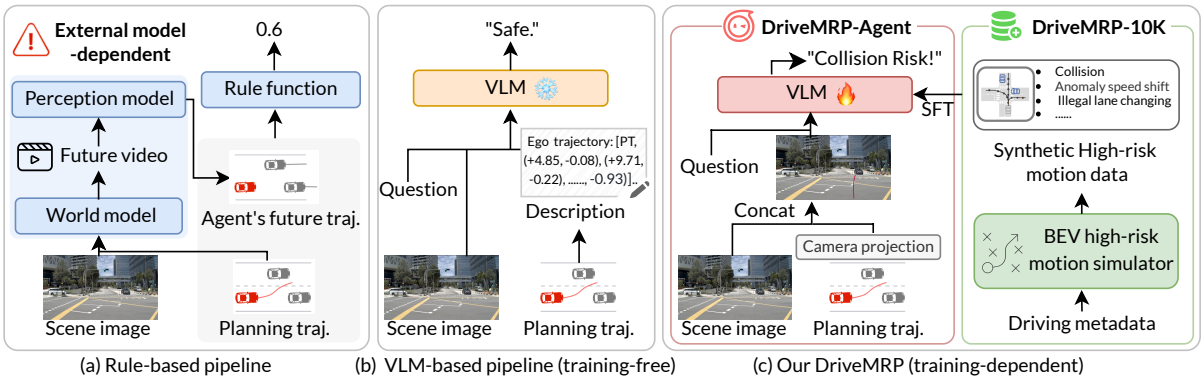
DriveMRP: Enhancing Vision–Language Motion Risk Prediction via Controllable High-Risk Motion Synthesis

Anonymous ACL submission



(a) Rule-based pipeline (b) VLM-based pipeline (training-free) (c) Our DriveMRP (training-dependent)

Figure 1: (a) **Rule-based pipeline**: relies on external world and perception models and scores the plan via a rule function; it is sensitive to upstream errors and offers limited interpretability. (b) **Zero-shot VLM**: feeds the trajectory as numeric text, leading to a coordinate–image modality gap and insufficient OOD generalization. (c) **Our DriveMRP**: a **BEV**-based high-risk motion simulator synthesizes **DriveMRP-10K**; Visualized-Trajectory Prompting (VTP) projects future waypoints onto the front view; on this basis, a backbone-flexible **DriveMRP-Agent** is trained with **SFT** and adapted via cross-distribution **few-shot tuning**. This simple approach generalizes well to the real world and accurately performs interpretable risk prediction.

Abstract

The autonomous driving field has achieved significant progress, largely driven by abundant real-world data. However, accurately predicting the safety of the ego vehicle's future motion policy in long-tail scenarios remains a major challenge due to the inherent uncertainty of dynamic environments and the limited coverage of existing data. In this work, we enhance the motion risk prediction capability of vision-language models (VLMs) using controllable high-risk motion data synthesis. Specifically, we introduce a bird's-eye-view motion simulation framework that models risks from the ego vehicle, other agents, and the environment, producing plug-and-play high-risk motion data for VLM training. Around this framework, we establish a "data synthesis → model training → generalization validation" pipeline, and train a backbone-flexible motion risk prediction model, **DriveMRP-Agent**, which reasons about potential risks via multi-step, logic-guided visual question answering. Extensive experiments show that fine-tuning with our synthesized high-risk motion data substantially improves multiple VLM baselines, raising risk recognition accuracy from 27.39% to 88.83% on the DriveMRP-10K test set. Moreover, in zero-shot testing on our in-house real high-risk motion data, DriveMRP-Agent achieves 68.25% risk recognition accuracy. Cross-distribution few-shot adaptation further boosts cross-domain accuracy to 84.50%, indicating strong generalization to real risk scenarios.

1 Introduction

End-to-end autonomous driving has made rapid progress in recent years (Hu et al., 2023; Jiang et al., 2023; Wang et al., 2024b; Sima et al., 2024; Tian et al., 2024). However, accurately predicting the safety of the ego vehicle's future motion in long-tail scenarios remains challenging because dynamic environments are uncertain and risky data are sparse. In such situations, systems often generate multiple candidate trajectories of varying quality. Current evaluations mostly rely on rule functions (Dosovitskiy et al., 2017; Li et al., 2022; Dauner et al.,

2024; Caesar et al., 2021) or model-based scalar scores (Wang et al., 2024c; Gao et al., 2024; Huang et al., 2024) and lack explanations of risk categories and causes, which hinders risk prevention for end-to-end algorithms and makes them unsuitable for issuing warnings or operational cues as driver assistance. Therefore, identifying motion risk types and explaining their causes is fundamental to reliable driver assistance and continual improvement of decision-making algorithms.

Building strong motion-risk prediction faces three main difficulties: (i) dynamic environments are highly uncertain, especially when interpreting unstructured aspects (Min et al., 2024; Seo et al., 2023) (e.g., weather (Zhang et al., 2023)) and unpredictable traffic participants (Chen et al., 2024a; Zhou et al., 2024a); (ii) high-risk events are long-tailed, so learning-based methods are susceptible to out-of-distribution states and struggle to generalize to unseen hazards (Li et al., 2024b); and (iii) definitions, rules, and thresholds of "risk" vary across environments and systems, creating a need for a controllable synthesis pipeline that aligns heterogeneous risk definitions while providing diverse high-risk motion data.

As shown in Figure 1(a), traditional rule-based methods (e.g., Drive-WM (Wang et al., 2024c)) typically rely on external world models and perception modules to predict other agents and then compute scores via predefined rules; this pipeline is sensitive to upstream prediction/perception errors and structured-scene assumptions. By contrast, recent work shown in Figure 1(b) (e.g., HE-Drive (Wang et al., 2024a)) leverages VLMs for unstructured scenes, but often adopts a zero-shot setting and provides trajectory coordinates as "numeric text," leaving a coordinate–image modality gap that makes waypoint–environment spatial relations difficult to model.

In this work, we enhance VLM motion risk prediction by synthesizing long-tailed high-risk motion data. We propose a BEV motion simulation method that models risks from ego behavior, other agents, and environment constraints, while exposing rules and thresholds as configurable parameters. In parallel, we design **DriveMRP-Agent**, a plug-and-play, backbone-flexible risk prediction framework whose final performance depends on the visual-spatial ability of the chosen VLM. DriveMRP-Agent uses Visualized-Trajectory Prompting (VTP) to overlay future waypoints as visual elements onto the front-view image, bridging the modality gap between numeric waypoints and image representations; it then performs multi-step visual QA from scene understanding to motion analysis and risk determination. We further analyze domain gaps and use cross-distribution few-shot adaptation to transfer synthetic training to real risky scenarios.

In summary, our main contributions are as follows:

1. A controllable synthesis framework for motion risk—covering risk grammar, incident attribution, and quality control—that spans typical risk types with pluggable rules and constructs the paired dataset **DriveMRP-10K**.

2. A VLM-based risk prediction agent that combines VTP and multi-step logical QA to produce explainable estimates of both risk categories and causal factors.

3. Empirical evidence that zero-shot VLMs are unreliable for risk recognition; accordingly, we propose an "SFT + cross-distribution few-shot-tuning" training paradigm that narrows the domain gap and achieves stable, significant improvements in cross-domain tests.

2 Related Work

Rule-based Motion Reward Estimation. In autonomous driving, rule-based reward functions typically rely on simulators (Dosovitskiy et al., 2017; Caesar et al., 2021; Li et al., 2022; Dauner et al., 2024) to evaluate vehicle dynamics and compliance. However, existing simulators provide limited structured annotations (e.g., they lack precise lane-level solid/dashed markings, comprehensive traffic-light sensing, and diverse weather modeling), making behaviors such as red-light violations and illegal lane changes difficult to assess. Moreover, rule-based methods depend on upstream object detection and trajectory prediction, so their overall capability is bounded by these modules and tends to generalize poorly to complex real-world scenes.

World Model-based Motion Reward Estimation. Recent works explore using world models for motion reward estimation. *Drive-WM* (Wang et al., 2024c) generates future scenes via action-conditioned video prediction and then applies external perception modules (Li et al., 2024a; Liao et al., 2022) to extract structured representations for rule-based scoring, inheriting both the annotation

dependence of rule-based methods and the domain limitations of the perception modules trained on specific datasets (Caesar et al., 2020). *Vista* (Gao et al., 2024) directly evaluates candidate actions through a video prediction model pre-trained on expert data, but tends to favor expert-like trajectories and struggles to generalize beyond the training distribution. Both approaches generally output scalar rewards without explaining which factors lead to potential risks. Motivated by this interpretability gap, we leverage vision–language models (VLMs) for more transparent risk evaluation.

VLM-based Motion Evaluation. Pre-trained vision–language models (VLMs), by virtue of their understanding and reasoning over open-world scenes, have shown potential in end-to-end autonomous driving (Gopalkrishnan et al., 2024; Zhou et al., 2024b; Tian et al., 2024; Sima et al., 2024; Wang et al., 2024b; Mao et al., 2023; Han et al., 2025). Many works focus on scene semantics and instruction understanding, as well as exploring the feasibility of multi-modal QA and planning assistance in driving contexts. Nevertheless, fine-grained motion risk prediction and evaluation remain less studied. Among existing attempts, *HE-Drive* (Wang et al., 2024a) dynamically reweights traditional rule metrics using a VLM to adjust risk scores, but it still heavily depends on hand-crafted rules and zero-shot capability, limiting representation of complex interactions and near-critical conditions. *Gen-Drive* (Huang et al., 2024) uses a VLM to construct trajectory preference pairs for training an independent reward model, which can capture relative preferences, but its samples mainly stem from a biased trajectory-generation process with insufficient coverage of high-risk/critical distributions, making it prone to failure in rare scenarios. *Simlingo* (Renz et al., 2025) enumerates multiple candidate actions within the same visual scene via Action Dreaming and annotates "instruction–action–risk," strengthening instruction-constrained risk judgment; however, real accidents often do not presuppose explicit instructions, and its data primarily come from *Carla* (Dosovitskiy et al., 2017) sensors, which differ in distribution from real data, so systematic validation on real risk data remains relatively limited.

Unlike the above methods, we propose a BEV-based motion simulation method that synthesizes key hazardous trajectories with matched causal explanations, and adopt a "synthetic SFT and cross-distribution few-shot adaptation" paradigm for robust real-world generalization. Our goal is not to estimate scalar rewards, learn trajectory preferences, or score candidate actions under the same protocol as *Drive-WM* (Wang et al., 2024c), *HE-Drive* (Wang et al., 2024a), or *Gen-Drive* (Huang et al., 2024); instead, DriveMRP predicts motion-risk categories and generates natural-language causal explanations. Therefore, direct numerical comparison with these methods is not fair due to differences in output format, label space, input assumptions, and evaluation protocol. We instead conduct controlled comparisons with appropriate VLM baselines under our own task setting, so as to isolate the contributions of synthetic risk supervision and VTP.

3 Method

3.1 Overview

We formalize motion risk prediction as follows: given front-view camera images and the ego vehicle's future trajectory $\mathcal{M} = \{(x_n, y_n)\}_{n=0}^{N}$ sampled in BEV at fixed step Δt (horizon $H = N\Delta t$), the model outputs (i) a scene description, (ii) a risk category, and (iii) a causal explanation. To this end, §3.2 presents a high-risk motion synthesis pipeline with category labels and causal explanations; §3.3 introduces DriveMRP-Metric for evaluation; §3.4 proposes DriveMRP-Agent, which bridges the coordinate–image modality gap via Visualized-Trajectory Prompting (VTP) under an "SFT and cross-distribution few-shot adaptation" paradigm (Figure 2).

