refinement.

Structured diffusion decoder. The decoder refines trajectory queries using occupancy, object, and radar conditions, which respectively provide free-space and obstacle constraints, instance-level object representations, and direct measurements of position and motion. Radar clusters are obtained from 4D radar points using DBSCAN (Schubert et al., 2017). Feature adapters project each condition into the trajectory-query feature dimension.

At decoder layer ℓ, the trajectory queries interact with each condition through cross-attention, and the attended context is incorporated through a gated residual update:

$$\mathbf{C}_k^{\ell} = \mathrm{MCA}_k\big(\mathrm{LN}(\mathbf{Q}_{k,\mathrm{in}}^{\ell}), \mathbf{M}_k, \mathbf{M}_k\big) \,, \tag{6}$$

$$\mathbf{g}_k^{\ell} = \sigma\big(\mathrm{MLP}_k[\mathbf{Q}_{k,\mathrm{in}}^{\ell}; \mathbf{C}_k^{\ell}]\big) \,, \tag{7}$$

$$\mathbf{Q}_{k,\mathrm{out}}^{\ell} = \mathrm{LN}\big(\mathbf{Q}_{k,\mathrm{in}}^{\ell} + \mathbf{g}_k^{\ell} \odot \mathbf{C}_k^{\ell}\big) \,. \tag{8}$$

where $k \in \{\mathrm{OCC}, \mathrm{OD}, \mathrm{R}\}$ denotes the condition type, $\mathbf{Q}_{k,\mathrm{in}}^{\ell}$ and $\mathbf{Q}_{k,\mathrm{out}}^{\ell}$ are the trajectory queries before and after interaction with condition k, and $\mathbf{M}_k$ contains the corresponding adapted condition

features. $\mathrm{MCA}_k(\cdot,\cdot,\cdot)$ denotes multi-head cross-attention. The resulting condition context $\mathbf{C}_k^\ell$ is modulated by the query-conditioned gate $\mathbf{g}_k^\ell$ before being incorporated through the residual update. The operators $\sigma(\cdot)$ and $\odot$ denote the sigmoid function and elementwise multiplication, respectively. After the condition interactions, the trajectory queries are updated by an FFN with a residual connection, yielding $\widetilde{\mathbf{Q}}^\ell$. The diffusion timestep then modulates these queries through FiLM:

$$\mathbf{Q}^\ell = \widetilde{\mathbf{Q}}^\ell \odot (\mathbf{1} + \boldsymbol{\gamma}_i^\ell) + \boldsymbol{\beta}_i^\ell, \tag{9}$$

where $\widetilde{\mathbf{Q}}^\ell$ denotes the trajectory queries after the conditional interactions and FFN update, and $\boldsymbol{\gamma}_i^\ell$ and $\boldsymbol{\beta}_i^\ell$ are the scale and bias generated from the diffusion-timestep embedding. A regression head and a scoring head then predict a denoised candidate trajectory $\widehat{\boldsymbol{\tau}}_m$ and its score s_m, respectively. The regression head directly predicts a coordinate correction to the noisy trajectory rather than noise. At inference, the final-layer decoder predictions update the candidates at the next DDIM timestep.

Radar-based risk calibration. In addition to trajectory refinement, 4D radar clusters also provide physical cues for risk-aware candidate selection. For radar cluster j, let $\mathbf{p}_j$ and $\mathbf{v}_j$ denote its planar center position and compensated radial-velocity vector, respectively. Assuming constant velocity, its center at planning step h is predicted as $\mathbf{p}_{j,h} = \mathbf{p}_j + t_h \mathbf{v}_j$, where $t_h = h\Delta t$ and $\Delta t = 0.5\,\mathrm{s}$. For the m-th candidate trajectory, let $\mathbf{e}_{m,h}$ and $\mathbf{v}_{m,h}$ denote the ego position and velocity at step h, and let $d_{m,j,h}$ denote the minimum planar distance between the ego and radar-cluster footprints, with $d_{m,j,h} = 0$ for overlap. The closing speed is

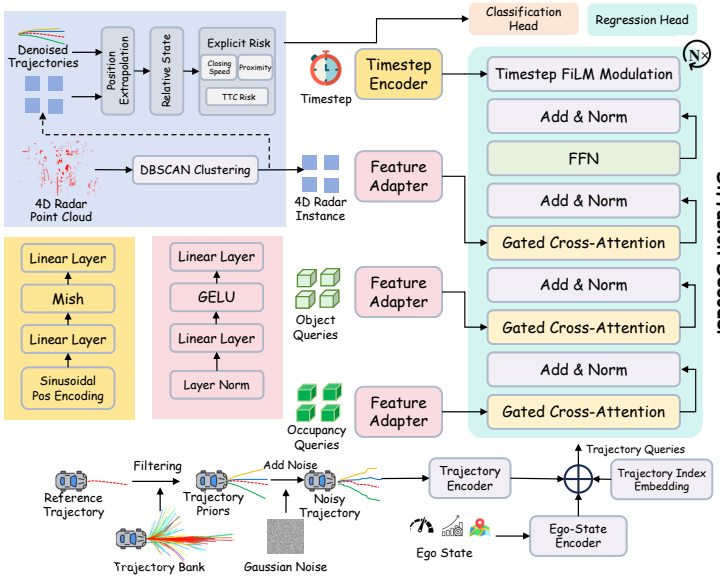


Figure 4: **Trajectory arbitration.**

$$c_{m,j,h} = \max\left((\mathbf{v}_{m,h} - \mathbf{v}_j)^\top \frac{\mathbf{p}_{j,h} - \mathbf{e}_{m,h}}{\|\mathbf{p}_{j,h} - \mathbf{e}_{m,h}\|_2 + \varepsilon}, 0 \right). \tag{10}$$

where ε is a small constant for numerical stability. The corresponding TTC is

$$\mathrm{TTC}_{m,j,h} = \begin{cases} t_h, & d_{m,j,h} = 0, \\ t_h + d_{m,j,h}/c_{m,j,h}, & d_{m,j,h} > 0,\ c_{m,j,h} > 0. \end{cases} \tag{11}$$

The minimum TTC over all clusters and waypoints is converted into candidate risk:

$$r_m = \begin{cases} \exp\left(-\mathrm{TTC}_m^{\min}/\tau_{\mathrm{TTC}}\right), & \mathrm{TTC}_m^{\min} \le T_{\mathrm{h}}, \\ 0, & \text{otherwise}, \end{cases} \tag{12}$$

where $\mathrm{TTC}_m^{\min} = \min_{j,h} \mathrm{TTC}_{m,j,h}$ and T_{h} is the planning horizon. A smaller TTC yields a larger risk. During training, r_m is used to construct a risk-aware soft ranking target, while at inference it calibrates the candidate scores before final selection.

4.4 Training Objectives

RCVLA is trained in three stages. First, the radar encoder is pretrained with 3D detection and semantic occupancy supervision. We then freeze the radar encoder and the LLM and train only RQ-Former on Cap4DR with masked autoregressive language modeling for radar–language alignment.

