Integrating Vision-Language Models for Enhanced Scene Understanding in Autonomous Driving

Thesis submitted in partial fulfillment
of the requirements for the degree of

Master of Science
in
Computer Science and Engineering
by Research

by

Tushar Choudhary
2019111019
tushar.choudhary@research.iiit.ac.in

International Institute of Information Technology
Hyderabad - 500 032, INDIA
June 2024

International Institute of Information Technology
Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled **"Integrating Vision-Language Models for Enhanced Scene Understanding in Autonomous Driving"** by **Tushar Choudhary**, has been carried out under my supervision and is not submitted elsewhere for a degree.

_____ _____

Date Adviser: Prof. K. Madhava Krishna

To my parents, grandparents, and friends.

Acknowledgments

First and foremost, I extend heartfelt gratitude to my advisor, Dr. K. Madhava Krishna, for providing me with the opportunity to work at the Robotics Research Center and for his invaluable guidance on numerous challenging problems. Without his support, this work would not have been achievable.

I also wish to thank Dr. Arun K. Singh, Dr. Krishna Murthy Jatavallabhula, and Dr. Siddharth Srivastava for their guidance during my research journey. Their insights into new research directions and technical expertise have been invaluable. Our discussions on the latest trends and applications of robotics to the problems we aim to solve have been particularly enlightening.

Sincere appreciation goes to Dr. Anoop M. Namboodiri, Dr. Praveen Paruchuri, Dr. Ravi Kiran, and all the other professors at IIIT Hyderabad for their invaluable contributions through the courses they taught. Their expertise and guidance have played a pivotal role in laying the strong foundation upon which my research endeavors have been built. I am truly grateful for their dedication to nurturing academic excellence.

Furthermore, I thank my seniors Basant Sharma, Sarthak Sharma, and Shivam Chandhok for their guidance and support at different stages of my research journey. Their profound knowledge in the field and willingness to assist were instrumental in our research endeavors.

Deepest gratitude goes to my colleagues Anant Garg, Pranjal Paul, and Vikrant Dewangan, with whom I've conducted my research endeavors. Their unwavering dedication, including late-night efforts, has been instrumental in our collective achievements. I am immensely thankful for the enriching discussions, invaluable insights, and diverse perspectives within and beyond academia. Additionally, I would like to thank Aakash Aanegola, Anushka Jain, and Shubham Priyadarshan for their support at various steps.

My journey at IIIT owes much to the companionship of all my friends and peers at IIIT Hyderabad, with whom I faced both challenges and successes. Their steadfast support made my time at IIIT Hyderabad truly memorable.

Finally, I express heartfelt gratitude to my parents for their unwavering support and encouragement over the past several years, without which none of this would have been achievable.

Abstract

Autonomous driving (AD) systems require a comprehensive understanding of their surroundings to navigate safely without human intervention. This involves interpreting intricate scenes, including object interactions and anticipating future implications, which are essential for making informed decisions. However, existing AD systems often depend on task-specific models trained on limited datasets, limiting their adaptability to diverse real-world scenarios. Recent advancements in Large Language and Large Vision Language models (LLMs and LVLMs) offer a promising solution to overcome these limitations by providing general-purpose scene understanding capabilities.

This work introduces Talk2BEV, a large vision-language model interface designed for bird's-eye view (BEV) maps in autonomous driving contexts. While previous perception systems for autonomous driving have mainly focused on predefined (closed) sets of object categories and driving scenarios, Talk2BEV integrates recent advances in general-purpose language and vision models with BEV map representations. This integration eliminates the need for task-specific models, allowing a single system to handle various autonomous driving tasks, including visual and spatial reasoning, predicting traffic actors' intentions, and decision-making based on visual cues. Talk2BEV has been extensively evaluated on a wide range of scene understanding tasks that require the interpretation of free-form natural language queries and grounding these queries to the visual context embedded in the language-enhanced BEV map. Notably, this approach requires no additional training, offering flexibility and enabling rapid deployment across different domains and tasks.

Additionally, this work extends to a lightweight Vision Language Network (VLN) aimed at addressing the challenge of estimating a goal point location based on a given language command as an intermediate representation. A generalized open-set LLM or a human driver can understand an autonomous driving scenario and suggest an appropriate action, which can then be consumed by the VLN to predict an optimized associated goal point, subsequently used by downstream planners. This extension enhances explainability and efficiency in autonomous driving tasks as we have the action and goal-point as an intermediate input/output.

These contributions aim to advance the development of generalizable perception systems for autonomous vehicles by emphasizing the integration of language understanding with visual reasoning capabilities.

Contents

Chapter Page

1 Introduction . . . 1

2 Related Work . . . 5
2.1 Large Vision Language Models . . . 5
2.2 3D Vision-Language Models . . . 5
2.3 Vision-Language Models for Autonomous Driving . . . 6
2.4 Visual Grounding in Scene Understanding and Navigation . . . 6
2.5 Concurrent Work . . . 6

3 Language-Enhanced Bird's Eye View Maps . . . 8
3.1 Introductory Overview . . . 8
3.2 Language-Enhanced Maps . . . 9
3.3 Interfacing with LLMs . . . 12
3.4 Implementation Details . . . 14
3.5 Dataset and Benchmarking . . . 14
3.5.1 Ground-truth language-enhanced maps . . . 16
3.5.2 Question Generation and Evaluation Metrics . . . 16
3.6 Results . . . 17
3.6.1 Quantitative Results . . . 18
3.6.2 Qualitative Results . . . 21
3.7 Chapter Summary . . . 22

4 Grounding Action Commands to Goal Points . . . 23
4.1 Introduction & Overview . . . 23
4.2 Goal Point Prediction . . . 24
4.3 Additional Predictions . . . 25
4.4 Complex Commands and Scene Understanding . . . 26
4.5 Transforming to BEV for Downstream Tasks . . . 26
4.6 Results . . . 27
4.7 Chapter Summary . . . 29

5 Conclusion . . . 31

Bibliography . . . 34

List of Figures

Figure Page

1.1 **Absence of Intricate Details in BEV Maps**: The figure illustrates the information loss in Bird's Eye View (BEV) maps compared to camera images from the vehicle. BEV maps, while providing accurate spatial information about the layout of objects relative to the ego-vehicle, are highly abstracted and lack intricate perspective details. As seen in the figure, the semantic class only conveys the presence of a vehicle on the map; however, the rich intricate details of different objects are lost. 2

1.2 **Talk2BEV** constructs *language-enhanced bird's-eye view (BEV) maps* using (a) BEV representations constructed from vehicle sensors (Multi-View Images, LiDAR), and (b) Aligned vision-language features for each object, which can be directly used as context within large vision-language models (LVLMs) to query and *interact* with the objects in the scene. These maps embed knowledge about object semantics, material properties, affordances, and spatial concepts and can be queried for visual reasoning, spatial understanding, and making decisions about potential future scenarios, crucial for autonomous driving applications. Furthermore, we introduce the first benchmark *Talk2BEV-Bench* for evaluating LVLMs in AD applications, covering a diverse set of question categories with human-annotated ground truth. 3

3.1 **Overall Talk2BEV Pipeline**: We first predict the bird's-eye view (BEV) map from multi-view images using a standard off-the-shelf model. Then, we construct the language-enhanced map by augmenting the predicted BEV with aligned image-language features for each object from large vision-language models (LVLMs). To achieve this, for each object in the BEV, we project it onto the image (using LiDAR-camera extrinsics), extract a bounding box, and caption the cropped bounding box using an off-the-shelf LVLM. Each object in the language-enhanced map now encodes geometric cues (position, area, centroid), and semantic cues (object and image descriptions). These joint features can then be used as context for the LLMs to answer object-level and scene-level queries. . 9

3.2 **Sample serialized JSON structure** to display the information extracted from the BEV and perspective images for a particular object. Each object is associated with one JSON, and the entire scene is a list of such JSONs. As discussed earlier, for each object, we extract the position, area, and crop descriptions, which are presented in more detail in Fig 3.3. 11

3.3 **Sample instance of crop and background descriptions** associated with an object. The crop description includes a brief description of the object, status of vehicle indicators, the vehicle direction/view in the image, any visible text using OCR, and a general background description with the weather and any anomalies. 11

3.4 ***Talk2BEV-Bench* Creation**: To develop this benchmark, we use the NuScenes Ground Truth BEV annotations and generate object and scene-level descriptions using dense Captioners (GRiT [1]), and Text-Recognition (PaddleOCR [2]) models. The Ground Truth BEV is then passed to an LLM like GPT4 to generate diverse questions including, but not limited to- Spatial Reasoning, Instance Attribute, Visual Reasoning and Instance Counting. 12

3.5 **LLM System Prompts: (a)** Generic question generation prompt for the LLM [3]. **(b)** System prompt for response generation. **(c)** Details the type-specific commands added to generate questions along each evaluation dimension. **(d)** Displays the response format JSON with a brief explanation provided to the LLM on how to fill each key of the JSON. 13

3.6 **Spatial Operators**: To compute the distance between the bulldozer and the white truck, the Language Enhanced Maps for the objects are interpreted by an LLM like GPT4 to invoke relevant spatial operators in our framework with appropriate object IDs as arguments. 13

3.7 **Talk2BEV in free-form conversation.** We illustrate a free-form query q_{ff} and a sequential conversation with our Talk2BEV framework. There is a car in front of the ego-vehicle (highlighted in red) that is reversing into a parking spot. Talk2BEV identifies that the parking lights are on and, based on this visual information and the spatial location of the car in front, deems it unsafe to continue moving forward. 18

3.8 **Composition of Spatial Operators**: To find the nearest 2 vehicles in front, LLM like GPT-4 composes the spatial operators. 20

