MD4AD: THE WHOLE DEAL

Tianzhi Li* **Junhong Zhou*** **Patrick Chen*** **Daniel Yang***
{tianzhil, junhong2, bochunc, danielya}@andrew.cmu.edu

ABSTRACT

In this project, we introduce a novel extension of EM-VLM4AD, a lightweight multi-frame Vision-Language Model (VLM) designed for visual question answering in autonomous driving. To tackle the limitations of prior vision encoders in capturing spatial context, we replace the traditional image patch encoder with BEVFusion, a bird's-eye-view (BEV) feature extractor that fuses camera and LiDAR data. We show that this BEV-enhanced architecture significantly improves spatial reasoning and interpretability, while maintaining low computational cost—making it suitable for real-time applications in driving scenarios.

1 [2 POINTS] INTRODUCTION AND PROBLEM DEFINITION (1-1.25 PAGES)

Motivation. As autonomous vehicles transition from research labs to public roads, ensuring both robust perception and transparent decision-making is essential. In high-stakes environments like driving, it is not sufficient to detect objects or predict trajectories; systems must also be able to explain their reasoning. Vision-Language Models (VLMs) offer a pathway toward interpretable AI by enabling models to answer natural language questions about the scene using visual and spatial understanding.

Problem Setup. We focus on the task of **multi-modal Visual Question Answering (VQA)** in driving scenarios. The input consists of multiple synchronized camera views and LiDAR point clouds, along with a question posed in natural language. The model must output an interpretable text response grounded in both visual and spatial context. For example:

- *"What is the status of the pedestrian to the left of the ego vehicle?"*
- *"Will the red car stop at the intersection?"*

These questions require both fine-grained visual perception and spatial reasoning about object interactions and motion dynamics.

Challenges in Prior Work. Current multimodal approaches typically fall into one of two categories: (1) heavy-weight LLMs like GPT-3.5 or BLIP-2 that require immense compute and are unsuitable for real-time use, or (2) lightweight models that operate on single-view frames and miss important spatial cues. Additionally, common vision backbones (e.g., ViT, CLIP) process images independently, lacking mechanisms to effectively incorporate LiDAR data or multi-camera context.

Proposed Solution. To address these challenges, we propose a new variant of EM-VLM4AD—a lightweight vision-language model optimized for driving—that introduces the following innovations:

1. **BEVFusion backbone:** Instead of using patch-based image encoders, we adopt BEVFusion to integrate LiDAR and multi-camera features into a bird's-eye-view spatial representation.
2. **Multi-view attention pooling:** For images, we implement gated attention to aggregate features across all views, enabling the model to reason over the full 3D scene.

* Everyone Contributed Equally – Alphabetical order

3. **Efficient T5-based decoding:** Our language model head is based on T5 (Base and Quantized-Large), offering a strong performance-memory trade-off for deployment.

Together, these changes create a system that is not only capable of multi-modal spatial reasoning, but also lightweight enough to be applied in real-time scenarios.

High-Level Objective. Our model serves as a bridge between perception and natural-language understanding. It aims to assist downstream systems—or even human operators—by producing semantically meaningful, interpretable, and contextually grounded answers to vision-language queries in driving scenes.

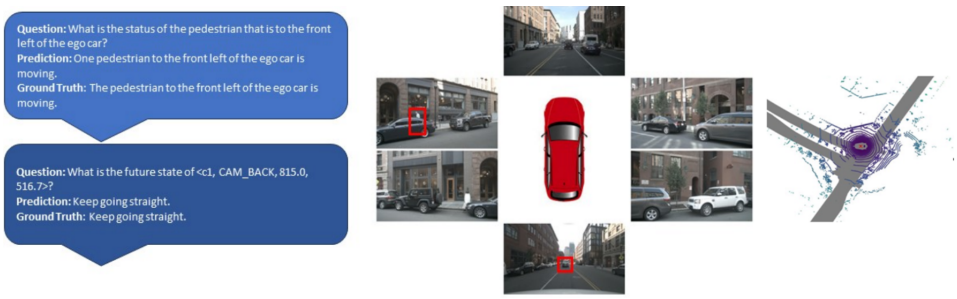


Figure 1: Example Questions and Visual Inputs for VQA in Driving.

Contributions.

- We enhance EM-VLM4AD by integrating BEVFusion, allowing rich LiDAR and camera-based spatial encoding.
- We achieve the strongest overall balance between fluency (BLEU-4, METEOR) and semantic alignment (ROUGE-L, CIDEr).
- We show our model is computationally efficient while improving language output quality over prior baselines.

2 [5 POINTS] RELATED WORK AND BACKGROUND (5 PAPERS PER PERSON)

Related Datasets We build our model and experiments on the **DriveLM-nuScenes** dataset. DriveLM is a recently proposed benchmark for Graph Visual Question Answering (GVQA) in autonomous driving Sima et al. (2025). It enables multi-stage reasoning over tasks such as perception, prediction, and planning by generating dense, structured QA graphs from real-world driving data. DriveLM-nuScenes in particular is semi-rule-based, combining automated QA generation with human-in-the-loop filtering, and provides over 91 QAs per frame, significantly outpacing earlier driving VQA datasets.

Table 1 provides a comparison between DriveLM and other related datasets in terms of frame count, annotation richness, task coverage, and logic structure. While previous datasets focus on perception or captioning, DriveLM uniquely supports complex, graph-based question answering across the autonomous driving stack.

Dataset	Source Dataset	# Frames	Avg. QA / Frame	Perception	Prediction	Planning	Logic
nuScenes-QA Qian et al. (2024)	nuScenes	34,149	13.5	460k**	✗	✗	None
nuPrompt Yang et al. (2023)	nuScenes	34,149	1.0	35k*	✗	✗	None
HAD Tang et al. (2023)	HDD	25,549	1.8	25k	✗	20k	None
BDD-X Bai et al. (2021)	BDD	26,228	1.1	26k	✗	✗	None
LingoQA Chen et al. (2023a)	LingoQA	28,000	15.3	-	-	-	None
DRAMA Yao et al. (2022)	DRAMA	17,785	5.8	85k	✗	17k	Chain
Rank2Tell Zhou et al. (2023)	Rank2Tell	5,800	-	-	✗	-	Chain
DriveLM-CARLA† Sima et al. (2025)	CARLA	64,285	24.4	697k**	311k**	558k**	Graph
DriveLM-CARLA‡ Sima et al. (2025)	CARLA	5,721	24.8	63k**	28k**	51k**	Graph
DriveLM-nuScenes Sima et al. (2025)	**nuScenes**	**4,871**	**91.4**	144k*	153k	146k	**Graph**

Table 1: Comparison of DriveLM-nuScenes with Existing Vision-Language Datasets. *: semi-rule-based labeling with human annotation. **: fully-rule-based annotation. †: full dataset. ‡: keyframe-only dataset.

Unimodal Baselines To isolate the impact of visual and spatial inputs, we evaluate a unimodal Q-only baseline, where the model receives only the natural language question as input—without any image or LiDAR data. Specifically, we retain the original input format by concatenating a zero tensor in place of image embeddings, ensuring that the merged multi-modal embedding shape remains unchanged. This setting tests the language model's ability to infer answers solely from question priors.

