

# What’s Hidden Matters: Identifying Planning-Critical Occluded Agents using Vision-Language Models

Amirhosein Chahe<sup>1,2</sup> Tyler Naes<sup>1</sup> Jovin D’sa<sup>1</sup> Faizan M. Tariq<sup>1</sup> Sangjae Bae<sup>1</sup>  
Lifeng Zhou<sup>2</sup> David Isele<sup>1</sup>

**Abstract**—Autonomous vehicles must safely navigate complex environments where planning-critical agents may be hidden from view. Current approaches often treat all occlusions with uniform conservatism, yielding needlessly defensive driving, or they infer hidden spaces without estimating the impact on the planner. This work bridges the critical gap between perception and planning by enabling Vision-Language Models (VLMs) to identify and reason about the specific hidden agents that are most critical to the ego-vehicle’s trajectory. We introduce a novel framework that uses Planning KL-divergence (PKL), an information-theoretic metric, to systematically identify and rank occluded agents based on their impact on the ego vehicle’s plan. Using this planning-aware ranking, we employ an expert VLM (GPT-5) to generate rich, structured annotations that capture the visual evidence and reasoning required for this task. We apply this framework to the nuScenes dataset to create a new benchmark focused on high-impact scenarios. We conduct comprehensive experiments on a wide range of general-purpose and domain-adapted VLMs, demonstrating that fine-tuning on our PKL-guided data yields dramatic performance improvements across all models. Notably, our results show that smaller, fine-tuned models significantly outperform their much larger zero-shot counterparts, and that our PKL-guided data selection strategy improves performance by approximately 30% over random sampling. Our work presents the first systematic approach for training VLMs to focus on planning-critical occlusions, enabling more semantically grounded and efficient risk assessment in autonomous driving.

## I. INTRODUCTION

Safe navigation requires that autonomous vehicles reason not only about what their sensors can see but also about what they cannot see. While recent advances in perception have enabled impressive object detection capabilities [1], [2], a fundamental challenge remains: not all occluded regions pose equal risk to the ego vehicle’s trajectory. Current approaches either treat all occlusions conservatively, leading to overly cautious behavior that disrupts traffic flow, or attempt to generatively complete the scene by predicting the contents of occluded areas, which can produce unreliable and potentially dangerous predictions [3], [4].

The gap between perception and planning becomes particularly pronounced when dealing with occluded agents. To help bridge this gap, planner-aware metrics like Planning

KL-divergence (PKL) were introduced to quantify how perception errors impact downstream planning [5]. We propose leveraging this concept for a different task: to identify **planning-critical hidden agents** whose presence would force a significant change in the ego vehicle’s trajectory. However, effectively identifying these high-risk scenarios requires a deep, contextual understanding that goes beyond raw geometry. Vision-Language Models (VLMs) are well suited to this challenge, bringing semantic grounding and reasoning beyond geometry [6]–[8]. Recent work has demonstrated VLMs’ ability to handle uncertainty through techniques like conformal prediction [9] and to provide interpretable risk assessments [10]. Yet, no prior work has systematically addressed the critical challenge of determining which hidden agents actually impact safe trajectory planning using planning-aware metrics and semantic reasoning. In this paper, we present the first comprehensive framework for identifying and reasoning about planning-critical occluded agents in the autonomous driving domain using VLMs. We make three key contributions:

- 1) We introduce a systematic method to **identify and rank planning-critical occluded agents** by applying Planning KL-Divergence (PKL) analysis to the nuScenes dataset [11], creating a targeted collection of high-impact scenarios.
- 2) We construct a new benchmark for occlusion reasoning by using GPT-5 [12] to generate **structured annotations** that detail the semantic context, likely agent hypotheses, and spatial reasoning for these critical events.
- 3) We fine-tune and benchmark a diverse set of VLMs on this dataset, demonstrating that training on **planning-aware data** significantly improves their ability to identify and reason about high-risk occluded agents across all model scales.

This entire data generation and annotation pipeline is illustrated in Figure 1. Our work bridges the critical gap between perception uncertainty and planning safety by enabling semantically-grounded and planning-aware reasoning about occluded agents in complex urban environments.

## II. RELATED WORK

Our research is positioned at the intersection of three key areas in autonomous driving: occlusion handling, the application of VLMs to driving scenarios, and risk-aware motion planning.

<sup>1</sup>Honda Research Institute (HRI), San Jose, CA 95134, USA.

<sup>2</sup>Drexel University, Philadelphia, PA 19104, USA.

All work was done while A. Chahe was employed by HRI. Contact: [ac4462@drexel.edu](mailto:ac4462@drexel.edu), [sbae@honda-ri.com](mailto:sbae@honda-ri.com), [disele@honda-ri.com](mailto:disele@honda-ri.com).