3.9 **Qualitative Results:** A BEV corresponding to a scene with multiple vehicles at an interchange. Talk2BEV is able to identify emergency vehicles (such as the *police car* shown here). The captions for a police car and a construction vehicle from Language-Enhanced maps constructed with different LVLMs (BLIP-2, InstructBLIP-2, MiniGPT-4) have been visualized. We show the corresponding BEV captions produced by various LVLMs and their performance across 4 questions from *Talk2BEV-Bench* relevant to these 2 objects. The correct answer for each question is highlighted in green. 22

4.1 **Overall pipeline of the proposed approach**: Given the visual frame and a linguistic action command, the network predicts a segmentation map corresponding to the referenced navigable region and an associated goal point. 24

4.2 **Scene Understanding**: Pairing the goal-point prediction with a Large Vision-Language Model such as GPT-4V on the front camera image allows the system to operate in a self-reliant mode, eliminating the need for the user to assess and select the most suitable action. The vision-language model is capable of determining the optimal course of action from a variety of potential driving maneuvers. In the illustrated example, it accurately identifies an obstruction ahead and suggests "switching to the left lane" to continue moving forward safely. 27

4.3 **Model Predictions:** The goal-point is shown in green and segmentation masks are shown in red. The model can predict reasonable outputs for different kinds of commands, identifying various lanes, other traffic actors, different objects on the roadside, and road signs. 28

4.4 **Complex command**: The closed-loop simulation illustrates a compound command divided into two atomic commands, executed sequentially. Navigation initially follows the first atomic command. After reaching the first goal, the second atomic command is executed to reach the final goal. 29

List of Tables

Table Page

3.1 **List of spatial operators**: Here, `objs` denotes the list of objects in the BEV, and o_{id} refers to the object with `object_id` equal to id. Operators that do not require `object_id` as input operate on the ego-vehicle. 15

3.2 **Overall Accuracy on MCQ Queries** (q_{mcq}). Performance of Talk2BEV with Language Enhanced Map constructed with different LVLMs (BLIP-2, InstructBLIP-2, MiniGPT-4) and BEV variants (LSS and GT) on Multiple Choice Questions (MCQs). 19

3.3 **Object Category-wise Evaluation:** Performance of Talk2BEV with Language-Enhanced Map constructed with different LVLMs (BLIP-2, InstructBLIP-2, MiniGPT-4) and BEV variants (LSS and GT) on queries q_{mcq} for different vehicle categories. 19

3.4 **Accuracy vs Weather:** Accuracy of various pairs of LVLMs and BEV variants on MCQ queries q_{mcq} across different weather conditions. 20

3.5 **Impact of Spatial Operators**: When relying directly on the LLM's abilities to reason about distances, orientations, and areas, we notice a significant performance drop (Talk2BEV w/o SO). Providing access to primitive spatial operators via API calls enables strong performance in terms of Jaccard index (higher is better) and distance error (lower is better) metrics. 21

Chapter 1

Introduction

For safe navigation without human intervention, autonomous driving (AD) systems must understand the visual world to make informed decisions. This involves not only recognizing specific object categories but also understanding their interactions with the environment, both currently and in the future. A crucial aspect of autonomous driving is comprehensive scene understanding to interpret information for reasoning and decision-making. Many critical tasks in autonomous driving require detailed scene comprehension, especially to handle unforeseen scenarios. This includes understanding local details like objects' properties and semantics, as well as global aspects such as their spatial orientation and location relative to markers like lanes and traffic lights. Therefore, perception systems need general-purpose representations that work across a wide range of scenarios. It is essential for AD systems to effectively learn and integrate object- and scene-level information for visual reasoning, spatial understanding, and decision-making.

In autonomous driving, Bird's Eye View (BEV) maps are very popular. Figure 1.1 represents a conventional BEV map in autonomous driving. It is a segmentation image of the top view of the region near the ego-vehicle. In a standard BEV map, the ego-vehicle is located at the center and includes layout information about objects from different semantic classes such as vehicles, pedestrians, and drivable regions. BEV maps are commonly used because they can be predicted directly using multi-view images from the ego-vehicle and provide accurate spatial information about the layout. The precision of BEV maps is well-defined; for example, in NuScenes [4], the resolution is 0.5 meters per pixel, allowing rough determination of object locations relative to the ego-vehicle. This information can be used for navigation and planning.

However, BEV maps are highly abstracted and lose intricate details present in perspective images, preserving information only for semantic classes. Given just the BEV map, you cannot discern details about the actual vehicle apart from its class, since the perspective information is lost, making it impossible to differentiate or further describe vehicle classes. In a way, there is a decoupling where spatial information is well-present in BEV maps, but intricate perspective details are missing. These details are present in front camera images, but they do not capture spatial information that well.

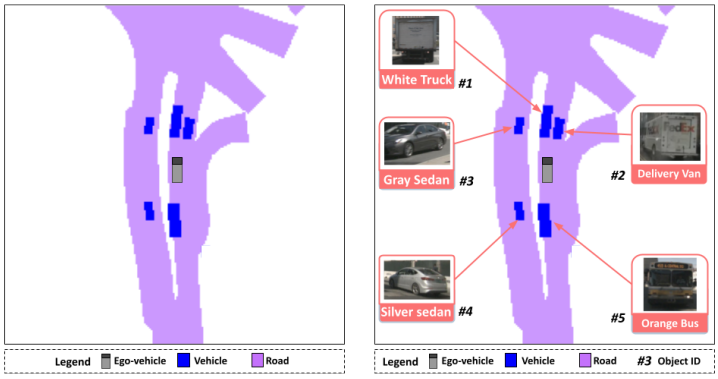

Figure 1.1 Absence of Intricate Details in BEV Maps: The figure illustrates the information loss in Bird's Eye View (BEV) maps compared to camera images from the vehicle. BEV maps, while providing accurate spatial information about the layout of objects relative to the ego-vehicle, are highly abstracted and lack intricate perspective details. As seen in the figure, the semantic class only conveys the presence of a vehicle on the map; however, the rich intricate details of different objects are lost.

We aim to create a representation that enables more holistic scene understanding. This will allow us to enhance conventional BEV maps by incorporating perspective information. By consolidating all relevant information from both the perspective and spatial domains in one place, we can tackle visual question-answering (VQA) tasks that require information from both camera data and BEV maps (as shown in Fig. 1.2). To address this problem, we plan to leverage recently developed vision-language models that have shown impressive capabilities.

Recent advances in large language models (LLMs) [5, 6, 7, 8, 9] and large vision-language models (LVLMs) [3, 10, 11, 12] offer a promising alternative for perception in AD. These models, pretrained on web-scale data, can perform all the aforementioned tasks and more, particularly by handling unforeseen scenarios. They demonstrate the potential to build general-purpose image understanding systems capable of interpreting and reasoning about both object- and scene-level information. Fine-tuned on human instructions, these models possess common sense reasoning and can follow natural language instructions. They can interpret human intent, reason based on visual cues, and make sensible decisions. Additionally, these models are vision-language aligned, enabling zero-shot capabilities and knowledge transfer to novel objects, tasks, and diverse unseen scenarios encountered in the real world.

Hence, one core motivation of the work presented is how to most efficiently integrate the capabilities of LLMs with the scene representations traditionally used in autonomous driving, making the perception stack more generalizable and introducing reasoning.

To this end, I worked on a pipeline called Talk2BEV, which leverages LLMs and LVLMs to generate language-enhanced maps for AD that enable holistic scene understanding and reasoning across a broad range of road scenarios. Our framework interfaces LVLMs with bird's-eye view (BEV) maps—top-down semantic maps of the road plane and traffic actors that are widely used in AD systems [13, 14, 15, 16]—to enable visual reasoning, spatial understanding, and decision-making. A BEV map is augmented

Figure 1.2 Talk2BEV constructs *language-enhanced bird's-eye view (BEV) maps* using (a) BEV representations constructed from vehicle sensors (Multi-View Images, LiDAR), and (b) Aligned vision-language features for each object, which can be directly used as context within large vision-language models (LVLMs) to query and *interact* with the objects in the scene. These maps embed knowledge about object semantics, material properties, affordances, and spatial concepts and can be queried for visual reasoning, spatial understanding, and making decisions about potential future scenarios, crucial for autonomous driving applications. Furthermore, we introduce the first benchmark *Talk2BEV-Bench* for evaluating LVLMs in AD applications, covering a diverse set of question categories with human-annotated ground truth.

with aligned image-language features for each object in the scene. These features can then be directly passed as (visual) context to an LLM, enabling the model to answer a wide range of questions about the scene and make decisions about potential future scenarios using the vast knowledge base acquired by the LVLM during training. It is observed that these LVLMs can interpret object semantics, material properties, affordances, and spatial concepts, making them an ideal alternative to domain-specific models and allowing us to present a unified framework for AD instead of using separate closed-set components.

Notably, this approach does not require any BEV-specific or vision-language training/finetuning; it uses existing pretrained LLMs and LVLMs. This allows our approach to be flexibly and rapidly deployed across a wide range of domains and tasks, and to easily adapt to newer LLMs and LVLMs as better models become available. This also prevents our approach from being task-specific or dataset-dependent and enables generalization to diverse scenarios encountered in the real world. To objectively evaluate LVLMs for perception in the AD context and to expedite further research, I also developed Talk2BEV-Bench: a benchmark for the evaluation of large vision and language models for autonomous driving systems on a range of tasks, encompassing object-level and scene-level visual understanding.