Second, we initialize the image and radar encoders, IQ-Former, and RQ-Former from pretrained weights and fine-tune RCVLA-Sem on OmniHD-QA. The base LLM remains frozen, while its LoRA adapters are trainable. The objective jointly supervises language generation, reference planning, and structured scene queries:

$$\mathcal{L}_{\text{sem}} = \mathcal{L}_{\text{LM}} + \mathcal{L}_{\text{wp}} + \lambda_{\text{Det}}\mathcal{L}_{\text{Det}} + \lambda_{\text{OCC}}\mathcal{L}_{\text{OCC}}, \tag{13}$$

where $\mathcal{L}_{\text{LM}}$ is the next-token prediction loss and $\mathcal{L}_{\text{wp}}$ is an L_1 loss. The detection and occupancy losses, $\mathcal{L}_{\text{Det}}$ and $\mathcal{L}_{\text{OCC}}$, are defined following (Zheng et al., 2026a).

Third, we freeze RCVLA-Sem and train RCVLA-Phys for trajectory refinement and risk-aware candidate ranking. For M denoised candidates, let $\mathbf{d}$, $\mathbf{r}$, and $\mathbf{s}$ denote their average displacement errors to the ground-truth trajectory $\boldsymbol{\tau}^*$, radar risk, and predicted logits, respectively. The candidate with the smallest trajectory error $m^* = \arg\min_m d_m$, is selected for regression. For candidate ranking, we construct a risk-aware soft target $\boldsymbol{\pi}$ and calibrate the predicted distribution $\mathbf{p}$ as

$$\boldsymbol{\pi} = \text{softmax}\left(-\frac{\mathbf{d} + \lambda_{\text{r}}\mathbf{r}}{T_{\text{c}}}\right), \tag{14}$$

$$\mathbf{p} = \text{softmax}(\mathbf{s} - \lambda_{\text{s}}\mathbf{r}), \tag{15}$$

where λ_{r} and λ_{s} control the contribution of radar risk to the soft target and predicted logits, respectively, and T_{c} is the temperature parameter. The overall objective jointly optimizes trajectory regression and candidate ranking:

$$\mathcal{L}_{\text{phys}} = \lambda_{\text{reg}}\,\text{SmoothL1}(\widehat{\boldsymbol{\tau}}_{m^*}, \boldsymbol{\tau}^*) - \sum_{m=1}^{M} \pi_m \log p_m, \tag{16}$$

where $\widehat{\boldsymbol{\tau}}_{m^*}$ is the selected denoised trajectory and λ_{reg} balances the regression term. The regression candidate is determined solely by trajectory error, while radar risk affects only candidate ranking and score calibration.

5 EXPERIMENTS

Dataset and evaluation. Cap4DR is used only for RQ-Former pretraining, while all downstream experiments are conducted on OmniHD-QA, with 358,185 training and 161,976 test samples. We evaluate scene description with standard caption metrics, key-object reasoning with velocity error and intention accuracy, occupancy reasoning with accuracy, and trajectory planning with open-loop L2 error and collision rate. Baselines include LLaVA-v1.5-7B (Liu et al., 2023), Qwen2.5-VL-7B-Instruct (Bai et al., 2025b), Qwen3-VL-8B-Instruct (Bai et al., 2025a), InternVL2.5-8B (Chen et al., 2024), InternVL3-8B (Zhu et al., 2025), OmniDrive (Wang et al., 2025), Constant-Velo, BEV-Planner (Li et al., 2024), and UniAD (Hu et al., 2023). The general VLM baselines perform zero-shot inference, while OmniDrive and UniAD are trained without map supervision. Further dataset and metric details are provided in Appendix C.

Table 1: Scene description performance. B-n: BLEU-n; R-L: ROUGE-L; MET.: METEOR. Higher is better.

Method	B-1	B-2	B-3	B-4	R-L	CIDEr	MET.
LLaVA-v1.5	14.50	8.08	4.90	2.91	19.53	7.25	19.31
Qwen2.5-VL	28.66	15.34	9.00	5.29	22.57	22.00	30.36
Qwen3-VL	30.53	17.31	10.74	6.78	24.35	24.38	33.57
InternVL2.5	20.28	11.79	7.37	4.55	23.01	15.17	24.92
InternVL3	26.85	14.82	8.82	5.28	23.05	20.45	28.28
OmniDrive	39.80	28.48	21.90	17.38	36.98	92.49	43.13
RCVLA-Sem	**41.82**	**30.22**	**23.40**	**18.69**	**38.28**	**102.41**	**44.91**

Table 2: Occupancy and key-object reasoning performance. Occ., Dyn., Sta., and Drv. are class accuracies; A_{occ} and A_{int} are aggregate accuracies. All values are percentages except V_{err} (m/s).

Method	Occ.↑	Dyn.↑	Sta.↑	Drv.↑	A_{occ} ↑	V_{err} ↓	A_{int} ↑
LLaVA-v1.5	45.21	37.06	56.55	43.88	44.25	11.04	16.09
Qwen2.5-VL	47.89	25.48	56.56	15.17	35.00	6.16	**75.63**
Qwen3-VL	49.37	43.76	58.06	44.55	47.78	8.71	64.55
InternVL2.5	56.30	45.31	45.24	90.46	61.51	5.83	75.22
InternVL3	51.13	72.19	43.47	90.48	65.57	6.02	70.99
OmniDrive	70.30	82.44	69.44	**92.15**	78.81	3.61	75.38
RCVLA-Sem	**72.93**	**82.57**	**75.29**	91.67	**80.41**	**2.82**	74.25

Implementation details. We use six synchronized 640×640 camera views and multi-view 4D radar points, with Vicuna-7B as the language model. IQ-Former and RQ-Former each contain 256 learnable queries and six encoder layers. RCVLA-Phys uses five trajectory candidates, a two-layer

diffusion decoder, and two DDIM refinement steps. All stages are trained with AdamW on four NVIDIA L20 GPUs. Complete architecture and optimization settings are provided in Appendix C.2.

5.1 Main Results

Scene cognition. RCVLA-Sem combines visual semantics with radar-derived spatial and motion cues for scene description. In Table 1, it improves CIDEr over OmniDrive from 92.49 to 102.41, with gains across all captioning metrics. Table 2 shows that RCVLA-Sem improves occupancy accuracy from 78.81% to 80.41% and reduces key-object velocity error from 3.61 to 2.82 m/s. Drivable-area and intention accuracies remain lower than OmniDrive by 0.48 and 1.13 points, respectively. Figure 5 provides qualitative question-answering examples, illustrating how RCVLA-Sem combines visual semantics with radar-derived spatial and motion cues for scene understanding.

Table 3: Open-loop planning on OmniHD-QA. L2 error and collision rate at each horizon; lower is better.