We implement this baseline using two encoder-decoder backbones: **T5-Base** and **T5-Q-Large**, yielding two variants—Q-only(Base) and Q-only(Q-Large). T5-Base contains 223M parameters, while T5-Q-Large has 757M, enabling a comparison of capacity and efficiency. Interestingly, despite lacking any visual context, both unimodal variants achieve surprisingly competitive performance: BLEU-4 scores of 45.5 and 45.4, and CIDEr scores of 3.09 and 3.10, respectively (Table 2). These results underscore the strong language priors inherent in the dataset.

However, the Q-only models consistently underperform in tasks requiring fine-grained visual grounding and spatial reasoning, highlighting the necessity of multi-modal fusion. Our proposed models, which incorporate both images and LiDAR features, outperform all baselines across ROUGE-L and CIDEr, demonstrating the value of enriched spatial inputs for real-world driving scene understanding.

Methods	**BLEU-4** ↑	**METEOR** ↑	**ROUGE-L** ↑	**CIDEr** ↑	**Parameters** ↓
Q-only$_{\text{Base}}$	45.5	34.1	70.5	3.09	223M
Q-only$_{\text{Q-Large}}$	45.4	34.0	70.1	3.10	757M

Table 2: Performance of Unimodal Q-only Baselines using T5-Base and T5-Q-Large on DriveLM-nuScenes VQA Tasks.

Prior Work Recent advances in vision-language models (VLMs) have enabled impressive performance across a variety of general-purpose reasoning tasks. However, their application to autonomous driving remains constrained by several limitations: high computational cost, inability to operate in real-time, insufficient support for multi-modal sensor inputs (such as LiDAR and camera), and a lack of structured scene-level reasoning.

Table 3 summarizes 20 relevant approaches, comparing their core strengths and drawbacks. Existing methods largely fall into two categories: (1) heavy-weight LLMs (e.g., GPT-3.5, BLIP-2) that deliver strong language understanding but are unsuitable for onboard or low-latency deployment; and (2) lightweight or single-view methods that lack spatial context and do not integrate multi-modal data, particularly LiDAR and multi-camera streams. Moreover, most do not provide support for temporally grounded planning or interpretable decision-making.

Our work builds upon the strengths of these models while addressing their limitations through a lightweight architecture that performs multi-view fusion of synchronized camera images and LiDAR point clouds into a BEV (bird's-eye-view) representation. This enables accurate and interpretable graph-based question answering (QA) across perception, prediction, and planning tasks in driving scenarios.

Relevant techniques Our approach draws inspiration from several core techniques in vision-language and autonomous driving research. First, we leverage **encoder-decoder transformer architectures**, such as T5, which enable flexible cross-modal reasoning through joint sequence modeling of language and visual embeddings. To support multi-camera spatial understanding, we adopt **multi-view feature fusion mechanisms**—particularly gated attention pooling for images—to aggregate contextual information from diverse camera viewpoints. For integrating LiDAR, we build on methods like **BEVFusion**, which aligns point cloud features with image-derived spatial grids for dense scene representation.

To improve training efficiency and parameter scalability, we employ **Low-Rank Adaptation (LoRA)** modules in the T5 language model. Specifically, we insert lightweight trainable adapters into the attention and MLP sublayers of T5, allowing us to fine-tune only a small subset of parameters while freezing the original backbone. This approach reduces computational cost and mitigates overfitting, enabling more efficient adaptation to domain-specific VQA tasks.

In addition, our QA formulation benefits from **graph-based reasoning**, where the underlying model is trained to infer structured outputs grounded in object interactions and temporal planning. This builds upon trends in **multi-modal VQA** and **driving decision understanding**, extending them toward interpretable and deployable systems. Finally, to benchmark the importance of multi-modal inputs, we incorporate **zero-embedding masking techniques** in our unimodal baselines (Q-only), following practices in **ablation-based analysis** to isolate language-only performance.

Paper	Advantage	Disadvantage
BLIP-2 Li et al. (2023a)	Strong zero-shot vision-language reasoning	Heavy model, unsuitable for real-time use
GPT-3.5 + CLIP OpenAI (2023)	Powerful language modeling	High inference latency
DriveGPT4 Xu et al. (2023)	Generates both answers and low-level control	Uses only single-view images
DriveMLM Wang et al. (2023)	Aligns multimodal inputs including LiDAR and planning state	Very large backbone (LLaMA-7B + ViT-g)
LLM-Driver Chen et al. (2023b)	Fuses vectorized representations with LLM	Requires custom grounding
VL-T5 Cho et al. (2021)	Lightweight text generation from images + text	Single-image only, lacks LiDAR support
InstructBLIP Dai et al. (2023)	General-purpose instruction-tuned VLM	Expensive encoder + language model
CLIP Radford et al. (2021)	Powerful vision-text embedding	Not task-optimized, lacks generation
ViLT Kim et al. (2021)	Transformer-based early fusion V+L	Less accurate on downstream tasks
LLaVA Liu et al. (2023)	Open language-vision model with chat capability	Requires strong image encoder
Visual ChatGPT Wu et al. (2023)	Interactive multi-modal answering	Relies on external visual tools
MIVC Wu et al. (2024)	Attention pooling over visual patches	Still slow due to ViT backbone
MiniGPT-4 Zhu et al. (2023)	Lightweight GPT-4 mimic using BLIP	Brittle on complex tasks
VILA Zhao et al. (2023)	Strong on instruction-following VQA	Limited support for LiDAR or multi-frame
EM-VLM4AD Gopalkrishnan et al. (2024)	Efficient multi-frame T5-based VLM	ViT encoder lacks 3D reasoning
BEVFormer Li et al. (2022)	Strong temporal BEV reasoning	Not natively multi-modal
BEVFusion Liu et al. (2022b)	Fuses LiDAR + image into BEV	Not designed for VQA
TransFusion Bai et al. (2022)	Accurate 3D object detection from LiDAR	No VQA or text interface
Point-BERT Yu et al. (2022)	Point cloud self-supervised learning	No cross-modal capabilities
DALL-E Ramesh et al. (2021)	Visual generation from text	Not interactive or grounded in scene understanding

Table 3: Analysis of 20 Related Prior Works in Vision-Language and Autonomous Driving.

3 [1 POINTS] TASK SETUP AND DATA

Task Definition. We focus on the task of **graph-based Visual Question Answering (VQA)** in the autonomous driving context, where the goal is to answer natural language questions about driving scenes using both visual and spatial information. Each QA pair is associated with multi-view images, LiDAR point clouds, and object-level annotations. The answers must be interpretable and grounded in scene elements, such as traffic participants, road layout, and dynamic motion.

Dataset. We use the **DriveLM-nuScenes** dataset Sima et al. (2025), which extends the nuScenes dataset with rich visual question answering annotations, having totally 445,279 QA pairs. The QA annotations cover a wide spectrum of driving scenarios and decision-making elements. Each sample includes:

- 6 synchronized camera images (front, back, left, right, front-left, front-right)
- A LiDAR sweep in the same timestamp
- 91.4 QA pairs per frame on average

Question Types. DriveLM questions are grouped into three major categories: **Perception**, **Prediction**, and **Planning**, with further subcategories illustrated in Figure 2 and Figure 3. Perception questions (e.g., "What objects are to the left of the ego vehicle?") account for 38% of the data, covering tasks like object identification, traffic sign recognition, and occlusion reasoning. Prediction questions (28%) involve future motion, object interaction, and attention. Planning questions (24%) cover safe action reasoning, importance ranking, and high-level decision making.