However, for tasks such as goal point prediction or trajectory prediction, LLMs and LVLMs may lead to infeasible outputs due to the inability to perform optimized mathematics. Instead, a simpler approach would be to gain an intermediate representation that is interpretable. Following this line of thought, in further extensions, I noticed from Talk2BEV that LLMs are able to reason out and suggest action commands. Afterwards, the aim was to explore a goal-oriented approach where LLM-suggested high-level language commands can be mapped to a desired goal. The advantages of such an approach, similar

to those discussed in [17, 18, 19], are that goal-directed planning improves explainability in autonomous driving. Predicting just the goal position allows the use of smaller and lightweight networks, requiring less data for training and resulting in faster inference times.

Hence, I also worked on a lightweight Vision Language Network (VLN) that takes in an instruction command along with the front camera image and predicts the associated goal point on the image along with some additional predictions. The visual grounding of the goal point is achieved by utilizing a transformer-based model inspired by [20] that accepts the embedding of multimodal input obtained from a pretrained LLM and outputs a language-aligned region along with a goal state. The VLN can also be extended to make additional predictions, such as identifying the need to come to a rest state and any references based on the language command, which can be useful to any downstream planner. To enhance the reasoning capability of our model, I experimented with its integration with a VQA model, which automates the navigation instructions as an extended application.

In summary, the work discussed in the thesis is as follows:

- The Talk2BEV pipeline, which augments BEV maps with language to enable general-purpose visuolinguistic reasoning for AD scenarios.
- This framework does not require any training or fine-tuning, relying instead on pretrained image-language models. This allows for generalization to a diverse collection of models, scenarios, and tasks. It seamlessly works in tandem with modern LLMs to enable scene understanding and decision-making for potential future situations critical for autonomous driving.
- The Talk2BEV-Bench, a benchmark for evaluating LVLMs for AD applications with human-annotated ground truth for object attributes, semantics, visual reasoning, spatial understanding, and decision-making.
- A VLN that takes guiding action commands and predicts a relevant goal point on the image along with some additional predictions. This VLN pipeline can be preceded by an LLM to generate the suggested action and can be followed by a downstream planner to generate the trajectory to the predicted goal point.

Chapter 2

Related Work

2.1 Large Vision Language Models

The recent advancements in Large Language Models (LLMs) have spurred the development of Large Vision Language Models (LVLMs) [5, 6, 7, 8, 9]. These LVLMs are specifically tailored to tackle vision-language tasks by integrating both textual and visual information. Unlike traditional language models that primarily process text, LVLMs can comprehend and generate text based on accompanying images.

To gauge the effectiveness and performance of LVLMs, several evaluation frameworks have been devised. Prominent among them are Multimodal Model Evaluation (MME) [21], Multimodal Model Benchmark (MMBench) [22], LVLM-Ehub [23], and SEED-Bench [24]. These frameworks employ diverse evaluation metrics, including question answering and image captioning tasks, to assess the LVLMs' capabilities across different dimensions such as existence, color, count, and position.

Our evaluation methodology follows a similar vein to SEED-Bench [24] and MMBench [22]. We leverage GPT-4, a cutting-edge language model, to facilitate both question formulation and evaluation processes. By harnessing GPT-4's natural language processing capabilities, we aim to conduct comprehensive assessments of LVLMs, shedding light on their strengths and areas for potential enhancement.

2.2 3D Vision-Language Models

LVLMs have also started to be employed in scene understanding tasks such as object localization [25, 26, 27], scene captioning [28, 29], and 3D Visual Question Answering utilizing multi-view images [30, 31] or point clouds [32, 33]. The 3D-LLM [34] integrates LLMs into point clouds generated from multi-view images, bridging 2D models to 3D, thereby aiding in object spatial reasoning and geometry understanding. In contrast, the Point-LLM [35] is trained exclusively on point clouds, eliminating the need for images in the training process.

2.3 Vision-Language Models for Autonomous Driving

Language prompts have recently been applied to autonomous driving to improve scene understanding from a human standpoint. One of the primary tasks in autonomous driving is object segmentation referring. Approaches like CityScapes-Ref [36] and Talk2Car [37] attempt it on CityScapes [38] and NuScenes [4] respectively. ReferKITTI [39] accumulates temporal information and performs object referral along with Multi-Object Tracking (MOT) on the KITTI dataset. NuPrompt [40] extends it to 3D by generating language prompts referring to 3D bounding boxes in the scene. They use RoBERTa [41] as their language encoder and perform end-to-end training. Unlike ReferKITTI and NuPrompt, we do not use temporal information and use inputs from the current time step only. Our work offers substantial improvements over this by blending state-of-the-art LLMs and LVLMs with BEV maps, while requiring no training or fine-tuning.

2.4 Visual Grounding in Scene Understanding and Navigation

Visual grounding aims to associate a natural language query with the most relevant visual elements or objects in a visual scene. Visual grounding tasks were previously approached as referring expression comprehension (REC), which involved localizing a bounding box based on the natural language expression. Traditional REC approaches typically involve two phases [42, 43, 44, 45]. Initially, they identify candidate regions within the input image using pre-trained object detectors [46, 47]. Subsequently, they select the most suitable candidate region based on the provided referring expression. Conversely, one-stage methods [48, 49, 50, 51, 52] integrate linguistic and visual features within a network and directly predict the target box [50, 48, 49]. However, bounding box prediction is imprecise in capturing the shape of the region and is insufficient for navigation tasks. Another approach is to localize the objects by their pixel-level segmentation mask, formally known as Referring Image Segmentation (RIS). RIS methods also use different strategies to fuse information from different modalities [53, 54, 55, 56]. [20] uses an RIS approach for the task of identifying navigable regions on the drivable areas based on a language command. However, the work is limited to scene understanding and does not include navigation simulations, as trajectory planning relies on precise goal-point location, which they do not address.

2.5 Concurrent Work

Efforts such as NuScenes-QA [57] address Visual Question Answering (VQA) in autonomous driving by crafting scene graphs and question templates. Their evaluation demands end-to-end training and exact answer matching. Other efforts have focused on training end-to-end vision-language-action models [58] on large amounts of aligned multimodal data.

Additionally, existing works [59, 60, 61, 62, 63, 64, 65] utilize LLMs to reason about driving scenes and predict control inputs. The majority of such works attempt to directly estimate the control actions.

However, these approaches are susceptible to errors due to their heavy reliance on VLMs' capability to give optimized control outputs. This reliance may lead to non-smooth motion for intricate cases such as parking or highway merging, which require complex combinations of control actions.

In contrast to concurrent work, the Talk2BEV pipeline represents a significant advancement by providing zero-shot scene comprehension through the utilization of Large Vision and Language Models (LVLMs) designed for autonomous driving. It also introduces a comprehensive benchmark, named Talk2BEV-Bench, aimed at evaluating LVLMs' effectiveness in scene understanding in autonomous driving scenarios. Additionally, unlike existing visual grounding approaches, we extend our framework to include goal-point prediction for autonomous driving along with the preexisting segmentation mask prediction using the RIS technique. We demonstrate that the predicted goal-point is reasonable and can be used for planning. Moreover, the model is well-suited for autonomous driving since it provides additional contextual states such as final velocity and identifying object references based on the textual command, which might be helpful for downstream tasks.

Chapter 3

Language-Enhanced Bird's Eye View Maps

The conventional Bird's Eye View (BEV) maps offer a comprehensive top-down perspective, displaying the positions of both dynamic and static objects within a scene. They provide a spatial understanding of objects across different semantic classes to efficiently convey the required semantic information crucial for navigation and planning. However, they lack the intricate details and contextual information present in perspective images. Unlike perspective images, which offer specific viewpoints of a scene, layout maps present a simplified, abstract representation, often at the expense of fine-grained visual cues. These details are indispensable for tasks requiring precise comprehension and interpretation of the environment, such as scene understanding and interaction with the environment.

I worked on a pipeline that we named Talk2BEV, which acts as an end-to-end framework enabling communication with a Bird's Eye View (BEV) map for visual reasoning about objects and spatial aspects. Our framework facilitates conversation with entities within the BEV to enhance comprehension of their attributes and features, fostering a holistic understanding of the map. Additionally, our spatial module allows for the calculation and solving of feasible tasks based on the current map. Fig. 3.1 provides an overview of our pipeline.

3.1 Introductory Overview

The key idea of Talk2BEV is to enhance the conventional bird's-eye view (BEV) map with general-purpose vision-language features derived from pretrained LVLMs. A BEV map, denoted $\mathcal{O}$, is a top-view multi-channel grid encoding semantic information of various relevant classes such as *vehicles*, *pedestrians*, *lanes*, and *lane markings*. In this work, we use only the *vehicle* and *pedestrian* classes to extract LVLM features, and the *road* classes are shown only for visualization purposes. In a standard BEV frame, the ego-vehicle is represented at the origin, which is assumed to be the center of the BEV image. Given multi-view RGB images $\mathcal{I}$ and a LiDAR point cloud $\mathcal{X}$, a BEV can be obtained using a number of off-the-shelf approaches such as [13, 16, 14, 66, 67].