©2026 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

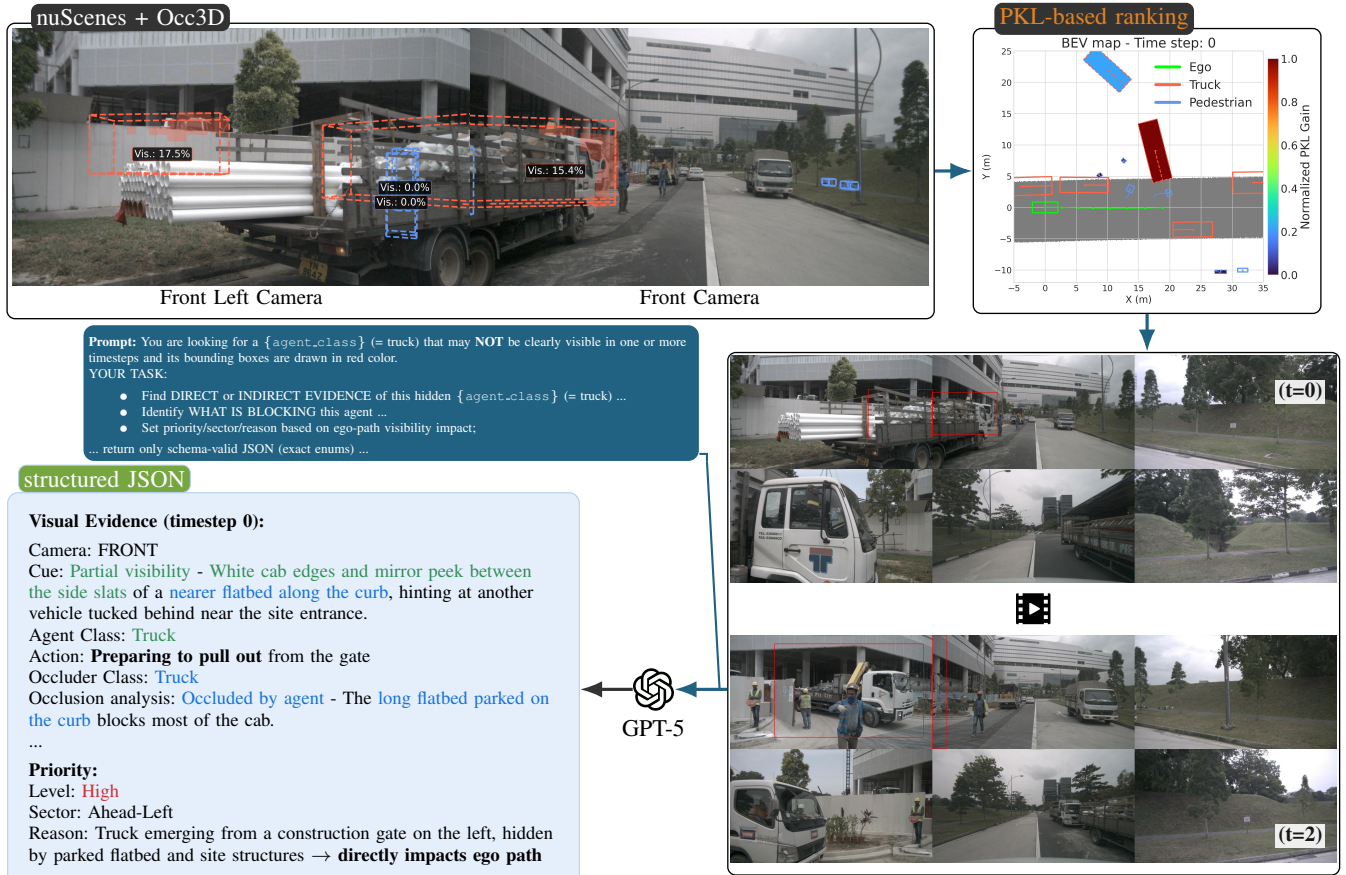


Fig. 1: **PKL-guided dataset generation pipeline.** The process begins with nuScenes data and Occ3D visibility analysis (top left), which feeds into PKL computation for identifying planning-critical hidden agents. The selected scenes with temporal multi-camera panoramas with agent bounding box overlay (bottom right,  $t=[0,2]$ ) are provided to GPT-5 alongside a structured prompt (center left) containing the ground-truth agent class and task instructions. GPT-5 operates as a semantic annotator at this final stage, producing structured JSON annotations (bottom left) with visual evidence, agent hypotheses, occlusion analysis, and survey priority.

### A. Occlusion Handling in Autonomous Driving

Reasoning about occluded regions is a long-standing and critical challenge for ensuring the safety of autonomous vehicles (AVs). Traditional approaches often adopt a conservative stance, treating any occluded space as potentially occupied by a hidden obstacle. These methods, frequently employing set-based prediction and reachability analysis, guarantee safety by planning trajectories that avoid potential collisions under worst-case assumptions [13], [14]. While provably safe, this often leads to overly cautious or “freezing” behaviors, hindering traffic efficiency, especially in dense urban environments [15]. Other methods have explored Partially Observable Markov Decision Processes (POMDPs) to model the uncertainty associated with occlusions and plan maneuvers that balance safety with progress, such as “creeping” forward to gain more visibility [16], [17]. However, these methods often lack the semantic understanding to differentiate risks based on context, thereby ignoring scene semantics and treating all blind spots equivalently.

More recently, with the rise of deep learning, Bird’s-Eye-View (BEV) perception models have become a popular paradigm for representing the 3D world from 2D camera

inputs [1], [2]. Several works have attempted to address occlusions by predicting occupancy grids, effectively guessing the state of unseen areas [3], [4], [18]. While these methods can reconstruct detailed scene geometry, they often struggle with a critical limitation: they treat all occlusions equally and may generate predictions that are not dynamically plausible or critical to the AV’s immediate plan. Recent works such as CorrBEV have begun to leverage language prototypes as a form of prior knowledge to enhance the features of partially occluded objects [19]. This emerging use of semantic, language-grounded priors to solve core perception challenges motivates our deeper exploration into the role of large-scale VLMs, which we discuss next.

### B. VLMs for Driving

Recent work adapts VLMs to driving along three fronts. *BEV-centric interfaces* treat the map as an image for language-grounded reasoning, e.g., Talk2BEV [20] and BEV-Driver [21]. *3D-aware/agentive systems* fuse multi-view perception with language models or unified world models for grounding and closed-loop control, including OmniDrive [8] and NuGrounding [22]. *Structured reasoning* frames the

driving stack as a causal graph, as in DriveLM [23] and ELM [24]. While these methods target general scene understanding, BEV QA, or end-to-end control, none explicitly focus on identifying *which occluded agents are most critical to the immediate plan*. Our work fills this gap by coupling PKL-guided selection with structured VLM outputs for occlusion reasoning.

### C. Planner-Aware Metric

To bridge the perception-planning gap, a new class of **planner-aware metrics** has been proposed [5], [25], [26]. Our methodology is grounded in an information-theoretic approach to risk assessment. We adopt the Planning KL-Divergence (PKL) metric [5], originally proposed to evaluate the real-world impact of perception errors on a planner. PKL measures the divergence between a trajectory plan made with imperfect perception and an ideal plan made with ground-truth information.

While PKL was designed for evaluating perception systems, we repurpose it as a powerful tool for data curation and annotation. By measuring the change in the ego-vehicle’s trajectory plan with and without knowledge of a specific hidden agent, we can quantify that agent’s “importance.” This allows us to create a dataset that is not just about any occluded agent, but specifically about the *most planning-critical* ones, providing a focused and challenging benchmark for occlusion reasoning.

## III. METHOD

We present a systematic approach to identify and annotate planning-critical occluded agents in autonomous driving scenarios. Our method combines visibility analysis, planning-based importance ranking, and structured annotation generation to create a comprehensive dataset for training VLMs.

### A. Problem Formulation

Let  $\mathcal{A} = \{a_1, \dots, a_N\}$  denote the set of all agents in a driving scene at time  $t$ . We partition this set into two subsets: visible agents  $\mathcal{V} \subseteq \mathcal{A}$  detected by the ego vehicle’s perception system and hidden agents  $\mathcal{H} \subseteq \mathcal{A}$  not detected due to occlusions or sensor limitations, such that  $\mathcal{A} = \mathcal{V} \cup \mathcal{H}$ .

Given a driving scene with visible agents  $\mathcal{V}$  and a  $T$ -second window of multi-camera images, our goal is to train a VLM to generate a rich, structured description for the most planning-critical hidden agents. The model outputs a comprehensive set of attributes, covering identification, contextual analysis, and risk assessment:

- **Agent Identification:** The model predicts the **agent’s class** (e.g., car, pedestrian) and provides a textual description of the **visual cue** supporting its hypothesis.
- **Contextual Analysis:** It identifies the **occluding object**, categorizes the **occlusion type**, and localizes the event to a specific **camera view**.
- **Risk Assessment:** The model predicts the hidden agent’s **action hypothesis**, assigns a **priority level** (high, medium, or low), and recommends a **Priority Sector** on the BEV map (e.g., Ahead-Left).

This detailed output provides an interpretable assessment of the latent risks within a scene.

### B. Planning-Based Importance Ranking

To identify which hidden agents are planning-critical, we employ Planning KL-divergence (PKL) [5]. The PKL measures how much the ego vehicle’s planned trajectory distribution changes when given different sets of object detections:

$$\text{PKL}(\mathcal{H}) = \sum_{0 < \Delta \leq T} D_{\text{KL}}(p_{\theta}(x_{t+\Delta} | \mathcal{A}_{\leq t}) \| p_{\theta}(x_{t+\Delta} | \mathcal{V}_{\leq t})) \quad (1)$$

where  $T$  is the planning horizon,  $p_{\theta}$  is the learned trajectory distribution,  $x_{t+\Delta}$  is the ego vehicle’s poses,  $\mathcal{A}_{\leq t}$  represents the set of all agents up to time  $t$ ,  $\mathcal{V}_{\leq t}$  represents the visible agent set up to time  $t$ , and  $D_{\text{KL}}$  denotes the Kullback-Leibler divergence.

