NuScenes-QA: A Multi-Modal Visual Question Answering Benchmark for Autonomous Driving Scenario

Tianwen Qian[1], Jingjing Chen[2†], Linhai Zhuo[2], Yang Jiao[2], Yu-Gang Jiang[2]

[1]Academy for Engineering and Technology, Fudan University
[2]Shanghai Key Lab of Intelligent Information Processing, School of Computer Science, Fudan University
{twqian19, chenjingjing, lhzhuo19, ygj}@fudan.edu.cn, yjiao23@m.fudan.edu.cn

Abstract

We introduce a novel visual question answering (VQA) task in the context of autonomous driving, aiming to answer natural language questions based on street-view clues. Compared to traditional VQA tasks, VQA in autonomous driving scenario presents more challenges. Firstly, the raw visual data are multi-modal, including images and point clouds captured by camera and LiDAR, respectively. Secondly, the data are multi-frame due to the continuous, real-time acquisition. Thirdly, the outdoor scenes exhibit both moving foreground and static background. Existing VQA benchmarks fail to adequately address these complexities. To bridge this gap, we propose NuScenes-QA, the first benchmark for VQA in the autonomous driving scenario, encompassing 34K visual scenes and 460K question-answer pairs. Specifically, we leverage existing 3D detection annotations to generate scene graphs and design question templates manually. Subsequently, the question-answer pairs are generated programmatically based on these templates. Comprehensive statistics prove that our NuScenes-QA is a balanced large-scale benchmark with diverse question formats. Built upon it, we develop a series of baselines that employ advanced 3D detection and VQA techniques. Our extensive experiments highlight the challenges posed by this new task. Codes and dataset are available at https://github.com/qiantianwen/NuScenes-QA.

Introduction

Autonomous driving is a rapidly developing field with immense potential to improve transportation safety and efficiency with advancements in sensor technologies and computer vision. As the increasing maturity of traditional perceptual techniques such as 3D object detection (Liu et al. 2023; Jiao et al. 2023b) and tracking (Chen et al. 2023), autonomous driving systems are progressing towards enhanced interpretability and flexible human-car interactivity. In this context, visual question answering (VQA) (Antol et al. 2015) can play a critical role. On one hand, VQA possesses interactive and entertainment, enabling passengers to perceive their surroundings through language and enhancing the user experience of intelligent driving systems. On the other hand, users can verify the correctness of perception system through question answering, fortifying their trust in its capabilities.

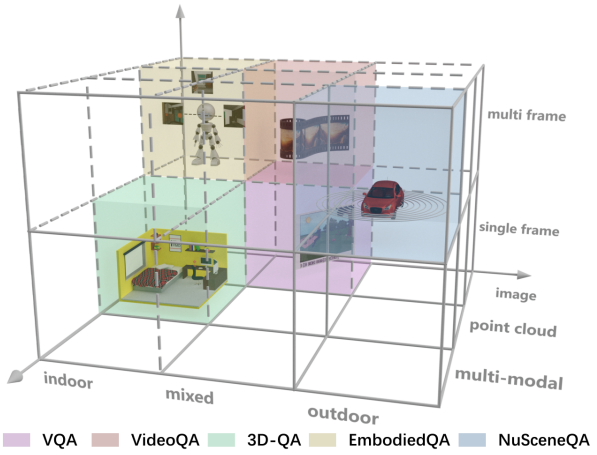


Figure 1: NuScenes-QA is a multi-modal, multi-frame, outdoor dataset that differs significantly from other VQA benchmarks in terms of visual data.

Despite the notable progress made by the VQA community, models trained on existing VQA datasets (Goyal et al. 2017; Hudson and Manning 2019) have limitations in addressing the complexities of autonomous driving scenario. This limitation is primarily caused by the difference in visual data between self-driving scenario and existing VQA benchmarks. For instance, to answer question like "*Are there any moving pedestrians in front of the stopped bus?*", it is necessary to locate and identify the bus, pedestrians, and their status accurately. This requires the model to effectively leverage the complementary information from images and point clouds to understand complex scenes and capture object dynamics from multiple frames of data streams. Therefore, it is essential to explore VQA in the context of multi-modal, multi-frame and outdoor scenario. However, existing VQA benchmarks cannot satisfy all these conditions simultaneously, as illustrated in Fig. 1. For instance, although 3D-QA (Azuma et al. 2022) and the self-driving scenario both focus on understanding the structure and spatial relationships of objects, 3D-QA is limited to single-modal (*i.e.*, point cloud),

†Correspondence to: Jingjing Chen.

single-frame, and static indoor scenes. The same goes for other benchmarks. To bridge this gap, we construct the first VQA benchmark specifically designed for autonomous driving scenario, named NuScenes-QA. NuScenes-QA is different from all other existing VQA benchmarks in terms of visual data characteristics, presenting new challenges for both VQA and autonomous driving community.

The proposed NuScenes-QA is built upon nuScenes (Caesar et al. 2020), which is a popular 3D perception dataset for autonomous driving. We automatically annotate the question-answer pairs using the CLEVR benchmark (Johnson et al. 2017) as inspiration. To be specific, we consider each keyframe annotated in nuScenes as a "scene" and construct a related scene graph. The objects and their attributes are regarded as the nodes in the graph, while the relative spatial relationships between objects are regarded as the edges, which are calculated based on the 3D bounding boxes annotated in nuScenes. Additionally, we design different types of question templates manually, including counting, comparison, and existence, etc. Based on these constructed templates and scene graphs, we sample different parameters to instantiate the templates, and use the scene graph to infer the correct answers, thus automatically generating question-answer pairs. Eventually, we obtained a total of 460K question-answer pairs on 34K scenes from the annotated nuScenes training and validation split, with 377K pairs for training and 83K for testing.