We then further enhance this conventional BEV map with LVLM features, which can serve as context for responding to various user queries q. The BEV representations $\mathcal{O}$ are enriched with scene and object

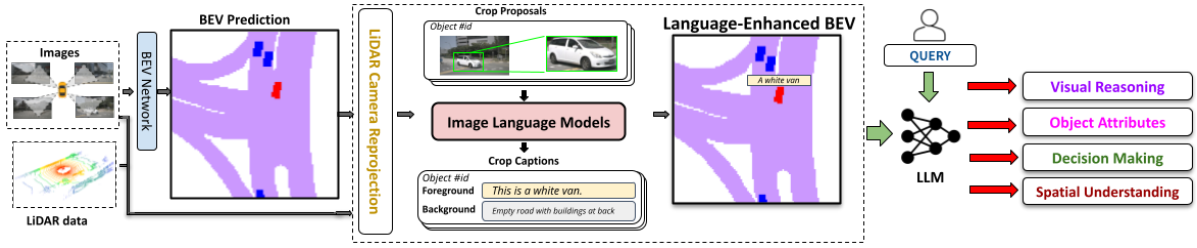

Figure 3.1 Overall Talk2BEV Pipeline: We first predict the bird's-eye view (BEV) map from multi-view images using a standard off-the-shelf model. Then, we construct the language-enhanced map by augmenting the predicted BEV with aligned image-language features for each object from large vision-language models (LVLMs). To achieve this, for each object in the BEV, we project it onto the image (using LiDAR-camera extrinsics), extract a bounding box, and caption the cropped bounding box using an off-the-shelf LVLM. Each object in the language-enhanced map now encodes geometric cues (position, area, centroid), and semantic cues (object and image descriptions). These joint features can then be used as context for the LLMs to answer object-level and scene-level queries.

descriptions to create a Language-Enhanced map $\mathbf{L}(\mathcal{O})$. We utilize powerful LVLMs such as BLIP-2 [10], MiniGPT-4 [11], and InstructBLIP-2 [12] to generate free-form captions and descriptions for different object crops and complete perspective images in a zero-shot manner. To achieve this, our three-phase pipeline (refer to Fig. 3.1) operates as follows:

1. We first predict a BEV map using a standard off-the-shelf model [13], which takes as input multi-view images captured by vehicle sensors.

2. For each object o_i in this BEV map, we generate aligned image-language features using an LVLM [10, 12, 11]. These features are then passed into the language model of an LVLM to extract object metadata. The object data, along with geometric information present in the BEV, forms the language-enhanced map, $\mathbf{L}(\mathcal{O})$.

3. Finally, given a user query, we prompt an LLM (e.g., GPT-4 [3]) to interpret the query, parse the language-enhanced BEV as needed, and produce a response to the query.

These steps will now be discussed in detail in the later subsections.

3.2 Language-Enhanced Maps

Generating BEV: To begin, we have N multi-view images $\mathcal{I} = \{\mathbf{I}_n\}$ of the scene, where $n \in \{1, \ldots, N\}$, and the corresponding LiDAR scan $\mathcal{X}$. We use these images and LiDAR scan as input and first predict a BEV representation using the LSS (Lift-Splat-Shoot) method [13]. LSS takes N multi-view images of a scene as input and generates corresponding BEV segmentation map $\mathcal{O}$.

BEV-Image Correspondence: Next, we localize each object in the predicted BEV within the multi-view images used to generate the BEV map. Given a LiDAR scan (a point cloud) $\mathcal{X}$ and the positions of the objects in the Bird's Eye View, for each object, we select the K points $x_k \in \mathcal{X}$, where k ranges from 1 to K, that are closest to the object's segmentation mask center. These points are then transformed from the LiDAR frame to the camera frame and projected onto the camera image with the best overlap to obtain the correspondence between the object patch in the BEV map and the image region.

Language Enhancement: We then use a point-queryable segmentation model, such as FastSAM [68], with the point prompt as the mean of the projected image points to generate image masks specific to the BEV segmentation patches. For each object mask, we crop a tightly fitting bounding box b_i around it. Both the cropped image and original image are then passed onto the LVLM to generate object descriptions. At this stage, we only pass these images through the visual encoders to obtain image-language features that may later be passed as context tokens into language decoders. The descriptions for each object encompass both object-level and scene-level details as explained later. These generated metadata are then augmented with the BEV map in the form of a text entry (see a sample JSON-structured entry in Fig. 3.3).

Object Descriptions: Our language-enhanced map augments additional object details in a BEV by computing the image regions corresponding to each object and deriving textual descriptions to combine with the pre-existing spatial details in the conventional BEV. For each object i, we then have (a) displacement along the BEV X and Y axes (in m) from the ego-vehicle, (b) object area (in m^2), (c) a text description of the object, and (d) a text description of the background. LVLMs are specifically prompted to generate detailed descriptions of objects, and their outputs typically encode the type, color, and utility of the vehicle, status of the vehicle indicators, any text displayed on the vehicle, etc., along with a detailed summary of the entire perspective image, which includes details on weather, traffic conditions, any anomalies, etc.

To obtain rich captions for our images, we utilize state-of-the-art Large Vision Language Models (LVLMs) such as BLIP-2 [10], MiniGPT-4 [11], InstructBLIP-2 [12], etc. These models feature a frozen visual encoder (ViT/Q-former), a language backbone (Vicuna), and a linear projection layer, typically used for fine-tuning. These LVLMs demonstrate excellent zero-shot generalization capabilities across various vision-language-based tasks, including complex reasoning and conversation. Given a region crop r_i, these networks provide crop-level details c_i as described earlier in a zero-shot manner. We later demonstrate in detail that for our task, all LVLMs exhibit similar performance, and any LVLM of choice can be used.

The obtained language-enhanced BEV maps contain both spatial and perspective information, facilitating free-form question answering, visual and spatial reasoning, and the execution of potential tasks using spatial operators. Additionally, this information is encoded in a serialized JSON data format that a Large-Language-Model (LLM) like GPT4 can parse. When presented with a user query, the LLM can use this data for general-purpose reasoning and decision-making regarding present and potential future situations across various autonomous driving scenarios.

Figure 3.2 Sample serialized JSON structure to display the information extracted from the BEV and perspective images for a particular object. Each object is associated with one JSON, and the entire scene is a list of such JSONs. As discussed earlier, for each object, we extract the position, area, and crop descriptions, which are presented in more detail in Fig 3.3.

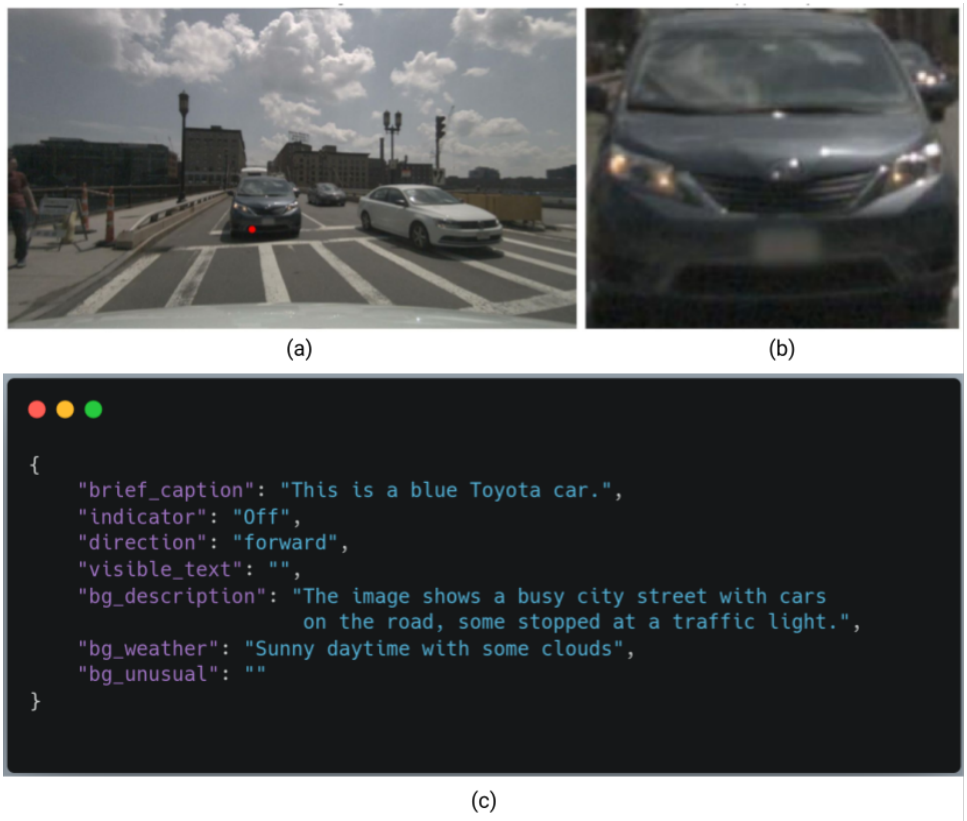

Figure 3.3 Sample instance of crop and background descriptions associated with an object. The crop description includes a brief description of the object, status of vehicle indicators, the vehicle direction/view in the image, any visible text using OCR, and a general background description with the weather and any anomalies.

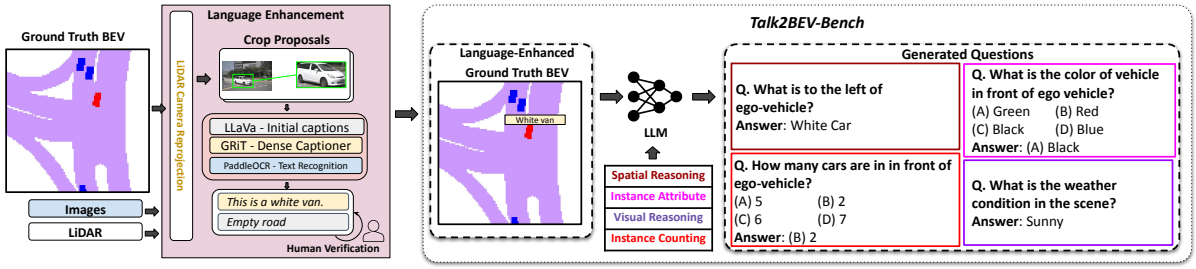

Figure 3.4 *Talk2BEV-Bench* Creation: To develop this benchmark, we use the NuScenes Ground Truth BEV annotations and generate object and scene-level descriptions using dense Captioners (GRiT [1]), and Text-Recognition (PaddleOCR [2]) models. The Ground Truth BEV is then passed to an LLM like GPT4 to generate diverse questions including, but not limited to- Spatial Reasoning, Instance Attribute, Visual Reasoning and Instance Counting.