3.2 High-Risk Motion Data Synthesis Framework

We build a simulation-driven data synthesis framework. Building on nuPlan (Caesar et al., 2021), we controllably synthesize key hazardous trajectories and pair them with risk categories and causal supervision to support evaluation in DriveMRP-Metric and training of DriveMRP-Agent. The system abstracts risk into a set of configurable rules and thresholds such as risk types, triggering conditions, and trajectory duration. Based on this risk grammar, we align heterogeneous risk definitions across different systems and synthesize plug-and-play high-risk motion data. Each sample integrates images, trajectories, structured risk labels, and QA pairs to target both scene understanding and motion risk prediction. Data generation adopts a human-

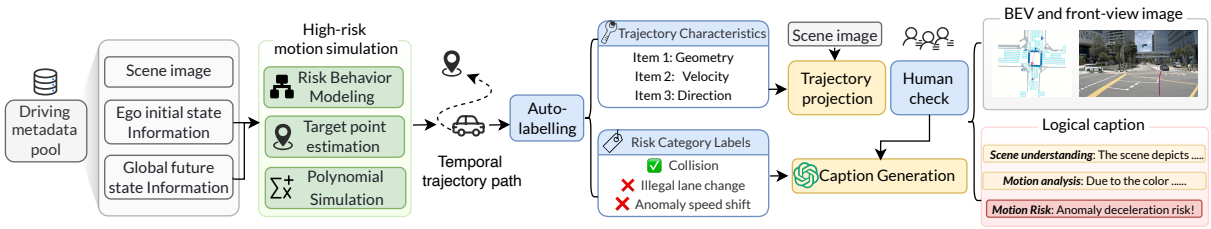


Figure 2: The proposed high-risk motion data synthesis framework consists of the following stages: high-risk motion simulation based on polynomial simulation, automatic labeling of motion attributes, quality check with human-in-the-loop, and caption generation.

in-the-loop and GPT-4o (Achiam et al., 2023) co-design. As shown in Figure 2, the pipeline involves high-risk trajectory simulation, automated annotation of motion attributes, GPT-4o generation of cause texts under label constraints, and final scene QA aligned to the task template.

3.2.1 High-Risk Motion Simulation

High-risk motion simulation is central to data synthesis. We systematically design and generate high-risk scenarios along three dimensions: ego behavior, interactions with other agents, and environmental/map constraints.

High-Risk Scenario Selection. Based on statistics of real traffic accidents and hazardous driving behaviors, we select one or more representative risky events for each dimension. For ego behavior, we focus on Hard Braking and Abnormal Acceleration to assess precise understanding of ego states; for interactions, we simulate Collision events to reflect prediction of other agents' intents (e.g., vehicles, pedestrians) and recognition of interaction risks; for environment/map constraints, we select illegal lane-change scenarios to test understanding and compliance with static elements such as lane markings and drivable regions.

Rule Definition for High-Risk Scenarios. To ensure realism and clear evaluation criteria, each scenario is precisely specified. For example, collisions use a minimum safe-distance threshold; hard acceleration/deceleration use magnitude and duration thresholds; abnormal lane changes are judged by the geometric relationship between the vehicle trajectory and lane lines/curbs. Vehicle dynamics are considered (normal speed ranges, trajectory smoothness/continuity) to guarantee physical feasibility and logical consistency.

Simulation of High-Risk Scenarios. For each original nuPlan (Caesar et al., 2021) clip, given initial states (position, speed, acceleration) and short-horizon environmental evolution, we generate targeted risky cases under the predefined rules. Concretely, we first infer key target points/areas likely to trigger the risky event, based on scenario rules and initial/goal states. Then, to balance dynamics feasibility with flexibility/diversity, we adopt a polynomial-based trajectory generator: using initial speed, acceleration, position, intended duration, and the inferred risky target as constraints, we solve a unique polynomial trajectory that satisfies them; finally, we densely sample the polynomial to obtain a smooth, complete trajectory pointing toward the preset risky state.

3.2.2 Automated Labeling and Quality Check

After synthesizing large quantities of high-risk trajectories, we perform detailed automated labeling and strict human quality control.

Automated Feature Extraction and Labeling. Within the simulator context, we automatically extract detailed kinematic features from each trajectory, including high-precision coordinates (x, y) at each time step, instantaneous speed, acceleration, curvature, and other geometric/physical attributes to support downstream risk analysis and training.

Trajectory Projection and Human-in-the-Loop Quality Control. With global-view trajectories, we project BEV trajectories onto the ego front-view image plane using the scene's camera intrinsics/extrinsics and coordinate transforms. This step is crucial for vision-based risk perception. Using the scenario definitions, human reviewers filter massive preliminary samples, removing physically implausible cases (e.g., instantaneous motion violating Newtonian mechanics), semantic mismatches, or trajectories not cleanly/projectably visible due to occlusion/truncation. After this process, we retain about 10,000 high-quality, representative high-risk motion samples.

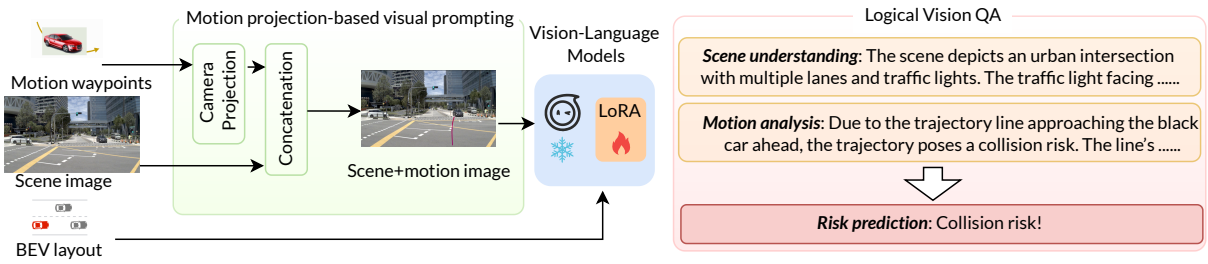


Figure 3: **DriveMRP-Agent.** Inputs include the ego front-view image and future waypoints. VTP bridges the numeric waypoint and image representation modality gap while the optional BEV layout provides global geometric priors for ambiguous boundaries. Built on Qwen2.5-VL-7B (Bai et al., 2025), the agent follows a logical QA chain from scene understanding to motion analysis to risk prediction.

3.2.3 Caption Generation

Given risk-category labels, ego front-view images, and detailed kinematic features, we prompt GPT-4o (Achiam et al., 2023) to integrate these three sources into a unified textual description (caption). Each caption typically contains three parts: (i) a comprehensive scene description (road type, traffic density, weather if available, and key static/dynamic elements); (ii) an in-depth analysis of the planned motion, explaining potential risk points and possible violations of traffic rules or safety principles; and (iii) a clear final risk prediction stating the likely risk type (e.g., collision, hard braking, illegal lane change). After automatic caption generation, we further conduct a human consistency check and remove samples whose descriptions or trajectory analyses are inconsistent with the visual evidence, projected trajectory, risk label, risk definition, or basic traffic logic. This structured approach yields a large-scale multimodal dataset with images, trajectories, text, and risk labels for VLM training and risk prediction research.

3.3 DriveMRP-Metric

DriveMRP-Metric evaluates *risk-scene reasoning* and *motion-risk prediction*.

Risk-Scene Reasoning. For textual reasoning outputs, we use ROUGE-1-F1, ROUGE-2-F1, ROUGE-L-F1 (Chin-Yew, 2004) and BERTScore (Zhang et al., 2019) to evaluate lexical overlap, phrase-level consistency, and semantic similarity between generated explanations and reference explanations. In addition, for synthetic samples with available simulation metadata and risk grammar, we conduct an evidence-card logic evaluation. Each sample is paired with structured evidence, including the supported risk label, speed trend, target object, and expected causal keypoints. We use Qwen3.6-27B (Bai et al., 2025) as the judge to evaluate generated outputs in terms of label consistency, causal-cue matching, rule validity, and hallucination. This provides a complementary measure of whether the model's reasoning process is reasonable and consistent with the risk logic.

Motion-Risk Prediction. We use Accuracy, Recall, and F1 Score (Goutte and Gaussier, 2005) to evaluate category-level correctness, coverage, and precision–recall trade-off under class imbalance. Full evidence-card definitions and judge settings are provided in Appendix F.

3.4 DriveMRP-Agent

3.4.1 Architecture

DriveMRP-Agent is instantiated with Qwen2.5-VL-7B (Bai et al., 2025), while the framework can be plugged into other VLM backbones. As shown in Figure 3, inputs are ego front-view images plus Visualized-Trajectory Prompting: future BEV waypoints are projected onto the front view as visible elements, with time and speed encoded by color, thereby bridging the numeric waypoint and image representation gap. Optional BEV rendering and waypoint coordinates provide global geometric priors for ambiguous-boundary cases. A unified visual encoder is followed by vision-language alignment and decoding along a guided path from scene understanding to motion analysis to risk prediction. The final performance still depends on the backbone's visual-spatial ability.

3.4.2 Training Strategy

To balance performance and efficiency, we apply LoRA (Hu et al., 2022) for parameter-efficient fine-tuning and use chain-of-thought (CoT) (Wei et al., 2022) prompting to elicit intermediate reasoning. Training has two stages. First, Synthetic SFT uses DriveMRP-10K to learn the reasoning routine and

Table 1: Overall performance on the DriveMRP-10K test set. Risk-scene reasoning includes ROUGE-L-F1 and BERTScore as reference-similarity metrics, and LogicScore and Faithful Rate as evidence-card faithfulness metrics. "–" indicates that the metric is not available for that baseline; complete evidence-card scores are in Appendix F.1.

Method	**Risk-Scene Reasoning** ↑				**Motion-Risk Prediction** ↑		
	ROUGE-L-F1	BERTScore	LogicScore	Faithful Rate	Accuracy	Recall	F1 score
Open-source VLMs							
EM-VLM4AD-Base	11.09	45.70	–	–	–	–	–
InternVL2.5-8B	30.94	69.65	40.73	29.53	25.80	15.50	19.11
Qwen2.5VL-7B	30.71	68.72	41.60	27.66	27.39	16.19	16.43
Qwen3VL-8B	29.55	68.69	39.76	32.71	32.45	19.63	24.46
Kimi-VL-16B	32.21	70.06	42.36	29.59	28.99	18.27	22.07
Closed-source VLMs							
GPT-4o	31.97	69.62	38.62	27.39	29.79	18.28	22.00
Gemini-2.0-Flash	29.55	66.79	42.32	33.36	32.45	20.11	25.09
DriveMRP-Agent	**52.49**	**81.06**	**89.87**	**88.03**	**88.83**	**75.69**	**79.11**

output format from scene to waypoints to risk; GPT-4o-generated texts are label-conditioned explanatory supervision, not the source of risk labels. Second, Cross-Distribution Few-Shot Adaptation lightly updates LoRA parameters on small real-world samples to reduce the domain gap. During inference, the model follows a QA template covering scene understanding, motion analysis, and risk conclusion.

4 Experiments

4.1 Implementation Details

Synthetic domain. Risk evaluation for autonomous driving has long been constrained by the lack of annotated data covering hazardous trajectories. To address this, we construct **DriveMRP-10K** with 10,000 samples synthesized from nuPlan (Caesar et al., 2021) and split 80/20 for training and testing. Each sample follows a unified template including images, trajectory, risk labels, and QA pairs. Trajectories are sampled in the BEV coordinate system at a fixed step and projected to the front view via VTP. Each sample includes cause explanations generated under label constraints and human quality control to satisfy DriveMRP-Metric for scene understanding and risk reasoning.

Real domain. To assess real-world usability and cross-domain generalization, we curate an independent real-world dataset of 2,000 risky clips, sourced from actual vehicle logs with human-annotated risk labels while preserving original camera imaging characteristics. This dataset supports two experiments: (i) zero-shot evaluation; (ii) few-shot adaptation with a stair-step setting of 10 to 80 samples per class for fine-tuning, holding out the rest strictly for testing. Because the real and synthetic domains differ in camera intrinsics/extrinsics, streetscape textures, and imaging aspect ratios, this enables objective assessment of cross-domain generalization.

Model selection. We evaluated multiple VLMs with diverse model architectures and parameters. The proprietary models included GPT-4o (Achiam et al., 2023) and Gemini-2.0-Flash (Comanici et al., 2025). The open-source models included EM-VLM4AD-Base (Gopalkrishnan et al., 2024), InternVL2.5-8B (Chen et al., 2024b), Qwen2.5VL-7B (Bai et al., 2025), Qwen3VL-8B (Bai et al., 2025), Kimi-VL-16B (Team et al., 2025), and LLaVA-1.5-7B (Liu et al., 2024).

4.2 Main Results

Our results show that DriveMRP-Agent excels at fine-grained scene understanding and precise risk identification. As presented in Table 1, compared with state-of-the-art general-purpose VLMs and driving-specialized VLMs, DriveMRP-Agent consistently leads across multiple composite metrics. This demonstrates both the effectiveness and robustness of the method, and further indicates that vanilla VLMs (including strong closed-source models) are not equivalent to usable risk evaluators. As shown in Figure 4(a), qualitative examples exhibit DriveMRP-Agent's ability to identify hazardous trajectories in extreme scenarios and to provide clear, interpretable rationales, which competing methods generally lack.