In addition to the dataset, we also develop baseline models using the existing 3D perception (Huang et al. 2021; Yin, Zhou, and Krahenbuhl 2021; Jiao et al. 2023a) and visual question answering (Anderson et al. 2018; Yu et al. 2019) techniques. These models fall into three categories: image-based, point cloud-based, and multi-modal fusion-based. The 3D detection models are used to extract visual features and provide object proposals, which are then combined with question features and fed into the question answering model for answer decoding. While our experiments show that these models outperform the question-only blind model, their performance still significantly lags behind models that use ground truth object labels as inputs. This indicates that combining existing technologies is not sufficient for intricate street views understanding. Thus, NuScenes-QA poses a new challenge, urging further research in this realm.

Overall, our contributions can be summarized as follows:

- We introduce a novel visual question answering task in autonomous driving scenario, which evaluates current deep learning based models' ability to understand and reason with complex visual data in multi-modal, multi-frame, and outdoor scenes. To facilitate this task, we contribute a large-scale dataset, NuScenes-QA, consisting of 34K complex autonomous driving scenes and 460K question-answer pairs.
- We establish several baseline models and extensively evaluate the performance of existing techniques for this task. Additionally, we conduct ablation experiments to analyze specific techniques that are relevant to this task, which provide a foundation for future research.

Related Works

Visual Question Answering

There are various datasets available for VQA, including image-based datasets such as VQA2.0 (Goyal et al. 2017), CLEVR (Johnson et al. 2017), and GQA (Hudson and Manning 2019), as well as video-based datasets such as TGIF-QA (Jang et al. 2017) and TVQA (Lei et al. 2018). For the image-based VQA, earlier works (Lu et al. 2016; Anderson et al. 2018; Qian et al. 2022a) typically use CNNs to extract image features, and RNNs to process the question. Then, joint embddings of vision and language obtained through concatenation or other operations (Kim, Jun, and Zhang 2018) are input to the decoder for answer prediction. Recently, many Transformer-based models (Tan and Bansal 2019; Zhang et al. 2021) have achieved state-of-the-art performance through large-scale vision-language pre-training. Differing to image-based VQA, VideoQA (Jang et al. 2017; Zhu et al. 2017) places greater emphasis on mining the temporal contextual from videos. For example, Jiang *et al.* (Jiang et al. 2020) proposed a question-guided spatial-temporal contextual attention network, and Qian *et al.* (Qian et al. 2022b) suggested first localizing relevant segments in a long-term video before answering.

3D Visual Question Answering

3D Visual Question Answering (3D-QA) is a novel task in the VQA field that focuses on answering questions about 3D scenes represented by point cloud. Unlike traditional VQA tasks, 3D-QA requires models to understand the geometric structure and the spatial relations of objects in a indoor scene. Recently, many 3D-QA datasets have been constructed. For example, the 3DQA dataset (Ye et al. 2022), which is based on ScanNet (Dai et al. 2017), has manually annotated 6K question-answer pairs. Similarly, ScanQA (Azuma et al. 2022) has utilized a question generation model along with manual editing to annotate 41K pairs on the same visual data. Despite these advancements, current 3D-QA models face limitations in solving more complex autonomous driving scenario, which involve multi-modalities, multi-frames, and outdoor scenes.

Vision-Language Tasks in Autonomous Driving

Language systems are pivotal for communication between the passengers and vehicles. Pioneering efforts have explored language-guided visual understanding in this context. For instance, Deruyttere *et al.* proposed the Talk2Car (Deruyttere et al. 2019), which is the first object referral dataset with natural language commands for self-driving cars. Wu *et al.* developed a benchmark with scalable expressions named Refer-KITTI (Dongming et al. 2023) based on the self-driving dataset KITTI (Geiger, Lenz, and Urtasun 2012). It aims to track multiple targets based on language descriptions. In contrast, our proposed NuScene-QA stands out in two ways. Firstly, it tackles high-level question answering, demanding both understanding and reasoning. Secondly, NuScenes-QA provides richer visual information, including images and point clouds.