	L2 (m)				Collision (%)			
Method	1 s	2 s	3 s	Avg.	1 s	2 s	3 s	Avg.
Qwen2.5-VL	1.239	1.774	2.488	1.834	2.870	6.389	11.111	6.790
Qwen3-VL	0.224	0.581	1.059	0.621	0.370	2.623	6.420	3.138
InternVL2.5	0.255	0.668	1.570	0.831	0.463	2.039	6.457	2.986
InternVL3	2.411	4.646	7.086	4.714	19.136	23.796	31.636	24.856
UniAD	0.403	0.750	1.171	0.775	0.525	1.821	3.519	1.955
Constant-Velo	0.126	0.337	0.646	0.370	0.031	0.247	2.222	0.833
BEV-Planner	0.176	0.339	0.585	0.367	0.031	**0.062**	0.864	0.319
OmniDrive	0.121	0.334	0.648	0.368	**0.000**	0.123	1.420	0.514
RCVLA-Sem	0.117	0.317	0.608	0.348	0.031	0.216	1.481	0.576
RCVLA-Phys	**0.109**	**0.236**	**0.431**	**0.259**	**0.000**	0.123	**0.401**	**0.175**

Table 4: RCVLA-Sem ablation results. Accuracy and collision values are percentages. RLP denotes radar-language pretraining.

Setting	CIDEr↑	V_{err} ↓	Acc_{int} ↑	Acc_{occ} ↑	L2↓	Coll.↓
w/o 4D radar	98.70	3.50	68.89	79.48	**0.344**	**0.391**
w/o RLP	101.60	3.45	70.97	**80.63**	0.345	**0.391**
Token addition	100.45	3.00	72.07	80.21	0.354	0.761
Token concatenation	102.20	2.96	73.80	**80.63**	0.353	0.525
Full	**102.41**	**2.82**	**74.25**	80.41	0.348	0.576

Trajectory planning. As shown in Table 3, RCVLA-Phys achieves the lowest average L2 error and collision rate among the compared methods, with 0.259 m and 0.175%, respectively. Compared with RCVLA-Sem, trajectory arbitration reduces the average L2 error from 0.348 to 0.259 m and the collision rate from 0.576% to 0.175%. Although RCVLA-Sem already outperforms BEV-Planner in average L2 error (0.348 vs. 0.367 m), its collision rate remains higher (0.576% vs. 0.319%), indicating that accurate reference trajectories do not necessarily correspond to lower collision risk. With RCVLA-Phys, the L2 error is further reduced at all three horizons, while the 3 s collision rate drops from 1.481% to 0.401%. Qualitative examples of trajectory refinement are shown in Figure 6.

5.2 Ablation Studies

RCVLA-Sem ablation. Table 4 compares different RCVLA-Sem settings. The full model achieves the best CIDEr, key-object velocity error, and intention accuracy. Compared with removing 4D radar, it improves CIDEr from 98.70 to 102.41 and reduces velocity error from 3.50 to 2.82 m/s. Removing RLP or replacing the proposed interaction with token addition or concatenation also degrades semantic performance. These results support the roles of radar-language alignment and cross-modal interaction in scene cognition.

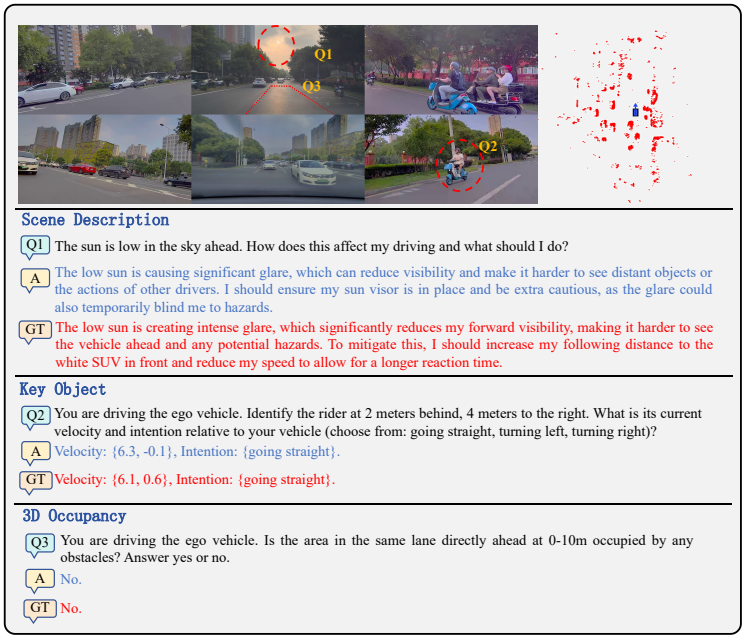


Figure 5: Qualitative question-answering results.

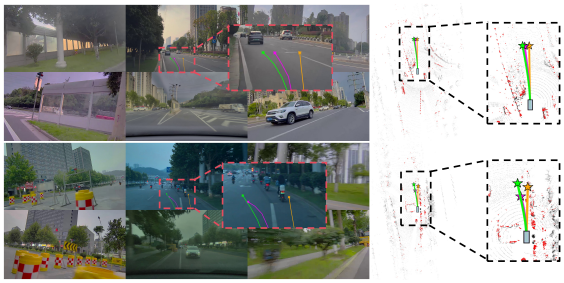


Figure 6: Qualitative examples of trajectory arbitration. Orange, magenta, and green denote the reference, arbitrated, and GT trajectories.

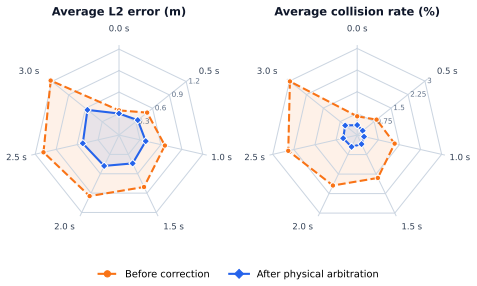


Figure 7: Effect of reference delay on trajectory arbitration. Orange and blue denote the reanchored reference and arbitrated trajectory.

However, the gains in semantic reasoning do not translate into a lower collision rate, suggesting that semantic-level radar integration alone is insufficient for trajectory safety and motivating the dedicated trajectory arbitration in RCVLA-Phys.

RCVLA-Phys components. Table 5 evaluates the contribution of occupancy queries, object queries, radar clusters, and radar risk calibration. Removing occupancy queries causes the largest overall degradation, increasing average L2 error from 0.259 to $0.321\,\mathrm{m}$ and collision rate from 0.175% to 0.597%. Removing object queries increases the two metrics to $0.320\,\mathrm{m}$ and 0.309%, respectively. Removing radar clusters raises the collision rate to 0.442%, while removing risk calibration increases it to 0.340%. These results show that all four components contribute to trajectory arbitration, with both radar clusters and risk calibration providing clear gains in collision reduction.

Table 5: Ablation of RCVLA-Phys.

Setting	L2 (m)↓	Collision (%)↓
w/o occupancy queries	0.321	0.597
w/o object queries	0.320	0.309
w/o radar clusters	0.297	0.442
w/o risk calibration	0.296	0.340
Full	**0.259**	**0.175**

Robustness to delayed references. We evaluate RCVLA-Phys under delayed reference trajectories. During training, reference delays are sampled from 0 to $3\,\mathrm{s}$ at $0.5\,\mathrm{s}$ intervals, while object, occupancy, and radar conditions remain current. Expired waypoints are removed and the remaining trajectory is reanchored to the current ego frame before arbitration. As shown in Table 6, average L2 error increases from 0.301 to $0.564\,\mathrm{m}$ as the delay reaches $3\,\mathrm{s}$, while the collision rate rises to 0.532%. Figure 7 shows that current scene conditions can partially compensate for delayed references, but the degradation increases with reference staleness.