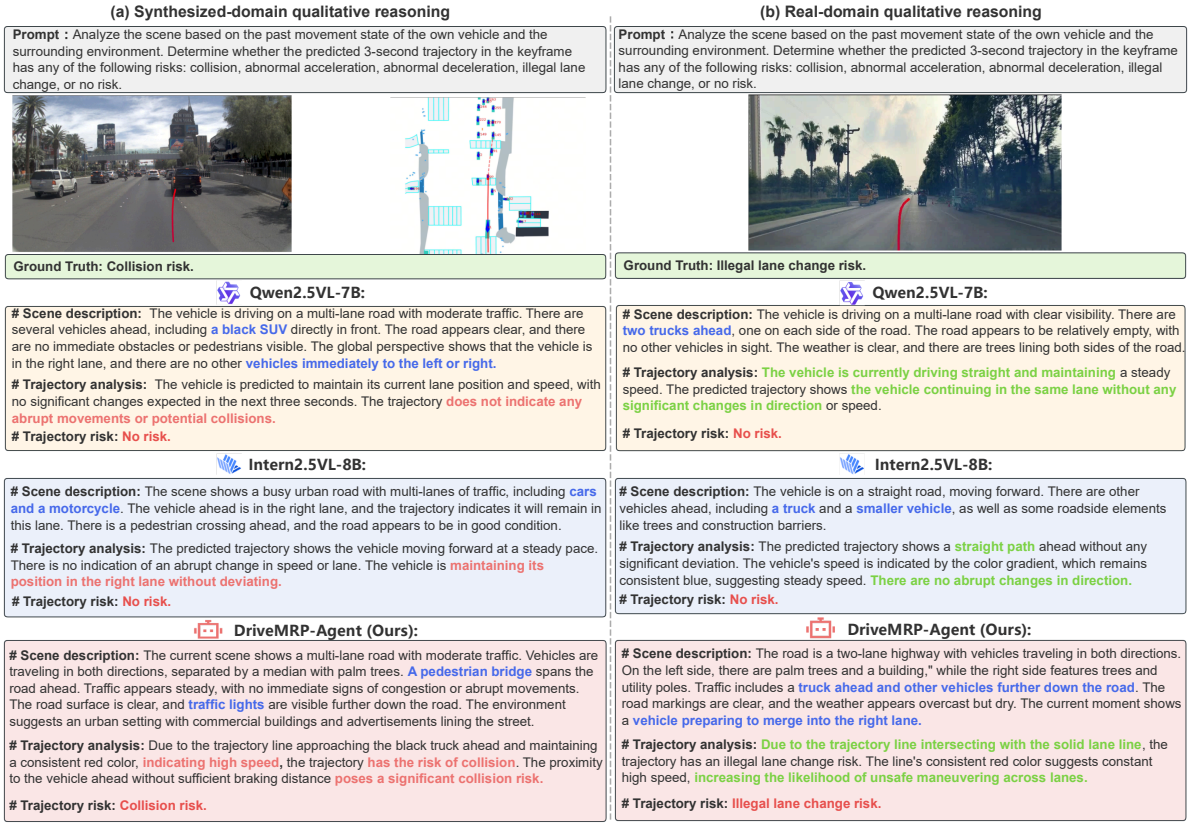


Figure 4: Qualitative reasoning results of DriveMRP-Agent compared with baseline VLMs. Each case shows the prompt, ground-truth risk label, the VTP overlay of future waypoints on the front view, and model outputs including scene description, trajectory analysis, and predicted risk (with causes). (a) Synthesized-domain qualitative reasoning. (b) Real-domain qualitative reasoning.

Table 2: Performance of VLMs trained without and with DriveMRP-10K. LogicScore is evaluated by the evidence-card judge on DriveMRP-10K.

Method	ROUGE-L-F1 ↑	LogicScore ↑	Accuracy ↑	F1 score ↑
LLaVA-1.5-7B	27.23	23.43	22.34	0.85
+DriveMRP-10K	**45.57**	**55.43**	**59.04**	**25.99**
Qwen3VL-8B	29.55	39.76	32.45	24.46
+DriveMRP-10K	**44.16**	**72.52**	**70.48**	**67.72**
Qwen2.5VL-7B	30.71	41.60	27.39	16.43
+DriveMRP-10K	**52.49**	**89.87**	**88.83**	**79.11**

Effectiveness of synthetic data. As shown in Table 2, training on DriveMRP-10K substantially boosts scene understanding and risk recognition across multiple mainstream VLMs. For the Qwen2.5-VL backbone, the LogicScore also rises from 41.60 to 89.87, suggesting that synthetic risk supervision improves not only label prediction but also evidence-grounded reasoning.

Generalization to the real domain. Table 3 evaluates unseen real-risk clips covering collisions, lane-change line violations, emergency braking, and related hazards. DriveMRP-Agent is trained only on DriveMRP-10K in the zero-shot setting and still improves over general VLMs by about 30% absolute accuracy, reaching 68.25%. With few-shot tuning, accuracy rises to 84.50%. As shown in Figure 4(b), real-domain examples further validate the method under domain shift.

4.3 Ablation Studies

Input-channel ablations for DriveMRP-Agent. We ablate three key channels: (1) BEV layout, (2) front-view image, and (3) VTP in Table 4. Results show that VTP is most critical: removing the projection and keeping only "BEV + front view" yields clear drops on both "scene description" and "risk classification," indicating that overlaying future waypoints as visual elements in the front view effectively bridges the modality gap and strengthens spatio-temporal reasoning between waypoints and environment. By contrast, the BEV channel contributes as a gain factor, suggesting its global geometric prior brings relatively limited additional benefits under ambiguous boundaries, occlusions, or incomplete visual cues. Together, these comparisons verify the necessity and effectiveness of our multi-channel, explicitly geometry-aligned design

Table 3: Real-domain generalization comparison on our in-house dataset of real-world risky scenarios. We compare zero-shot transfer, trained solely on synthetic data, against few-shot tuning with 80 samples per risk category. Real-domain evidence cards are not reported because the real clips do not contain the structured simulator metadata needed to construct reliable evidence cards; see Appendix F.2.

Method	**Scene Understanding** ↑				**Motion Risk Prediction** ↑		
	ROUGE-1-F1	ROUGE-2-F1	ROUGE-L-F1	BERTScore	Accuracy	Recall	F1 score
Open-source VLMs							
InternVL2-8B	53.83	19.50	33.01	71.41	35.00	19.65	20.54
Qwen2.5VL-7B	50.84	18.78	32.40	70.49	38.25	22.83	26.38
Qwen3VL-8B	52.36	17.90	32.80	70.65	41.75	26.69	31.54
Kimi-VL-16B	55.07	22.27	35.41	72.07	35.75	20.44	22.36
Closed-source VLMs							
GPT-4o	57.59	22.43	36.74	73.42	40.25	24.44	28.12
Gemini-2.0-Flash	51.81	18.45	33.14	69.99	38.75	28.21	35.93
DriveMRP-Agent (Zero-shot)	**62.38**	**30.75**	**42.68**	**76.43**	**68.25**	**51.57**	**61.97**
DriveMRP-Agent (Few-shot)	**70.38**	**44.97**	**54.81**	**81.53**	**84.50**	**66.93**	**72.63**

Table 4: Ablation of three input channels in DriveMRP-Agent: (1) BEV layout, (2) front-view image, and (3) VTP. The first row keeps the trajectory as text coordinates without VTP.

(1)	**(2)**	**(3)**	**Scene Understanding**	**Motion Risk Prediction**	
			ROUGE-L-F1 ↑	Accuracy ↑	F1 score ↑
✓	✓		52.20	85.90	76.89
	✓	✓	52.28	87.23	78.12
✓	✓	✓	**52.49**	**88.83**	**79.11**

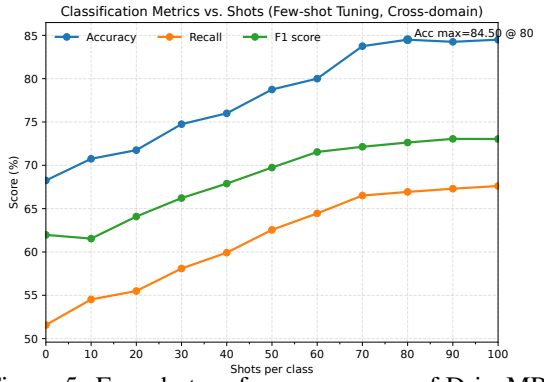


Figure 5: Few-shot performance curve of DriveMRP-Agent tuning with different shots on real-risk scenarios.

for fine-grained risk recognition.

Few-shot tuning strategy. Figure 5 shows a few-shot tuning curve of DriveMRP-Agent on real-risk scenarios by fine-tuning with 10-100 samples per class. Performance steadily improves and plateaus around 80 samples per class, where overall accuracy reaches 84.50%. This indicates that small target-domain labels can substantially shrink the composite domain gap by correcting imaging distribution differences (camera parameters, streetscape style, aspect ratio) and partly aligning label semantics (more contextual, fuzzy boundaries in real data vs. rule-based, discretized definitions in synthetic data). This provides practical guidance for few-shot hyperparameters and data sampling toward rapid deployment of DriveMRP-Agent in real applications, while further corroborating strong cross-domain generalization.

5 Conclusion

We introduce an integrated framework for autonomous driving motion risk prediction by addressing data scarcity and modality misalignment. Our BEV motion synthesis framework models risk across ego behavior, traffic interactions, and environmental constraints to construct DriveMRP-10K. On this basis, the backbone-flexible DriveMRP-Agent employs visual trajectory prompting to align numeric waypoints with image representations and uses multi-step logical QA for interpretable risk reasoning. Experiments show that our approach outperforms general and driving-specialized VLMs, transfers to real high-risk clips under zero-shot evaluation, and benefits from cross-domain few-shot adaptation, validating the effectiveness of the proposed synthetic-to-real learning loop.

6 Limitations

While DriveMRP-Agent enhances motion risk prediction, it is not a low-level closed-loop controller and should not be used as the sole basis for safety-critical vehicle control or certification. Its main limitations are: (i) **sensory scope**: front-view VTP is efficient but may miss side/rear blind spots, although Appendix G shows a preliminary surround-

view extension; (ii) **spatial bottleneck**: current VLMs still have limited precise 3D geometric understanding, so performance depends on the chosen backbone's visual-spatial ability; and (iii) **open-loop protocol**: our evaluation focuses on candidate-trajectory risk diagnosis rather than continuous closed-loop planning. Future work will pursue multi-view/temporal fusion, stronger geometry-language representations, model compression, and integration into closed-loop planners with human oversight.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

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

Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. 2020. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631.

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

Qitong Chen, Dong Zhao, Congzhi Liu, Meng Yang, and Yehui Shi. 2024a. Autonomous vehicles in mixed-autonomy traffic: game theoretic human-like decision making countermeasures. *Complex Engineering Systems*, 4(4):N–A.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, and 1 others. 2024b. 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*, pages 24185–24198.

Lin Chin-Yew. 2004. Rouge: A package for automatic evaluation of summaries. In *Proceedings of the Workshop on Text Summarization Branches Out, 2004*.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, and 1 others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*.

Daniel Dauner, Marcel Hallgarten, Tianyu Li, Xinshuo Weng, Zhiyu Huang, Zetong Yang, Hongyang Li, Igor Gilitschenski, Boris Ivanovic, Marco Pavone, and 1 others. 2024. Navsim: Data-driven non-reactive autonomous vehicle simulation and benchmarking. *Advances in Neural Information Processing Systems*, 37:28706–28719.

Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. 2017. Carla: An open urban driving simulator. In *Conference on robot learning*, pages 1–16. PMLR.

Shenyuan Gao, Jiazhi Yang, Li Chen, Kashyap Chitta, Yihang Qiu, Andreas Geiger, Jun Zhang, and Hongyang Li. 2024. Vista: A generalizable driving world model with high fidelity and versatile controllability. *arXiv preprint arXiv:2405.17398*.

Akshay Gopalkrishnan, Ross Greer, and Mohan Trivedi. 2024. Multi-frame, lightweight & efficient vision-language models for question answering in autonomous driving. *arXiv preprint arXiv:2403.19838*.

Cyril Goutte and Eric Gaussier. 2005. A probabilistic interpretation of precision, recall and f-score, with implication for evaluation. In *European conference on information retrieval*, pages 345–359. Springer.

Wencheng Han, Dongqian Guo, Cheng-Zhong Xu, and Jianbing Shen. 2025. Dme-driver: Integrating human decision logic and 3d scene perception in autonomous driving. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 3347–3355.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, and 1 others. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3.

Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, and 1 others. 2023. Planning-oriented autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17853–17862.

Zhiyu Huang, Xinshuo Weng, Maximilian Igl, Yuxiao Chen, Yulong Cao, Boris Ivanovic, Marco Pavone, and Chen Lv. 2024. Gen-drive: Enhancing diffusion generative driving policies with reward modeling and reinforcement learning fine-tuning. *arXiv preprint arXiv:2410.05582*.

Bo Jiang, Shaoyu Chen, Qing Xu, Bencheng Liao, Jiajie Chen, Helong Zhou, Qian Zhang, Wenyu Liu, Chang Huang, and Xinggang Wang. 2023. Vad: Vectorized scene representation for efficient autonomous driving. *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8306–8316.

Quanyi Li, Zhenghao Peng, Lan Feng, Qihang Zhang, Zhenghai Xue, and Bolei Zhou. 2022. Metadrive: Composing diverse driving scenarios for generalizable reinforcement learning. *IEEE transactions on pattern analysis and machine intelligence*, 45(3):3461–3475.

Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Qiao Yu, and Jifeng Dai. 2024a. Bevformer: learning bird's-eye-view representation from lidar-camera via spatiotemporal transformers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Zhiqi Li, Zhiding Yu, Shiyi Lan, Jiahan Li, Jan Kautz, Tong Lu, and Jose M Alvarez. 2024b. Is ego status all you need for open-loop end-to-end autonomous driving? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14864–14873.

Bencheng Liao, Shaoyu Chen, Xinggang Wang, Tianheng Cheng, Qian Zhang, Wenyu Liu, and Chang Huang. 2022. Maptr: Structured modeling and learning for online vectorized hd map construction. *arXiv preprint arXiv:2208.14437*.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2024. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26296–26306.

Jiageng Mao, Yuxi Qian, Junjie Ye, Hang Zhao, and Yue Wang. 2023. Gpt-driver: Learning to drive with gpt. *arXiv preprint arXiv:2310.01415*.

Chen Min, Shubin Si, Xu Wang, Hanzhang Xue, Weizhong Jiang, Yang Liu, Juan Wang, Qingtian Zhu, Qi Zhu, Lun Luo, and 1 others. 2024. Autonomous driving in unstructured environments: How far have we come? *arXiv preprint arXiv:2410.07701*.

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

Junwon Seo, Jungwi Mun, and Taekyung Kim. 2023. Safe navigation in unstructured environments by minimizing uncertainty in control and perception. *arXiv preprint arXiv:2306.14601*.

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

Kimi Team, Angang Du, Bohong Yin, Bowei Xing, Bowen Qu, Bowen Wang, Cheng Chen, Chenlin Zhang, Chenzhuang Du, Chu Wei, and 1 others. 2025. Kimi-vl technical report. *arXiv preprint arXiv:2504.07491*.

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

Junming Wang, Xingyu Zhang, Zebin Xing, Songen Gu, Xiaoyang Guo, Yang Hu, Ziying Song, Qian Zhang, Xiaoxiao Long, and Wei Yin. 2024a. Hedrive: Human-like end-to-end driving with vision language models. *arXiv preprint arXiv:2410.05051*.

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

Yuqi Wang, Jiawei He, Lue Fan, Hongxin Li, Yuntao Chen, and Zhaoxiang Zhang. 2024c. Driving into the future: Multiview visual forecasting and planning with world model for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14749–14759.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Yuxiao Zhang, Alexander Carballo, Hanting Yang, and Kazuya Takeda. 2023. Perception and sensing for autonomous vehicles under adverse weather conditions: A survey. *ISPRS Journal of Photogrammetry and Remote Sensing*, 196:146–177.

Lijun Zhou, Tao Tang, Pengkun Hao, Zihang He, Kalok Ho, Shuo Gu, Wenbo Hou, Zhihui Hao, Haiyang Sun, Kun Zhan, and 1 others. 2024a. Ua-track: Uncertainty-aware end-to-end 3d multi-object tracking. *arXiv preprint arXiv:2406.02147*.

Yunsong Zhou, Linyan Huang, Qingwen Bu, Jia Zeng, Tianyu Li, Hang Qiu, Hongzi Zhu, Minyi Guo, Yu Qiao, and Hongyang Li. 2024b. Embodied understanding of driving scenarios. In *European Conference on Computer Vision*, pages 129–148. Springer.

Overview. This appendix provides a more systematic description of implementation details and supplementary experiments that could not be fully elaborated in the main paper, and is organized as follows.

Section A formally refines the motion risk prediction task, clarifying the notation for inputs and outputs, as well as the Visualized-Trajectory Prompting (VTP) representation adopted in DriveMRP-Agent.

Section B focuses on synthetic data and the risk grammar: it first describes how we select scenarios from nuPlan and unify trajectory sampling, then presents the risk-category distribution of DriveMRP-10K, and finally details the simulation rules and physical constraints for four types of high-risk behaviors, together with the label-driven explanation generation based on GPT-4o and the multi-stage quality control pipeline.

Section C introduces the real high-risk dataset, including the on-vehicle data collection and annotation process, and analyzes the main domain gaps between the real and synthetic domains in terms of risk definition and imaging geometry.

Section D provides implementation details of DriveMRP-Metric, defining text-similarity metrics (ROUGE, BERTScore), risk classification metrics (Accuracy, Recall, F1), and evidence-card logic metrics (LogicScore, Faithful Rate).

Section E supplements the architecture and training configuration of DriveMRP-Agent, explaining the multi-channel visual inputs (including BEV layout and VTP trajectory projection), the unified instruction template design, and the key hyperparameters used in synthetic-domain SFT and real-domain few-shot tuning.

Section F presents evidence-grounded explanation faithfulness and counterfactual trajectory reasoning, focusing on whether model explanations and predictions are grounded in trajectory evidence.

Section G studies robustness and input extensions, including BEV perturbation and surround-view input.

Section H analyzes inference efficiency and scalability under different hardware, output modes, and backbone sizes.

Section I presents qualitative case studies on both synthetic and real domains, comparing DriveMRP-Agent and baseline models under different risk scenarios.

A Task Definition and Notation

In this work, we focus on a motion risk prediction task: given multi-modal observations of the current driving scene and a candidate future trajectory of the ego vehicle, the model needs to determine whether this trajectory poses any safety risk, and output both a corresponding risk category and a brief explanation of the underlying cause.

Concretely, the inputs can be divided into three parts.

1) Raw sensor observations. We use the current front-view RGB image frame to represent the visual appearance of the scene.

2) Ego trajectory information. The motion of the ego vehicle is represented by a sequence of discrete trajectory waypoints. We uniformly sample the planned future trajectory at a fixed time interval Δt, and denote the future trajectory over T_{fut} time steps as

$$\tau_{1:T_{\text{fut}}} = \{(x_k^{\text{ego}}, y_k^{\text{ego}})\}_{k=1}^{T_{\text{fut}}},$$

where $(x_k^{\text{ego}}, y_k^{\text{ego}})$ denotes the ego position at step k in the BEV ground-plane coordinate system. It is important to emphasize that, in the model inputs, we only provide these discrete 2D coordinates. In DriveMRP-Agent, the model only receives the VTP-projected trajectory image; velocity, acceleration, and other higher-order motion quantities are not explicitly provided.

3) Optional BEV scene representation. In some configurations, we further provide a BEV feature map rendered from HD maps and scene objects, denoted as B_t for the t-th time step. This representation encodes lane geometry, intersection topology, static obstacles, and approximate positions of surrounding traffic participants, and is used to help the model understand global geometric constraints and interaction patterns.

On the output side, for each sample, the model is required to produce two types of results simultaneously: (i) a discrete motion risk category $r \in \mathcal{R}$, where $\mathcal{R}$ includes the "safe" class as well as multiple high-risk modes; and (ii) a short natural-language explanation e that describes the current scene and highlights the key factors leading to the predicted risk judgment.

B Synthetic Data and Risk Grammar Details

B.1 Scenario and Trajectory Sampling Strategy

When constructing DriveMRP-10K, we do not directly use all scenarios from nuPlan; instead, we deliberately avoid almost static, low-interaction "boring segments." Specifically, we prioritize scenarios in which the ego vehicle is moving and exhibits clear interactions with the surrounding environment or other traffic participants, such as those with official tags like `starting_high_speed_turn` and `on_intersection`. Through this filtering process, we retain about 9,000 representative dynamic scenarios, which serve as the base scenario pool for subsequent risk synthesis.

At the trajectory level, we uniformly sample and synthesize within a fixed time horizon of "future 3 seconds." For each selected nuPlan scenario, we discretize the ego's future trajectory at 10 Hz, i.e., we record the ego position in the BEV coordinate system every 0.1 seconds, obtaining a reference trajectory sequence of length 30. In the risk synthesis stage, we generate high-risk trajectories on the same temporal grid of "3 seconds into the future at 10 Hz," so that synthesized trajectories are fully aligned with the original plans in terms of time steps and sampling frequency, and each trajectory is likewise represented by 30 two-dimensional coordinates.

B.2 Risk Category Distribution and Statistics of DriveMRP-10K

DriveMRP-10K contains 10,000 samples in total. Around the combination of "safe + four typical high-risk behaviors," we define five risk categories with the distribution summarized in Table 5.

Overall, as shown in Table 5, the dataset maintains an approximate ratio of "safe : high-risk $\approx 1:3$," where the safe class accounts for about 25% and the four high-risk behaviors collectively account for 75%. Within the high-risk portion, the sample counts of different behaviors are on a similar order of magnitude, which is intended to drive the model to learn diverse motion risk patterns.

We also attempted to synthesize an "off-road" risk category (e.g., the ego vehicle crossing road boundaries or entering clearly non-drivable regions). However, in our collected real high-risk clips, there are currently no corresponding off-road risk annotations. To keep the label space of the synthetic and real domains as consistent as possible, we ultimately did not include this risk type in the official label space of the current version of DriveMRP-10K, and the associated samples are not counted within the 10,000 samples described above. In the future, if real data with reliable off-road labels become available, this category can naturally be added as an extended risk type within the same risk grammar and evaluation framework.

Table 5: Distribution of risk categories in DriveMRP-10K.

Category	#Samples
Safe	2,500
Collision	2,000
Abnormal Acceleration	2,000
Abnormal Deceleration	2,000
Illegal Lane Change	1,500
Total	10,000

B.3 Risk Rules and Physical Constraints

In the high-risk motion simulation module, we explicitly construct a configurable "risk grammar" along three dimensions: ego behavior, interactions with other traffic participants, and environment/map constraints. In the current version, we mainly synthesize four representative types of risky scenarios: abnormal acceleration/deceleration, collision risk, illegal lane-change risk, and off-road / drivable-area exit. All risk scenarios are generated under a unified temporal configuration: starting from the current ego state and environment, we simulate within a 3.0 s prediction horizon and sample future trajectories at 0.1 s resolution, yielding 30 future trajectory waypoints. For each risk type, we first design triggering conditions and target regions (e.g., potential collision points, target lane centerlines, sampled locations inside non-drivable areas) based on its semantic definition and physical constraints, then use a polynomial trajectory generator to synthesize candidate trajectories that satisfy the initial state and target constraints under vehicle dynamics limits, and finally apply a unified set of risk-grammar checks to filter out trajectories that are physically infeasible or would trigger other risk types. In this way, we obtain semantically "relatively pure" high-risk scenarios for each category.

1) Abnormal acceleration/deceleration. This risk type models cases where the ego vehicle exhibits abnormally large acceleration or deceleration while traveling in a straight line, and is used to evaluate the model's sensitivity to abnormal ego longitudinal states. We start from the current ego state (position, speed, and heading) and synthesize straight-line trajectories strictly along the current heading direction. Concretely, we first check whether it is meaningful to trigger this type of anomaly based on the initial speed: when the ego speed is in the range 1–7 m/s, we allow synthesis of abnormal acceleration scenarios; when the ego speed is at least 10 m/s, we allow synthesis of abnormal deceleration scenarios, so as to avoid introducing unrealistic "abnormal" behaviors at near-zero or extreme speeds. For frames that pass this speed filter, we randomly sample multiple longitudinal acceleration magnitudes from the interval 8–15 m/s^2, and construct the corresponding acceleration vectors according to the current heading: for abnormal acceleration, the acceleration is applied along the ego's forward direction and starts at $t = 2.0$ s; for abnormal deceleration, the acceleration is applied in the opposite direction and starts at $t = 1.0$ s. During trajectory generation, we discretize over 0–3.0 s with a step of 0.1 s: before the trigger time, the ego moves in straight-line constant-speed motion; from the trigger time onward, it enters a constant-acceleration or constant-deceleration phase, whose duration is enforced to be at least 1.0 s; if the speed would exceed the upper bound 15 m/s or drop below 0 m/s during this phase, we clip it to $[0, 15]$ m/s and, once the limit speed or standstill is reached, continue with that constant speed. This yields a complete "straight-line constant speed + constant accel/decel + speed-limited cruising/stop" trajectory sequence. Finally, we apply the generic risk-grammar checks to all candidate abnormal trajectories, removing those that would lead to collisions with other traffic participants in the next 3 s or clearly leave the drivable area and thus fall into other risk categories. We retain only scenarios whose semantics are predominantly driven by longitudinal abnormal acceleration/deceleration while remaining safe along other dimensions.