We calculate the individual contribution, or **planning gain**, of each hidden agent  $h_i \in \mathcal{H}$  by measuring the reduction in PKL if that agent were to become visible:

$$\text{Gain}(h_i) = \text{PKL}(\mathcal{V}) - \text{PKL}(\mathcal{V} \cup \{h_i\}) \quad (2)$$

This gain quantifies how much revealing agent  $h_i$  would reduce the divergence between planned trajectories (i.e., the planning benefit of observing  $h_i$ ). We use this value to rank all hidden agents in a scene, creating a ground truth importance score.

### C. PKL-Guided Dataset Generation Pipeline

Our dataset creation pipeline, illustrated in Figure 1, consists of three main stages that integrate PKL-based importance ranking with VLM annotation.

1) *Hidden Agent Extraction:* We process the nuScenes dataset [11] using the Occ3D framework [4] to compute visibility scores for all annotated agents. For each agent  $a_i$ , we compute its visibility score:

$$v_i = \frac{A_{\text{visible}}^i}{A_{\text{bbox}}^i} \quad (3)$$

where  $A_{\text{visible}}^i$  is the area of the agent’s visible voxels splatted on the camera plane, and  $A_{\text{bbox}}^i$  is the area of the agent’s projected 3D bounding box. We classify agents with  $v_i < \tau$  as hidden agents, where  $\tau$  is a visibility threshold. In Figure 1’s example, the truck behind the flatbed has  $v_i = 0.15$ , clearly qualifying as a hidden agent.

2) *Planning-Critical Agent Selection:* For each scene, we compute PKL gains for all hidden agents using Equation 2. The BEV visualization in Figure 1 (top right) shows the PKL-based importance ranking, where warmer colors indicate higher planning impact. Among the multiple hidden agents in this scene, the truck preparing to pull out from the construction gate (shown in red) has the highest PKL gain, indicating it would most significantly impact the ego vehicle’s trajectory. We select scenes containing at least one hidden agent with substantial planning impact, ensuring our dataset focuses on safety-critical scenarios rather than routine occlusions.

3) *Structured Annotation Generation*: The key innovation of our approach lies in generating rich semantic annotations for planning-critical hidden agents. As shown in Figure 1 (bottom), we provide GPT-5 with temporal multi-camera panoramas spanning  $T$  seconds and the target hidden agent’s class and bounding box to generate structured descriptions. GPT-5 generates structured JSON annotations containing:

- **Visual Cue**: Evidence indicating the hidden agent  $\in$  {partial visibility, indirect motion responses, temporal traces, light signatures}, along with a textual description
- **Camera**: View where evidence is observed (e.g., “FRONT”)
- **Agent Class**: Hidden agent class  $\in$  {car, truck, trailer, bus, motorcycle, pedestrian, bicycle, unknown}
- **Action Hypothesis**: Predicted behavior (e.g., “Preparing to pull out from the gate”)
- **Occluder Class**: Primary occluding object’s nuScenes class (e.g., “Truck”)
- **Occlusion Type**: Occlusion pattern  $\in$  {occluded by agent, occluded by landmark, weather, lighting, other}, along with a textual description (e.g., “The long flatbed parked on the curb blocks most of the cab”)
- **Priority**: Including Priority Level  $\in$  {High, Medium, Low} and Sector in ego’s BEV  $\in$  {Ahead-Left, Ahead, Ahead-Right, Left, Right, Back-Left, Back, Back-Right}
- **Reasoning**: Explanation for the risk assessment

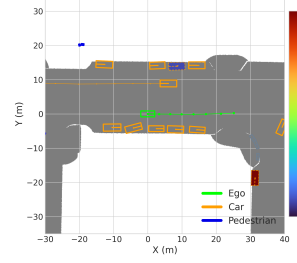
We iteratively refined the annotation process by optimizing prompts [12] to emphasize indirect visual cues over direct visibility, standardizing coordinate-system definitions for consistent localization, issuing detailed classification guidelines to reduce ambiguity. We manually audited a subset of samples to ensure annotation quality and consistency.

#### IV. EXPERIMENTS

We evaluate our approach on the nuScenes dataset, demonstrating that VLMs can effectively identify planning-critical occluded agents when fine-tuned on PKL-guided data. Our experiments show significant improvements across all model



(a) Panoramic front view at timestep 2



(b) BEV map

| Field     | InternVL3.5-Ins |           |            |         |
|-----------|-----------------|-----------|------------|---------|
|           | 4B              | GPT       | 38B        | SFT     |
| class     | car             | car       | pedestrian | car     |
| action    | turn_left       | turn_left | brake      | parked  |
| cam.view  | F.LEFT          | F.LEFT    | F.LEFT     | F.RIGHT |
| vis.cue   | partial         | indirect  | partial    | partial |
| occ.type  | landmark        | landmark  | agent      | agent   |
| occ.class | manmade         | manmade   | car        | car     |
| priority  | high            | high      | high       | medium  |
| sector    | A.Left          | A.Left    | A.Left     | A.Right |

(c) Model predictions

Fig. 2: A rainy intersection scene with an occluded car. Fine-tuning (SFT) the Intern models (from 4B to 38B) results in more plausible predictions compared to their zero-shot versions. Green = correct, red = incorrect.

families, with fine-tuned smaller models outperforming larger zero-shot counterparts.

#### A. Experimental Setup

1) *Dataset*: We construct a comprehensive dataset from nuScenes [11] using the PKL-guided pipeline described in Section III-C. To compute PKL scores, we train the planner following the methodology in [5] on the nuScenes training split, producing trajectory predictions over a 4 s horizon. PKL is then computed over  $T=2$  s windows within each scene. We set the visibility threshold  $\tau=0.4$ . Both the chunking window and threshold choices are validated in Section IV-D.1. We select scenes containing at least one hidden agent with substantial planning impact by sorting all scenes by their  $\text{PKL}(V)$  values and choosing the top 1,000 from the nuScenes training set and top 250 from the nuScenes validation set, yielding diverse occlusion scenarios from urban driving. This selection ensures our dataset focuses on safety-critical situations where hidden agents would significantly alter the ego vehicle’s trajectory if visible. For each selected scene, we use GPT-5 to generate structured JSON annotations

TABLE I: Zero-shot performance of base models. Avg. action is the mean of BLEU, ROUGE-L, and Embedding Sim. Bold = best, underline = second best.