Table 6: Robustness to reference trajectories

Delay	Avg. L2 (m)↓	Avg. Coll. (%)↓
0.0 s	0.301	0.347
0.5 s	0.335	0.243
1.0 s	0.383	0.243
1.5 s	0.439	0.347
2.0 s	0.478	0.451
2.5 s	0.519	0.498
3.0 s	0.564	0.532

6 Conclusion

In this work, we present RCVLA, a radar-camera VLA framework that exploits 4D radar in complementary roles for semantic reasoning and trajectory arbitration. We construct Cap4DR for radar-language alignment and OmniHD-QA for driving instruction tuning and evaluation. RCVLA-Sem performs radar-grounded semantic reasoning through camera-radar token interaction, supporting driving question answering and long-horizon reference planning while producing structured object and occupancy queries. RCVLA-Phys further refines reference-guided trajectory candidates using these queries and cluster-level radar measurements, with radar-derived TTC risk calibrating candidate selection. Experiments show that language-aligned radar tokens improve scene cognition, while cluster-level radar measurements and risk calibration further improve open-loop trajectory planning. Future work will explore predictive world models for longer-horizon reasoning and planning.

AI USE STATEMENT

Generative AI tools were used for limited assistance with manuscript organization, language editing, selected drafting, and adaptation to the ICLR template. They were not used to generate or modify the reported experimental results. All citations, technical descriptions, analyses, and revisions were reviewed and verified by the authors, who take full responsibility for the final manuscript.

REFERENCES

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*, 2025a.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025b.

Mario Bijelic, Tobias Gruber, Fahim Mannan, Florian Kraus, Werner Ritter, Klaus Dietmayer, and Felix Heide. Seeing through fog without seeing fog: Deep multimodal sensor fusion in unseen adverse weather. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11679–11689. IEEE, 2020.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24185–24198, 2024.

Minseong Choi, Seunghoon Yang, Seungho Han, Yeongseok Lee, Minyoung Lee, Keun Ha Choi, and Kyung-Soo Kim. Msc-rad4r: Ros-based automotive dataset with 4d radar. *IEEE Robotics and Automation Letters*, 8(11):7194–7201, 2023.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

Yuxin Fang, Quan Sun, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. Eva-02: A visual representation for neon genesis. *Image and Vision Computing*, 149:105171, 2024.

Runwei Guan, Liye Jia, Shanliang Yao, Fengyufan Yang, Sheng Xu, Erick Purwanto, Xiaohui Zhu, Ka Lok Man, Eng Gee Lim, Jeremy Smith, et al. Watervg: Waterway visual grounding based on text-guided vision and mmwave radar. *IEEE Transactions on Intelligent Transportation Systems*, 26(5):7275–7291, 2025a.

Runwei Guan, Jianan Liu, Liye Jia, Haocheng Zhao, Shanliang Yao, Xiaohui Zhu, Ka Lok Man, Eng Gee Lim, Jeremy Smith, and Yutao Yue. Nanomvg: Usv-centric low-power multi-task visual grounding based on prompt-guided camera and 4d mmwave radar. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 11789–11796. IEEE, 2025b.

Runwei Guan, Ruixiao Zhang, Ningwei Ouyang, Jianan Liu, Ka Lok Man, Xiaohao Cai, Ming Xu, Jeremy Smith, Eng Gee Lim, Yutao Yue, et al. Talk2radar: Bridging natural language with 4d mmwave radar for 3d referring expression comprehension. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 10884–10891. IEEE, 2025c.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2022a. URL https://openreview.net/forum?id=nZeVKeeFYf9.

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

Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, et al. Planning-oriented autonomous driving. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17853–17862. IEEE, 2023.

Jianzhu Huai, Binliang Wang, Yuan Zhuang, Yiwen Chen, Qipeng Li, and Yulong Han. Snail radar: A large-scale diverse benchmark for evaluating 4d-radar-based slam. *The International Journal of Robotics Research*, 44(12):1941–1958, 2025.

Bo Jiang, Shaoyu Chen, Bencheng Liao, Xingyu Zhang, Wei Yin, Qian Zhang, Chang Huang, Wenyu Liu, and Xinggang Wang. Senna: Bridging large vision-language models and end-to-end autonomous driving. *International Journal of Computer Vision*, 134(9):407, 2026.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International Conference on Machine Learning (ICML)*, pp. 19730–19742. PmLR, 2023.

Zhiqi Li, Zhiding Yu, Shiyi Lan, Jiahan Li, Jan Kautz, Tong Lu, and Jose M Alvarez. Is ego status all you need for open-loop end-to-end autonomous driving? In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14864–14873. IEEE, 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 *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12037–12047. IEEE, 2025.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in Neural Information Processing Systems (NeurIPS)*, 36:34892–34916, 2023.

Andras Palffy, Ewoud Pool, Srimannarayana Baratam, Julian FP Kooij, and Dariu M Gavrila. Multi-class road user detection with 3+ 1d radar in the view-of-delft dataset. *IEEE Robotics and Automation Letters*, 7(2):4961–4968, 2022.

Xiangyuan Peng, Kay Bierzynski, Lorenzo Servadei, and Robert Wille. Text4radar-v2x: Text-guided 4d radar for cooperative 3d object detection. In *Proceedings of the 2026 International Conference on Multimedia Retrieval*, pp. 1025–1034, 2026.

Erich Schubert, Jörg Sander, Martin Ester, Hans Peter Kriegel, and Xiaowei Xu. Dbscan revisited, revisited: why and how you should (still) use dbscan. *ACM transactions on database systems (tods)*, 42(3):1–21, 2017.

Chonghao Sima, Katrin Renz, Kashyap Chitta, Li Chen, Hanxue Zhang, Chengen Xie, Jens Beißwenger, Ping Luo, Andreas Geiger, and Hongyang Li. Drivelm: Driving with graph visual question answering. In *European Conference on Computer Vision (ECCV)*, pp. 256–274. Springer, 2024.

Shihao Wang, Zhiding Yu, Xiaohui Jiang, Shiyi Lan, Min Shi, Nadine Chang, Jan Kautz, Ying Li, and Jose M Alvarez. Omnidrive: A holistic vision-language dataset for autonomous driving with counterfactual reasoning. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22442–22452. IEEE, 2025.

Jun Zhang, Huayang Zhuge, Yiyao Liu, Guohao Peng, Zhenyu Wu, Haoyuan Zhang, Qiyang Lyu, Heshan Li, Chunyang Zhao, Dogan Kircali, et al. Ntu4dradlm: 4d radar-centric multi-modal dataset for localization and mapping. In *2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 4291–4296. IEEE, 2023.

Xinyu Zhang, Li Wang, Jian Chen, Cheng Fang, Guangqi Yang, Yichen Wang, Lei Yang, Ziying Song, Lin Liu, Xiaofei Zhang, et al. Dual radar: A multi-modal dataset with dual 4d radar for autononous driving. *Scientific data*, 12(1):439, 2025.