2) Collision risk. Collision-risk scenarios are designed to synthesize ego trajectories that are guaranteed to geometrically collide with a specific target within a 3 s window, in order to evaluate the model's ability to recognize risks in complex interaction and near-collision settings. Concretely, in the current frame, we first select all visible dynamic targets (such as lead vehicles, oncoming vehicles, pedestrians, and cyclists) within the ego's forward view cone, and obtain their future 3.0 s center trajectories from logged history. At the same time, we extract the ego's "reference future trajectory" (i.e., the planned trajectory from nuPlan) and uniformly resample it at 0.1–3.0 s to obtain 30 reference waypoints, which are later used to measure the similarity in shape between synthesized trajectories and the original plan. For each visible target, we traverse its future center trajectory and take the positions at $t = 0.1, 0.2, \ldots, 3.0$ s as potential "collision target points," and assume that the ego must reach each such point at the corresponding time T. To synthesize a physically feasible ego trajectory, we construct independent cubic polynomials in the BEV coordinate system for x and y, such that at $t = 0$ they satisfy the current ego position and velocity constraints, and at $t = T$ they exactly reach the specified target point. We then sample the resulting polynomial trajectory over 0–3.0 s at 0.1 s intervals, compute the instantaneous acceleration norm at each time step, and require it not to exceed a preset maximum acceleration limit (constraining both longitudinal and lateral components); if any time step violates this physical constraint, the candidate trajectory is immediately discarded. For trajectories that satisfy the acceleration bound, we further check, along the entire time sequence, whether the ego's shape (e.g., bounding box) intersects with the target's shape at any time step: only those that exhibit at least one geometric overlap within the 3 s window are considered "inevitable collision" candidates. In addition, we require all sampled points of the trajectory to remain inside the current forward view cone, so as to avoid synthesizing collisions that would be invisible to the visual input. Finally, for each target semantic category, we select a subset of representative trajectories from all candidates based on two criteria: high shape similarity to the original ego plan and earlier collision time. These trajectories both resemble realistic planned motions and indeed lead to collisions within 3 s, and are used as training and evaluation samples for the collision-risk category.

3) Illegal lane-change risk. Illegal lane-change scenarios aim to synthesize trajectories that are geometrically smooth and seemingly reasonable, but violate lane-keeping rules, i.e., the ego laterally cuts from the current lane into a neighboring

lane within the current field of view, without immediately causing collisions or going off-road, yet clearly breaking lane-line or lane-change regulations. Specifically, in the current frame, we first parse the current ego lane and the geometry of its left and right adjacent lanes from HD maps and perception outputs; if no candidate target lane exists on one side, we do not construct illegal lane-change scenarios in that direction.

For each available target lane, we extract a smooth lane centerline from its boundary polygon and treat it as the "desired driving trajectory" after the lane change. We then sample multiple candidate target points along the centerline segment ahead of the ego and within the current view cone, imposing a lower bound on their distance to the ego to filter out overly close, clearly unreasonable aggressive lane changes. From these candidates, we uniformly subsample a set of representative target positions. For each target position, we construct a cubic polynomial trajectory over a fixed 3.0 s horizon that satisfies the initial and final state constraints: at $t = 0$ it matches the current ego position and velocity, and at $t = 3.0$ s it reaches (or lies near) the target point, while spatially achieving a smooth lateral transition from the current lane to the target lane. For each candidate lane-change trajectory, we sample it at 0.1 s intervals over 0–3.0 s and check, at each time step, that both longitudinal and lateral accelerations do not exceed a preset maximum acceleration limit and that all sampled points remain within the current view cone; if any sample violates these physical or visibility constraints, the trajectory is deemed infeasible and discarded. On top of this, we invoke the generic risk-grammar checks (such as `CheckCollision()` and `CheckOffroad()`) to filter out trajectories that would result in collisions or leave the drivable area within 3 s, and keep only those that successfully transition from the current lane into the adjacent lane while violating lane-line or lane-change rules as illegal lane-change risk scenarios.

4) Off-road / drivable-area exit. Off-road risk scenarios describe motion patterns where the ego vehicle deviates from its lane in the future and enters clearly non-drivable regions (such as green belts, medians, and pedestrian areas), and are used to assess the model's understanding and sensitivity to the boundary between drivable and non-drivable areas. In the current frame, we first use HD maps and perception results to extract the geometric contours of all non-drivable regions within the ego's forward field of view; if no obvious non-drivable region is present, we consider that no off-road scenario can be constructed for this frame. Otherwise, we uniformly or randomly sample multiple target points inside these non-drivable regions, and treat them as potential "off-road target locations." For each sampled target point, we enumerate several possible arrival times T on the 0.1–3.0 s temporal grid, and for each (target, T) pair generate a trajectory from the current ego state that reaches this point at time T (using a cubic-polynomial generator similar to the above), while checking over the full 3.0 s window that the acceleration always remains below a preset threshold to ensure dynamical feasibility. For each candidate trajectory, we define an off-road risk metric: we compute the proportion of sampled trajectory points that lie inside non-drivable regions and record the first time step at which the trajectory leaves the drivable area. Only when this off-road ratio exceeds a specified threshold and the first off-road time lies within the 3.0 s window is the trajectory regarded as a genuine off-road candidate. We then apply collision-checking in the same manner: if a trajectory has a potential collision with other traffic participants within the time window, it is treated as a collision-dominated scenario and excluded from the off-road category, in order to avoid label ambiguity. Finally, among all candidate trajectories that pass physical and collision filters, we select a small set of representative samples based on off-road ratio and first exit time, preferring those that leave the drivable area earlier and to a larger extent while remaining globally smooth and feasible. These samples are used as training and evaluation data for the off-road risk category.

B.4 GPT-4o Prompt for Synthetic Explanations

After completing high-risk trajectory simulation and structured risk-label generation, we further need to automatically produce natural-language explanations consistent with the labels for each sample, which serve as causal supervision when training DriveMRP-Agent. To this end, we use GPT-4o as a text generator under a label-conditioned setting: instead of letting GPT-4o decide whether a trajectory is risky, we explicitly provide the current front-view image, the projected visualization of the future trajectory, the corresponding risk label, and the risk definitions, and require the model to "explain backward" why this trajectory has been

Prompt:
###Objective
The image below shows a driving scenario, viewed from the vehicle's front. Lines depict the trajectory for the next three seconds, along with labels indicating the risk level of the future trajectory (artificially labeled risk levels). You need to first analyze the current scenario, then analyze the reasons for the risk levels in the trajectory labels based on the trajectory for the next four seconds. If a label is "No Risk," analyze the reasons for its lack of risk.
###Image at the Current Moment
The image shows a line depicting the trajectory for the next four seconds, with color representing the vehicle's speed. The closer the color is to blue, the lower the speed; the closer the color is to red, the higher the speed. According to projection principles, the distance from the bottom of the image (closer to the vehicle) to the middle of the image (farthest from the vehicle) (starting from the vehicle's position) represents the future trajectory moving from the current position to the furthest point over time.
###Trajectory Labels
Trajectory labels indicate the potential risks of driving along the trajectory indicated by the current line.
<**collision risk**>
###Risk Definition:
1. If the line color changes from blue to red, it indicates a rapid increase in vehicle speed within three seconds, i.e., abnormal acceleration.
2. If the line color changes from red to blue, it indicates that the vehicle speed rapidly decreases within three seconds, i.e., abnormal deceleration has occurred.
3. If the line approaches a vehicle, pedestrian, or obstacle within the field of vision, it indicates a collision risk.
4. If the trajectory line heads towards grass, sidewalks, or other inaccessible areas, it indicates an off-road risk.
5. If the trajectory line intersects with a lane line (solid line) within the field of view, it means that there is a risk of illegal lane change.
###Output
You only need to output the scene description and trajectory analysis. Trajectory analysis mainly analyzes the reasons why the trajectory is marked as a risk based on the scene description, the shape and color changes of the main line and trajectory line, and the risk definition. In the trajectory analysis, your description should be that due to XXX, the trajectory has XXX risk (or no risk). The output format of trajectory analysis is:
#Scene description: (no more than 150 words)
#Trajectory analysis: (no more than 70 words)
In addition, you do not need to output anything else

output:
#Scene description: The current scene shows a vehicle moving forward under a covered area with multiple cars parked or moving slowly ahead. Pedestrians are visible near the right side, and a taxi lane sign is present. The road markings indicate lanes for vehicles. The environment appears busy but orderly, with no immediate obstacles directly in front of the vehicle. The lighting suggests daytime conditions with clear visibility.
#Trajectory analysis: Due to the trajectory line approaching other vehicles ahead and the lack of sufficient space to maneuver safely, the trajectory has the risk of collision. The line's consistent color suggests steady speed, but proximity to other cars increases collision likelihood.

Risk Label: collision risk

Figure 6: Overview of the caption generation. Given a text prompt and the front-view image together with VTP, GPT-4o is used to generate a scene-level description that captures the key objects, interactions, and motion-related risk factors in the driving scenario.

assigned a given risk label based on the provided Risk Label and Risk Definition. Concretely, we feed GPT-4o with the front-view image overlaid with the next 3 seconds of future trajectory, where the trajectory color encodes speed magnitude (colors closer to blue indicate lower speeds, and colors closer to red indicate higher speeds), and on the text side we provide the discrete risk label of the sample along with a fixed list of risk-definition items. To ensure consistent output style and facilitate subsequent evaluation and parsing, we constrain GPT-4o, via a unified instruction template, to generate only two short text segments: one for scene description (*Scene description*), and one for trajectory analysis (*Trajectory analysis*), which respectively describe the overall driving environment and explain "why the current trajectory corresponds to the given risk label."

In the instruction, we explicitly specify the model's objective: it must first perform holistic scene understanding, and then, based on "the known risk label + trajectory shape and color changes + a unified set of risk definitions," provide a causal analysis, rather than re-predicting the risk category itself. To make this process more intuitive, we include a concrete example in the

supplementary material: as illustrated in the figure, given a sample with a certain risk label and its VTP overlaid image, GPT-4o produces a scene description that summarizes the road layout and surrounding agents, followed by a trajectory analysis that explicitly links the trajectory's shape, color evolution, and relative position to nearby targets to the specified risk. The generated output simultaneously adheres to the unified format constraints and tightly couples geometric and visual cues with the final risk conclusion, forming a clear causal chain "from visual evidence to risk judgment," which is exactly the type of explanatory supervision we aim to provide for DriveMRP-Agent.

B.5 Quality Control in the Synthesis Process

After high-risk motion simulation and text generation, we apply multiple rounds of quality control to the synthetic data, in order to ensure that samples in DriveMRP-10K are consistent in terms of physical plausibility, visual usability, and risk semantics. First, we perform manual spot checks and filtering on the synthesized ego trajectories, focusing on removing extremely unrealistic cases in dynamics or geometry, such as excessively large steering angles or abrupt sharp turns within a single frame, abnormally aggressive lateral shifts at high speed, or trajectories that oscillate back and forth drastically in a very short time; although such trajectories may pass simple acceleration bounds numerically, they are clearly incompatible with feasible real-vehicle motion patterns.

Second, we check the quality of VTP projection and discard those trajectory samples that cannot be properly displayed in the front-view image, including cases where coordinate anomalies cause the trajectory to be projected outside the image boundaries, where the trajectory lies almost entirely outside the camera's field of view, or where the projected trajectory is severely misaligned with the actual road geometry; these samples would heavily interfere with subsequent image-based risk recognition and are thus removed at the quality-control stage. Third, for illegal lane-change scenarios, because the HD maps provided by nuPlan do not explicitly distinguish between solid and dashed lane markings, we cannot fully rely on automatic rules to decide whether a lane change is "crossing a solid line" or not. Therefore, we perform additional manual screening over the constructed lane-change trajectories, preferentially keeping those whose lateral transitions clearly cross lane boundaries or ride on lane lines, and discarding some borderline cases whose geometry is closer to "legal lane changes" or "mild deviations," so as to strengthen the semantic purity of the illegal lane-change category. After these multiple rounds of filtering and manual refinement, we ultimately remove about 2,000 samples that do not meet our quality criteria and retain 10,000 samples that are physically plausible, visually usable, and semantically clean in terms of risk, which together constitute the DriveMRP-10K dataset used in this paper.