3.3 Interfacing with LLMs

Type of queries: The *Talk2BEV* system can handle multiple types of user queries. In this study, we categorize these queries into three distinct categories: free-form text queries, multiple-choice questions (MCQs) with a single correct answer, and spatial reasoning queries expressed through textual descriptions. Free-form text queries and spatial reasoning queries simulate the natural conversational interface intended for interaction with Talk2BEV, reflecting how users might naturally communicate with the system. Conversely, MCQs provide a structured format for assessment, enabling us to conduct objective evaluations of the system's performance. This evaluation methodology aligns with the standardized approach outlined in SEEDBench [24], ensuring consistency and comparability across evaluations.

Response format: Instead of directly generating free-form text outputs, we instruct the LLM used in Talk2BEV to produce a JSON-formatted output with four fields: (i) `inferred_query`, which rephrases the user query, providing the LLM's interpretation; (ii) `query_achievable`, indicating whether the query is achievable; (iii) `spatial_reasoning_functions`, denoting whether spatial reasoning functions are needed; and (iv) `explanation`, containing a brief explanation of how the LLM addressed the provided task. Figure 3.5 specifies the system prompts provided to the LLM, with GPT-4 as the specific model used. Employing this format presents dual advantages: firstly, it ensures that the LLM delivers information in an organized manner, structured into key-value pairs. Secondly, it facilitates chain-of-thought reasoning [69] by outlining the intermediate steps that lead to the final response, enhancing interpretability and user confidence in the system's outputs.

Spatial Operators: To facilitate spatial queries, we have implemented a set of primitive spatial operators [70] capable of performing simple spatial calculations on entities of the language-enhanced map $\mathbf{L}(\mathcal{O})$. These modules primarily take as input the object_id of the referenced objects, and optionally the distance (m) as input. The list of spatial operators is detailed in Table 3.1. Based on their return type, they fall into two main categories: (i) returning a list (of object IDs), and (ii) returning a distance. To enable the LLM to accurately perform spatial reasoning, we provide access to an API of primitive

f"""You will be given, as input a 2D road scene in Bird's Eye View, as a list. The ego-vehicle is at (0,0) facing along the positive X-axis. Each entry in the list describes one object in the scene. Please ask the user to input JSON. Once you have parsed the JSON and are ready to generate questions about the scene. Create a multi-choice questions about the image, and provide the choices and answer. Each question should have 4 options, out of which only 1 should be correct.
{type-specific-prompt}
"""

(a) Question Generation System Prompt (Generic)

f"""The input to the model is a 2D road scene in Bird's Eye View, described in a JSON format. The ego-vehicle is at (0,0) facing along the positive Y-axis. The "scene" key will have a list. Each entry in the list describes one object in the scene. Ask the user to user to prompt JSON. Once you have parsed the JSON and are ready to answer questions about the scene, say "I'm ready". The user will then begin to ask questions, and the task is to answer. For each user question, respond as:
{response-format}
"""

(b) Response Generation System Prompt

Evaluation Dimension	type-specific-prompt
Spatial Reasoning	*"The question should be about spatial relations between two objects. The questions should be mainly based on the coordinates of the two objects. To answer the questions, one should find the two mentioned objects, and find their relative spatial relation to answer the question."*
Instance Attribute	*"The questions should be about the attribute of a certain object, such as its color, shape or fine-grained type."*
Instance Counting	*"The questions should involve the number of appearance of a certain object. Start with "How many ....". The choices of the question should be numbers. To answer the question, one should find and count all of the mentioned objects in the image."*
Visual Reasoning	*"Create complex questions beyond describing the scene. To answer such questions, one should first understanding the visual content, then based on the background knowledge or reasoning, either explain why the things are happening that way, or provide guides and help to user's request. Make the question challenging by not including the visual content details in the question so that the user needs to reason about that first."*

(c) Type Specific Prompts

Response Format JSON (you here refers to GPT-4)	
key	***key description provided to GPT-4***
inferred_query	your interpretation of the user query in a succinct form
query_achievable	whether or not the user-specified query is achievable using the objects and descriptions provided in the scene
spatial_reasoning_functions	If the query needs calling one or more spatial reasoning functions, this field contains a list of function calls that conform to the API above. Else, this field contains an empty list.
explanation	A brief explanation of what the most relevant object is, and how it addresses the task.

(d) Response Format JSON

Figure 3.5 LLM System Prompts: **(a)** Generic question generation prompt for the LLM [3]. **(b)** System prompt for response generation. **(c)** Details the type-specific commands added to generate questions along each evaluation dimension. **(d)** Displays the response format JSON with a brief explanation provided to the LLM on how to fill each key of the JSON.

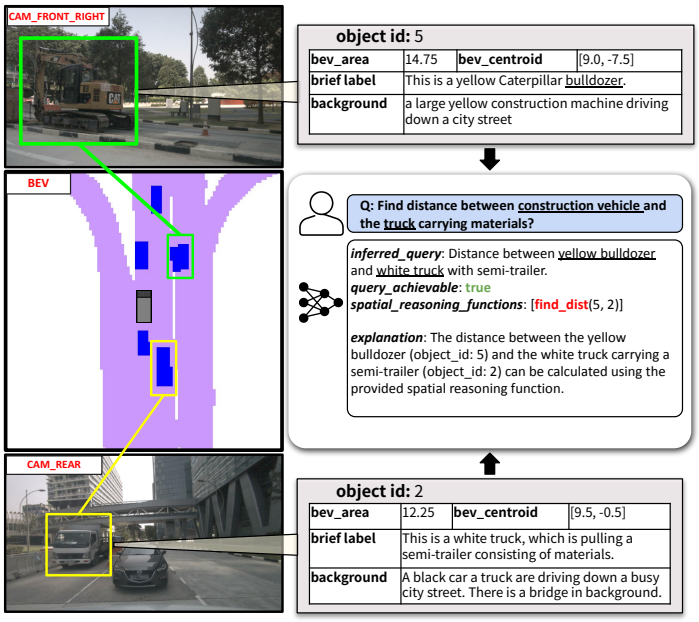

Figure 3.6 Spatial Operators: To compute the distance between the bulldozer and the white truck, the Language Enhanced Maps for the objects are interpreted by an LLM like GPT4 to invoke relevant spatial operators in our framework with appropriate object IDs as arguments.

spatial operators. The LLM's task is to parse the complex natural language spatial reasoning queries q_{sp} into a set of sequential spatial operations and invoke the relevant functions in correct order to find the answer. Whenever a user query involves spatial reasoning (locations, distances, orientations), the model is instructed to generate API calls that directly invoke relevant spatial operators, rather than attempting to calculate the output itself. We design their metrics based on their return type, as described in the later subsection. This also simplifies the evaluation and scoring of these LVLMs and establishes a standard format for it. An example usage of spatial operators is illustrated in Fig. 3.6, where we capture the distance between the construction vehicle and the truck carrying materials. Importantly, these vehicles are never co-visible in the same camera and require a BEV map for reasoning about them jointly.

3.4 Implementation Details

To generate Bird's Eye View (BEV) maps from multi-view images, we utilize the Lift-Splat-Shoot model [13]. Each BEV comprises a grid measuring 200×200, with each cell corresponding to a spatial resolution of $0.5m$. It's noteworthy that all our ground-truth BEV maps, used for evaluation purposes, maintain uniform resolution and grid dimensions.

In our experimentation, we explore the effectiveness of various Language Vision and Language Models (LVLMs) in augmenting different object captions within our Language Enhanced Maps $\mathbf{L}(\mathcal{O})$ with relevant visual features. Specifically, we consider BLIP-2 [10], MiniGPT-4 [11], and InstructBLIP-2 [12]. These visual features serve as contextual information for the language decoder of LVLM, aiding in the generation of descriptive object representations. For BLIP-2, we utilize the Flan5XXL [71] language decoder, while for InstructBLIP-2 and MiniGPT-4, we employ the Vicuna-13b language decoder [72].

Furthermore, user queries regarding the language-enhanced BEVs are addressed using GPT-4 for question answering tasks. Across all experiments, a temperature value of 0.7 is consistently used for LVLMs, while GPT-4 operates at a temperature of 0.0. All inferences are carried out using NVIDIA A100 hardware for efficient processing.

3.5 Dataset and Benchmarking

To evaluate the quality of our language-enhanced map and assess the spatial understanding and visual reasoning capabilities of our framework, we introduce Talk2BEV-Bench. This is the first benchmark designed specifically for evaluating Language Vision and Language Models (LVLMs) in the context of autonomous driving applications.

We start by generating ground-truth language-enhanced maps for 1000 scenes sourced from the NuScenes dataset [4]. Additionally, we create over 20,000 human-verified question-answer pairs as part of the SEEDBench [24]. In SEEDBench, each question comes with multiple answer choices, among which only one is correct. These questions are carefully crafted to assess various aspects such

Method	Description
`front_filter(objs)`	Objects to the front
`left_filter(objs)`	Objects to the left
`right_filter(objs)`	Objects to the right
`rear_filter(objs)`	Objects to the rear
`dist_filter(objs, X)`	Objects within **"X"** meters
`k_closest(objs, k)`	k closest objects
`k_farthest(objs, k)`	k farthest objects
`objs_in_dist(objs, id, dist)`	Objects within distance **"dist"** to o_{id}
`k_closest_to_obj(objs, id, k)`	k closest objects to o_{id}
`k_farthest_to_obj(objs, id, k)`	k farthest objects to o_{id}
`obj_distance(objs, id)`	Distance (in meters) to o_{id}
`find_dist(objs, id1, id2)`	Distance between objects o_{id1} and o_{id2}

Table 3.1 List of spatial operators: Here, `objs` denotes the list of objects in the BEV, and o_{id} refers to the object with `object_id` equal to id. Operators that do not require `object_id` as input operate on the ego-vehicle.

as understanding of object attributes, instance counting, visual reasoning, decision making, and spatial reasoning relevant to autonomous driving tasks.