Lianqing Zheng, Zhixiong Ma, Xichan Zhu, Bin Tan, Sen Li, Kai Long, Weiqi Sun, Sihan Chen, Lu Zhang, Mengyue Wan, et al. Tj4dradset: A 4d radar dataset for autonomous driving. In *2022 IEEE 25th international conference on intelligent transportation systems (ITSC)*, pp. 493–498. IEEE, 2022.

Lianqing Zheng, Sen Li, Bin Tan, Long Yang, Sihan Chen, Libo Huang, Jie Bai, Xichan Zhu, and Zhixiong Ma. Rcfusion: Fusing 4-d radar and camera with bird's-eye view features for 3-d object detection. *IEEE Transactions on Instrumentation and Measurement*, 72:1–14, 2023.

Lianqing Zheng, Jianan Liu, Runwei Guan, Long Yang, Shouyi Lu, Yuanzhe Li, Xiaokai Bai, Jie Bai, Zhixiong Ma, Hui-Liang Shen, and Xichan Zhu. Doracamom: Joint 3d detection and occupancy prediction with multi-view 4d radars and cameras for omnidirectional perception. *IEEE Transactions on Circuits and Systems for Video Technology*, 36(6):7630–7645, 2026a. doi: 10.1109/TCSVT.2026.3657111.

Lianqing Zheng, Long Yang, Qunshu Lin, Wenjin Ai, Minghao Liu, Shouyi Lu, Jianan Liu, Hongze Ren, Jingyue Mo, Xiaokai Bai, et al. Omnihd-scenes: A next-generation multimodal dataset for autonomous driving. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 48(6): 7032–7049, 2026b.

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

APPENDIX

A OVERVIEW

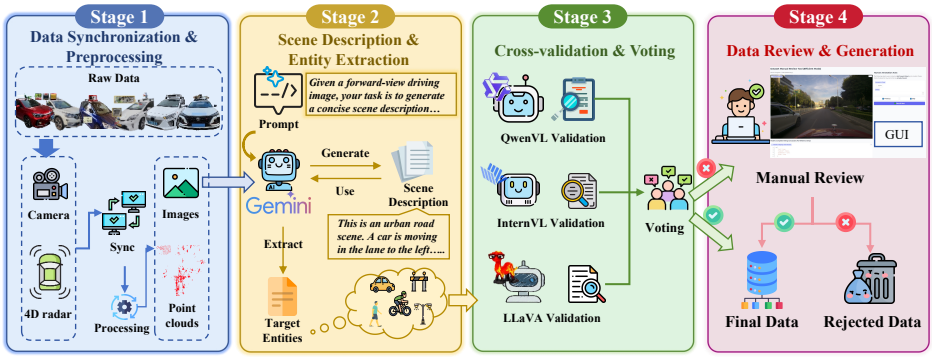
We provide additional details on dataset construction, evaluation protocols, and implementation, together with per-horizon ablations, delayed-reference results, runtime analysis, and qualitative examples.

B DATASET CONSTRUCTION

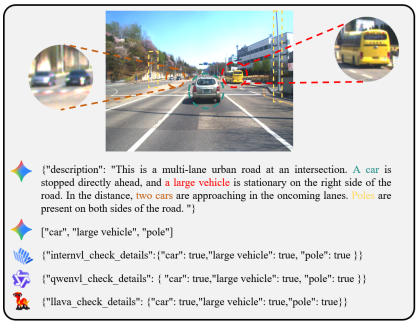
B.1 CAP4DR CONSTRUCTION

Sources and synchronization. Cap4DR is built from front-view images and 4D radar data collected from NTU4DRadLM (Zhang et al., 2023), SNAIL Radar (Huai et al., 2025), MSC-RAD4R (Choi et al., 2023), Dual Radar (Zhang et al., 2025), View-of-Delft (Palffy et al., 2022), and TJ4DRadSet (Zheng et al., 2022). These datasets cover urban roads, elevated roads, tunnels, and campus environments. We use the original camera–radar synchronization when it is available. Otherwise, each radar frame is paired with the nearest camera frame, and pairs with a time difference greater than $0.1\,\mathrm{s}$ are removed.

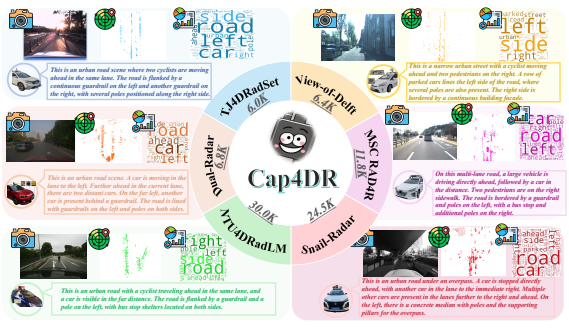
All radar data are converted to a common forward–left–up coordinate system. The available RCS or SNR channel is normalized separately for each source dataset. When ego motion compensation is not provided, ego velocity is estimated from the radial-velocity field using RANSAC. The resulting synchronized data are denoted as $\mathcal{S} = \{(\mathcal{P}_i, \mathbf{I}_i)\}_{i=1}^{N}$.



(a) Construction pipeline



(b) Description validation



(c) Dataset composition and examples

Figure A1: Cap4DR construction pipeline and examples. (a) The four-stage pipeline for camera–radar synchronization, description generation, entity validation, and manual review. (b) An example of description and entity validation. (c) Dataset composition and representative scenes from the six source datasets.

Description generation and review. Gemini-2.5-Pro (Comanici et al., 2025) is prompted to generate a concise description of each front-view image using geometric traffic entities and relative spatial relations. Attributes such as color, texture, text, and illumination are excluded, together with unsupported inferences. A second prompt extracts the entity nouns from the description and maps them to a canonical vocabulary. Qwen2.5-VL, InternVL2.5, and LLaVA-v1.5 then independently verify whether each extracted entity is supported by the paired image. For entity e_{ij}, the validation score is

$$v_{ij} = \sum_{m=1}^{3} V_m(\mathbf{I}_i, e_{ij}), \qquad V_m(\mathbf{I}_i, e_{ij}) \in \{0, 1\}. \tag{17}$$

Entities with $v_{ij} \geq 2$ are accepted automatically. Samples containing lower-scoring entities are sent for manual review and retained only when all flagged entities are confirmed. This procedure verifies image-level support for the extracted radar-relevant entities, but does not establish point-to-entity correspondence in the radar data. The resulting dataset contains 86,016 radar–image–text triples.

B.2 OmniHD-QA Generation

OmniHD-QA is constructed from OmniHD-Scenes (Zheng et al., 2026b) and covers four driving tasks. Scene description QA uses generated text, while key object states, occupancy, and future trajectories are derived from structured annotations and deterministic rules.

Scene description QA. Each sample uses a stitched six-view image together with the observed class labels as semantic priors. We first generate an ego-centric description of the road layout, weather, lighting, traffic signals, surrounding traffic participants, and potential hazards, using only information supported by the input images. We then generate up to ten QA pairs from each description, focusing on scene content, spatial relations, and traffic elements relevant to driving.