C Real-World High-Risk Dataset Details

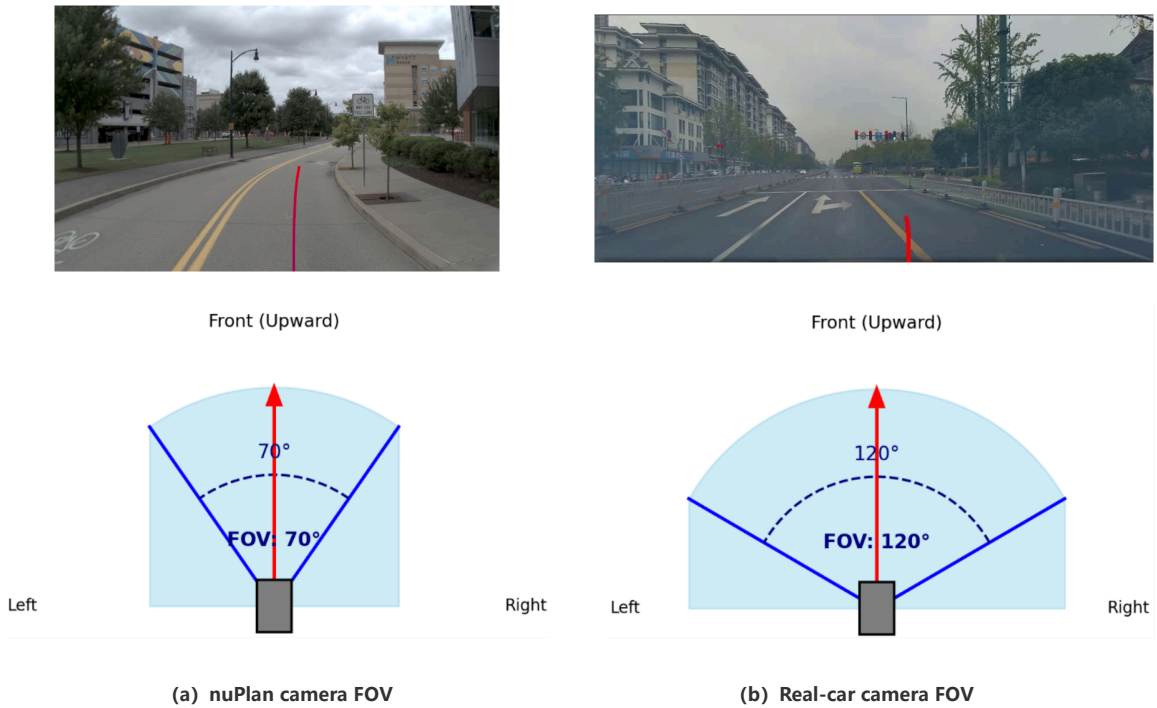
C.1 Data Source and Collection Protocol

Our real-world high-risk data mainly comes from autonomous driving experiments conducted on test vehicles. We deploy in-house autonomous driving models onto real cars and run them on urban roads and surrounding environments, with a safety driver continuously monitoring and ready to take over at any time. Whenever the safety driver subjectively perceives a potential risk in the current autonomous behavior (e.g., following too closely, aggressive lane changes, or insufficient distance to surrounding traffic participants), a logging mechanism is triggered: sensor images, ego states, and the model's predicted future trajectories around that time are cached, and annotators then assign appropriate risk labels to the recorded trajectories. The resulting samples thus focus on "high-risk behaviors that actually arise during model operation under real road conditions."

For the language component—scene understanding and risk analysis texts—we follow exactly the same procedure as in the synthetic domain: for each real high-risk sample, we use the same prompt as in Section B.4, feeding GPT-4o with the front-view image overlaid with the future trajectory, the pre-assigned risk label, and the risk definitions, and asking it to generate structured "#Scene description" and "#Trajectory analysis" texts. In this way, we obtain language descriptions in the real domain that share the same format and structure as those in the synthetic domain, which facilitates unified evaluation and training.

C.2 Domain Gap Between Synthetic and Real Data

Although we align the risk types and text formats as much as possible, there remains a significant domain gap between the synthetic and real domains,



(a) nuPlan camera FOV

(b) Real-car camera FOV

Figure 7: Comparison of front-view images and camera fields of view (FOV) between the nuPlan environment and the real-world data.

mainly in terms of how risk is defined and how scenes are imaged. First, on risk definition, synthetic data is entirely driven by pre-designed rules and thresholds, such as judging "collision or off-road" from geometric relations, so its labels follow strictly deterministic criteria. In contrast, risk labels in the real dataset come from the combined judgment of human safety drivers and annotators, placing more emphasis on driving intuition" and overall context; for borderline cases, their decisions often do not perfectly match the rule-based definitions. This difference between rule-driven" and "human-intuition-driven" labeling constitutes a major semantic gap that the model must overcome when transferring across domains.

Second, in terms of visual imaging, the two domains also differ noticeably, as illustrated in Fig. 7: real-world data is captured at an original resolution of 3840×2160 and uniformly resized to 960×540 for model input, whereas nuPlan's front-view camera images have an original resolution of 1600×900. Moreover, the horizontal field of view (FOV) is inconsistent: the real vehicle's front camera has an FOV of about $120°$, while nuPlan's front camera FOV is around $70°$. This implies that, even for similar future trajectories, the projected spatial scale on the image plane, the extent of visible surroundings, and the overall composition will differ systematically, which in turn affects how the model perceives distance" and closeness" to surrounding objects. In addition, real data is influenced by road types, camera mounting positions, and environmental lighting conditions, leading to differences in image composition and texture style compared to the nuPlan simulation environment.

Taken together, the discrepancies in risk semantics (human vs. rule-based) and imaging geometry/resolution/viewpoint form a substantial domain gap. This helps explain why the model's zero-shot performance in the real domain still has room for improvement, and why the proposed few-shot tuning is able to significantly reduce this gap.

D DriveMRP-Metric: Implementation and Extended Metrics

D.1 Language Metrics for Scene Understanding and Textual Explanations

For the scene-understanding and risk-explanation parts, we treat the model's full output text (including both the scene description and the trajectory analysis) as a single natural-language sequence,

Table 6: Few-shot tuning results of DriveMRP-Agent on the real-domain benchmark. "#Shots/class" denotes the number of labeled cases per risk category used for tuning.

#Shots/class	Scene Understanding ↑				Motion Risk Prediction ↑		
	ROUGE-1-F1	ROUGE-2-F1	ROUGE-L-F1	BERTScore	Accuracy	Recall	F1 score
0 (zero-shot)	62.38	30.75	42.68	76.43	68.25	51.57	61.97
10	66.14	38.81	48.99	79.19	70.75	54.52	61.54
20	68.00	41.04	50.89	80.15	71.75	55.49	64.09
30	69.11	42.74	52.24	80.74	74.75	58.09	66.22
40	69.63	43.59	53.27	81.10	76.00	59.92	67.89
50	69.81	44.09	53.69	81.21	78.75	62.55	69.74
60	69.91	44.12	53.72	81.22	80.00	64.45	71.54
70	70.19	44.23	53.76	81.23	83.75	66.51	72.14
80	**70.38**	**44.97**	**54.81**	**81.53**	**84.50**	**66.93**	**72.63**
90	70.33	44.97	54.44	81.47	84.25	67.30	73.05
100	70.44	44.84	54.44	81.45	84.50	67.61	73.04

and evaluate its proximity to the reference text using standard text-generation metrics. Suppose there are N samples in total; for the i-th sample, let the model output be denoted by p_i and the reference text by r_i.

1) ROUGE. We use the F1 scores of ROUGE-1, ROUGE-2, and ROUGE-L to measure n-gram coverage and longest-common-subsequence consistency. Let $\mathrm{rouge1_f}(p_i, r_i)$, $\mathrm{rouge2_f}(p_i, r_i)$, and $\mathrm{rougeL_f}(p_i, r_i)$ denote the ROUGE-1/2/L F1 values for the i-th sample, respectively. The three metrics are then defined as:

$$\text{ROUGE-1-F1} = \frac{1}{N}\sum_{i=1}^{N} \mathrm{rouge1_f}(p_i, r_i),$$

$$\text{ROUGE-2-F1} = \frac{1}{N}\sum_{i=1}^{N} \mathrm{rouge2_f}(p_i, r_i),$$

$$\text{ROUGE-L-F1} = \frac{1}{N}\sum_{i=1}^{N} \mathrm{rougeL_f}(p_i, r_i).$$

In practice, we apply stemming to the texts to reduce the impact of simple inflectional variations.

2) BERTScore-F1. To capture semantic similarity, we adopt the F1 variant of BERTScore. Let $\phi(\cdot)$ be a pretrained BERT encoder that maps a text to token-level contextual embeddings. For each pair (p_i, r_i), BERTScore computes cosine similarities between predicted tokens and reference tokens, and takes bidirectional maximum matching to obtain precision P_i, recall R_i, and an F1 score $\mathrm{F1}_i$. The overall BERTScore-F1 is defined as:

$$\text{BERTScore-F1} = \frac{1}{N}\sum_{i=1}^{N} \mathrm{F1}_i.$$

In our implementation, we use `bert-base-uncased` as the encoder and perform batch computation on GPU.

D.2 Risk Classification Metrics

In DriveMRP-Metric, risk recognition is not produced by a separate classification head, but instead is inferred from the model-generated explanation text: we parse the "trajectory risk" segment from the output and compare it with the corresponding "trajectory risk" segment in the reference explanation, yielding a discrete correct/incorrect judgment at the category level. Formally, let there be M valid samples in the evaluation set. For the i-th sample, we parse the risk-category label y_i from the reference text and parse the predicted label $\hat{y}_i$ from the model output. Based on this label pair $(y_i, \hat{y}_i)$, we define three standard classification metrics.

1) Risk_Accuracy. The overall accuracy measures how often the model outputs the correct risk conclusion, and is defined as:

$$\text{Risk_Accuracy} = \frac{1}{M}\sum_{i=1}^{M} \mathbb{1}[\hat{y}_i = y_i],$$

where $\mathbb{1}[\cdot]$ is the indicator function, which equals 1 when the predicted and true labels are identical, and 0 otherwise.

2) Risk_Recall. Since we care about the coverage over different risk categories, we use macro-averaged recall. Let $\mathcal{C}$ be the set of all risk categories that actually appear in the ground-truth labels. For each category $c \in \mathcal{C}$, we can compute its recall $\mathrm{Recall}(c)$; the macro recall is then:

$$\text{Risk_Recall} = \frac{1}{|\mathcal{C}|}\sum_{c\in\mathcal{C}} \mathrm{Recall}(c),$$

where $\mathrm{Recall}(c) = \mathrm{TP}_c/(\mathrm{TP}_c + \mathrm{FN}_c)$, and TP_c and FN_c denote the number of true positives and false negatives of category c, respectively. This metric reflects, on average, how often different risk types are missed (i.e., the model's "miss rate" across classes).

3) Risk_F1. Similarly, we compute the F1 score for each category and then take the macro average as the overall Risk_F1:

$$\mathrm{Risk_F1} = \frac{1}{|\mathcal{C}|} \sum_{c \in \mathcal{C}} \mathrm{F1}(c),$$

where

$$\mathrm{F1}(c) = \frac{2 \cdot \mathrm{Precision}(c) \cdot \mathrm{Recall}(c)}{\mathrm{Precision}(c) + \mathrm{Recall}(c)},$$

and $\mathrm{Precision}(c)$ and $\mathrm{Recall}(c)$ are the precision and recall of category c, respectively. Macro-averaged F1 balances precision and recall across classes under label imbalance, and provides a more comprehensive view of the model's risk-recognition quality over different categories.

D.3 Evidence-card Logic Metrics

In addition to reference-similarity metrics and risk-classification metrics, we report two evidence-card logic metrics on synthetic samples where structured simulation metadata and risk grammar are available. These metrics are designed to evaluate whether the generated reasoning is consistent with the underlying risk evidence.