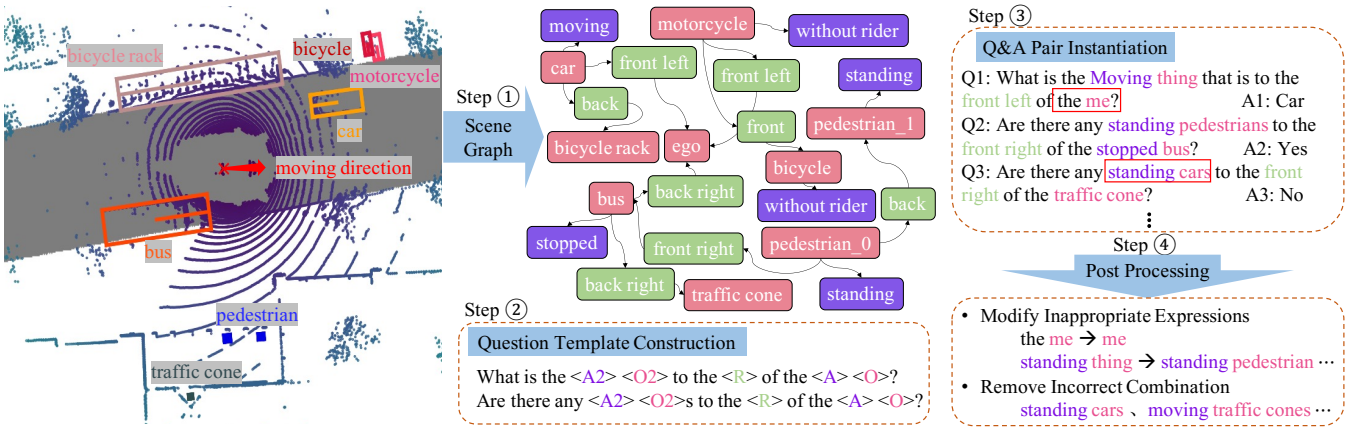


Figure 2: Data construction flow of NuScenes-QA. First, the scene graphs are generated using the annotated object labels and 3D bounding boxes. Then, we design question templates manually, and instantiate the question-answer pairs with them. Finally, the generated data are filtered based on certain rules.

NuScenes-QA Dataset

Our primary contribution is the NuScenes-QA dataset, which we will introduce in detail in this section. We provide a comprehensive overview of the dataset construction, including scene graph development, question template design, question-answer pair generation, and post-processing. In addition, we analyze the statistical characteristics of the NuScenes-QA dataset, such as the distribution of question types, lengths, and answers.

Data Construction

For question-answer pairs generation, we adapted an automated method inspired by CLEVR (Johnson et al. 2017). This method requires two types of structured data: scene graphs generated from 3D annotations, containing object category, position, and relationships; alongside manually crafted question templates that specify the question type, expected answer type, and reasoning required to answer it. By combining these structured data, we automatically generate question-answer pairs. These pairs are then filtered and validated through post-processing programs to construct the complete dataset. Fig. 2 illustrates the overall data construction pipeline.

Scene Graph Construction A scene graph (Johnson et al. 2015) is defined as an abstract representation of a visual scene, where nodes in the graph represent objects in the scene, and edges represent relationships between objects.

In nuScenes, the collected data is annotated with a frequency of 2Hz, and each annotated frame is referred as a "keyframe". We consider each keyframe as a "scene" in NuScenes-QA. The existing annotations include object categories and their attributes in the scene, as well as the 3D bounding boxes of the objects. These annotated objects and their attributes are directly used as nodes in the graph. However, relationships between objects are not provided in the original annotations, so we developed a rule for calculating object relationships. Given that spatial position relationships are crucial in autonomous driving scenario, we define six relationships between objects, namely *front*, *back*, *front left*, *front right*, *back left*, and *back right*. To determine object relationships, we first project 3D bounding boxes onto the Bird's-Eye-View (BEV). Subsequently, we calculate the angle between the vector connecting the centers of two bounding boxes and the forward direction of the ego-car. The formula is given by

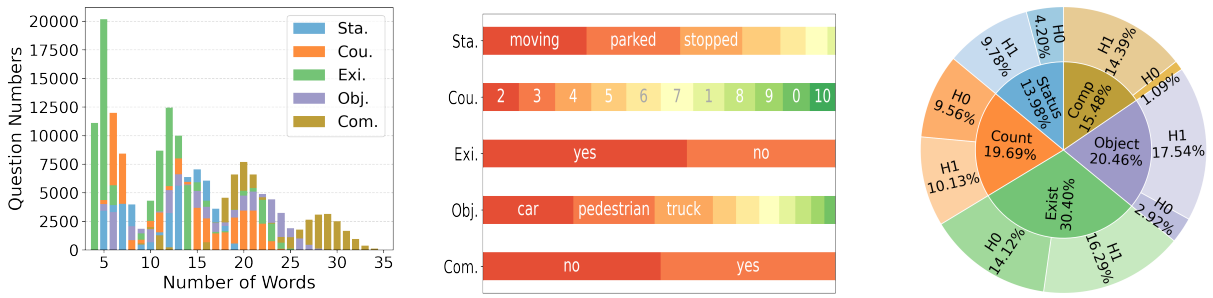
$$\theta = \cos^{-1} \frac{(B_1[:2] - B_2[:2]) \cdot V_{ego}[:2]}{\|B_1[:2] - B_2[:2]\| \|V_{ego}[:2]\|}, \tag{1}$$

where $B_i = [x, y, z, x_{size}, y_{size}, z_{size}, \varphi]$ is the 3D bounding box of object i, and $V_{ego} = [v_x, v_y, v_z]$ represents the speed of the ego car. Based on the angle range, the relationship between two objects is defined as

$$relation = \begin{cases} front & \text{if } -30^\circ < \theta <= 30^\circ \\ front\ left & \text{if } 30^\circ < \theta <= 90^\circ \\ front\ right & \text{if } -90^\circ < \theta <= -30^\circ \\ back\ left & \text{if } 90^\circ < \theta <= 150^\circ \\ back\ rigth & \text{if } -150^\circ < \theta <= -90^\circ \\ back & \text{else.} \end{cases} \tag{2}$$

We define the forward direction of the car as 0° and counterclockwise angle as positive. At this point, we can convert the annotations of nuScenes into the scene graphs we need, as illustrated in step one of Fig. 2.

Question Template Design We devised templates manually for question generation. For instance, the question "*What is the moving thing to the front left of the stopped bus?*" can be abstracted as the template "*What is the <A2><O2>to the <R>of the <A1><O1>?*", with *<A>*, *<O>*, and *<R>* as parameters for instantiation, representing attribute, object, and relationship, respectively. Additionally, we can express the same semantic with another form like "*There is a <A2><O2>to the <R>of the <A1><O1>, what is it?*".