Key object QA. Key object QA focuses on traffic participants most relevant to the ego vehicle and their motion. Visible actors with valid future tracks are selected based on their relative position, distance, and motion state, and only the most relevant candidates are retained. The answers provide each selected actor's planar velocity and driving intention, categorized as turning left, turning right, or going straight according to its future trajectory.

Occupancy QA. Occupancy QA converts dense voxel annotations into region-level binary questions. Semantic labels are grouped into dynamic road users, static obstacles, drivable surface, sidewalk, and background. The surrounding space is partitioned into ego-centric regions at different longitudinal ranges. For each region, the questions ask whether dynamic or static obstacles are present and whether the area is drivable.

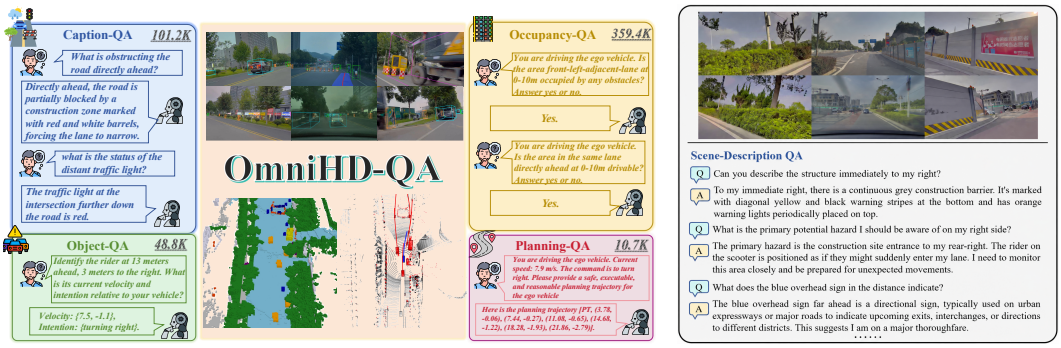
Planning QA. Planning QA pairs each sample with a driving instruction, the current ego state, and a future trajectory expressed in the current ego coordinate frame. The instruction is derived from the recorded future motion and provides the high-level driving intent for trajectory generation. Only samples with valid future poses are retained, and all compared methods receive the same instruction and ego states.

Data split. OmniHD-QA follows the original data split of OmniHD-Scenes. Table A1 summarizes the task composition and the number of samples in each split.

Table A1: OmniHD-QA task composition and data split.

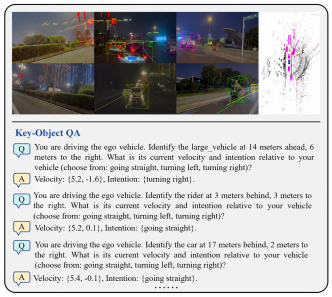
Task	Pairs	Split	Pairs
Scene description	101,221	Train	358,185
Key object	48,812	Test	161,976
Occupancy	359,407	Total	520,161
Planning	10,721		

Figure A2 illustrates the composition of OmniHD-QA and representative examples for the four tasks. Scene description QA focuses on road conditions, traffic signals, and driving-relevant scene information. Key object QA describes the motion of nearby traffic participants. Occupancy QA queries the state of local regions, while planning QA represents future motion as a sequence of waypoints.

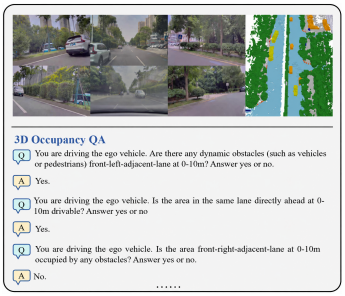


(a) Dataset composition

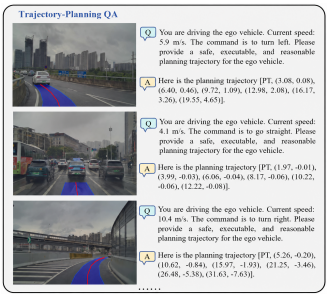
(b) Scene-description QA



(c) Key-object QA



(d) 3D occupancy QA



(e) Trajectory-planning QA

Figure A2: OmniHD-QA composition and representative examples for scene description, key object reasoning, 3D occupancy, and trajectory planning.

C Reproducibility and Evaluation Protocol

C.1 Metric Definitions

Captioning and classification metrics are reported with higher values indicating better performance, while velocity error, trajectory L2 error, and collision rate are lower-is-better metrics.

Scene-description metrics. We evaluate scene description using BLEU-1 to BLEU-4, ROUGE-L, CIDEr, and METEOR against the reference description. These metrics capture complementary aspects of textual agreement between the generated and reference descriptions.

Key-object metrics. For each evaluated object, we report planar velocity error and intention accuracy. Let $\mathbf{v}_i$ and $\widehat{\mathbf{v}}_i$ denote the annotated and predicted planar velocities. The mean velocity error is

$$\mathrm{V}_{\mathrm{err}} = \frac{1}{N_{\mathrm{obj}}} \sum_{i=1}^{N_{\mathrm{obj}}} \left\| \widehat{\mathbf{v}}_i - \mathbf{v}_i \right\|_2 . \tag{18}$$

Motion intention is categorized as turning left, turning right, or going straight, and intention accuracy is reported as the percentage of correctly predicted labels.

Occupancy metrics. Occupancy QA is evaluated for general occupancy, dynamic obstacles, static obstacles, and drivable area. We report the accuracy of each category together with the micro-averaged accuracy over all occupancy questions.

Trajectory-planning metrics. Let $\widehat{\mathbf{p}}_{i,t}$ and $\mathbf{p}_{i,t}$ denote the predicted and logged ego positions for sample i at waypoint t. Since waypoints are spaced at 0.5 s, the L2 error at horizon $h \in \{1, 2, 3\}$ s is computed over all waypoints up to that horizon:

$$\mathrm{L2}_h = \frac{1}{N_p} \sum_{i=1}^{N_p} \frac{1}{n_h} \sum_{t=1}^{n_h} \left\| \widehat{\mathbf{p}}_{i,t} - \mathbf{p}_{i,t} \right\|_2, \qquad n_h = 2h. \tag{19}$$

A sample is counted as a collision at horizon h if the predicted ego footprint overlaps an annotated surrounding vehicle at any waypoint up to that horizon. The collision rate is

$$\mathrm{CR}_h = \frac{100}{N_p} \sum_{i=1}^{N_p} C_i^h, \tag{20}$$

where $C_i^h \in \{0, 1\}$ denotes the collision indicator. The average L2 error and collision rate are computed as the arithmetic means over the 1, 2, and 3 s horizons.

RCVLA-Sem predicts 12 reference waypoints over 6 s. The reported planning results use only the first six waypoints, corresponding to the first 3 s, which is also the segment refined by RCVLA-Phys.

C.2 Architecture and Optimization Settings

Figure A3 illustrates the shared representation path. Image and radar features are used by both the query formers and the structured object and occupancy decoders.

Table A2 summarizes the main architecture and training settings. RCVLA-Sem generates a 12-waypoint reference trajectory over 6 s, and RCVLA-Phys refines its first six waypoints over the next 3 s. During training, the diffusion timestep is sampled uniformly from $i \in \{0, \ldots, 49\}$. At inference, RCVLA-Phys starts from step 8 and performs two deterministic DDIM updates. The final trajectory is selected using $\arg\max_m (s_m - \lambda_s r_m)$. Candidate trajectories are formed from the semantic reference and selected trajectory anchors, and are refined using structured scene queries and radar-cluster measurements before risk-aware ranking.