Templates and Reasoning Complexity. QA examples in Table 4 highlight the dataset's diversity and reasoning depth. For instance:

- **Perception:** "What are the objects to the front left of the ego vehicle?"
- **Prediction:** "Will this vehicle likely change motion state based on another?"
- **Planning:** "What is the most probable action for the ego vehicle given the current situation?"

Many questions require spatial grounding using camera coordinates (e.g., CAM_FRONT, 689.2, 527.5) and entail multi-object relational reasoning, temporal logic, or safety evaluation.

Key Objects and Object-Level QA Distribution. To further understand what drives the QA composition, we analyze which types of objects are most frequently referenced and how different object categories relate to specific QA types. Figure 4 (left) shows the distribution of key object categories, such as vehicles, pedestrians, and traffic elements (e.g., lights, markings, signs). The middle and right plots break down how question types vary when the subject is a traffic element versus other categories. Notably, traffic-related QAs emphasize spatial grounding and signal semantics (e.g., meaning, position), whereas questions about other objects skew toward motion prediction, planning, and occlusion.

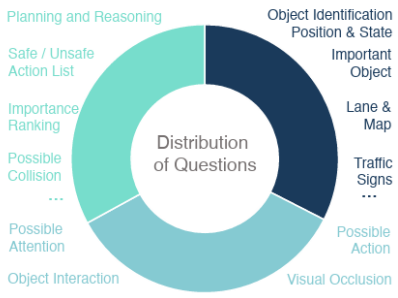


Figure 2: High-level Question Types Grouped into Perception, Prediction, and Planning.

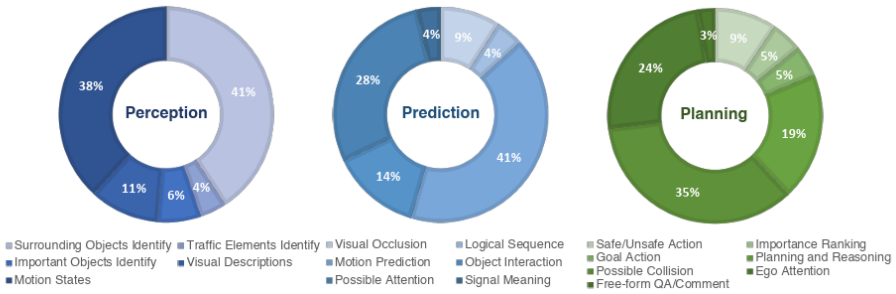


Figure 3: Subcategory Breakdown of VQA tasks in DriveLM-nuScenes.

Category	Example Template (Q & A)
Perception	Q: What objects are to the front-left of the ego car? A: Two barriers.
Prediction	Q: Will object <c1> change motion due to <c2>? A: No.
Planning	Q: What action should the ego vehicle take in this situation? A: Decelerate without braking.

Table 4: Examples of QA Templates at Different Reasoning Levels in DriveLM.

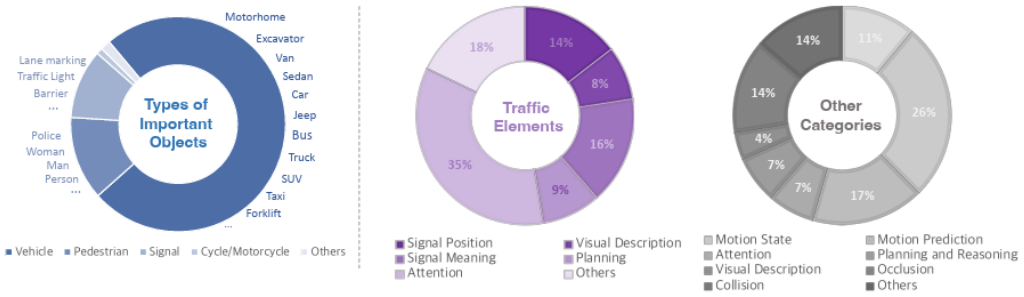


Figure 4: (Left) Distribution of Key Object Types Referenced in DriveLM. (Middle) Breakdown of QA Types Associated with Traffic Elements. (Right) Distribution of QA Types for Other Object Categories.

4 [1 POINTS] BASELINES

4.1 PREVIOUS DATASET/BASELINES

At the early stage of this project, we adopted the **NuScenes-QA** dataset Qian et al. (2024) as our main VQA benchmark. This dataset extends the original nuScenes perception dataset by adding over **450,000 question-answer pairs** across more than **34,000 driving frames**, supporting multi-modal inputs such as RGB images, LiDAR sweeps, and scene-level metadata.

Each QA pair in NuScenes-QA is designed to probe a specific type of reasoning: *Existence*, *Counting*, *Object Recognition*, *Status*, or *Comparison*. Questions are further categorized as either *Zero-Hop* (H0) or *One-Hop* (H1), with H1 questions requiring spatial or relational inference across multiple scene elements.

To evaluate VQA performance on this dataset, we implemented several previously proposed baselines, including both unimodal and multi-modal fusion methods. All models use either the **Bottom-Up Top-Down (BUTD)** attention module or the **Modular Co-Attention Network (MCAN)** as the reasoning backend, fed by features extracted from BEVDet (camera), CenterPoint (LiDAR), or both.

Baseline Summary:

- **Q-Only:** Only textual question is used; all visual inputs are masked.
- **BEVDet+BUTD / MCAN:** Vision-only baseline using BEV feature maps from multi-camera images.
- **CenterPoint+BUTD / MCAN:** LiDAR-only baseline using 3D bounding boxes and spatial cues.
- **MSMDFusion+BUTD / MCAN:** Multi-sensor fusion combining both BEVDet and CenterPoint.

Table 5 reports the Top-1 accuracy across five question categories and two reasoning levels (H0: zero-hop, H1: one-hop) on the NuScenes-QA dataset. Models that incorporate multi-modal fusion (e.g., MSMDFusion+MCAN) consistently outperform unimodal or Q-only baselines, especially in complex reasoning tasks like status and object comparison. The results highlight the importance of both rich visual input and attention-based fusion mechanisms for improving reasoning accuracy in autonomous driving VQA.

Models	**Exist**			**Count**			**Object**			**Status**			**Comparison**			**Acc**
	H0	H1	All	H0	H1	All	H0	H1	All	H0	H1	All	H0	H1	All	
Q-Only+BUTD	81.7	78.3	79.9	18.7	18.8	18.8	63.2	40.0	43.4	57.2	49.2	52.0	81.0	64.5	66.0	54.0
BEVDet+BUTD	87.2	80.6	83.7	21.7	20.0	20.9	69.4	45.2	48.8	55.0	50.5	52.0	76.1	66.8	67.6	57.0
CenterPoint+BUTD	87.3	80.8	83.8	21.6	20.2	20.9	67.7	43.5	47.0	67.7	51.1	54.7	76.6	65.1	66.1	56.8
MSMDFusion+BUTD	89.4	81.4	85.1	25.3	21.3	23.2	73.3	48.7	52.3	67.4	55.4	59.5	81.6	67.2	68.5	59.8
Q-Only+MCAN	81.7	78.6	80.1	18.8	18.8	18.8	64.9	40.9	44.5	56.9	45.6	49.5	80.5	65.9	67.3	54.2
BEVDet+MCAN	87.2	81.7	84.2	21.8	19.2	20.4	73.0	47.4	51.2	64.1	49.9	54.7	75.1	66.7	67.4	57.9
CenterPoint+MCAN	87.1	82.4	84.6	21.7	20.8	21.2	69.8	50.9	53.7	64.5	56.3	59.1	75.5	66.8	67.6	59.3
MSMDFusion+MCAN	**89.0**	**82.3**	**85.4**	**23.4**	**21.1**	**22.2**	**75.3**	**50.6**	**54.3**	**69.0**	**56.2**	**60.6**	**78.8**	**68.8**	**69.7**	**60.4**

Table 5: Original Baseline Models on NuScenes-QA Showing Top-1 Accuracy across Question Types. H0 = Zero-Hop; H1 = One-Hop.