(a) Question Length Distribution (b) Answer Distribution (c) Category Distribution

Figure 3: Statistical distributions of questions and answers in the NuScenes-QA training split.

Ultimately, NuScenes-QA holds 66 diverse question templates, divided into 5 question types: *existence*, *counting*, *query-object*, *query-status*, and *comparison*. In addition, to better evaluate the models reasoning performance, we also divide the questions into *zero-hop* and *one-hop*. Specifically, zero-hop questions require no reasoning between objects, *e.g.*, "*What is the status of the <A><O>?*". One-hop questions involve one step spatial reasoning, *e.g.*, "*What is the status of the <A2><O2>to the <R>of the <A1><O1>?*". Comprehensive template details are available in the supplementary material.

Q&A Pair Generation and Filtering Given the scene graphs and question templates, instantiating a question-answer pair is straightforward: we select a template, assign parameter values through depth-first search, and deduce the ground truth answer on the scene graph. Moreover, we dismiss ill-posed or degenerate questions. For instance, the question is ill-posed if the scene do not contain any cars or pedestrians when *<O1>==pedestrian* and *<O2>==car* is assigned for the template "*What is the status of the <O2>to the <R>of the <A1><O1>?*".

It is important to note that post-processing, as depicted in step 4 of Fig. 2, addresses numerous unsuitable expressions. For example, we added the ego-car as an object in the scene, it is referred to as "*me*" in questions. This led to some inappropriate instances like "*the me*" or "*there is a me*" when *<O>* is assigned "*me*". We revised such expressions. In addition, during the instantiation, inappropriate *<A>+<O>* combinations like "*standing cars*" and "*parked pedestrians*" were eliminated through rules. Also, we removed questions with counting answers greater than 10 to balance the answer distribution.

Data Statistics

In total, NuScenes-QA provides 459,941 question-answer pairs across 34,149 visual scenes, with 376,604 questions from 28,130 scenes for training, and 83,337 questions from 6,019 scenes for testing. To the best of our knowledge, NuScenes-QA is currently the largest 3D related question answering dataset. Detailed comparison of 3D-QA datasets can be found in supplementary materials.

Fig. 3 depicts various statistical distributions of NuScenes-QA. Fig. 3(a) showcases a broad spectrum of question lengths (5 to 35 words). Fig. 3(b) and 3(c) present answer and question category distributions, revealing the balance of NuScenes-QA. A balanced dataset can prevent models from learning answer biases or language shortcuts, which are common in many other VQA benchmarks (Antol et al. 2015; Azuma et al. 2022).

Method

Along with the proposed dataset, we provide several baselines based on existing 3D detection and VQA techniques.

Task Definition

Given a visual scene S, and a question Q, the task of visual question answering aims to select an answer $\hat{a}$ from the answer space $\mathcal{A} = \{a_i\}_{i=1}^{N}$ that best answers the question. Therefore, the task can be formulated as:

$$\hat{a} = \arg\max_{a \in \mathcal{A}} P(a \mid S, Q). \tag{3}$$

For NuScenes-QA, visual scene data encompass multi-view images I, point clouds P, and any frames I_i and P_i before the current frame in the data sequences. We can further decompose the Eq. 3 into:

$$\begin{aligned} P(a \mid S, Q) &= P(a \mid I, P, Q) \\ I &= \{I_i,\ T - t < i \leq T\} \\ P &= \{P_i,\ T - t < i \leq T\}, \end{aligned} \tag{4}$$

where T is the index of current frame and t is the number of previous frame used in the model. It is also possible to use only single modality or single frame data for prediction.

Framework Overview

The overall framework of our proposed baseline is illustrated in Fig. 4 and mainly consists of three components. The first is the feature extraction backbone, which includes both image and point cloud feature extractor. The second part is the region proposal module for object embedding, and the last component is the QA-head for answer prediction.

Initially, the surrounded-view images and point clouds are fed into the feature extraction backbone, with features projected onto the Bird's-Eye-View (BEV). Subsequently, 3D bounding boxes inferred by a pre-trained detection model

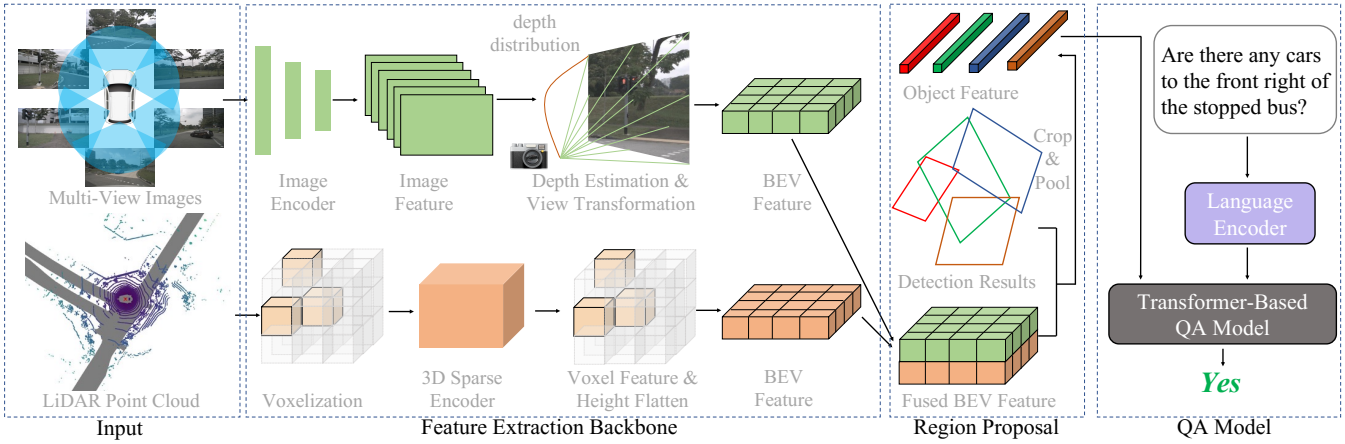


Figure 4: Framework of baseline. The multi-view images and point clouds are first processed by the feature extraction backbone to obtain BEV features. Then, the objects embeddings are cropped based on the detected 3D bounding boxes. Finally, these objects features are fed into the question-answering head along with the given question for answer decoding.