It's important to note that we deliberately exclude any free-form query q_{ff} or its corresponding answer. Instead, we focus solely on structured queries and responses, restricting free-form queries to qualitative analysis.

The process of generating questions and responses involves several steps. Initially, we extract ground-truth Bird's Eye View (BEV) maps from the NuScenes dataset. For each object in these maps, we obtain captions through a refinement process carried out by human annotators. Subsequently, we utilize GPT-4, a state-of-the-art language model, to generate questions and initial responses for each question based on the extracted captions. These questions and responses undergo further validation by human annotators to ensure accuracy and relevance. The final set of Multiple Choice Questions (MCQs) produced through this iterative process forms the basis of our benchmark.

This question and answer curation approach is visually depicted in Fig. 3.4, where an example set of generated questions is presented alongside a ground-truth language-enhanced BEV map.

3.5.1 Ground-truth language-enhanced maps

We start by using the BEV maps provided in the NuScenes ground-truth dataset to identify objects of interest and obtain their corresponding image crops through LiDAR-camera projection. For each object o_i and its associated region r_i, we employ the following approach to extract the image-language description.

Crop captions: To initiate our description generation, we utilize LLaVA [73] and GRiT [1]. These tools assist in capturing intricate object details within the local crop. Additionally, we integrate the text-recognition model PaddleOCR [2] into our pipeline. This model aids in detecting and extracting text from the body of numerous vehicles, enhancing our understanding of the vehicle's type and category. This comprehensive process ultimately improves the benchmark quality.

Background information: In addition to extracting crop-level features, we also gather crucial contextual information from the complete camera image. This information, which may not be visible in the BEV segmentation and object-level crops, holds significant importance in an autonomous driving context. Examples of this additional information include street/road signs, barriers/cones, weather conditions, time of day, and any other unusual elements within the scene.

At this stage, we proceed with human verification. Human annotators play a pivotal role in combining and refining the generated foreground and background captions. The caption generation process is further elucidated in Figure 3.4.

3.5.2 Question Generation and Evaluation Metrics

In our evaluation, we focus on four types of tasks related to visual and spatial understanding:

- *Instance Attributes*: These questions inquire about objects and their attributes, such as color, size, or shape.
- *Instance Counting*: This task involves determining the number of objects described in the query, which helps assess the model's ability to comprehend quantities.
- *Visual Reasoning*: These questions test general visual understanding beyond specific attributes or counting, encompassing tasks like identifying patterns or relationships among objects.
- *Spatial Reasoning*: This category involves questions related to spatial concepts such as location, distance, or orientation, which are crucial for understanding scenes and environments accurately.

The evaluation benchmark, known as *Talk2BEV-Bench*, primarily comprises two question types:

- q_{mcq} (Multiple-Choice Questions): These questions cover all categories except spatial operators. The benchmark provides the correct answer for each multiple-choice question.

- q_{sp} (Spatial Queries): For spatial queries, the benchmark returns either a *list* of objects or a *distance* value, depending on the query. When the return type is a list, it contains objects extracted from the ground truth BEV along with relevant spatial operators. When it's a distance, it provides the precise distance of the query from a reference point.

We use different evaluation metrics based on the type of questions:

1. **Multiple-Choice Questions (q_{mcq})**:
    - **Accuracy**: We evaluate the accuracy of the model's response by comparing it with the single correct option provided by the benchmark.
2. **Spatial Reasoning (q_{sp})**:
    - **Return Type: *List***:
        - **Intersection-Over-Union (IoU)**: This metric measures the degree of overlap between the list obtained during evaluation and the ground truth list, providing insights into the model's ability to identify spatial relationships accurately.
    - **Return Type: *Distance***:
        - **Distance Error**: We utilize a linear relative error metric, ranging between 0 and 1, to quantify the difference between the distance obtained from the benchmark and the distance response generated during evaluation. This metric helps assess the precision of spatial distance estimation by the model.

Each metric yields a score between 0 and 1, indicating the model's performance on the respective task. For multiple-choice questions (q_{mcq}), the generated response is compared against the correct option from the benchmark to determine accuracy. For spatial queries (q_{sp}), the response can be either a *list of objects* or *distances*, depending on the nature of the query. When the return type is a list, we use the Intersection-Over-Union (IoU) metric to evaluate spatial relationships, whereas for distances, we employ Distance Error to assess the accuracy of distance estimation.

3.6 Results

For our Talk2BEV pipeline, we evaluate quantitatively on questions from Talk2BEV-Bench and find that:

1. Talk2BEV addresses a broad set of visual and spatial understanding tasks by leveraging language-enhanced maps.
2. Access to an API of primitive spatial operators significantly improves performance on spatial reasoning tasks.

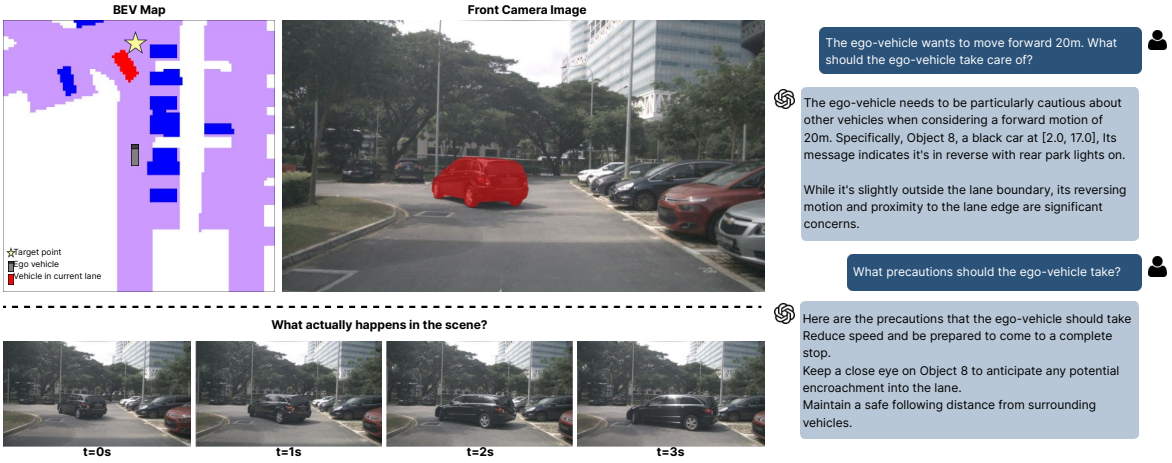

Figure 3.7 Talk2BEV in free-form conversation. We illustrate a free-form query q_{ff} and a sequential conversation with our Talk2BEV framework. There is a car in front of the ego-vehicle (highlighted in red) that is reversing into a parking spot. Talk2BEV identifies that the parking lights are on and, based on this visual information and the spatial location of the car in front, deems it unsafe to continue moving forward.

3. The zero-shot nature of Talk2BEV allows seamless switching of LVLMs, enabling easy integration across more performant LVLMs.

We also present qualitative results on challenging scenarios from NuScenes [4], indicating the ability of Talk2BEV to interpret the BEV layout at a granularity that allows predicting potential risky driving maneuvers and recourse.

We also present goal point qualitative results on the CARLA dataset. We show that the model is able to predict reasonable outputs for different kinds of commands. It can identify various lanes, other traffic actors, different objects on the roadside, and road signs.

3.6.1 Quantitative Results

We first assess the performance of Talk2BEV on questions from Talk2BEV-Bench. In Table 3.2, we report the performance across task subsets and LVLMs used. To distinguish errors originating from incorrect BEV predictions versus inaccurate LVLM captions, we also present results from an oracle approach that leverages the ground-truth BEV map.

When using BEV maps output by LSS [13], we find that InstructBLIP-2 achieves the best performance in *instance attribute* recognition and *visual reasoning* compared to BLIP-2 and MiniGPT-4. In contrast, for *instance counting*, the MiniGPT-4 based $\mathbf{L}(\mathcal{O})$ map achieves the best accuracy. Overall, we notice that MiniGPT-4 achieves the best average performance across different types of questions.

We observe that *instance attribute* and *visual reasoning* tasks are more sensitive to the quality of LVLM captions compared to other question categories, which is expected given the complexity of these tasks compared to *instance counting*.

We also note that errors in the BEV have only a minor impact on performance (3%). This is mainly because, although predicted BEVs may not capture the exact shape of different traffic actors in the BEV map, there is still significant overlap, and the predicted area and the BEV-to-image projection correctly land on the object region for most objects.

BEV	LVLM	Instance Attribute	Instance Counting	Visual Reasoning	Avg
LSS	BLIP-2	0.50	0.83	0.47	0.60
	InstructBLIP-2	**0.54**	0.80	**0.50**	0.62
	MiniGPT-4	0.50	**0.90**	0.49	**0.63**
GT	BLIP-2	0.51	0.83	0.47	0.60
	InstructBLIP-2	**0.55**	0.80	0.50	0.62
	MiniGPT-4	**0.55**	**0.91**	**0.51**	**0.66**

Table 3.2 Overall Accuracy on MCQ Queries (q_{mcq}). Performance of Talk2BEV with Language Enhanced Map constructed with different LVLMs (BLIP-2, InstructBLIP-2, MiniGPT-4) and BEV variants (LSS and GT) on Multiple Choice Questions (MCQs).