4.2 THE PROBLEM OF THE PREVIOUS DATASET

Despite its scale and diversity, a significant portion of the NuScenes-QA dataset suffers from a critical flaw: most QA pairs are misaligned with the corresponding sensor data in the nuScenes dataset. Questions are often paired with incorrect camera images, LiDAR sweeps, or RADAR frames, leading to faulty training signals and unreliable model predictions. Below are representative examples of these misalignments:

- **Example 1:** For example, one QA pair asks: "There is a truck behind the trailer; does it have the same status as the truck to the front-left of the stopped car?" However, the associated images show no truck near the trailer, and the ground-truth answer is "yes", which is inconsistent with the scene.

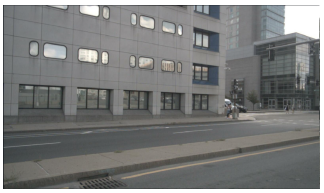
(a) The Trailer is at the Front-Left of the Ego Car.

(b) No Truck Exists around the Trailer.

(c) No Truck Exists to the Back of the Trailer.

Figure 5: The Front-Left Image (left), Back-Left Image (middle), and the Back Image (right) to the Ego Car.

- **Example 2:** Another example asks: "The truck to the front-left of me is in what status?" with the ground-truth answer "moving". However, the associated images clearly show no truck present in front of the ego vehicle, further highlighting the misalignment between the QA pair and the corresponding sensor data.



(a) No Truck Exists to the Front-Left of the Ego Car.

(b) No Truck Exists in the Front of the Ego Car.

(c) No Truck Exists to the Front-Right of the Ego Car.

Figure 6: The Front-Left Image (left), Front Image (middle), and the Front-Right Image (right) to the Ego Car.

- **Example 3:** Similarly, another QA pair asks: "What is the status of the pedestrian to the back right of me?" with the ground-truth answer "not standing". Yet, the sampled camera images show no pedestrian in that region. This misalignment complicates training and undermines supervision quality.

(a) No Pedestrian Exists to the Back-Left of the Ego Car.

(b) No Pedestrian Exists in the Back of the Ego Car.

(c) No Pedestrian Exists to the Back-Right of the Ego Car.

Figure 7: The Back-Left Image (left), Back Image (middle), and the Back-Right Image (right) to the Ego Car.

4.3 New Dataset/Baselines

Due to the critical misalignment issues between QA pairs and sensor data in the NuScenes-QA dataset, we adopt a new dataset: **DriveLM-nuScenes** Gopalkrishnan et al. (2024). Unlike

NuScenes-QA, DriveLM provides accurate matching between each QA pair and the corresponding sensor data—including multi-view images, LiDAR, and object annotations—ensuring reliable visual grounding and interpretable outputs.

DriveLM-nuScenes contains 4,871 keyframes and 91.4 QA pairs per frame on average. Each QA is associated with a full set of six camera images and a synchronized LiDAR scan, enabling the model to reason over complete 360° driving scenes. Questions span a wide range of tasks: perception, prediction, and planning, covering object status, spatial reasoning, occlusions, and safe action planning.

We benchmark the following baselines on the new dataset with detailed metrics in Table 6:

- **Q-only (T5-Base / T5-Q-Large):** These models receive only the textual question as input. Visual embeddings are replaced with zero tensors, maintaining the same embedding size. T5-Base has 223M parameters, while T5-Q-Large has 757M, allowing comparison of language model capacity.
- **EM-VLM4AD (T5-Base / T5-Q-Large):** These multimodal models integrate image features with T5-based language models. The "Base" variant uses a lightweight encoder, while "Q-Large" adopts a larger T5 model with stronger linguistic capacity.
- **DriveLM-Agent:** A vision-language model based on BLIP-2 and Flan-T5-XL, which takes a front-view RGB image and a text prompt as input. The model performs trajectory-aware reasoning by decoding future waypoints as token sequences, using a specially designed inverse mapping to embed real-world trajectory coordinates into the language model's token space.

Model	**BLEU-4** ↑	**METEOR** ↑	**ROUGE-L** ↑	**CIDEr** ↑	**Params** ↓
Q-only$_{\text{Base}}$	45.50	34.10	70.50	3.09	223M
Q-only$_{\text{Q-Large}}$	45.40	34.03	70.10	3.10	757M
EM-VLM4AD$_{\text{Base}}$	45.36	34.49	**71.98**	**3.20**	235M
EM-VLM4AD$_{\text{Q-Large}}$	40.11	34.34	70.72	3.10	769M
DriveLM-Agent Gopalkrishnan et al. (2024)	**53.09**	**36.19**	66.79	2.79	3.96B

Table 6: Comparison of Baseline Models on DriveLM-nuScenes. Q-only models use text only; others incorporate image features.

5 [3 POINTS] PROPOSED MODEL (>1 PAGE)

Our proposed model fuses image and LiDAR modalities using BEVFusion, generating geometrically grounded visual features that capture both semantic appearance and spatial geometry. As shown in Figure 8, we use a dual-branch encoder: a camera encoder extracts features from multi-view RGB images, while a LiDAR encoder encodes point clouds. These are aligned in a shared Bird's Eye View (BEV) space and fused using spatial-aware convolution.

The resulting BEV features are flattened and concatenated with a language embedding sequence (corresponding to the input question tokens). This joint representation is then passed to a pre-trained T5 model (either Base or Q-Large), which acts as a multimodal decoder. We fine-tune this architecture end-to-end to generate natural language answers for graph-based QA tasks.

This unified pipeline enables spatially grounded reasoning across perception, prediction, and planning tasks, benefiting from the complementary strengths of image appearance and LiDAR depth information.

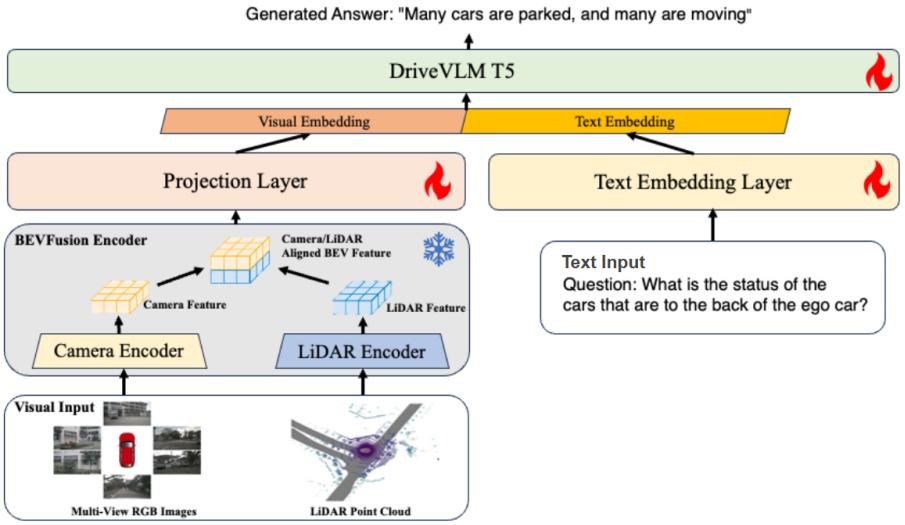


Figure 8: Proposed Model Architecture

5.1 LOSS FUNCTIONS

In the Visual Question Answering (VQA) task, the model output is a sequence of token IDs representing the predicted text. During training, we tokenize the ground-truth answers into token ID sequences as well. We then apply the cross-entropy loss between the predicted token distribution and the ground-truth token IDs. The loss function is formulated as:

$$\mathcal{L}_{\text{CE}} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{t=1}^{T_i}\log p_\theta(y_t^{(i)} \mid y_{<t}^{(i)}, x^{(i)})$$

where N is the batch size, T_i is the length of the i-th ground truth token sequence, $y_t^{(i)}$ is the ground truth token at time step t, $y_{<t}^{(i)}$ are the previous ground truth tokens, $x^{(i)}$ is the input (e.g., features and question), and p_θ is the model's predicted probability distribution parameterized by θ.

5.2 CHANGES TO TRAINING DATA

The DriveLM dataset we used is originally derived from the nuScenes dataset Caesar et al. (2019), which contains multi-modal raw data including 6-camera RGB images and LiDAR sweeps for each key sample. In the original form of DriveLM-nuScenes dataset, each QA pair is annotated with the question, answer, and six corresponding camera images from the nuScenes scene. However,

to enable geometrically grounded reasoning and leverage complementary spatial information, we augment the original QA pairs by incorporating aligned LiDAR sweeps for each sample. This extension allows our model to benefit from both visual appearance cues (from images) and accurate depth geometry (from LiDAR), which is critical for spatial reasoning in driving scenarios.

For the baseline EM-VLM4AD model, only the images are used as input. To extract visual features, it adopts the patch embedding strategy from ViT Dosovitskiy et al. (2020). Given an input RGB image $I \in \mathbb{R}^{3\times H\times W}$, the image is divided into patches, flattened, and passed through a linear projection followed by positional encoding. This results in a sequence of latent image embeddings:

$$V_i \in \mathbb{R}^{S_I \times H_I}$$

where S_I is the number of image patches (i.e., sequence length), and H_I is the hidden dimension of each patch embedding. We use a ViT-B/32 model pretrained on ImageNet Deng et al. (2009) to generate these embeddings.

Our model uses BEVFusion Liu et al. (2022a) as the core feature fusion module. BEVFusion unifies multi-modal features into a shared bird's-eye view (BEV) representation, preserving both geometric precision and semantic richness. Specifically, we employ a Swin-T Liu et al. (2021) network as the image encoder and a VoxelNet Zhou & Tuzel (2017) as the LiDAR encoder. These encoders produce camera and LiDAR features that are projected into the BEV space. The resulting BEV-aligned features from both modalities are fused, serving as input to the downstream QA model.

5.3 Training Setup

We train two versions of our model based on different backbone sizes: one using T5-Base and the other using T5-Large with LoRA fine-tuning (T5-Q-Large). All training is performed on a single NVIDIA RTX A6000 GPU.

The T5-Base model is trained for eight epochs, taking approximately 20 hours to complete, while the T5-Q-Large model requires about two days to finish training for eight epochs. We note that, due to the lightweight nature of our method, the T5-Base version can be easily accommodated within a single T4 GPU instance, allowing for free evaluation on platforms such as Google Colab.

For all models, we use a learning rate of 5×10^{-4}, a weight decay of 0.05, an exponential learning rate scheduler, and a batch size of 4.

5.4 Hyperparameters and their effects

We study the impact of several key hyperparameters on the model's final performance, measured by BLEU-4, METEOR, ROUGE-L, and CIDEr. Below, we detail the experimental settings and findings for three important hyperparameters as the following:

1. Learning Rate. We analyze the effect of different learning rates ($1\mathrm{e}{-5}$, $1\mathrm{e}{-4}$, and $5\mathrm{e}{-4}$) on the final performance of the T5-Base model with BEVFusion features under both non-pretrained and pretrained settings.

Without pretraining (Table 7), a small learning rate of $1\mathrm{e}{-5}$ results in poor performance across all metrics, suggesting underfitting and slow convergence. Increasing the learning rate to $1\mathrm{e}{-4}$ yields significant improvements in BLEU-4, METEOR, ROUGE-L, and CIDEr. A further increase to $5\mathrm{e}{-4}$ provides marginal gains in BLEU-4 and ROUGE-L, but slightly reduces METEOR, indicating that while the model benefits from faster learning dynamics, overly aggressive updates may hinder fine-grained alignment. This suggests that $5\mathrm{e}{-4}$ is effective in promoting expressive outputs, but $1\mathrm{e}{-4}$ provides a slightly more balanced optimization.

LR	BLEU-4	METEOR	ROUGE-L	CIDEr
$1e{-}5$	40.54	32.05	68.69	2.75
$1e{-}4$	50.04	**36.79**	73.59	**3.32**
$5e{-}4$	**50.57**	36.52	**73.74**	3.31

Table 7: Effect of Learning Rate on Final Metrics without Pretraining Stage 8 epochs.

With pretraining (Table 8), the learning rate has a more nuanced impact. The lowest learning rate ($1e{-}5$) achieves the best BLEU-4 (49.66), indicating accurate token-level alignment with ground-truth captions. However, the highest learning rate ($5e{-}4$) leads to the best METEOR (36.82), ROUGE-L (74.56), and CIDEr (3.36) scores, highlighting that a moderately larger learning rate helps the model generate more fluent, diverse, and semantically rich descriptions. This contrast suggests that pretrained models are more stable and can leverage higher learning rates to improve generalization, while smaller rates may restrict learning to exact matching.

LR	BLEU-4	METEOR	ROUGE-L	CIDEr
$1e{-}5$	**49.66**	36.03	73.23	3.31
$1e{-}4$	48.86	35.99	74.13	3.33
$5e{-}4$	48.95	**36.82**	**74.56**	**3.36**

Table 8: Effect of Learning Rate on Final Metrics with Pretraining Stage 8 epochs.

In summary, without pretraining, a moderate learning rate ($1e{-}4$ or $5e{-}4$) is necessary for effective convergence. With pretraining, higher learning rates like $5e{-}4$ allow the model to better exploit the initialization and produce richer captions, though at the slight expense of exact match accuracy.

2. Switch Between T5-Base/T5-Q-Large LLM models. We explore the effect of switching between T5-Base and T5-Q-Large as the backbone language model for our method. As shown in Table 9, we train two models with different LLM sizes: T5-Base and T5-Q-Large.

Specifically, we use two different pre-trained versions of the T5 language model: **T5-Base**, which contains approximately 223 million parameters, and an **8-bit quantized version of T5-Large**, which has around 750 million parameters. Using these pre-trained LMs, we perform fine-tuning to adapt the language model to the concatenated BEV features and text embeddings.

For the T5-Base model, we find that fine-tuning the entire model leads to the best performance. In contrast, for the quantized T5-Large model, we employ LoRA-Fine-Tuning-Aware Quantization Li et al. (2023b), which minimizes quantization error by carefully initializing the LoRA weights during training.