are used to crop and pool object features. Finally, the QA-model takes the question features and the object features as input for cross-modal interaction to predict the answer.

Input Embedding

Question Embedding For a question $Q = \{w_i\}_{i=1}^{n_q}$ that contains n_q words, we first tokenize it and initialize the tokens with pre-trained GloVe (Pennington, Socher, and Manning 2014) embeddings. The sequence is then fed into a single-layer biLSTM (Hochreiter and Schmidhuber 1997) for word-level context encoding. Each word feature $\mathbf{w}_i$ is represented by the concatenation of the forward and backward hidden states of the biLSTM, denoted as:

$$\mathbf{w}_i = [\overrightarrow{\mathbf{h}_i};\ \overleftarrow{\mathbf{h}_i}] \in \mathbb{R}^d, \tag{5}$$

and the question embedding is represented as $\mathbf{Q} \in \mathbb{R}^{n_q \times d}$.

Visual Feature Extraction We adopt leading-edge 3D detection techniques for visual feature extraction. As shown in Fig. 4, it entails two branches: image stream and point cloud stream. For multi-view images, ResNet (He et al. 2016) with FPN (Lin et al. 2017) is used as the backbone for multi-scale feature extraction. Then, in order to make the feature spatial-aware, we estimate the depth of the pixels in the images and lift them to 3D virtual points with a view transformer inspired by LSS (Philion and Fidler 2020). Finally, pooling along the Z-axis compresses the feature in voxel space, producing the BEV featmap $\mathbf{M}_I \in \mathbb{R}^{H \times W \times d_m}$.

For point clouds, we first partition 3D space into voxels, transforming raw point clouds into binary voxel grids (Zhou and Tuzel 2018). Subsequently, 3D sparse convolutional neural network (Graham, Engelcke, and Van Der Maaten 2018) is applied to the voxel grid for feature representation. Similar to the image features mentioned earlier, Z-axis pooling yields the point cloud BEV featmap $\mathbf{M}_P \in \mathbb{R}^{H \times W \times d_m}$. Combining $\mathbf{M}_I$ and $\mathbf{M}_P$, we can aggregate them to obtain multi-modal featmap $\mathbf{M} \in \mathbb{R}^{H \times W \times d_m}$.

Object Embedding Following 2D detection works (Ren et al. 2015), we crop and pool the features in bounding boxes as the object embedding. However, unlike standard 2D bounding boxes aligned with the coordinate axis in images, projecting 3D boxes to BEV yields rotated boxes unsuited for standard RoI Pooling. To this end, we make some modifications. Firstly, we project the 3D box $B = [x, y, z, x_{size}, y_{size}, z_{size}, \varphi]$ into the BEV featmap:

$$x_m = \frac{x - R_{pc}}{F_v \times F_o}, \tag{6}$$

where,F_v, F_o and R_{pc} represent the voxel factor, out size factor of the backbone, and the point cloud range, respectively. All box parameters follow the Eq. 6 to transform into BEV space except the heading angle φ. Then, based on the center and size of the box, we can easily calculate the four vertices $V = \{x_i, y_i\}_{i=0}^{3}$. Secondly, we calculate the rotated vertex $V^{'}$ via the heading angle φ:

$$\begin{bmatrix} x_i^{'} \\ y_i^{'} \end{bmatrix} = \begin{bmatrix} \cos\varphi & -\sin\varphi \\ \sin\varphi & \cos\varphi \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix} \tag{7}$$

Finally, we use cross product algorithm to identify pixel membership in the rotated rectangle. Then, we perform mean pooling on the features of all the pixels inside the rectangle to obtain the object embedding $\mathbf{O} \in \mathbb{R}^{N \times d_m}$. Algorithm details can be found in supplementary materials.

Answer Head and Training

We adopt the classical VQA model MCAN (Yu et al. 2019) as our answer head. It leverages stacked self-attention layers to model the language and visual context independently, along with stacked cross-attention layers for cross-modal feature interaction. The fused features are then projected to the answer space for prediction via basic MLP layers.

During the training phase, we extract the object embeddings using a pre-trained 3D detection model offline. And answer head is trained with the standard cross-entropy loss.