| Model      | agent class |       | camera view |       | visual cue |       | occlusion type |       | occluder class |       | priority |       | sector |       | bleu  | action |       | sim.  | overall |           |
|------------|-------------|-------|-------------|-------|------------|-------|----------------|-------|----------------|-------|----------|-------|--------|-------|-------|--------|-------|-------|---------|-----------|
|            | acc.        | f1    | acc.        | f1    | acc.       | f1    | acc.           | f1    | acc.           | f1    | acc.     | f1    | acc.   | f1    |       | rouge  | acc.  |       | avg. f1 | avg. acc. |
| Qwen-3B    | 0.328       | 0.375 | 0.280       | 0.262 | 0.092      | 0.022 | 0.496          | 0.329 | 0.220          | 0.151 | 0.088    | 0.054 | 0.128  | 0.089 | 0.000 | 0.000  | 0.249 | 0.233 | 0.183   | 0.083     |
| Qwen-7B    | 0.432       | 0.501 | 0.252       | 0.229 | 0.124      | 0.093 | 0.508          | 0.374 | 0.248          | 0.204 | 0.192    | 0.074 | 0.352  | 0.324 | 0.000 | 0.003  | 0.253 | 0.301 | 0.257   | 0.085     |
| Qwen-32B   | 0.512       | 0.558 | 0.296       | 0.307 | 0.492      | 0.508 | 0.612          | 0.582 | 0.292          | 0.274 | 0.252    | 0.181 | 0.300  | 0.308 | 0.000 | 0.009  | 0.270 | 0.394 | 0.388   | 0.093     |
| Qwen-72B   | 0.516       | 0.565 | 0.204       | 0.125 | 0.640      | 0.587 | 0.624          | 0.583 | 0.348          | 0.320 | 0.364    | 0.323 | 0.384  | 0.338 | 0.000 | 0.012  | 0.266 | 0.440 | 0.406   | 0.093     |
| Intern-1B  | 0.644       | 0.627 | 0.000       | 0.000 | 0.000      | 0.000 | 0.000          | 0.000 | 0.000          | 0.000 | 0.000    | 0.000 | 0.000  | 0.000 | 0.000 | 0.001  | 0.246 | 0.092 | 0.090   | 0.082     |
| Intern-2B  | 0.324       | 0.424 | 0.212       | 0.190 | 0.800      | 0.726 | 0.496          | 0.329 | 0.248          | 0.169 | 0.184    | 0.057 | 0.356  | 0.342 | 0.000 | 0.012  | 0.246 | 0.374 | 0.319   | 0.086     |
| Intern-4B  | 0.648       | 0.639 | 0.088       | 0.108 | 0.264      | 0.342 | 0.264          | 0.280 | 0.084          | 0.084 | 0.168    | 0.202 | 0.112  | 0.131 | 0.000 | 0.009  | 0.266 | 0.233 | 0.255   | 0.092     |
| Intern-8B  | 0.496       | 0.544 | 0.240       | 0.227 | 0.628      | 0.563 | 0.648          | 0.617 | 0.260          | 0.224 | 0.548    | 0.388 | 0.248  | 0.201 | 0.000 | 0.013  | 0.271 | 0.438 | 0.395   | 0.095     |
| Intern-14B | 0.328       | 0.416 | 0.168       | 0.156 | 0.648      | 0.579 | 0.692          | 0.658 | 0.392          | 0.340 | 0.184    | 0.057 | 0.208  | 0.135 | 0.000 | 0.008  | 0.277 | 0.374 | 0.334   | 0.095     |
| Intern-GPT | 0.708       | 0.659 | 0.188       | 0.168 | 0.080      | 0.012 | 0.436          | 0.295 | 0.120          | 0.104 | 0.204    | 0.103 | 0.204  | 0.131 | 0.000 | 0.016  | 0.273 | 0.277 | 0.210   | 0.096     |
| Intern-38B | 0.220       | 0.295 | 0.096       | 0.043 | 0.712      | 0.605 | 0.608          | 0.551 | 0.368          | 0.316 | 0.192    | 0.080 | 0.152  | 0.083 | 0.000 | 0.010  | 0.278 | 0.335 | 0.282   | 0.096     |
| Gemma-4B   | 0.504       | 0.558 | 0.268       | 0.237 | 0.088      | 0.014 | 0.436          | 0.333 | 0.168          | 0.160 | 0.480    | 0.394 | 0.380  | 0.358 | 0.008 | 0.011  | 0.224 | 0.332 | 0.293   | 0.081     |
| Gemma-12B  | 0.300       | 0.391 | 0.280       | 0.263 | 0.076      | 0.029 | 0.464          | 0.425 | 0.228          | 0.221 | 0.344    | 0.317 | 0.176  | 0.122 | 0.000 | 0.007  | 0.242 | 0.267 | 0.253   | 0.083     |
| Gemma-27B  | 0.484       | 0.527 | 0.276       | 0.264 | 0.108      | 0.053 | 0.536          | 0.473 | 0.324          | 0.284 | 0.484    | 0.416 | 0.240  | 0.226 | 0.000 | 0.001  | 0.245 | 0.350 | 0.320   | 0.082     |
| MiniCPM    | 0.468       | 0.525 | 0.144       | 0.102 | 0.208      | 0.262 | 0.568          | 0.509 | 0.336          | 0.288 | 0.548    | 0.388 | 0.216  | 0.201 | 0.000 | 0.000  | 0.247 | 0.355 | 0.325   | 0.082     |
| DriveMM    | 0.004       | 0.008 | 0.000       | 0.000 | 0.004      | 0.008 | 0.000          | 0.000 | 0.000          | 0.000 | 0.000    | 0.000 | 0.000  | 0.000 | 0.000 | 0.000  | 0.275 | 0.001 | 0.002   | 0.092     |
| DriveLM    | 0.148       | 0.236 | 0.180       | 0.187 | 0.536      | 0.558 | 0.004          | 0.008 | 0.000          | 0.000 | 0.184    | 0.057 | 0.132  | 0.156 | 0.000 | 0.022  | 0.271 | 0.169 | 0.172   | 0.098     |
| DriveGPT4  | 0.000       | 0.000 | 0.460       | 0.311 | 0.712      | 0.592 | 0.424          | 0.266 | 0.028          | 0.037 | 0.180    | 0.056 | 0.308  | 0.276 | 0.000 | 0.000  | 0.000 | 0.302 | 0.220   | 0.000     |

as described in Section III-C.3. Our final dataset comprises 1,250 scene windows (1,000 training, 250 validation), resulting in 22,500 camera frames (18,000 training, 4,500 validation). Each window spans  $T=2$  seconds with 3 frames sampled at 1-second intervals across 6 cameras in a  $3 \times 2$  grid configuration. This yields 3,750 structured JSON annotations (3,000 training, 750 validation) generated by GPT-5, with one annotation per timestep.

2) *Evaluation Metrics*: We evaluate model performance against the ground truth annotations across the structured fields described in our method: Visual Cue, Camera, Agent Class, Action Hypothesis, Occluder Class, Occlusion Type, Priority Level, and Sector. To measure performance, we use the following metrics:

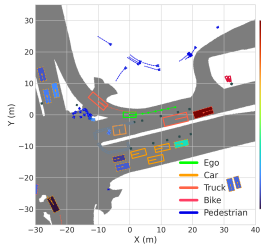
**Classification Metrics**: For all categorical fields (Agent Class, Camera, Occluder Class, Occlusion Type, Priority Level, and Sector), we compute Accuracy and F1 scores.

**Text Similarity Metrics**: For the Action Hypothesis field, we report BLEU, ROUGE-L, and embedding similarity [27].

**Overall Performance**: We report average accuracy and average F1 across all categorical fields, and average action score (mean of BLEU, ROUGE-L, and embedding similarity). These provide complementary views of classification consistency and text generation quality.