For each synthetic test sample, we construct an evidence card from the simulation log and risk grammar. The evidence card records the supported risk label, speed trend, target object or lane-line cue, and expected causal keypoints. We then use Qwen3.6-27B (Bai et al., 2025) as the judge to evaluate the model output according to four aspects: label consistency, causal-cue matching, rule validity, and hallucination.

LogicScore. LogicScore measures the overall consistency between the generated explanation and the structured evidence card. It aggregates the judge's assessment of whether the predicted risk label, causal cue, and risk-rule reasoning are compatible with the evidence recorded in the simulation log. A higher LogicScore indicates stronger agreement between the model's reasoning process and the risk logic.

Faithful Rate. Faithful Rate measures the proportion of outputs that are judged to be faithful to the evidence card. An output is counted as faithful when it gives a risk conclusion and causal explanation that are consistent with the supported risk label, the relevant scene or trajectory cue, and the corresponding risk rule, without introducing unsupported or contradictory causes.

These two metrics are reported for DriveMRP-10K, where structured evidence cards can be reliably constructed. For the real-domain benchmark, we do not report evidence-card scores because the real-world clips do not contain complete simulator metadata or causal event logs needed to construct reliable evidence cards.

In summary, DriveMRP-Metric combines reference-similarity metrics, risk-classification metrics, and evidence-card logic metrics. ROUGE and BERTScore-F1 evaluate the textual similarity of scene descriptions and explanations; $\mathrm{Risk_Accuracy}$, $\mathrm{Risk_Recall}$, and $\mathrm{Risk_F1}$ measure category-level motion-risk prediction; and LogicScore/Faithful Rate further assess consistency with structured risk evidence on DriveMRP-10K.

E DriveMRP-Agent Architecture and Training Details

E.1 Model Architecture

DriveMRP-Agent is built on top of Qwen2.5-VL-7B as the base model, while the overall framework is plug-and-play and can be instantiated with other vision–language backbones; its final performance still depends on each backbone's visual-spatial ability. The input to the model consists of two channels: (i) an image modality, including a global-view BEV rendering and an ego-view front camera image; and (ii) a trajectory-visualization modality, where the ego vehicle's future 3-second trajectory, represented by 30 discrete waypoints in the BEV coordinate system, is projected onto the front-view image to form Visualized-Trajectory Prompting (VTP).

In implementation, we project the BEV waypoints from world coordinates onto the camera plane using the camera's intrinsic and extrinsic parameters, and draw the resulting trajectory on the front-view image. A color gradient from blue to red is used to indicate the change in speed from low to high along the trajectory. We only render line segments whose projected endpoints fall within a rea-

sonable region of the image, so as to avoid invalid or out-of-view visualizations. In the final input to DriveMRP-Agent, the visual stream thus consists of two images: an optional BEV-view global layout image, encoding priors about surrounding vehicles and lane structures, and a front-view image overlaid with the next 3 seconds of the ego trajectory, encoding local appearance and the trajectory–environment relationship.

E.2 Training Hyperparameters and Strategy

For training, we adopt a two-stage scheme of synthetic-domain supervised fine-tuning (SFT) followed by real-domain few-shot adaptation, and use LoRA for parameter-efficient fine-tuning. In the first stage, we perform SFT on DriveMRP-10K to align the base Qwen2.5-VL-7B model with our task. Concretely, we use the `llama-factory` framework and train on 8 H20 GPUs, with `finetuning_type` set to `lora`, `lora_rank` set to 8, and `lora_target` set to `all`, i.e., inserting LoRA adapters into all trainable modules to obtain sufficient expressive power under a moderate parameter budget. The training dataset is our constructed DriveMRP-10K. For optimization, we use AdamW with an initial learning rate of 1.0×10^{-4}, a cosine learning-rate schedule, and a total of 3.0 training epochs. During training, we enable `bf16` precision and FlashAttention-2 to improve efficiency.

In the second stage, we perform cross-domain few-shot adaptation on real high-risk data to reduce the gaps between the synthetic and real domains in imaging, distribution, and label semantics. In this stage, we still update only the LoRA parameters while freezing the backbone of Qwen2.5-VL-7B, which reduces overfitting risk and facilitates deployment and model migration. The few-shot configurations follow the main paper: for each risk category, we use 10, 20, …, 80 labeled samples for fine-tuning and plot performance curves under different shot settings, while keeping the remaining majority of real-world samples strictly as a held-out test set that is never used for training or model selection. Hyperparameters are largely inherited from the SFT stage without substantial changes.

In practice, this "synthetic pre-training + real-domain adaptation" strategy significantly improves real-domain risk recognition performance while preserving the synthetic-domain capability. As shown in Table 6, with 80 samples per class, the risk-classification accuracy in the real domain improves from about 68% to 84.5%.

F Evidence-Grounded and Counterfactual Reasoning

This section evaluates two complementary aspects of trajectory-conditioned reasoning: (i) whether the generated explanation is consistent with structured causal evidence, and (ii) whether the prediction changes when the future trajectory changes while the visual scene is fixed.

F.1 Evidence-Grounded Explanation Faithfulness

For DriveMRP-10K, we construct evidence cards for the test samples using simulator metadata and the risk grammar. Each evidence card contains four groups of fields: (1) the supported risk label, (2) kinematic evidence such as speed trend and acceleration/deceleration pattern, (3) geometric evidence such as the target object, lane-line, or drivable-area cue, and (4) expected causal keypoints for the corresponding risk logic.

Judge input and criteria. For each model output, the judge receives the VTP image, the generated answer, and the corresponding evidence card. We use Qwen3.6-27B (Bai et al., 2025) as the judge and ask it to assess five dimensions: (i) risk-label consistency, (ii) primary causal-cue match, (iii) causal-rule validity, (iv) evidence grounding, and (v) contradiction-free reasoning. We report LogicScore, Faithful Rate, Causal Keypoint F1, Wrong-Cause Rate, and Contradiction Rate. GPT-4o GT Caption is included only as a supervision-quality check: it is label-conditioned and should not be interpreted as a zero-shot predictor.

Table 7: Evidence-grounded explanation faithfulness evaluation on DriveMRP-10K. Lower is better for Wrong Cause and Contradiction; higher is better for the other metrics. GPT-4o GT Caption is used for supervision-quality checking rather than as a zero-shot predictor.

Output	Accuracy	LogicScore	Faithful	Keypoint F1	Wrong Cause	Contradiction
Qwen3VL zero-shot	36.97	39.76	32.71	30.66	4.26	28.99
Qwen2.5-VL zero-shot	28.19	41.60	27.66	24.76	0.53	3.19
GPT-4o zero-shot	30.59	38.62	27.39	23.83	3.19	15.43
DriveMRP-Agent	**88.83**	**89.87**	**88.03**	**85.00**	**0.80**	2.66
GPT-4o GT Caption*	100.00	98.38	97.87	94.98	2.13	**1.60**

Table 7 shows that DriveMRP-Agent obtains much higher LogicScore, Faithful Rate, and Causal Keypoint F1 than zero-shot VLMs. This indicates that its explanations are largely consistent with the

structured evidence and causal rules, complementing the reference-similarity and risk-classification metrics reported in the main paper.

F.2 Scope of Evidence-Card Evaluation

We apply evidence-card logic evaluation only to DriveMRP-10K. The reason is that a reliable evidence card requires structured causal metadata: the risk rule that generated the sample, the target object or lane-line cue, the speed/acceleration trend, and the expected causal keypoints. These fields are available in the synthetic pipeline because the risk grammar and simulator state are controlled. In contrast, the in-house real-world clips contain camera images, predicted trajectories, and human risk labels, but do not provide exhaustive object-level interaction metadata, lane-line legality metadata, or a complete causal event log for each risky maneuver. Constructing evidence cards for real data would therefore require additional human annotation of causal evidence and could introduce unreliable or subjective cards. For this reason, Table 3 keeps the original real-domain metrics and does not report Logic/Faith scores; we treat real-domain evidence-card annotation as future work.

F.3 Counterfactual Trajectory Reasoning

To test whether DriveMRP-Agent reacts to the future trajectory rather than memorizing the visual scene, we construct about 80 counterfactual trajectory sets. Within each set, the front-view scene is held fixed, but the future trajectory overlay is replaced so that the same image corresponds to different risk types. This scene-fixed, trajectory-swapped protocol directly targets trajectory-conditioned reasoning: a scene-only shortcut cannot solve it, because the visual background is unchanged and the correct label depends on the replaced trajectory.

Table 8: Counterfactual trajectory test with the scene held fixed and only the future trajectory changed.

Method	ROUGE-1-F1	ROUGE-2-F1	ROUGE-L-F1	BERTScore	Accuracy	Recall	F1 score
DriveMRP-Agent	68.47	41.02	51.79	80.91	87.72	71.80	76.63

The aggregate accuracy in Table 8 should be read together with the scene-fixed protocol rather than as a standard i.i.d. accuracy number. For each counterfactual set, the visual background is unchanged while the VTP trajectory and risk label vary; thus, when two variants in the same set have different labels, correctly classifying both variants necessarily requires a prediction switch caused by the trajectory overlay. The high performance under this protocol indicates that the model is sensitive to trajectory evidence instead of relying only on scene appearance.

G Robustness and Input Extensions

This section examines sensitivity to imperfect geometric priors and the extensibility of VTP beyond a single front camera.

G.1 BEV Perturbation Robustness

To simulate imperfect upstream geometric priors, we perturb BEV lane-line elements with random noise and line distortions while keeping the front-view image and VTP trajectory unchanged. This setting tests whether the model collapses when the optional BEV prior is noisy, rather than evaluating a different trajectory or a different risk label.

Table 9: BEV perturbation robustness.

Setting	Accuracy ↑	Recall ↑	F1 score ↑
Clean BEV	88.83	75.69	79.11
Perturbed BEV	87.50	74.08	78.23

G.2 Surround-View Extension

We also evaluate a surround-view input setting to test whether VTP extends beyond the single-front-view default. This experiment keeps the same risk categories and evaluation protocol, but provides additional camera views as visual context. We treat it as an extension rather than the default setting because it increases inference cost.

Table 10: Surround-view extension. Front-view VTP remains the default efficient setting; surround-view input provides additional visual context with higher latency.

Method	Accuracy ↑	Recall ↑	F1 score ↑
DriveMRP-Agent	88.83	75.69	79.11
DriveMRP-Agent + surround view	92.11	76.50	77.60

Table 9 shows that BEV perturbation causes only a modest drop (1.33 accuracy and 0.88 F1), supporting the claim that DriveMRP-Agent does not rely exclusively on a clean BEV prior. Table 10 shows that the framework can be extended to multi-view input. We treat surround-view as an enhanced setting for blind spots and complex interactions rather than the default deployment setting, because it increases inference cost.

H Inference Efficiency and Scalability

We measure inference latency under different hardware, input views, and output modes. We distinguish label-only mode, which outputs the risk category and is more suitable for online warning, from CoT explanation mode, which generates a full scene description and trajectory analysis and is more suitable for explainable analysis, offline auditing, and training-data annotation. These measurements complement the accuracy-oriented experiments by clarifying the deployment trade-off between speed, input coverage, and explanation length.

H.1 Latency under Different Inputs and Output Modes

Table 11: Inference latency under different hardware, input views, and output modes.

Hardware	Input	Output	Tokens	Latency	Memory
H100×2	Front	Label	10.0	0.18s	54.4GB
H100×2	Front	CoT	163.4	0.35s	25.2GB
H100×2	Surround	Label	10.0	0.80s	54.4GB
H100×2	Surround	CoT	162.7	1.02s	54.4GB
4090×2	Front	Label	10.0	0.36s	25.2GB
4090×2	Front	CoT	162.0	0.55s	25.2GB
4090×2	Surround	Label	10.0	0.85s	42.3GB
4090×2	Surround	CoT	150.8	1.30s	42.3GB

H.2 Smaller-Backbone Scalability

Table 12: Smaller-backbone latency comparison for front-view label-only inference.

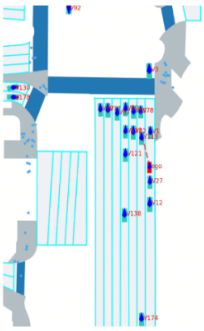
Model	Hardware	Latency/sample	Peak memory
DriveMRP-Agent-7B	H100×2	0.18s	54.4GB
DriveMRP-Agent-3B	H100×2	0.16s	38.2GB
DriveMRP-Agent-7B	4090×2	0.36s	25.2GB
DriveMRP-Agent-3B	4090×2	0.29s	14.8GB

The front-view label-only mode has potential for online risk warning, while CoT mode is slower but more informative for explanation and audit scenarios. Surround-view input improves contextual coverage but incurs substantially higher latency, motivating our choice of front-view VTP as the default efficient configuration. Smaller 3B backbones reduce memory and latency, suggesting a practical path for future deployment-oriented optimization.