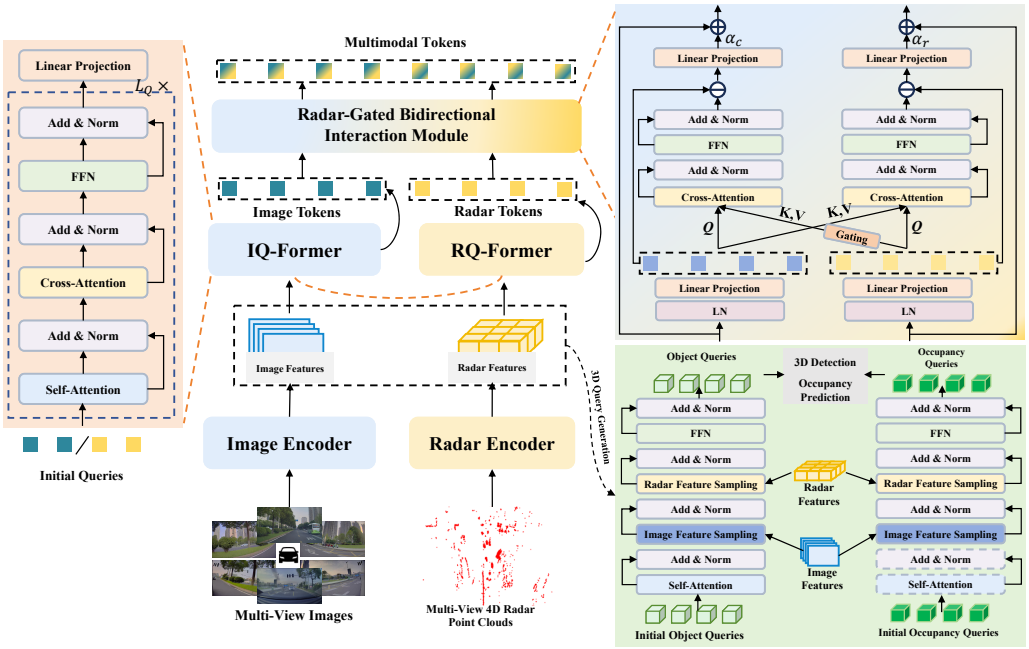


Figure A3: Construction of multimodal tokens and structured queries. Image and radar features are converted into language-aligned tokens and structured object and occupancy queries for RCVLA-Phys.

All stages are trained with AdamW on four NVIDIA L20 GPUs. Radar perception pretraining runs for 24 epochs with a learning rate of 10^{-3} and a batch size of 16, followed by radar-language alignment for 3 epochs with the same learning rate and a batch size of 8. RCVLA-Sem is fine-tuned for 6 epochs with a learning rate of 3×10^{-5} and a batch size of 4. For this stage, we use a layerwise learning-rate decay of 0.9 for EVA-02-L. RCVLA-Phys is trained for 12 epochs with a learning rate of 5×10^{-4} and a batch size of 256.

Table A2: Architecture and training settings.

Component	Configuration
Images	Six views, 640×640 pixels
Language model	Vicuna-7B with frozen base weights
IQ-/RQ-Former	256 queries, 256-D, 6 layers, 4096-D LLM projection
Token interaction	512-D interaction space
Structured heads	300 object queries, $60 \times 40 \times 4$ occupancy grid
LoRA	$r = 128$, $\alpha = 16$
Semantic reference	12 waypoints over 6 s at 0.5 s intervals
RCVLA-Phys	5 candidates, 512-D, 2 decoder layers
Arbitration conditions	64 object, 300 occupancy, and up to 32 radar queries
Diffusion schedule	1,000-step scaled-linear schedule
Risk parameters	$\tau_{\mathrm{TTC}} = 2.0\,\mathrm{s}$, $\lambda_{\mathrm{r}} = 0.3$, $T_{\mathrm{c}} = 0.3$, $\lambda_{\mathrm{s}} = 1$
Loss weights	$\lambda_{\mathrm{Det}} = \lambda_{\mathrm{OCC}} = 0.1$, $\lambda_{\mathrm{reg}} = 8$

D Additional Ablation Studies

D.1 Structured Conditions and Risk Calibration

Table A3 reports the per-horizon RCVLA-Phys ablation corresponding to Table 5. Removing occupancy queries causes the largest overall degradation, indicating the importance of scene-level spatial constraints for trajectory refinement. Removing object queries also increases L2 error, while removing radar clusters or risk calibration mainly increases the collision rate. These results show that structured scene queries and radar information provide complementary conditions for trajectory arbitration.

Table A3: Per-horizon RCVLA-Phys component ablation.

	L2 (m)				Collision (%)			
Setting	1 s	2 s	3 s	Avg.	1 s	2 s	3 s	Avg.
w/o occupancy queries	0.140	0.299	0.524	0.321	0.278	0.463	1.049	0.597
w/o object queries	0.147	0.301	0.514	0.320	0.031	**0.123**	0.772	0.309
w/o radar clusters	0.136	0.277	0.479	0.297	0.093	0.247	0.988	0.442
w/o risk calibration	0.132	0.273	0.483	0.296	0.031	0.247	0.741	0.340
Full RCVLA-Phys	**0.109**	**0.236**	**0.431**	**0.259**	**0.000**	**0.123**	**0.401**	**0.175**

D.2 Order of Structured Conditions

Table A4 compares different serial orders of the three structured conditions. OCC→OD→Radar achieves the lowest average L2 error and collision rate, suggesting that introducing scene-level occupancy constraints first, followed by object information and current radar measurements, provides a more effective refinement sequence. The differences are more evident at the 3 s horizon, and we use this order in the final model.

Table A4: Per-horizon serial condition-order ablation.

	L2 (m)				Collision (%)			
Order	1 s	2 s	3 s	Avg.	1 s	2 s	3 s	Avg.
Radar→OD→OCC	0.138	0.283	0.484	0.302	**0.000**	0.278	0.617	0.298
Radar→OCC→OD	0.134	0.276	0.484	0.298	0.031	0.154	0.802	0.329
OCC→Radar→OD	0.123	0.254	0.448	0.275	0.031	0.247	0.741	0.340
OD→OCC→Radar	0.117	0.245	0.437	0.266	0.031	0.216	0.648	0.298
OD→Radar→OCC	0.113	0.243	0.439	0.265	**0.000**	0.185	0.432	0.206
OCC→OD→Radar	**0.109**	**0.236**	**0.431**	**0.259**	**0.000**	**0.123**	**0.401**	**0.175**

D.3 Reference-Guided Candidate Construction

The nearest-only strategy combines the semantic reference with its closest trajectory anchors, while nearest–diverse additionally introduces anchors with different motion patterns. As shown in Table A5, adding limited diversity improves both trajectory error and collision rate for $M = 5$. Increasing the candidate set to 10 or 15 does not provide consistent further gains, indicating that excessive candidates may introduce redundant trajectory modes. We therefore use five nearest–diverse candidates in the final model.