(a) Panoramic front view at timestep 0



(b) BEV map

| Field     | Gemma-3-12B  |              | Gemma-3-27B |         |
|-----------|--------------|--------------|-------------|---------|
|           | Ins          | SFT          | Ins         | SFT     |
| class     | truck        | truck        | car         | truck   |
| action    | accelerate   | moving forw. | accelerate  | stopped |
| cam.view  | FRONT        | FRONT        | F.RIGHT     | FRONT   |
| vis.cue   | temp. trace  | partial      | temp. trace | partial |
| occ.type  | landmark     | agent        | agent       | agent   |
| occ.class | construction | truck        | truck       | truck   |
| priority  | high         | high         | medium      | high    |
| sector    | Ahead        | Ahead        | A.Right     | A.Right |

(c) Model predictions

Fig. 3: An urban construction scene with a hidden truck. Fine-tuning (SFT) significantly improves the Gemma models’ predictions. Green = correct, red = incorrect.

3) *Implementation Details*: All models are fine-tuned using LoRA [28] with rank 16, targeting all linear layers including the alignment module and language model components. We employ the ms-swift framework [29] with AdamW optimizer (learning rate  $2e-5$ ), batch size 4 per GPU across 4 H100 GPUs, and train for 3 epochs. For inference, we use structured JSON output generation to ensure consistent formatting across all models.

## B. Baseline Models

We evaluate two categories of VLMs:

### General-Purpose VLMs:

- **Qwen2.5-VL** [30]: 3B, 7B, 32B, 72B parameters.
- **InternVL3.5** [31]: 1B, 2B, 4B, 8B, 14B, 38B, GPT-OSS-20B configurations.

- **Gemma-3** [32]: 4B, 12B, 27B variants.
- **MiniCPM-V 4.5** [33]: 7B parameter.

### Domain-Adapted Models:

- **DriveMM** [34]: Enhanced LLaVA-7B with driving-specific adaptations.
- **DriveLM** [23]: Mini-InternVL2-4B fine-tuned on DriveLM dataset [35].
- **DriveGPT4** [36]: Mini-InternVL2-4B fine-tuned on DriveGPT4 dataset [35].

## C. Main Results

1) *Zero-Shot Performance*: Table I presents zero-shot performance of base (Instruct) and domain-adapted models. Key observations:

*Scale Effects*: Larger models generally achieve better zero-shot performance within each family. Qwen-72B achieves the highest average accuracy (0.440) among open models, followed by InternVL3.5-8B (0.438). *Cross-Family Variations*: Despite size differences, performance varies significantly across model families. InternVL3.5-8B (8B parameters) matches Qwen-72B’s performance, suggesting architectural differences play a crucial role.

*Domain Adaptation Limitations*: Surprisingly, domain-adapted models show mixed results. While DriveLM achieves the highest average action score (0.098), DriveMM fails almost completely (0.001 avg. accuracy), and DriveGPT4 shows no action understanding. This suggests that general driving knowledge alone is insufficient for this task.

2) *Fine-Tuning Impact*: Table II demonstrates the transformative effect of supervised fine-tuning on our PKL-guided dataset:

*Dramatic Improvements*: Fine-tuning yields substantial gains across all models, with average accuracy improvements ranging from +0.09 (Qwen-32B) to +0.45 (InternVL3.5-1B). Smaller models benefit disproportionately: InternVL3.5-1B improves from 0.092 to 0.545 in average accuracy (+493%). Figure 3 provides a qualitative example where fine-tuning enables correct identification of a hidden truck and its priority level, while base (instruct) models fail to capture these critical details.

*Efficiency Gains*: Fine-tuned smaller models consistently outperform larger zero-shot variants. Notably, Qwen-3B-SFT (0.478 avg. accuracy) surpasses Qwen-72B zero-shot (0.440) while being  $24\times$  smaller, demonstrating the value of task-specific training. Figure 2 illustrates this phenomenon within the InternVL3.5 family, where the fine-tuned version consistently outperforms larger base (instruct) variants.

*Component-Specific Gains*: Fine-tuning benefits vary substantially across evaluation targets. Visual cue detection shows the largest gain (+950%), followed by priority prediction (+518%), occluder classification (+390%), camera view (+386%), and action prediction (+138% in embedding similarity). This ordering suggests that fine-tuning is most impactful for skills absent in pretraining (e.g., reasoning about indirect occlusion cues and inferring the agent actions).

TABLE II: Supervised fine-tuned model performance. Values in parentheses show improvement over base models. Avg. action is the mean of BLEU, ROUGE-L, and Embedding Sim. Bold = best, underline = second best.

| Model      | agent class |       | camera view |       | visual cue |       | occlusion type |       | occluder class |       | priority |       | sector |       | bleu  | action rouge | sim.  | avg. acc.           | overall avg. fl     | act avg.            |
|------------|-------------|-------|-------------|-------|------------|-------|----------------|-------|----------------|-------|----------|-------|--------|-------|-------|--------------|-------|---------------------|---------------------|---------------------|
|            | acc.        | f1    | acc.        | f1    | acc.       | f1    | acc.           | f1    | acc.           | f1    | acc.     | f1    | acc.   | f1    |       |              |       |                     |                     |                     |
| Qwen-3B    | 0.708       | 0.683 | 0.196       | 0.184 | 0.812      | 0.741 | 0.628          | 0.601 | 0.352          | 0.331 | 0.544    | 0.387 | 0.104  | 0.125 | 0.319 | 0.364        | 0.546 | 0.478 (+.24)        | 0.436 (+.25)        | 0.410 (+.33)        |
| Qwen-7B    | 0.756       | 0.741 | 0.340       | 0.322 | 0.804      | 0.728 | 0.624          | 0.600 | 0.412          | 0.370 | 0.548    | 0.494 | 0.320  | 0.342 | 0.329 | 0.407        | 0.569 | 0.543 (+.24)        | 0.514 (+.26)        | 0.435 (+.35)        |
| Qwen-32B   | 0.660       | 0.667 | 0.272       | 0.273 | 0.744      | 0.654 | 0.620          | 0.603 | 0.412          | 0.385 | 0.468    | 0.447 | 0.204  | 0.228 | 0.320 | 0.372        | 0.547 | 0.483 (+.09)        | 0.465 (+.08)        | 0.413 (+.32)        |
| Intern-1B  | 0.756       | 0.692 | 0.452       | 0.304 | 0.804      | 0.728 | 0.488          | 0.361 | 0.288          | 0.187 | 0.548    | 0.388 | 0.476  | 0.327 | 0.319 | 0.377        | 0.553 | 0.545 (+.45)        | 0.427 (+.34)        | 0.416 (+.33)        |
| Intern-2B  | 0.768       | 0.704 | 0.464       | 0.326 | 0.756      | 0.674 | 0.576          | 0.521 | 0.392          | 0.332 | 0.548    | 0.388 | 0.508  | 0.442 | 0.329 | 0.390        | 0.566 | 0.573 (+.20)        | 0.484 (+.16)        | 0.428 (+.34)        |
| Intern-4B  | 0.752       | 0.684 | 0.428       | 0.335 | 0.800      | 0.724 | 0.608          | 0.591 | 0.412          | 0.373 | 0.480    | 0.430 | 0.476  | 0.407 | 0.316 | 0.427        | 0.584 | 0.565 (+.33)        | 0.506 (+.25)        | 0.443 (+.35)        |
| Intern-8B  | 0.752       | 0.684 | 0.396       | 0.336 | 0.792      | 0.716 | 0.596          | 0.555 | 0.364          | 0.300 | 0.564    | 0.467 | 0.492  | 0.415 | 0.319 | 0.416        | 0.585 | 0.565 (+.13)        | 0.496 (+.10)        | 0.440 (+.35)        |
| Intern-14B | 0.740       | 0.657 | 0.468       | 0.384 | 0.796      | 0.720 | 0.636          | 0.616 | 0.464          | 0.425 | 0.556    | 0.498 | 0.496  | 0.435 | 0.334 | 0.422        | 0.582 | <b>0.594 (+.22)</b> | 0.534 (+.20)        | <b>0.446 (+.35)</b> |
| Intern-GPT | 0.736       | 0.643 | 0.488       | 0.349 | 0.712      | 0.592 | 0.528          | 0.407 | 0.308          | 0.177 | 0.532    | 0.387 | 0.468  | 0.329 | 0.320 | 0.423        | 0.580 | 0.539 (+.26)        | 0.412 (+.20)        | 0.441 (+.34)        |
| Genma-4B   | 0.740       | 0.670 | 0.468       | 0.299 | 0.804      | 0.730 | 0.620          | 0.612 | 0.340          | 0.301 | 0.408    | 0.377 | 0.228  | 0.196 | 0.313 | 0.378        | 0.546 | 0.515 (+.18)        | 0.455 (+.16)        | 0.412 (+.33)        |
| Genma-12B  | 0.740       | 0.682 | 0.460       | 0.347 | 0.800      | 0.726 | 0.676          | 0.665 | 0.428          | 0.378 | 0.576    | 0.512 | 0.440  | 0.386 | 0.305 | 0.395        | 0.570 | 0.589 (+.32)        | 0.528 (+.28)        | 0.423 (+.34)        |
| Genma-27B  | 0.728       | 0.704 | 0.364       | 0.368 | 0.804      | 0.728 | 0.652          | 0.653 | 0.440          | 0.425 | 0.520    | 0.494 | 0.404  | 0.390 | 0.295 | 0.379        | 0.556 | 0.559 (+.21)        | <b>0.538 (+.22)</b> | 0.410 (+.33)        |
| MiniCPM    | 0.756       | 0.671 | 0.416       | 0.339 | 0.792      | 0.716 | 0.612          | 0.585 | 0.420          | 0.366 | 0.552    | 0.403 | 0.484  | 0.396 | 0.315 | 0.414        | 0.589 | 0.576 (+.22)        | 0.497 (+.17)        | 0.439 (+.36)        |
| DriveLM    | 0.656       | 0.602 | 0.428       | 0.342 | 0.652      | 0.581 | 0.488          | 0.481 | 0.224          | 0.198 | 0.520    | 0.396 | 0.432  | 0.297 | 0.284 | 0.355        | 0.534 | 0.486 (+.32)        | 0.414 (+.24)        | 0.391 (+.29)        |