Models	Exist			Count			Object			Status			Comparison			Acc
	H0	H1	All	H0	H1	All	H0	H1	All	H0	H1	All	H0	H1	All	
Q-Only	81.7	77.9	79.6	17.8	16.5	17.2	59.4	38.9	42.0	57.2	48.3	51.3	79.5	65.7	66.9	53.4
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	67.7	57.0
CenterPoint+BUTD	87.7	81.1	84.1	21.9	20.7	21.3	70.2	45.6	49.2	62.8	52.4	55.9	81.6	68.0	69.2	58.1
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
GroundTruth+BUTD	98.9	87.2	92.6	76.8	38.7	57.5	99.7	71.9	76.0	98.8	81.9	87.6	98.1	76.1	78.1	79.2
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	87.7	82.3	84.8	22.5	19.1	20.8	71.3	49.0	52.3	66.6	56.3	59.8	82.4	68.8	70.0	59.5
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
GroundTruth+MCAN	99.6	95.5	97.4	52.7	39.9	46.2	99.7	86.2	88.2	99.3	95.4	96.8	99.7	90.2	91.0	84.3

Table 1: Results of different models on the NuScenes-QA test set. We evaluate top-1 accuracy across the overall test split and different question types. H0 denotes zero-hop and H1 denotes one-hop. C denotes camera, L denotes LiDAR.

Experiments

To validate the challenge of NuScenes-QA, we assess baseline performance in various configurations: camera-only or LiDAR-only single-modality models, camera-lidar fusion models, and diverse answering heads. We conduct ablation studies on crucial steps of the baseline, including BEV feature cropping and pooling strategies, as well as the influence of detected 3D bounding boxes. Furthermore, visualization samples are showcased in supplementary material.

Evaluation Metrics

Questions in NuScenes-QA span 5 categories based on query format: 1) **Exist**, querying the existence of a object in the scene; 2) **Count**, object counting under specified conditions; 3) **Object**, object recognition based on language description; 4) **Status**, querying the status of a specified object; 5) **Comparison**, specified objects or their status comparison. Additionally, questions are also divided into two groups based their complexity of reasoning: zero-hop (denoted as **H0**) and one-hop (denoted as **H1**). We adopt the Top-1 accuracy as our evaluation metric, follow the practice of many other VQA works (Antol et al. 2015; Azuma et al. 2022), and evaluate the performance of different question types separately.

Implementation Details

For the feature extraction backbone, we use the pre-trained detection model following the original settings (Huang et al. 2021; Yin, Zhou, and Krahenbuhl 2021; Jiao et al. 2023a). The dimension of the QA model d_m is set to 512, and MCAN adopts a 6-layer encoder-decoder version. As for training, we used the Adam optimizer with an initial learning rate of 1e-4 and half decaying every 2 epochs. All experiments are conducted with a batch size of 256 on 2 NVIDIA GeForce RTX 3090 GPUs. More details can be found in supplementary material.

Quantitative Results

Compared Methods As mentioned earlier, our task can be divided into three settings: camera-only, LiDAR-only, camera+LiDAR. To explore the impact of different modalities on the question-answering performance, we select representative backbone for each setting. We choose **BEVDet** (Huang et al. 2021) for camera-only setting, which proposed a novel paradigm of explicitly encoding the perspective-view features into the BEV space. **CenterPoint** (Yin, Zhou, and Krahenbuhl 2021) is selected for LiDAR-only setting. It introduced a center-based object keypoint detector and has shown excellent performance in both detection accuracy and speed. For the multi-modal model, we opt for **MSMDFusion** (Jiao et al. 2023a), which leverages depth and fine-grained LiDAR-camera interaction, achieving state-of-the-art results on the nuScenes detection benchmark for single model.

Regarding the QA-head, we select two classic models, **BUTD** (Anderson et al. 2018) and **MCAN** (Yu et al. 2019). BUTD advocates for computing bottom-up and top-down attention on salient regions of the image. MCAN stacks self-attention and cross-attention modules for vision-language feature interaction. To establish the upper bound of the QA models, we employ perfect perceptual results, *i.e.*, ground-truth object labels. Specifically, we use GloVe for objects and their status embedding, noted as **GroundTruth** in Table 1. Additionally, we design a **Q-Only** baseline to investigate the impact of language bias. Q-Only can be considered as a blind model that ignores the visual input.

Results and Discussions According to the results shown in Table 1, we have the following observations that are worth discussing.

1. It is evident that visual data play a critical role in the performance of our task. When comparing the Q-Only baseline to others, we find that it only achieves an accuracy of 53.4%, which is significantly lower than that of other models. For instance, MSMDFusion+MCAN performs 7% better. This indicates that model cannot achieve good performance solely rely on language shortcuts, but needs to leverage rich visual information.

2. Referring to the bottom part of Table 1, we can see that the LiDAR-based CenterPoint outperforms the camera-based BEVDet, achieving accuracy of 57.9% and 59.5%, respectively. We attribute this performance gap to the task characteristics. Images possess detailed texture information, point clouds excel in structure and spatial representation. Our proposed NuScenes-QA emphasizes more on the understanding of structure and spatial relationships of objects. On the other hand, the fusion-based model MSMDFusion

Variants	Question Types					All
	Exi.	Cou.	Obj.	Sta.	Com.	
Det w/o boxes	84.8	20.8	52.3	59.8	70.0	59.5
Det w/ boxes	84.3	21.7	53.0	57.7	67.2	58.9
GT w/o boxes	91.2	38.8	61.1	80.3	76.8	70.8
GT w/ boxes	97.4	46.2	88.2	96.8	91.0	84.3

Table 2: Ablation comparison between model trained with and without detection boxes feature.

attains the best performance with an accuracy of 60.4%, demonstrating the camera and LiDAR data are complementary. Further work can explore how to better exploit the complementary information of multi-modal data. Of course, our baselines still have a long way to go compared to the GroundTruth (achieving an accuracy of 84.3%).

3. According to Table 1, QA-head has a significant impact on the performance. With the same detection backbone, we observed that the QA-head based on MCAN outperforms BUTD by a large margin. For example, the overall accuracy of CenterPoint+MCAN is 59.5%, 1.4% higher than CenterPoint+BUTD. A dedicated QA-head designed for NuScenes-QA may lead to a greater improvement. We leave this as future work.