Performance across Object Categories: To assess variance in performance across object categories, we report per-category statistics in Table 3.3. We note that 2-Wheeler vehicles, including bicycles and motorcycles, consistently showed lower performance compared to other categories. This is mainly due to their smaller BEV segmentation predictions, making it more difficult to accurately back-project when there are minor inconsistencies in the predicted positions. On the contrary, larger vehicles such as trucks and construction vehicles consistently outperformed cars in most cases. This can be attributed to their larger BEV segmentations, which enable more accurate back projections.

BEV	LVLM	2-Wheeler	Cars	Trucks	Construction
LSS	BLIP-2	0.56	0.60	0.67	0.67
	InstructBLIP-2	0.52	0.58	0.73	0.61
	MiniGPT-4	0.48	0.59	0.67	0.72
	Average	0.52	0.59	0.69	0.67
GT	BLIP-2	0.56	0.60	0.68	0.67
	InstructBLIP-2	0.56	0.58	0.74	0.67
	MiniGPT-4	0.56	0.66	0.72	0.72
	Average	0.56	0.61	0.71	0.68

Table 3.3 Object Category-wise Evaluation: Performance of Talk2BEV with Language-Enhanced Map constructed with different LVLMs (BLIP-2, InstructBLIP-2, MiniGPT-4) and BEV variants (LSS and GT) on queries q_{mcq} for different vehicle categories.

Impact of weather conditions: The NuScenes dataset mainly comprises sunny daytime frames, but our evaluation also considered rainy and cloudy conditions. As shown in Table 3.4, the performance drops significantly with BLIP-2 under rainy conditions compared to sunny conditions. However, there is only

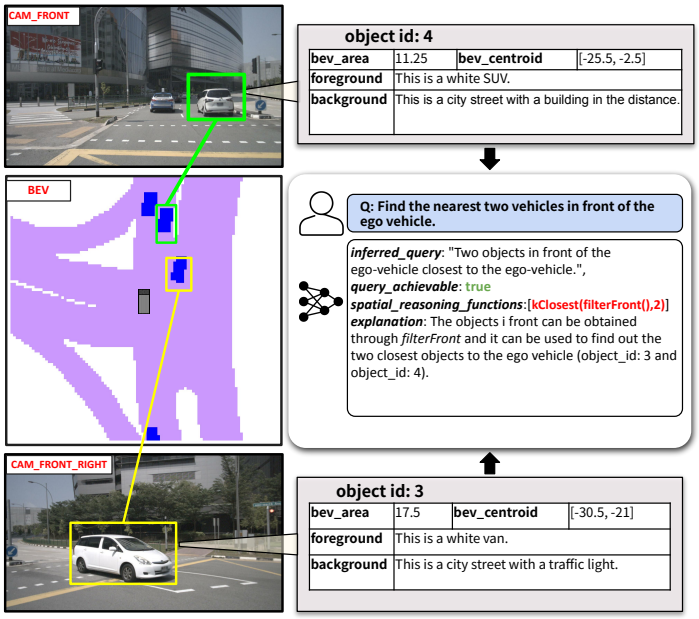

Figure 3.8 Composition of Spatial Operators: To find the nearest 2 vehicles in front, LLM like GPT-4 composes the spatial operators.

a slight decline in the performance of InstructBLIP-2 and MiniGPT-4 under rainy conditions compared to sunny weather. The decline in performance can be attributed to SAM's challenges in object boundary segmentation from reduced visibility and LVLMs' struggles with unclear object crops. These issues compromise the accuracy of the Language-Enhanced map, leading to decreased overall performance, with BLIP-2 witnessing a more significant decline compared to other models.

BEV	LVLM	Rain/Cloudy	Clear Weather
LSS	BLIP-2	0.46	0.61
	InstructBLIP-2	0.60	0.62
	MiniGPT-4	**0.60**	**0.63**
GT	BLIP-2	0.48	0.61
	InstructBLIP-2	0.61	0.62
	MiniGPT-4	**0.64**	**0.66**

Table 3.4 Accuracy vs Weather: Accuracy of various pairs of LVLMs and BEV variants on MCQ queries q_{mcq} across different weather conditions.

Spatial Operators: To assess the impact of explicit spatial operators available to our model via an API, we evaluate the performance of our system with and without spatial operators in Table 3.5. For reference, we also implement a baseline method, *Random*, which uniformly randomly guesses distances and relevant objects. Note that spatial reasoning queries are evaluated using the Jaccard index or distance error based on the nature of the query as explained in Sec. 3.5.2. While Talk2BEV without spatial op-

	Jaccard Index ↑	Distance Error ↓
Random	0.16	0.44
Talk2BEV w/o SO*	0.25	0.22
Talk2BEV with SO*	**0.83**	**0.13**

*SO: Spatial Operators

Table 3.5 Impact of Spatial Operators: When relying directly on the LLM's abilities to reason about distances, orientations, and areas, we notice a significant performance drop (Talk2BEV w/o SO). Providing access to primitive spatial operators via API calls enables strong performance in terms of Jaccard index (higher is better) and distance error (lower is better) metrics.

erators demonstrates markedly better performance compared to the *Random* baseline, the model seems to struggle with spatial reasoning queries, often encountering large errors. We see that Talk2BEV integrated with our spatial operators achieves significant performance improvements (58% improvement in Jaccard index, 0.09 m reduction in distance error) compared to directly using the LLM (here, GPT-4 [3]) for spatial reasoning. Hence, incorporating spatial operators enhances Talk2BEV's capability to tackle spatial reasoning challenges, providing the LLM with contextual depth and directing its attention to relevant components.

3.6.2 Qualitative Results

Fig. 3.7 illustrates a free-form back-and-forth conversation with the LLM. In this scene, the user intends to move forward by 20 m and thus asks the LLM about potential obstructions. In front of the ego-vehicle, a vehicle is reversing and parking into a spot. The Language-Enhanced map $\mathbf{L}$ has the information embedded that its reverse light is on, and also its position in the BEV. Based on these factors, the LLM is able to reason its intention accurately and warn the user to take caution regarding the SUV. The LLM's prediction is verified by the activity of the vehicle visualized into the future, $t = 0$ to $t = 3$s.

Fig. 4.3 displays results with captions for all LVLM baselines: BLIP-2 [10], InstructBLIP-2 [12], and MiniGPT-4 [11]. We highlight two objects from the BEV, which are significant from an autonomous driving standpoint: a police car with "Police" written on its bonnet, parked in front of the ego-vehicle, and an orange construction truck with a crane located to the rear right of the ego-vehicle. We notice that the map constructed with BLIP-2 identifies both objects as white trucks, leading to incorrect answers to *Talk2BEV-Bench* questions. In contrast, maps constructed with both MiniGPT-4 and InstructBLIP-2 identify the foreground object correctly, leading to comparatively more correct answers than the BLIP-2 variant. This indicates that the language-enhanced map encoding object attributes, especially for the object crop, is critical for overall performance. For the crane, the detail from the InstructBLIP-2-based map is more specific (i.e., 'Orange Crane with cab') than 'Large Orange Crane' from the MiniGPT-4 variant. This distinction also leads to the InstructBLIP-2 variant answering a question correctly under 'Instance Counting' about the count of white cars, while the other two models provide incorrect answers. This

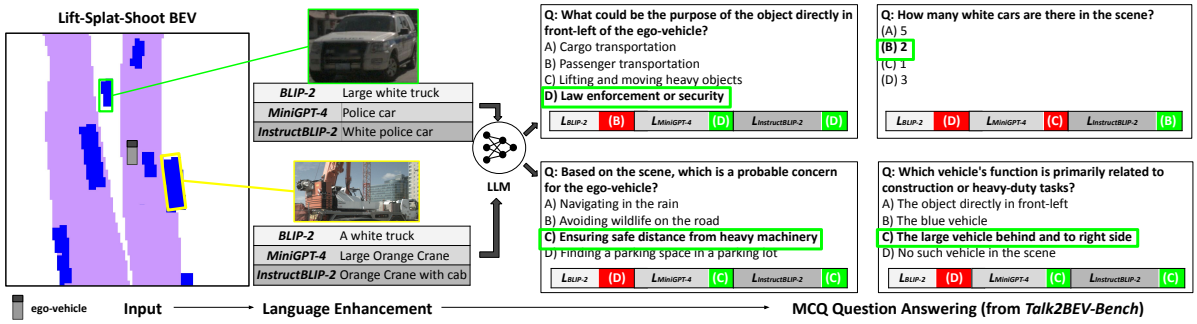

Figure 3.9 Qualitative Results: A BEV corresponding to a scene with multiple vehicles at an interchange. Talk2BEV is able to identify emergency vehicles (such as the *police car* shown here). The captions for a police car and a construction vehicle from Language-Enhanced maps constructed with different LVLMs (BLIP-2, InstructBLIP-2, MiniGPT-4) have been visualized. We show the corresponding BEV captions produced by various LVLMs and their performance across 4 questions from *Talk2BEV-Bench* relevant to these 2 objects. The correct answer for each question is highlighted in green.

demonstrates that our proposed workflow successfully leverages the aligned visual features obtained from LVLMs and that the features embed enough object-level semantics within the language-enhanced map to correctly answer questions.

3.7 Chapter Summary

In this work, I focused on improving Bird's Eye View (BEV) maps for autonomous driving by integrating perspective information. The method involved using vision language models to create Language-Enhanced BEV maps. This included associating objects in the BEV map with corresponding image crops and generating captions for foreground and background elements using language models.