#### D. Ablation Studies

1) *PKL and Visibility Threshold Sensitivity*: We validate the robustness of two key hyperparameters in our pipeline. Figure 4 (left) shows the per-scene PKL convergence analysis on the nuScenes training set. Both mean and median PKL estimates stabilize after approximately 2 seconds (5 samples per scene), confirming that our  $T=2$ s windows provide reliable importance estimates without requiring exhaustive sampling.

Figure 4 (right) reports the PKL Recovery Ratio ( $\frac{\text{Gain}(h_i)}{\text{PKL}(\mathcal{V})}$ ) on the nuScenes validation set across  $\tau \in [0.1, 0.6]$ . Mean and median recovery remain stable at 0.6-0.75. However,  $\tau$  also controls a trade-off: lower values retain fewer, more severely occluded agents, while higher values include more agents. Our  $\tau=0.4$  balances dataset size with occlusion severity, aligns with the nuScenes lowest visibility bin (0–40%), and lies within the stable recovery region.

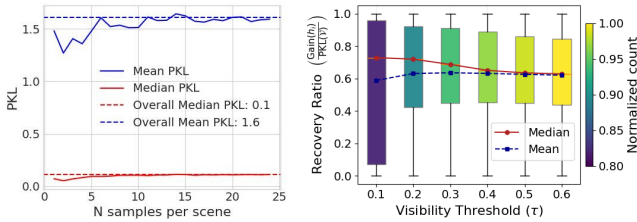


Fig. 4: Robustness of PKL estimation and visibility threshold. (Left) Per-scene PKL convergence: mean and median stabilize after  $\sim 5$  samples. (Right) PKL Recovery Ratio across  $\tau \in [0.1, 0.6]$ : recovery remains stable while lower thresholds yield fewer but more severely occluded agents.

2) *Impact of PKL-Guided Selection*: Table III validates our PKL-guided data selection strategy by comparing against random hidden agent selection. Both models were evaluated on the same 250-sample validation set from nuScenes to ensure fair comparison. For Qwen-7B, PKL-guided selection improves average accuracy from 0.405 to 0.543 (+34%). Similarly, InternVL3.5-8B improves from 0.429 to 0.565 in accuracy (+32%). Figure 5 provides a qualitative example of this performance gap; it illustrates a common failure case for models trained on data with randomly selected hidden agents, which often misidentify agent priority and wrongly

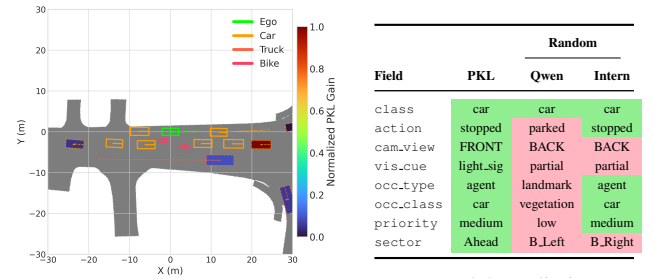
point to less important hidden agents. The consistent  $\sim 30\%$  improvement demonstrates that focusing on planning-critical agents during training is essential. The gains in action and class metrics indicate a better semantic understanding of occlusion patterns. This validates our hypothesis that the model benefits significantly from training on examples where hidden agents substantially impact planning decisions.

TABLE III: Impact of PKL-guided data selection. Top row: Random selection baseline. Bottom row: PKL selection. Bold values indicate an improvement over the baseline (deltas in parentheses).

| Model     | Sel.  | Average Class       |                     |                     | Action              |
|-----------|-------|---------------------|---------------------|---------------------|---------------------|
|           |       | Acc.                | Prec.               | F1                  | Avg.                |
| Qwen-7B   | Rand. | 0.405               | 0.446               | 0.350               | 0.410               |
|           | PKL   | <b>0.543(+.138)</b> | <b>0.533(+.087)</b> | <b>0.514(+.164)</b> | <b>0.435(+.025)</b> |
| Intern-8B | Rand. | 0.429               | 0.419               | 0.382               | 0.406               |
|           | PKL   | <b>0.565(+.136)</b> | <b>0.486(+.067)</b> | <b>0.496(+.114)</b> | <b>0.440(+.034)</b> |



(a) Panoramic front view at timestep 2



(c) Model predictions.

Fig. 5: A challenging nighttime scene. Qwen-7B and Intern-8B, when trained on randomly selected data, struggle with predicting priority and location compared to their PKL-guided counterparts. Green = correct, red = incorrect.

3) *Fine-Tuning Strategy Analysis*: Table IV compares fine-tuning only the language model versus our full approach (LLM + alignment module). Improvements are modest but consistent for Qwen-7B, with gains across all metrics including the largest improvement in action score (+0.007).