Table A5: Per-horizon candidate-construction ablation.

		L2 (m)				Collision (%)			
Strategy	M	1 s	2 s	3 s	Avg.	1 s	2 s	3 s	Avg.
Nearest only	15	0.176	0.387	0.682	0.415	0.062	0.185	0.679	0.309
Nearest only	10	0.174	0.369	0.636	0.393	0.062	0.340	0.864	0.422
Nearest only	5	0.123	0.256	0.453	0.277	0.216	0.432	0.833	0.494
Nearest–diverse	15	0.117	0.251	0.456	0.275	0.062	0.216	0.586	0.288
Nearest–diverse	10	0.145	0.288	0.495	0.309	0.062	0.185	0.586	0.278
Nearest–diverse	**5**	**0.109**	**0.236**	**0.431**	**0.259**	**0.000**	**0.123**	**0.401**	**0.175**

D.4 Robustness to Delayed Semantic References

Table A6 reports the per-horizon results for delayed semantic references. During training, the reference delay is sampled uniformly from 0 to 3 s at 0.5 s intervals. For each delay, expired waypoints are removed from the historical 6 s reference, the remaining trajectory is reanchored, and the following 3 s segment is refined by RCVLA-Phys. Because the maximum delay is 3 s, all settings are evaluated on the same subset after excluding the first 3 s of each sequence. As the delay increases from 0 to 3 s, the average L2 error increases from 0.301 to 0.564 m, while the collision rate remains relatively low, indicating that current scene conditions can partially compensate for outdated references, although the effect weakens as the reference becomes older.

Table A6: Per-horizon planning with delayed semantic references.

	L2 (m)				Collision (%)			
Delay	1 s	2 s	3 s	Avg.	1 s	2 s	3 s	Avg.
0.0 s	0.146	0.281	0.475	0.301	0.069	0.347	0.625	0.347
0.5 s	0.166	0.316	0.524	0.335	0.035	0.208	0.486	0.243
1.0 s	0.194	0.364	0.593	0.383	0.035	0.208	0.486	0.243
1.5 s	0.228	0.420	0.670	0.439	0.035	0.243	0.764	0.347
2.0 s	0.252	0.459	0.723	0.478	0.035	0.382	0.938	0.451
2.5 s	0.280	0.500	0.777	0.519	0.069	0.417	1.007	0.498
3.0 s	0.306	0.545	0.839	0.564	0.069	0.486	1.042	0.532

Figure A4 shows representative rain, night-turning, and car-following scenes under different reference delays. Across these cases, the refined trajectories remain smooth and generally follow the road geometry despite increasingly outdated references. The degradation becomes more visible as the delay increases, consistent with the quantitative trends in Figure 7.

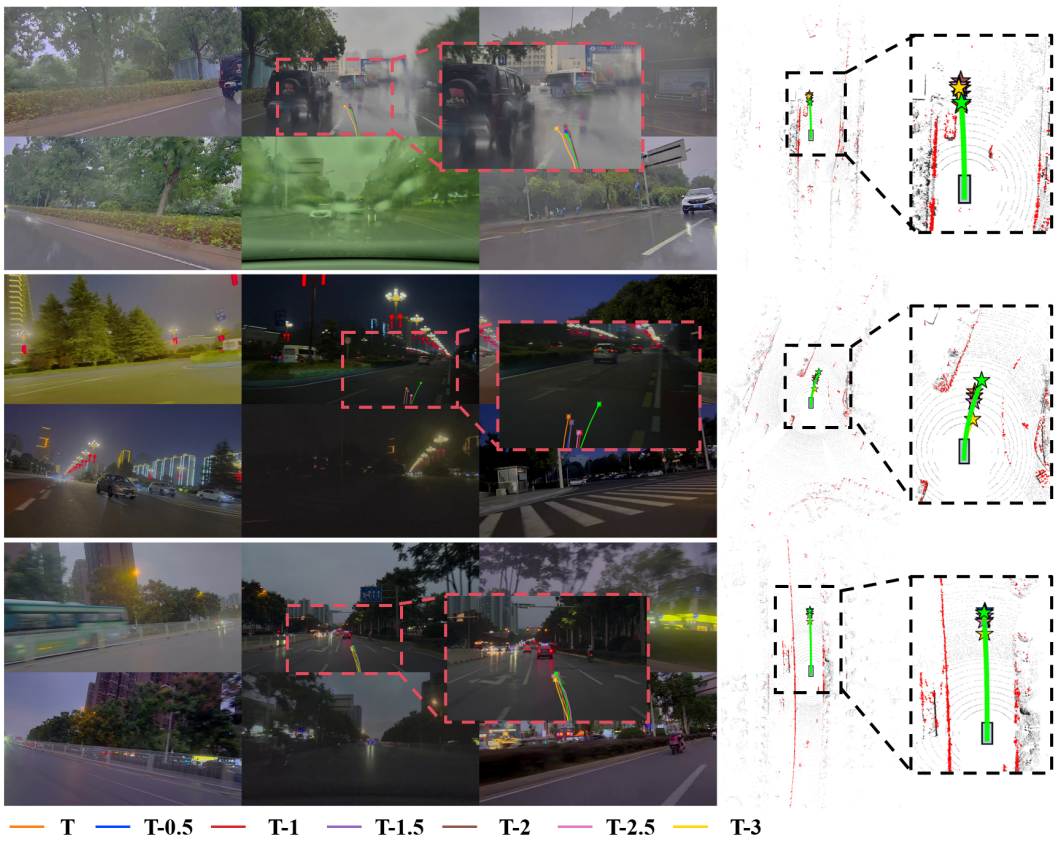


Figure A4: Qualitative trajectory arbitration with delayed semantic references. Representative rain, night-turning, and car-following scenes are shown for reference delays from 0 to 3 s.

D.5 Inference Efficiency

Table A7 reports the average inference latency on a single NVIDIA L20 GPU. The runtime of RCVLA-Sem varies across QA tasks because language responses are generated autoregressively, whereas planning directly regresses the reference trajectory from the planning-token representation. RCVLA-Phys requires only 12.50 ms for trajectory refinement and selection. The direct trajectory regression in RCVLA-Sem also reduces planning latency relative to OmniDrive, while the lightweight RCVLA-Phys module supports much more frequent short-horizon trajectory updates.

Table A7: Inference efficiency on a single NVIDIA L20 GPU.

Model	Task	Latency (ms)	FPS
RCVLA-Sem	Description QA	1666.92	0.60
RCVLA-Sem	Object QA	1108.22	0.90
RCVLA-Sem	Occupancy QA	555.32	1.80
RCVLA-Sem	Planning	551.02	1.81
RCVLA-Phys	Trajectory refinement	12.50	80.03
OmniDrive	Planning	2620.99	0.38

E Limitations

The planning evaluation uses open-loop geometric metrics and does not capture interactive agent responses, controller errors, or closed-loop recovery. Cap4DR descriptions are machine-generated and manually reviewed when needed, but residual errors and model bias may remain. Future work will consider closed-loop evaluation, broader geographic validation, and end-to-end latency.