Table 9 compares the performance of T5-Base and T5-Q-Large when fine-tuned with BEVFusion features. While T5-Q-Large achieves a slightly higher BLEU-4 score (**49.54**), indicating stronger surface-level n-gram overlap with ground truth answers, T5-Base achieves the best scores across METEOR (**36.82**), ROUGE-L (**74.56**), and CIDEr (**3.36**). These metrics suggest that T5-Base generates responses with better semantic coverage, longer-span fluency, and alignment with human-like answers. Despite its smaller size, T5-Base proves more effective in capturing essential information for driving-related QA, possibly due to more stable fine-tuning dynamics or better adaptation to structured visual input.

LLM Model	BLEU-4	METEOR	ROUGE-L	CIDEr
T5-Base	48.95	**36.82**	**74.56**	**3.36**
T5-Q-Large	**49.54**	36.73	73.51	3.29

Table 9: Effect of LLM Models on Final Metrics.

3. BEV Feature Projection Layer Complexity. We also investigate how the complexity of projection layers affects the model performance. In our original BEVFusion design, the projection layer consists of only a single linear layer that aligns the BEVFusion features with the text embeddings. We are interested in exploring whether using more complicated projection layers (multiple projection layers with more parameters) can lead to better performance comparing with the original one.

# of the projection layers	# of parameters	BLEU-4	METEOR	ROUGE-L	CIDEr
one layer	33,177,600	**48.95**	**36.82**	**74.56**	**3.36**
two layers	33,964,032	48.62	35.71	73.44	3.27

Table 10: Effect of of Projection Layers Complexity

As shown in the Table 10, we find that increasing the complexity of the projection layers, by adding an additional linear layer, slightly increases the number of parameters but does not lead to performance improvement. In fact, the model with a single projection layer achieves better or comparable results across most metrics (BLEU-4, ROUGE-L, and CIDEr). This means a simple single-layer projection is sufficient for aligning BEV features with text embeddings, and adding more layers introduces unnecessary complexity with the risk of overfitting.

6 [1 POINTS] RESULTS (1 PAGE)

We evaluate our models on the DriveLM-nuScenes benchmark using four standard language generation metrics: **BLEU-4**, **METEOR**, **ROUGE-L**, and **CIDEr**, as well as reporting the model **parameter size**.

- **BLEU-4** measures the precision of 4-gram overlaps between generated and reference text, rewarding exact matches over short sequences.
- **METEOR** considers both precision and recall, aligning words semantically, and penalizing fragmentation, making it more sensitive to synonyms and paraphrasing.
- **ROUGE-L** computes the longest common subsequence between the prediction and the ground truth, emphasizing overall structure matching.
- **CIDEr** evaluates consensus by weighting n-grams based on their importance across a set of references, and is particularly effective in measuring relevance for descriptive tasks.

As shown in Table 11, our models (**Ours$_{\text{Base}}$** and **Ours$_{\text{Q-Large}}$**) outperform prior baselines across most evaluation metrics. Notably, **Ours$_{\text{Base}}$** achieves the highest scores in METEOR (**36.8**), ROUGE-L (**74.6**), and CIDEr (**3.36**), demonstrating strong capabilities in generating semantically rich and consensus-aligned answers. In addition, **Ours$_{\text{Q-Large}}$** achieves the second-best BLEU-4 score (**49.5**), just behind DriveLM-Agent, while using significantly fewer parameters. These results highlight the effectiveness of our approach in balancing precision, fluency, and efficiency.

While BLEU-4 scores remain competitive, the significant improvements in CIDEr and ROUGE-L suggest that our generated outputs are richer and better aligned with human expectations.

Furthermore, despite using significantly fewer parameters than large baseline models such as DriveLM-Agent, our models maintain strong performance across all metrics, highlighting the efficiency and scalability of our approach.

Methods	BLEU-4 ↑	METEOR ↑	ROUGE-L ↑	CIDEr ↑	Parameters ↓
Q-only$_{\text{Base}}$	45.5	34.1	70.5	3.09	223M
Q-only$_{\text{Q-Large}}$	45.4	34.0	70.1	3.10	757M
DriveLM-Agent Sima et al. (2025)	**53.1**	36.2	66.8	2.79	3.96B
EM-VLM4AD$_{\text{Base}}$ Gopalkrishnan et al. (2024)	45.4	34.5	72.0	3.20	235M
EM-VLM4AD$_{\text{Q-Large}}$ Gopalkrishnan et al. (2024)	40.1	34.3	70.7	3.10	769M
Ours$_{\text{Base}}$	49.0	**36.8**	**74.6**	**3.36**	288M
Ours$_{\text{Q-Large}}$	49.5	36.7	73.5	3.29	822M

Table 11: Comparison of Our Methods with Baseline Models on DriveLM-nuScenes.

7 [3 POINTS] ANALYSIS (2 PAGES)

7.1 INTRINSIC METRICS

Intrinsic metrics are not directly tied to the task output but rather reflect the essential skills the model should possess. They may overlap with auxiliary losses but serve as independent indicators of model quality.

Our work focuses on a visual question answering (VQA) task under an autonomous driving scenario. Based on previous research on autonomous driving VQA tasks, such as nuScenes-QA Qian et al. (2024), we observed that higher detection and localization quality directly correlate with improved VQA performance. Therefore, we adopt two widely used metrics in autonomous driving to measure the quality of the extracted features, as detailed below.

Intrinsic Metric 1 Since our work is based on the nuScenes dataset Caesar et al. (2019), we adopt the mean Average Precision (mAP) as evaluation metrics. The mAP is defined as the mean of the average precision over ten classes, evaluated under distance thresholds of 0.5m, 1m, 2m, and 4m:

$$\text{mAP} = \frac{1}{C} \sum_{c=1}^{C} \text{AP}_c \tag{1}$$

Where C denotes the total number of classes (10 in the case of nuScenes), and AP_c represents the Average Precision for class c, computed over multiple distance thresholds (0.5m, 1m, 2m, 4m). The final mAP is obtained by averaging across all classes.

Intrinsic Metric 2 The nuScenes Detection Score (NDS) is a weighted combination of mAP and five True Positive error metrics: mean Average Translation Error (mATE), mean Average Scale Error (mASE), mean Average Orientation Error (mAOE), mean Average Velocity Error (mAVE), and mean Average Attribute Error (mAAE):

$$\text{NDS} = \frac{1}{10} \left[5 \cdot \text{mAP} + \sum_{m \in \mathcal{M}} (1 - \min(1, m)) \cdot 100 \right] \tag{2}$$

Where $\mathcal{M} = \{\text{mATE}, \text{mASE}, \text{mAOE}, \text{mAVE}, \text{mAAE}\}$ represents the five True Positive error metrics.

Each metric is better when smaller, so the score component is computed as $(1 - \min(1, m))$. The final NDS is a weighted combination of mAP and the five error metrics, normalized to a score between 0 and 100.

The results are summarized in Table 12. BEVFusion achieves the best performance in both mAP and NDS among all compared methods, which aligns with its superior results in the final VQA tasks.

Methods	MAP	NDS
CenterPoint (Simple model)	0.3754	0.4408
PETR (Simple model)	0.3830	0.3912
BEVFusion (Multimodal)	0.5209	0.4657

Table 12: A Complete Table of Intrinsic Metrics

7.2 Qualitative Analysis and Examples (full page tables – multiple pages for most projects)

We present both success and failure cases of our proposed method. Although our proposed method incorporates LiDAR inputs, we omit them here due to the difficulty of visualizing LiDAR data effectively.