4. In a horizontal comparison of Table 1, it is not difficult to find that counting is the most difficult among all question types. Our best baseline model achieved just 23.2% accuracy, much lower than other question types. Counting is historically tough in visual question answering, and some explorations (Zhang, Hare, and Prügel-Bennett 2018) have been made in traditional 2D-QA. Future efforts could involve counting modules to enhance its performance.

Ablation Studies

To validate effectiveness of different operations in our baselines, we conduct extensive ablation experiments on the NuScenes-QA test split using the CenterPoint+MCAN baseline combination.

Effects of Bounding Boxes Most 2D and 3D VQA models fuse the visual feature with object bounding box in the object embedding stage, making it position-aware. We follow this paradigm and evaluate the impact of 3D bounding boxes in our NuScenes-QA. Specifically, we project the 7-dimensional box $B = [x, y, z, x_{size}, y_{size}, z_{size}, \varphi]$ obtained from the detection model onto the same dimension as the object embeddings using MLP, and concatenate the two features as the final input for the QA head. As shown in Table 2, we are surprised to find that the performance varies significantly on different data. Adding box features for ground truth can increase the model's accuracy from 70.8% to 84.3%, a significant improvement of 13.5%. However, adding the detected boxes slightly decreased performance by 0.6%, which is counterproductive. We speculate that this phenomenon may be caused by two reasons. On one hand, the current 3D detection models are still immature, and the noise in the detected boxes hurts the QA model. On the other hand, the point cloud represented by XYZ itself has great position expression ability, and the gain from adding box features is not significant.

Crop	Question Types					All
	Exi.	Cou.	Obj.	Sta.	Com.	
Cir. Box	84.0	21.8	52.3	60.2	65.2	58.8
Rot. Box	84.8	20.8	52.3	59.8	70.0	59.5

Table 3: Ablation comparison of BEV feature crop strategies.

Pooling	Question Types					All
	Exi.	Cou.	Obj.	Sta.	Com.	
Max	84.2	20.7	51.6	58.0	69.7	58.9
Mean	84.8	20.8	52.3	59.8	70.0	59.5

Table 4: Ablation comparison of BEV feature pooling strategies.

BEV Feature Crop Strategy As mentioned earlier, due to the non-parallelism between the 3D boxes and the BEV coordinate axes, we cannot perform standard RoI pooling as in traditional 2D images. Therefore, we use cross product algorithm to determine pixels inside the rotated box for feature cropping. In addition to this method, we can also use a simpler approach, which directly uses the circumscribed rectangle of the rotated box parallel to the coordinate axes as the cropping region. Table 3 shows the performance comparison of these two crop strategy, where the circumscribed box is slightly inferior to the rotated box. The reason for this is that NuScenes-QA contains many elongated objects, such as bus and truck. These objects occupy a small area in the BEV space, but their circumscribed rectangles have a large range, making the object features over smoothing.

BEV Feature Pooling Strategy In terms of the feature pooling strategy for the cropped regions, we compared the classic Max Pooling and Mean Pooling operations. As illustrated in Table 4, Max Pooling achieved an accuracy of 58.9% under the same conditions, which is 0.6% lower than Mean Pooling. We speculate that this difference may be due to the fact that Max Pooling focuses on the texture features within the region, while Mean Pooling preserves the overall features. Our proposed NuScenes-QA mainly tests the model's ability of understanding the structure of objects and their spatial relationships in street views, and relatively ignores the texture of the objects. Thus, Mean Pooling has a slight advantage over Max Pooling.

Conclusion

In this paper, we apply VQA to the context of autonomous driving. We construct NuScenes-QA, the first large-scale multi-modal VQA benchmark for autonomous driving scenario. NuScenes-QA are generated automatically based on visual scene graphs and question templates, containing 34K scenes and 460K question-answer pairs. Alongside a series of baseline models, comprehensive experiments establish a solid foundation for future research. We strongly hope that NuScenes-QA can invigorate the evolution of multi-modal VQA and propel advancements in autonomous driving.

Acknowledgments

This work was supported in part by National Natural Science Foundation of China Project (No. 62072116) and Shanghai Science and Technology Program [Project No. 21JC1400600].

References

Anderson, P.; He, X.; Buehler, C.; Teney, D.; Johnson, M.; Gould, S.; and Zhang, L. 2018. Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 6077–6086.

Antol, S.; Agrawal, A.; Lu, J.; Mitchell, M.; Batra, D.; Zitnick, C. L.; and Parikh, D. 2015. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, 2425–2433.

Azuma, D.; Miyanishi, T.; Kurita, S.; and Kawanabe, M. 2022. ScanQA: 3D question answering for spatial scene understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 19129–19139.

Caesar, H.; Bankiti, V.; Lang, A. H.; Vora, S.; Liong, V. E.; Xu, Q.; Krishnan, A.; Pan, Y.; Baldan, G.; and Beijbom, O. 2020. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 11621–11631.

Chen, Y.; Liu, J.; Zhang, X.; Qi, X.; and Jia, J. 2023. VoxelNeXt: Fully Sparse VoxelNet for 3D Object Detection and Tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Dai, A.; Chang, A. X.; Savva, M.; Halber, M.; Funkhouser, T.; and Nießner, M. 2017. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 5828–5839.