I Qualitative Case Studies

We conduct qualitative case studies in both the synthetic and real domains to better understand how DriveMRP-Agent behaves compared with baseline VLMs. In the synthetic domain, we visualize representative examples from DriveMRP-10K, showing the ego front-view image with VTP overlaid (and the BEV layout). In the real domain, we present typical high-risk driving cases collected from our on-vehicle experiments using front-view images with VTP and ground-truth risk labels.

Prompt: Analyze the scene based on the past movement state of the own vehicle and the surrounding environment.
Determine whether the predicted 3-second trajectory in the keyframe has any of the following risks: collision, abnormal acceleration, abnormal deceleration, illegal lane change, or no risk.



Ground Truth: Collision risk

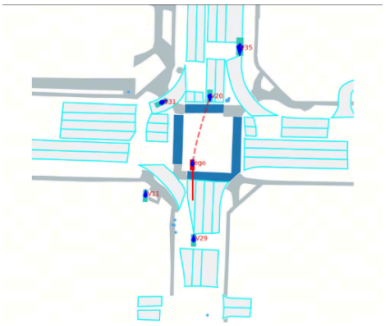
Qwen2.5VL-7B
#Scene description:
The vehicle is driving on a multi-lane road with moderate traffic, where **a red traffic light requires stopping**. A white car is directly ahead, and the road is clear of immediate obstacles or pedestrians. Urban surroundings include other vehicles in adjacent lanes, with clear visibility and no congestion.
#Trajectory analysis:
The vehicle is stationary behind the **white car**, and the predicted trajectory confirms it will **maintain its position and speed (zero movement)** for the next three seconds, aligning with the red light' s requirement to stop.
#Trajectory risk:
no risk

Intern2.5VL-8B
#Scene description:
The global perspective image depicts the vehicle's trajectory and speed changes over the next three seconds, while the vehicle's perspective shows a busy street with multiple vehicles and traffic lights. The **traffic light ahead is red**, indicating an imminent stop.
#Trajectory analysis:
The trajectory shows the vehicle slowing down to stop at the red light, with **speed gradually decreasing from red to blue—signifying normal deceleration** without abrupt movements.
#Trajectory risk:
no risk

DriveMRP
#Scene description:
The current scene shows a multi-lane road with moderate traffic, where vehicles are stopped at a **red light**. A white car is directly ahead, with other cars in adjacent lanes. **Pedestrians** are visible on the right sidewalk, separated by barriers. The traffic lights are red, requiring vehicles to stop. The urban environment features buildings, palm trees, and clear skies, indicating daytime. The road is well-maintained with clear lane markings.
#Trajectory analysis:
Due to the **trajectory approaching the black car ahead and the lack of significant color change (indicating insufficient deceleration), the trajectory poses a collision risk.** The vehicle may fail to slow down adequately before reaching the stopped car, increasing the likelihood of a rear-end collision.
#Trajectory risk:
collision risk

Figure 8: Case 1 in the Synthetic Domain

Prompt: Analyze the scene based on the past movement state of the own vehicle and the surrounding environment.
Determine whether the predicted 3-second trajectory in the keyframe has any of the following risks: collision, abnormal acceleration, abnormal deceleration, illegal lane change, or no risk.



Ground Truth:collision risk

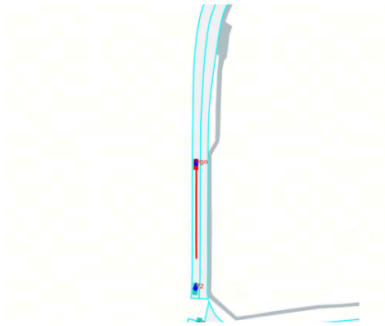
Qwen2.5VL-7B
#Scene description:
The ego vehicle is at a **traffic light-controlled intersection**, with a green arrow indicating it can proceed straight ahead. A pedestrian crossing is nearby, and a car (V29) is visible in the adjacent lane. The global perspective shows no other vehicles directly in the ego vehicle' s path.
#Trajectory analysis:
The ego vehicle is predicted to accelerate slightly and continue **straight through the intersection**, maintaining its lane position.
#Trajectory risk:
no risk

Intern2.5VL-8B
#Scene description:
The global perspective image shows the vehicle's trajectory and surrounding obstacles. The vehicle's perspective image depicts a clear intersection with a **green traffic light**, indicating it is about to proceed. A pedestrian crossing and other vehicles are present in the vicinity.
#Trajectory analysis:
The predicted trajectory moves straight ahead through the intersection at a speed consistent with the surrounding traffic flow, with no lane changes indicated. **The vehicle is in the correct lane for proceeding straight.**
#Trajectory risk:
no risk

DriveMRP
#Scene description:
The current scene shows an urban intersection with **multiple lanes and traffic lights.** The traffic light facing the vehicle is green, allowing forward movement. There are other vehicles visible ahead: **a white car turning left and a black car moving straight.** The road markings indicate a clear path forward, while pedestrians are seen crossing further down the road. The environment appears calm, with no immediate obstacles or hazards.
#Trajectory analysis:
Due to the **trajectory line approaching the black car ahead**, the trajectory poses a collision risk. **The line' s proximity to the vehicle suggests potential contact** if the vehicle continues along this path without adjusting its course or speed.
#Trajectory risk:
collision risk

Figure 9: Case 2 in the Synthetic Domain

Prompt: Analyze the scene based on the past movement state of the own vehicle and the surrounding environment.
Determine whether the predicted 3-second trajectory in the keyframe has any of the following risks: collision, abnormal acceleration, abnormal deceleration, illegal lane change, or no risk.



Ground Truth:Illegal lane change risk

Qwen2.5VL-7B
#Scene description:
The vehicle is driving on a **two-lane road** with clear lane markings, and no other vehicles are visible ahead, leaving the road appearing empty. The surroundings feature greenery and a building in the distance, and the vehicle is traveling at a moderate speed.
#Trajectory analysis:
The vehicle is predicted to **continue straight on its current path** while maintaining a steady speed. The trajectory shows no indication of sudden changes in direction or speed.
#Trajectory risk:
no risk

Intern2.5VL-8B
#Scene description:
The scene shows a **two-lane road** bordered by trees and a grassy area, with no other vehicles visible. A gentle curve lies ahead, and the road lacks lane markings **indicating a required lane change.** The environment is calm and free of immediate obstructions.
#Trajectory analysis:
The predicted trajectory shows a **straight path with slight acceleration**, denoted by the color transition from blue to red. The vehicle **maintains a consistent speed without abrupt changes,** aligning with the road' s curvature and conditions.
#Trajectory risk:
no risk.

DriveMRP
#Scene description:
The road is a **two-lane street** with clear lane markings and a slight curve ahead. The surroundings include **green grassy areas**, trees, and a few buildings in the distance. There are no visible vehicles or pedestrians on the road. The weather appears clear with some clouds in the sky. The road is relatively empty and well-maintained, with no immediate obstacles or hazards.
#Trajectory analysis:
Due to the trajectory line intersecting with the solid lane line, the trajectory poses an illegal lane change risk. The line' s consistent blue color indicates steady speed, but crossing into another lane without proper signaling or clearance presents a significant hazard.
#Trajectory risk:
Illegal lane change risk

Figure 10: Case 3 in the Synthetic Domain

Prompt: Analyze the scene based on the past movement state of the own vehicle and the surrounding environment.
Determine whether the predicted 3-second trajectory in the keyframe has any of the following risks: collision, abnormal acceleration, abnormal deceleration, illegal lane change, or no risk.

Ground Truth: abnormal deceleration risk

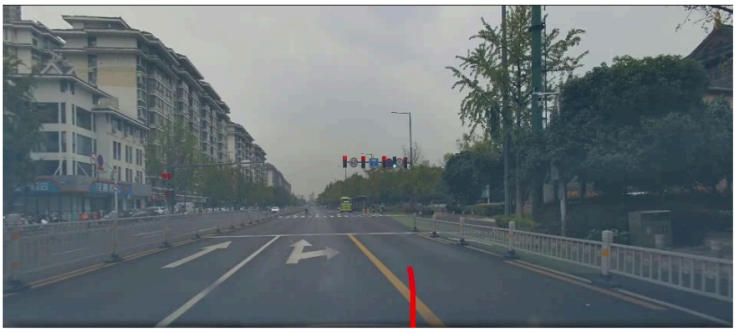
Qwen2.5VL-7B
 # Scene description:
The vehicle is driving on a multi-lane road with clear lane markings. **A white car** is ahead in the same lane, and **another vehicle** is visible further ahead. The weather is overcast, and no other vehicles are immediately surrounding the vehicle.
#Trajectory analysis:
The vehicle is maintaining a steady speed and trajectory within its lane. The predicted path shows the vehicle continuing straight without significant changes in direction or speed.
#Trajectory risk:
no risk

Intern2.5VL-8B
 # Scene description:
The vehicle is traveling on a multi-lane road with a pedestrian crossing. There is **a pedestrian walking across the road in front of the vehicle**. The vehicle's trajectory shows it moving straight ahead with no lane change, and the speed appears to be gradually increasing.
 # Trajectory analysis:
The vehicle is on a straight path, and the trajectory indicates a consistent forward motion without any lane changes. However, the pedestrian crossing the road presents a potential collision risk as the vehicle approaches.
 # Trajectory risk:
collision risk

DriveMRP
 # Scene description:
The scene depicts a multi-lane road with light traffic, where **a white van occupies the left lane and a pedestrian is crossing the right lane**. The environment features buildings on the left, trees on the right, and an overcast sky with a dry road surface. The pedestrian is in the process of moving across the lane, and the vehicle's trajectory is predicted to travel straight ahead.
 # Trajectory analysis:
The rapid color change from blue to red along the trajectory line indicates a sudden increase in speed, which paradoxically raises an abnormal deceleration risk. This contradiction suggests a potential misinterpretation of trajectory data—either the vehicle is accelerating toward a pedestrian (requiring urgent braking) or the color coding inversely represents speed (e.g., red = deceleration). In either case, the proximity to a crossing pedestrian and sudden speed change imply a high likelihood of unexpected braking to avoid a collision.
 # Trajectory risk:
abnormal deceleration risk

Figure 11: Case 1 in the Real Domain

Prompt: Analyze the scene based on the past movement state of the own vehicle and the surrounding environment.
Determine whether the predicted 3-second trajectory in the keyframe has any of the following risks: collision, abnormal acceleration, abnormal deceleration, illegal lane change, or no risk.



Ground Truth:Illegal lane change risk

Qwen2.5VL-7B
#Scene description:
The vehicle is currently driving on a multi-lane road with clear lane markings. There are no vehicles directly ahead, but there is **a bus** approaching from the opposite direction. **The traffic light is red**, indicating that the vehicle should stop. The surroundings include residential buildings and greenery.
#Trajectory analysis:
The vehicle is predicted to continue moving forward at a high speed, which is not aligned with the red traffic light and the need to stop.
#Trajectory risk:
Collision risk

Intern2.5VL-8B
#Scene description:
The image shows a vehicle approaching a traffic light intersection in an urban environment. The road is clear with visible lane markings, and there are **no pedestrians or other vehicles** in immediate proximity. **The traffic light ahead is red**, indicating the vehicle must stop.
#Trajectory analysis:
The predicted trajectory shows the vehicle maintaining a straight path, with no indication of lane changes. **The speed appears consistent, suggesting normal driving behavior.** The vehicle is positioned within its lane, and there are no obstacles or pedestrians that would pose a risk.
#Trajectory risk:
No risk

DriveMRP
#Scene description:
The image shows a multi-lane road with clear lane markings. The vehicle is currently in the rightmost lane, approaching an intersection with traffic lights ahead. There are buildings on the left and trees on the right. The road appears relatively empty, **with few vehicles visible in the distance.** The trajectory line is a straight path extending forward, indicating the vehicle's intended direction of travel.
#Trajectory analysis:
Due to the trajectory line intersecting with the lane line, the trajectory has an illegal lane change risk. The consistent red color suggests steady speed, but the crossing into another lane without proper signaling or clearance poses a significant hazard.
#Trajectory risk:
Illegal lane change risk

Figure 12: Case 2 in the Real Domain