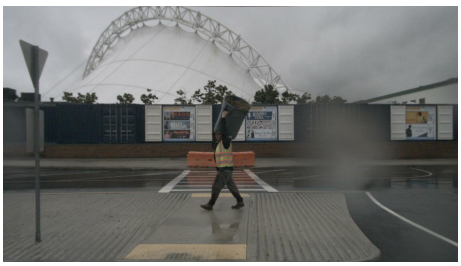
Our proposed method has shown some success in picking up visual hints to answer questions. For example, in the first example shows a scene where the ego vehicle is waiting at a red light, and correctly inferred that the target action of the ego vehicle is to remain stationary.

In the second example, where we asked *"What is the status of the pedestrian that is to the back left of the ego car?"* in pitch-black darkness, it is possible that our proposed method successfully leveraged the LiDAR input, which is unaffected by lighting conditions, and correctly responded *"One pedestrian is standing."*

Q: What is the target action of the ego vehicle?
GT Answer: Stationary.
Q-only: Go straight.
EM-VLM4AD: Go straight.
Our: Stationary.

CAM_BACK_LEFT

Q: What is the status of the pedestrian that is to the back left of the ego car?
GT Answer: One pedestrian is standing.
Q-only: The pedestrian to the back left of the ego car is moving.
EM-VLM4AD: The pedestrian to the back left of the ego car is moving.
Our: One pedestrian is standing.



CAM_BACK_LEFT

Q: What is the future state of <c2,CAM_BACK_LEFT,805.0,512.5>?
GT Answer: Keep going straight.
Q-only: Stationary.
EM-VLM4AD: Stationary.
Our: Keep going straight.

CAM_BACK_RIGHT

Q: What is the status of the car that is to the back right of the ego car?
GT Answer: One car is moving.
Q-only: One car is parked.
EM-VLM4AD: One car is parked.
Our: One car is moving.

Table 13: Cases Where Only Our Method Answered Correctly.

However, these successes are likely coincidental, as we observed a similar number of cases where each individual method was the only one to answer correctly while the other two failed. And in most scenarios, all methods answered incorrectly. This suggests that no method consistently outperforms the others across all scenarios, despite our proposed method having a marginally higher metric score.

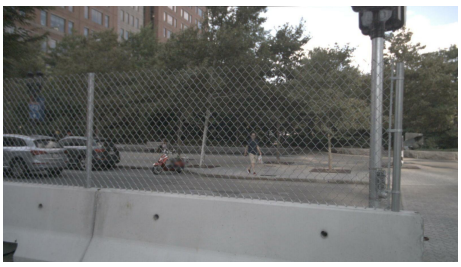
CAM_FRONT_RIGHT

Q: What is the visual description of <c1,CAM_FRONT_RIGHT,905.8,542.5>?
GT Answer: A woman wearing a black jacket.
Q-only: White sedan.
EM-VLM4AD: White van.
Our: White sedan.

CAM_BACK_LEFT

Q: Is <c3,CAM_BACK_LEFT,91.8,645.9> a traffic sign or a road barrier?
GT Answer: Yes.
Q-only: No.
EM-VLM4AD: No.
Our: No.

Table 14: Cases Where Every Method Failed.



CAM_BACK_LEFT

Q: Are there barriers to the back left of the ego car?
GT Answer: Yes.
Q-only: No.
EM-VLM4AD: Yes.
Our: No.

CAM_FRONT

Q: What is the visual description of <c4,CAM_FRONT,1428.3,524.2>?
GT Answer: Black SUV.
Q-only: White sedan.
EM-VLM4AD: Black SUV.
Our: White sedan.

Table 15: Cases Where Every EM-VLM4AD Answered Correctly.

8 LIMITATIONS

1. **Trivial QA Pairs in DriveLM Dataset:**
   Our ablation experiments revealed that our Q-only model, which predicts answers from questions without any camera or LiDAR input, performs surprisingly well. The primary reason is the fact that the question-answer templates lack diversity and are often insufficiently challenging. In many cases, the answer can be inferred directly from the question without requiring visual or spatial reasoning. As a result, models may achieve high performance by simply learning the language priors and answer patterns, rather than leveraging multimodal visual inputs.

Question	**GT Answer**	**Q-only**	**EM-VLM4AD** (w/ cameras)	**Our** (w/ cameras & lidar)
What object would consider <c3,CAM_FRONT_RIGHT, 1010.7,411.4> to be most relevant to its decision?	The ego vehicle.	The ego vehicle.	The ego vehicle.	The ego vehicle.
What actions taken by the ego vehicle can lead to a collision with <c1,CAM_FRONT_RIGHT, 215.8,664.2>?	Sharp right turn.	Sharp right turn.	Sharp right turn.	Slight right turn.
What is the moving status of object <c1,CAM_BACK, 813.3,546.7>?	Going ahead.	Going ahead.	Going ahead.	Going ahead.

Table 16: Trivially Correct Cases Caused by Poorly Constructed QA Pairs.

2. **Over-Reliance on Statistical Prior over Object Classes:**
   All models tend to ignore the visual input and instead rely on object frequency priors, favoring common object classes irrespective of the actual scene.

Question	**GT Answer**	**Q-only**	**EM-VLM4AD** (w/ cameras)	**Our** (w/ cameras & lidar)
What is the visual description of <c1,CAM_FRONT_RIGHT, 905.8,542.5>?	A woman wearing a black jacket.	White sedan.	White van.	White sedan.

Table 17: Models Favoring Frequent Object Classes over Visual Evidence.

3. **Lack of Explicit Spatial Grounding for Referenced Regions:**
   The visual features are fed into the language model as global embeddings of images (and lidar point clouds), without explicit mechanisms to retrieve the region of interest specified in the question (e.g., <c3,CAM_FRONT_RIGHT,1010.7,411.4>). This forces the language model to implicitly resolve spatial grounding from the global context, which is challenging given the lack of structured alignment between the referenced coordinates and the visual embeddings.
4. **Poor Discriminative Power of Metrics:**
   The evaluation metrics we adopt (i.e., BLEU-4, ROUGE-L, and CIDEr) primarily emphasize exact n-gram or subsequence matches, often failing to capture deeper semantic equivalence between predicted and reference answers. While METEOR partially mitigates this issue through synonym matching and stemming, it still heavily relies on surface-level overlaps. Consequently, the discriminative power of these metrics remains limited, reducing the reliability of our experimental comparisons.

9 FUTURE WORK

Building on the identified limitations, we outline key challenges to address and potential directions for improvement:

1. **Search for a High-Quality QA Dataset with Correct, Diverse, and Practical Questions:**
   We have experimented with two driving scenario QA datasets. **NuScenes-QA** Qian et al. (2024) contains a substantial amount of misaligned QA pairs with incorrect answers and unnatural question phrasing. The current dataset, **DriveLM** Gopalkrishnan et al. (2024), provides generally correct answers but still lacks diversity, complexity, and practical relevance. To make any meaningful advances in the QA task will require identifying or constructing a higher-quality dataset with well-formed, non-trivial questions grounded in real-world driving scenarios.
2. **Incorporating Temporal Context with Video Inputs:**
   The current setup answers questions based on single-frame camera and LiDAR inputs. This makes answering questions such as "Is the vehicle in front of me moving?" virtually impossible without relying on indirect visual cues (e.g., traffic lights). Incorporating video inputs would enable the model to capture temporal context directly and reason about dynamic scenes more effectively.
3. **Simultaneous Map Reconstruction with SLAM Methods:**
   To generate informed answers in driving scenarios requires accurate geometric and visual grounding. Recent advances in SLAM methods, such as DUSt3R Wang et al. (2024), enable real-time reconstruction of high-fidelity maps with precise geometry and visual appearance. Integrating such maps could significantly improve the quality of visual features and enhance the model's ability to answer spatially grounded QA tasks.