InternVL3.5-8B shows mixed results: accuracy and F1 improve marginally while precision slightly decreases, indicating that the benefit of aligner fine-tuning is model-dependent. Given the minimal computational overhead and consistent gains for Qwen, we recommend fine-tuning both components, though the LLM-only approach remains viable, particularly for architectures that show diminishing returns from aligner adaptation.

TABLE IV: Fine-tuning comparison. Top row: LLM-only baseline. Bottom row: LLM+Aligner. Bold values indicate an improvement over the baseline (deltas in parentheses).

| Model     | Variant  | Average Class        |                      |                      | Action               |
|-----------|----------|----------------------|----------------------|----------------------|----------------------|
|           |          | Acc.                 | Prec.                | F1                   | Avg.                 |
| Qwen-7B   | LLM      | 0.538                | 0.528                | 0.508                | 0.428                |
|           | +Aligner | <b>0.543</b> (+.005) | <b>0.533</b> (+.005) | <b>0.514</b> (+.006) | <b>0.435</b> (+.007) |
| Intern-8B | LLM      | 0.564                | 0.497                | 0.494                | 0.440                |
|           | +Aligner | <b>0.565</b> (+.001) | 0.486(−.011)         | <b>0.496</b> (+.002) | 0.440                |

4) *Impact on Visible-Agent Performance:* To verify that fine-tuning on our occlusion-focused dataset does not degrade performance on fully visible agents, we evaluate on the MME-RealWorld [37] autonomous driving subset, which contains predominantly unoccluded traffic scenarios. We compare base models against their counterparts fine-tuned on our PKL-guided dataset. As shown in Table V, DriveLM retains identical reasoning while overall and perception drop by only 1-2%. Qwen-3B shows even smaller degradation, with all scores decreasing by less than 1%. This confirms that our task-specific fine-tuning introduces no meaningful catastrophic forgetting of visible-agent capabilities across model families.

TABLE V: Performance on the MME-RealWorld autonomous driving subset (deltas in parentheses).

| Model               | Overall      | Reasoning    | Perception   |
|---------------------|--------------|--------------|--------------|
| DriveLM             | 0.482        | 0.403        | 0.511        |
| DriveLM(fine-tuned) | 0.474(−.008) | 0.403        | 0.500(−.011) |
| Qwen-3B             | 0.307        | 0.266        | 0.321        |
| Qwen-3B(fine-tuned) | 0.305(−.002) | 0.265(−.001) | 0.320(−.001) |

## E. Discussion

Our results reveal several important insights:

**Task-Specific Training Dominates:** The large improvements from fine-tuning outweigh the benefits of model scale or domain adaptation, highlighting the importance of curated, task-specific data. **Planning-Aware Data Selection Matters:** PKL-guided selection provides  $\sim 30\%$  better results than random selection, validating our approach of prioritizing occlusions that affect planning decisions.

**Architectural Efficiency:** Some model families (InternVL3.5) achieve comparable performance with fewer parameters, suggesting architectural innovations could further improve efficiency. Our fine-tuned models exhibit significant inference latency variations across architectures, with Qwen models achieving 47–79 ms per sample, Gemma models

requiring 74–99 ms, and InternVL variants ranging from 86–323 ms. These latencies suggest that smaller fine-tuned models are approaching real-time feasibility, particularly if the VLM runs as a parallel reasoning module at a lower frequency (e.g., 2–5 Hz) than the main perception stack. **Action Understanding:** The improvements in action prediction indicate that models learn to infer hidden agent intentions from indirect visual evidence, that is a crucial capability for safe autonomous driving. **Downstream Utility:** The structured output format enables concrete integration pathways. The priority level and sector fields can directly trigger driver warnings in an ADAS system (e.g., “High-priority hidden agent, Ahead-Left”). In multi-agent collaboration scenarios (V2V, drone-vehicle, V2X), the structured annotations provide a compact, standardized format for an ego vehicle to query another agent about planning-critical occlusions in its blind [38]. Priority levels and sectors can also be converted to cost-map modifications for downstream planners.

## F. Limitations and Future Work

Our investigation opens several important avenues for future work. Driving datasets like nuScenes are often limited by uniform collection processes, which reduces their diversity and excludes many real-world edge cases. The resulting long-tail distribution makes safety-critical occlusion events rare. A promising future direction is both more targeted data collection to capture a wider variety of high-risk scenarios, and cross-dataset validation on benchmarks such as nuPlan and Waymo to establish external validity. Our PKL-guided framework is dataset-agnostic in principle, it requires only 3D annotations and a planner, making such transfer straightforward. Additionally, the inference latency of large VLMs remains a challenge for real-time deployment. While our results show the promise of smaller fine-tuned models (e.g., Qwen-3B at 47 ms), further optimization is needed to meet strict latency requirements, such as running the VLM at a reduced frequency alongside the main perception stack or offloading to edge infrastructure. Finally, while our manual audit and strict schema enforcement mitigate annotation errors, reliance on a single LLM may still imprint systematic biases onto the dataset. Future work could explore inter-model agreement using a diverse ensemble of annotators, or human-in-the-loop verification at scale, to further quantify and reduce label noise. Looking forward, a natural next step is closed-loop evaluation, where the structured annotations feed directly into a planner to measure safety improvements beyond static QA metrics. This work also enables novel applications, from multi-agent to ADAS systems.

## V. CONCLUSION

In this work, we introduced a systematic framework to identify, annotate, and reason about planning-critical occluded agents in autonomous driving. By leveraging Planning KL-divergence (PKL) to curate a dataset of high-impact scenarios, we demonstrated that fine-tuning VLMs on this specialized data yields substantial performance gains across

all model families and scales. Our results show that this task-specific, planning-aware training is more critical than model size, enabling smaller models to outperform much larger zero-shot counterparts and proving that a focus on important occlusions leads to more effective learning. This approach bridges a critical gap between perception and planning, paving the way for autonomous systems that can reason more intelligently about risks and navigate complex, partially-observable environments with greater safety and efficiency.