Deruyttere, T.; Vandenhende, S.; Grujicic, D.; Van Gool, L.; and Moens, M.-F. 2019. Talk2car: Taking control of your self-driving car. *arXiv preprint arXiv:1909.10838*.

Dongming, W.; Wencheng, H.; Tiancai, W.; Xingping, D.; Xiangyu, Z.; and Shen, J. 2023. Referring Multi-Object Tracking. In *CVPR*.

Geiger, A.; Lenz, P.; and Urtasun, R. 2012. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, 3354–3361. IEEE.

Goyal, Y.; Khot, T.; Summers-Stay, D.; Batra, D.; and Parikh, D. 2017. Making the V in VQA Matter: Elevating the Role of Image Understanding in Visual Question Answering. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Graham, B.; Engelcke, M.; and Van Der Maaten, L. 2018. 3d semantic segmentation with submanifold sparse convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 9224–9232.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Hochreiter, S.; and Schmidhuber, J. 1997. Long short-term memory. *Neural computation*, 9(8): 1735–1780.

Huang, J.; Huang, G.; Zhu, Z.; Ye, Y.; and Du, D. 2021. Bevdet: High-performance multi-camera 3d object detection in bird-eye-view. *arXiv preprint arXiv:2112.11790*.

Hudson, D. A.; and Manning, C. D. 2019. GQA: A New Dataset for Real-World Visual Reasoning and Compositional Question Answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Jang, Y.; Song, Y.; Yu, Y.; Kim, Y.; and Kim, G. 2017. Tgif-qa: Toward spatio-temporal reasoning in visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2758–2766.

Jiang, J.; Chen, Z.; Lin, H.; Zhao, X.; and Gao, Y. 2020. Divide and conquer: Question-guided spatio-temporal contextual attention for video question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 11101–11108.

Jiao, Y.; Jie, Z.; Chen, S.; Chen, J.; Ma, L.; and Jiang, Y.-G. 2023a. MSMDFusion: Fusing LiDAR and Camera at Multiple Scales with Multi-Depth Seeds for 3D Object Detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Jiao, Y.; Jie, Z.; Chen, S.; Cheng, L.; Chen, J.; Ma, L.; and Jiang, Y.-G. 2023b. Instance-aware Multi-Camera 3D Object Detection with Structural Priors Mining and Self-Boosting Learning. *arXiv preprint arXiv:2312.08004*.

Johnson, J.; Hariharan, B.; Van Der Maaten, L.; Fei-Fei, L.; Lawrence Zitnick, C.; and Girshick, R. 2017. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2901–2910.

Johnson, J.; Krishna, R.; Stark, M.; Li, L.-J.; Shamma, D.; Bernstein, M.; and Fei-Fei, L. 2015. Image retrieval using scene graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3668–3678.

Kim, J.-H.; Jun, J.; and Zhang, B.-T. 2018. Bilinear attention networks. *Advances in neural information processing systems*, 31.

Lei, J.; Yu, L.; Bansal, M.; and Berg, T. L. 2018. Tvqa: Localized, compositional video question answering. *arXiv preprint arXiv:1809.01696*.

Lin, T.-Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; and Belongie, S. 2017. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2117–2125.

Liu, Z.; Tang, H.; Amini, A.; Yang, X.; Mao, H.; Rus, D.; and Han, S. 2023. BEVFusion: Multi-Task Multi-Sensor Fusion with Unified Bird's-Eye View Representation. In *IEEE International Conference on Robotics and Automation (ICRA)*.

Lu, J.; Yang, J.; Batra, D.; and Parikh, D. 2016. Hierarchical question-image co-attention for visual question answering. *Advances in neural information processing systems*, 29.

Pennington, J.; Socher, R.; and Manning, C. D. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 1532–1543.

Philion, J.; and Fidler, S. 2020. 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*, 194–210. Springer.

Qian, T.; Chen, J.; Chen, S.; Wu, B.; and Jiang, Y.-G. 2022a. Scene graph refinement network for visual question answering. *IEEE Transactions on Multimedia*.

Qian, T.; Cui, R.; Chen, J.; Peng, P.; Guo, X.; and Jiang, Y.-G. 2022b. Locate before Answering: Answer Guided Question Localization for Video Question Answering. *arXiv preprint arXiv:2210.02081*.

Ren, S.; He, K.; Girshick, R.; and Sun, J. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28.

Tan, H.; and Bansal, M. 2019. Lxmert: Learning cross-modality encoder representations from transformers. *arXiv preprint arXiv:1908.07490*.

Ye, S.; Chen, D.; Han, S.; and Liao, J. 2022. 3D question answering. *IEEE Transactions on Visualization and Computer Graphics*.

Yin, T.; Zhou, X.; and Krahenbuhl, P. 2021. Center-based 3d object detection and tracking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 11784–11793.

Yu, Z.; Yu, J.; Cui, Y.; Tao, D.; and Tian, Q. 2019. Deep modular co-attention networks for visual question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 6281–6290.

Zhang, P.; Li, X.; Hu, X.; Yang, J.; Zhang, L.; Wang, L.; Choi, Y.; and Gao, J. 2021. Vinvl: Revisiting visual representations in vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5579–5588.

Zhang, Y.; Hare, J.; and Prügel-Bennett, A. 2018. Learning to count objects in natural images for visual question answering. *arXiv preprint arXiv:1802.05766*.

Zhou, Y.; and Tuzel, O. 2018. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4490–4499.

Zhu, L.; Xu, Z.; Yang, Y.; and Hauptmann, A. G. 2017. Uncovering the temporal context for video question answering. *International Journal of Computer Vision*, 124: 409–421.