10 [1 POINTS] ETHICAL CONCERNS AND CONSIDERATIONS (UNINTENTIONAL, MALICIOUS, AND DUAL-USE)

Our work on multimodal question answering (QA) in autonomous driving scenarios introduces several ethical considerations that must be addressed to ensure responsible deployment and development.

UNINTENTIONAL HARM

Despite efforts to improve model accuracy, several limitations—such as dataset misalignments, over-reliance on statistical priors, and lack of temporal context—can lead to **incorrect or misleading answers**. In safety-critical domains like autonomous driving, these inaccuracies could result in poor system decisions or misinterpretations during system evaluation. For instance, misidentifying whether a pedestrian is present or if a vehicle is moving could negatively impact driving policies, either during simulation or in downstream planning modules.

MALICIOUS USE

The models developed could be **misused to manipulate or deceive autonomous systems**. For example, adversarial actors could craft specific questions or manipulate sensor inputs to elicit false responses from the QA system, potentially influencing the decision-making of autonomous vehicles. Additionally, QA systems trained on scene understanding could be repurposed to **infer sensitive details** about environments or individuals, especially when deployed in non-consensual settings such as unauthorized surveillance.

DUAL-USE CONCERNS

While our system is designed to improve **driving safety** and **scene understanding**, the underlying technologies (e.g., multimodal perception, scene graph extraction, QA models) have **dual-use potential**. These methods could be adapted for military or law enforcement purposes, such as **surveil-**

lance systems that analyze vehicular or pedestrian behavior at scale. This raises concerns about **privacy violations**, **mass surveillance**, or deployment in contexts lacking proper oversight.

MITIGATION STRATEGIES

To address these concerns, we propose several mitigation strategies:

- **Dataset auditing**: Ensure that datasets used for training and evaluation are well-aligned, diverse, and representative of real-world scenarios, minimizing biases and reducing the risk of unintended errors.
- **Robustness testing**: Incorporate adversarial testing frameworks to evaluate the model's resilience against manipulated inputs or misleading questions.
- **Transparency and explainability**: Develop mechanisms to provide **interpretable outputs** or rationales for model predictions, aiding human oversight and fostering trust.
- **Usage guidelines**: Define clear **ethical guidelines and usage policies** for deployment contexts, discouraging applications in scenarios that could violate privacy, safety, or human rights.

Addressing these ethical challenges is crucial for ensuring that the benefits of multimodal QA systems in autonomous driving are realized without causing harm or enabling misuse.

REFERENCES

Song Bai et al. Learning interpretable models for self-driving vehicles via bdd-x. In *ECCV*, 2021.

Yicheng Bai et al. Transfusion: Robust lidar-based 3d object detection with dense transformers. In *CVPR*, 2022.

Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yuxin Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11618–11628, 2019. URL https://api.semanticscholar.org/CorpusID:85517967.

Hang Chen et al. Lingoqa: Language-driven visual reasoning for autonomous vehicles. In *CVPR*, 2023a.

Long Chen et al. Driving with llms: Fusing object-level vector modality for explainable autonomous driving. *arXiv preprint arXiv:2310.01957*, 2023b.

Jaemin Cho et al. Unifying vision-and-language tasks via text generation. In *ICML*, 2021.

Wenliang Dai et al. Instructblip: Towards general-purpose vision-language models with instruction tuning. *arXiv preprint arXiv:2305.06500*, 2023.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, K. Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009. URL https://api.semanticscholar.org/CorpusID:57246310.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ArXiv*, abs/2010.11929, 2020. URL https://api.semanticscholar.org/CorpusID:225039882.

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

Wonjae Kim et al. Vilt: Vision-and-language transformer without convolution or region supervision. In *ICML*, 2021.

Junnan Li et al. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*, 2023a.

Tianyuan Li et al. Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. In *ECCV*, 2022.

Yixiao Li, Yifan Yu, Chen Liang, Pengcheng He, Nikos Karampatziakis, Weizhu Chen, and Tuo Zhao. Loftq: Lora-fine-tuning-aware quantization for large language models. *arXiv preprint arXiv:2310.08659*, 2023b.

Haotian Liu et al. Llava: Large language-and-vision assistant. *arXiv preprint arXiv:2304.08485*, 2023.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9992–10002, 2021. URL https://api.semanticscholar.org/CorpusID:232352874.

Zhijian Liu, Haotian Tang, Alexander Amini, Xinyu Yang, Huizi Mao, Daniela Rus, and Song Han. Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation. *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2774–2781, 2022a. URL https://api.semanticscholar.org/CorpusID:249097415.

Zhijian Liu et al. Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation. In *NeurIPS*, 2022b.

OpenAI. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Tianwen Qian, Jingjing Chen, Linhai Zhuo, Yang Jiao, and Yu-Gang Jiang. Nuscenes-qa: A multi-modal visual question answering benchmark for autonomous driving scenario. *arXiv preprint arXiv:2305.14836*, 2024. URL chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://arxiv.org/pdf/2305.14836.

Alec Radford et al. Learning transferable visual models from natural language supervision. *ICML*, 2021.

Aditya Ramesh et al. Zero-shot text-to-image generation. *arXiv preprint arXiv:2102.12092*, 2021.

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, 2025. URL https://arxiv.org/abs/2312.14150.

Yuming Tang et al. Human attention dataset (had) for driving. In *CVPR*, 2023.

Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy, 2024. URL https://arxiv.org/abs/2312.14132.

Wenhai Wang et al. Drivemlm: Aligning multimodal large language models with planning states. *arXiv preprint arXiv:2309.09952*, 2023.

Wenyi Wu et al. Mivc: Multi-instance visual components for vision-language models. In *WACV*, 2024.

Yuheng Wu et al. Visual chatgpt: Talking, drawing and editing with visual foundation models. *arXiv preprint arXiv:2303.04671*, 2023.

Zhenhua Xu et al. Drivegpt4: Interpretable end-to-end autonomous driving via llm. *arXiv preprint arXiv:2310.01412*, 2023.

Lei Yang et al. nuprompt: Benchmarking vision-language models for autonomous driving with prompting. *arXiv preprint arXiv:2311.05773*, 2023.

Yecheng Yao et al. Drama: A driving dataset for rare and multi-agent interactions. In *Conference on Robot Learning (CoRL)*, 2022.

Qiangeng Yu et al. Point-bert: Pre-training 3d point cloud transformers with masked point modeling. In *CVPR*, 2022.

Ziyang Zhao et al. Vila: Unified instruction-tuned vision-language foundation model. *arXiv preprint arXiv:2305.14588*, 2023.

Yanyan Zhou et al. Rank2tell: Grounded question answering via ranking and generation. In *CVPR*, 2023.

Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4490–4499, 2017. URL https://api.semanticscholar.org/CorpusID:42427078.

Deyi Zhu et al. Minigpt-4: Enhancing vision-language understanding with gpt-4 level capabilities. *arXiv preprint arXiv:2304.10592*, 2023.