## REFERENCES

- [1] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Qiao Yu, and Jifeng Dai. Bevformer: learning bird's-eye-view representation from lidar-camera via spatiotemporal transformers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.
- [2] Jonah Philion and Sanja Fidler. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16*, pages 194–210. Springer, 2020.
- [3] Jiayu Yang, Enze Xie, Miaomiao Liu, and Jose M. Alvarez. Parametric depth based feature representation learning for object detection and segmentation in bird's eye view. 7 2023.
- [4] Xiaoyu Tian, Tao Jiang, Longfei Yun, Yue Wang, Yilun Wang, and Hang Zhao. Occ3d: A large-scale 3d occupancy prediction benchmark for autonomous driving. *arXiv preprint arXiv:2304.14365*, 2023.
- [5] Jonah Philion, Amlan Kar, and Sanja Fidler. Learning to evaluate perception models using planner-centric metrics. 2020 ieee. In *CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14052–14061, 2020.
- [6] Daocheng Fu, Xin Li, Licheng Wen, Min Dou, Pinlong Cai, Botian Shi, and Yu Qiao. Drive like a human: Rethinking autonomous driving with large language models, 2023.
- [7] Amirhosein Chahe and Lifeng Zhou. Reasondrive: Efficient visual question answering for autonomous vehicles with reasoning-enhanced small vision-language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 3870–3879, 2025.
- [8] Shihao Wang, Zhiding Yu, Xiaohui Jiang, Shiyi Lan, Min Shi, Nadine Chang, Jan Kautz, Ying Li, and Jose M Alvarez. Omnidrive: A holistic llm-agent framework for autonomous driving with 3d perception, reasoning and planning. *arXiv preprint arXiv:2405.01533*, 2024.
- [9] Allen Z. Ren, Anushri Dixit, Alexandra Bodrova, Sumeet Singh, Stephen Tu, Noah Brown, Peng Xu, Leila Takayama, Fei Xia, Jake Varley, Zhenjia Xu, Dorsa Sadigh, Andy Zeng, and Anirudha Majumdar. Robots that ask for help: Uncertainty alignment for large language model planners. 9 2023.
- [10] Srikanth Malla, Chiho Choi, Isht Dwivedi, Joon Hee Choi, and Jiachen Li. Drama: Joint risk localization and captioning in driving. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 1043–1052, 2023.
- [11] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020.
- [12] Gpt-5 system card. Technical report, OpenAI, August 2025. Accessed: 2025-09-07.
- [13] Markus Koschi and Matthias Althoff. Set-based prediction of traffic participants considering occlusions and traffic rules. *IEEE Transactions on Intelligent Vehicles*, 6(2):249–265, June 2021. Publisher Copyright: © 2016 IEEE.
- [14] Piotr F. Orzechowski, Annika Meyer, and Martin Lauer. Tackling occlusions & limited sensor range with set-based safety verification. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, page 1729–1736. IEEE Press, 2018.
- [15] Patiphon Narksri, Hatem Darweesh, Eijiro Takeuchi, Yoshiki Ni-nomiya, and Kazuya Takeda. Occlusion-aware motion planning with visibility maximization via active lateral position adjustment. *IEEE Access*, 10:57759–57782, 2022.
- [16] Constantin Hubmann, Nils Quetschlich, Jens Schulz, Julian Bernhard, Daniel Althoff, and Christoph Stiller. A pomdp maneuver planner for occlusions in urban scenarios. In *2019 IEEE Intelligent Vehicles Symposium (IV)*, pages 2172–2179, 2019.
- [17] Kyle Hollins Wray, Bernard Lange, Arec Jamgochian, Stefan J. Witwicki, Atsuhide Kobashi, Sachin Hagaribommanahalli, and David Ilstrup. Pomdps for safe visibility reasoning in autonomous vehicles. In *2021 IEEE International Conference on Intelligence and Safety for Robotics (ISR)*, pages 191–195, 2021.
- [18] Xiyue Zhu, Vlas Zyrianov, Zhijian Liu, and Shenlong Wang. Mapprior: Bird's-eye view map layout estimation with generative models. 8 2023.
- [19] Ziteng Xue, Mingzhe Guo, Heng Fan, Shihui Zhang, and Zhipeng Zhang. Corbev: Multi-view 3d object detection by correlation learning with multi-modal prototypes. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 27413–27423, 2025.
- [20] Tushar Choudhary, Vikrant Dewangan, Shivam Chandhok, Shubham Priyadarshan, Anushka Jain, Arun K Singh, Siddharth Srivastava, Krishna Murthy Jatavallabhula, and K Madhava Krishna. Talk2bev: Language-enhanced bird's-eye view maps for autonomous driving. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 16345–16352. IEEE, 2024.
- [21] Katharina Winter, Mark Azer, and Fabian B Flohr. Bevdriver: Leveraging bev maps in llms for robust closed-loop driving. *arXiv preprint arXiv:2503.03074*, 2025.
- [22] Fuhao Li, Huan Jin, Bin Gao, Liaoyuan Fan, Lihui Jiang, and Long Zeng. Nugrounding: A multi-view 3d visual grounding framework in autonomous driving. *arXiv preprint arXiv:2503.22436*, 2025.
- [23] Chonghao Sima, Katrin Renz, Kashyap Chitta, Li Chen, Hanxue Zhang, Chengen Xie, Jens Beißwenger, Ping Luo, Andreas Geiger, and Hongyang Li. Drivelm: Driving with graph visual question answering. In *European conference on computer vision*, pages 256–274. Springer, 2024.
- [24] Yunsong Zhou, Linyan Huang, Qingwen Bu, Jia Zeng, Tianyu Li, Hang Qiu, Hongzi Zhu, Minyi Guo, Yu Qiao, and Hongyang Li. Embodied understanding of driving scenarios. *arXiv preprint arXiv:2403.04593*, 2024.
- [25] Yiluan Guo, Holger Caesar, Oscar Beijbom, Jonah Philion, and Sanja Fidler. The efficacy of neural planning metrics: A meta-analysis of pkl on nuscenes. *arXiv preprint arXiv:2010.09350*, 2020.
- [26] Boris Ivanovic and Marco Pavone. Injecting planning-awareness into prediction and detection evaluation. In *2022 IEEE Intelligent Vehicles Symposium (IV)*, pages 821–828. IEEE, 2022.
- [27] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019.
- [28] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.
- [29] Yuze Zhao, Jintao Huang, Jinghan Hu, Xingjun Wang, Yunlin Mao, Daoze Zhang, Zeyinzi Jiang, Zhikai Wu, Baole Ai, Ang Wang, Wenmeng Zhou, and Yingda Chen. Swift: a scalable lightweight infrastructure for fine-tuning, 2024.
- [30] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibao Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.
- [31] Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, et al. Internvl3. 5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*, 2025.
- [32] Gemma Team. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025.
- [33] Yuan Yao, Tianyu Yu, Ao Zhang, Chongyi Wang, Junbo Cui, Hongji Zhu, Tianchi Cai, Haoyu Li, Weilin Zhao, Zhihui He, et al. Minicpm-v: A gpt-4v level mllm on your phone. *arXiv preprint arXiv:2408.01800*, 2024.
- [34] Zhijian Huang, Chengjian Feng, Feng Yan, Baihui Xiao, Zequn Jie, Yujie Zhong, Xiaodan Liang, and Lin Ma. Drivemm: All-in-one large multimodal model for autonomous driving. *arXiv preprint arXiv:2412.07689*, 2024.
- [35] Zhangwei Gao, Zhe Chen, Erfei Cui, Yiming Ren, Weiyun Wang, Jinguo Zhu, Hao Tian, Shenglong Ye, Junjun He, Xizhou Zhu, et al. Mini-internvl: a flexible-transfer pocket multi-modal model with 5% parameters and 90% performance. *Visual Intelligence*, 2(1):1–17, 2024.

- [36] Zhenhua Xu, Yujia Zhang, Enze Xie, Zhen Zhao, Yong Guo, Kwan-Yee K Wong, Zhenguo Li, and Hengshuang Zhao. Drivegpt4: Interpretable end-to-end autonomous driving via large language model. *IEEE Robotics and Automation Letters*, 2024.
- [37] Yi-Fan Zhang, Huanyu Zhang, Haochen Tian, Chaoyou Fu, Shuangqing Zhang, Junfei Wu, Feng Li, Kun Wang, Qingsong Wen, Zhang Zhang, Liang Wang, Rong Jin, and Tieniu Tan. Mme-realworld: Could your multimodal llm challenge high-resolution real-world scenarios that are difficult for humans?, 2025.
- [38] Yaozu Wu, Dongyuan Li, Yankai Chen, Renhe Jiang, Henry Peng Zou, Wei-Chieh Huang, Yangning Li, Liancheng Fang, Zhen Wang, and Philip S. Yu. Multi-agent autonomous driving systems with large language models: A survey of recent advances, 2025.