To assess the effectiveness of the enhanced BEV maps, we conducted visual question answering tasks covering various query types, including free-form questions, multiple-choice questions, and spatial reasoning queries. The results showed moderate accuracy across tasks, with larger vehicles demonstrating better performance due to more precise lidar projection. Additionally, we successfully addressed spatial reasoning queries by utilizing pre-implemented functions to break down queries into smaller tasks.

Overall, the study provides a comprehensive approach to improving BEV maps in autonomous driving applications, with the aim of consolidating spatial and perspective information to enhance scene understanding capabilities.

Chapter 4

Grounding Action Commands to Goal Points

In my next work, I focused on developing a vision-language model designed to process a front camera image and an action command to produce a goal point on the image. This work extends the foundational efforts in the Talk2BEV pipeline, which primarily emphasized scene understanding. By integrating this new model with the previous pipeline, we aimed to leverage Talk2BEV's scene understanding capabilities and then use the vision-language network to convert the LLM-suggested actions into optimal goal points for downstream tasks such as planning.

4.1 Introduction & Overview

Most of the efforts and results in the Talk2BEV pipeline focused on scene understanding. Naturally, our idea was to extend this work to include navigation and planning. We experimented with stacking language-enhanced BEV JSONs of the last few frames so they could encode the relative motion of other vehicles with respect to the ego vehicle. Then we prompted the Large Language Model to understand the scene, select the best action, and provide an optimal goal point or trajectory suited for the scene.

However, it was observed that while the LLM could select the best action, it failed to provide optimally located goal points. This shortcoming arises because LLMs are based on next-token prediction and are not optimized for complex calculations such as determining goal points in a navigation setting. To address this, I developed a separate vision-language model that takes a front camera image from the ego vehicle and a language command to predict a goal point on the image corresponding to the command.

Our plan is to eventually integrate the Talk2BEV pipeline with this vision-language network. In this integrated system, the optimal action can be reasoned out by Talk2BEV, and the VLN network can predict a goal point for that action. This predicted goal point can then be used for planning and navigation tasks, ensuring that the vehicle can navigate effectively based on the scene understanding and the given language commands.

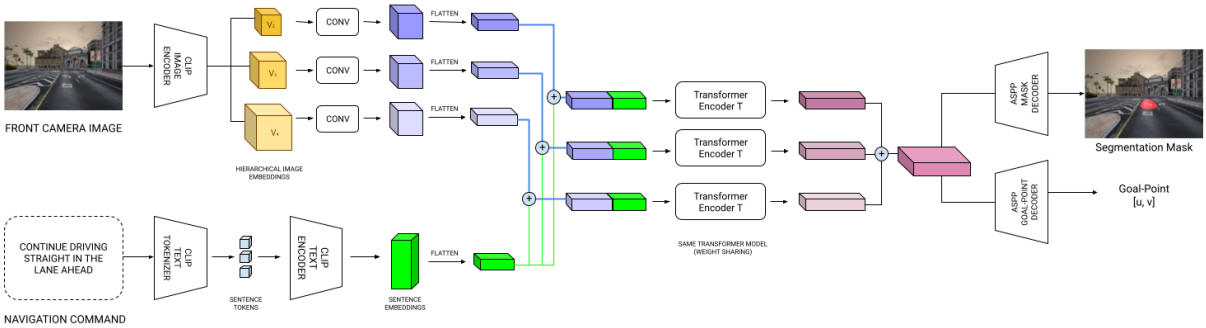

Figure 4.1 Overall pipeline of the proposed approach: Given the visual frame and a linguistic action command, the network predicts a segmentation map corresponding to the referenced navigable region and an associated goal point.

4.2 Goal Point Prediction

Our vision-language network takes two inputs: an image from the front camera and a corresponding linguistic command. The aim is to identify a navigable goal-point on the front camera image based on the provided action command. Fig. 4.1 shows the network architecture in detail. A detailed explanation follows below.

To encode the given navigation command, we tokenize the linguistic command using CLIP tokenizer and pass it through CLIP text encoder to obtain text embeddings $\mathbf{T}$. To get the image features from the given front camera image, we utilize the CLIP image encoder with ResNet-101 backbone. Hierarchical features are known to be beneficial for semantic segmentation; hence, we extract different visual feature $\mathbf{V_i} \in \mathbb{R}^{C_i \times H_i \times W_i}$ where $i \in \{2, 3, 4\}$ after the 2^{nd}, 3^{rd}, and 4^{th} layers of the ResNet backbone. Each $\mathbf{V_i}$ is passed through convolutional blocks ConvBlock$_i$ to bring them into a standard size with equal channel sizes, heights, and widths.

To capture the multi-modal context from the image and text features, we further use a transformer encoder adopted from the DETR architecture. All features $\mathbf{T}$, $\mathbf{V_2}$, $\mathbf{V_3}$, $\mathbf{V_4}$ are flattened, and text features are concatenated with different $\mathbf{V_i}$ individually to get multimodal features $\mathbf{M_i} = \mathbf{V_i} \oplus \mathbf{T}$. The $\mathbf{M_i}$ is then individually passed to the transformer encoder where the multi-headed self-attention layer helps in cross-modality interaction between the different kinds of features to obtain $\mathbf{X_i}$ as the encoder output with the same shape as $\mathbf{M_i}$.

We have two decoder heads, one each for the segmentation mask prediction and the goal point prediction task respectively. To predict the segmentation mask, $\mathbf{X_i}$ undergoes further reshape and restructure operations to reshape it into $\mathbb{R}^{C \times H \times W}$, resulting in $\mathbf{Z_i}$. For the segmentation mask prediction, we stack the $\mathbf{Z_i}$ from all layers to shape $\mathbb{R}^{C+C+C \times H \times W}$.

Both prediction heads use ASPP decoders from [74]. For segmentation mask prediction, ASPP outputs pass through a convolutional upsampling block that includes bilinear upsampling at specified stages to increase spatial resolution. The output finally undergoes sigmoid activation to produce binary masks.

In the goal point prediction decoder, it consists of convolutional layers followed by fully connected layers with the output reshaped to $\mathbb{R}^{2\times1}$ representing a pixel location on the image.

First, the segmentation mask prediction head is trained end-to-end with BCE loss between the predicted segmentation mask and the human-annotated ground truth segmentation mask. After a few epochs, the goal point prediction head is trained similarly end-to-end with a smooth L1 loss between the predicted goal point and human annotated ground truth goal point.

4.3 Additional Predictions

The results from the goal point prediction can be utilized for downstream tasks such as path planning. Path planning involves determining a feasible path for a vehicle or agent to follow. One method to achieve this is by using a neural network that predicts the path based on the current state and the final goal point, as well as the positions of other traffic agents in the bird's eye view (BEV) over the past few frames. However, trajectory predictions generated by such neural networks do not guarantee adherence to specific constraints, such as the final position, velocity, and other state variables.

To address this issue, we can employ optimization-based planners. These planners take into account the current state of the vehicle, along with additional constraints such as the desired final position and velocity. By solving for the least-cost trajectory, these planners ensure that the resulting path meets all specified requirements and constraints.

Since we used optimization-based planners in the downstream tasks, it was beneficial to predict additional final states to provide to the optimizer, thereby reducing its solution space. Predicting more information can streamline the optimization process, making it more efficient and accurate. Additionally, it is useful to determine whether the linguistic command refers to a stationary or non-stationary goal point. If the goal point is non-stationary, it may move over time, requiring us to query the goal point again after a few iterations to ensure accuracy.

To address these considerations, our goal-point prediction network also predicts two additional binary quantities: whether the final velocity is zero or non-zero, and whether another object is being referenced. The final velocity prediction can be inferred from the linguistic command since commands using "park," "stop," etc., require a final velocity of zero, while commands like "change lane," "turn," "follow," etc., generally imply a non-zero velocity. Predicting the final velocity helps the optimizer by providing more detailed information about the desired state at the end of the trajectory. Predicting whether an object is referenced allows the system to understand the context better, distinguishing between stationary and non-stationary goal points, and adjusting the planning process accordingly.

To ensure optimal performance in binary prediction tasks, we employ similar PyTorch classes to develop decoders utilizing the output features from the transformer encoder. These decoders have an input size of 1536 and consist of four hidden layers. The output layer, with a size of 1, is suitable for binary classification tasks. Each decoder comprises fully connected layers with ReLU activations, followed by a sigmoid activation to generate the final output.

The original goal prediction model remains static, while both decoder heads are trained using the Binary Cross-Entropy (BCE) loss function, which is tailored for binary classification challenges. This framework facilitates effective learning for distinguishing between zero and non-zero velocities, as well as between stationary and non-stationary goal points.

4.4 Complex Commands and Scene Understanding

To handle composite instructions serving cases where the final goal location is not visible in the current frame, we adapted this approach by decomposing the complex command into a list of atomic commands that need to be followed sequentially. For example, "switch to the left lane and then follow the black car" can be decomposed into "switch to the left lane" and "follow the black car". To decompose such complex commands, we constructed a list of atomic commands L, covering a wide range of simple actions such as lane changes, turns, speed adjustments, and object references. Upon receiving a complex command, we utilized the few-shot learning technique to prompt an LLM to decompose the given complex command into a sequential list of atomic commands l_i, from L. These atomic commands are then executed iteratively with our pipeline, with the predicted goal-point location serving as intermediate waypoints to help us reach the final goal point.

To generate high-level driving instructions tailored to the current scene, we can utilize powerful Large Vision-Language Models such as GPT-4V. By providing a front camera image along with a carefully crafted prompt that explains the driving context and available actions, GPT-4V can generate a suggested instruction command based on its analysis. This command is then forwarded to our pipeline for goal point prediction.

4.5 Transforming to BEV for Downstream Tasks

Usually, tasks like planning and navigation are handled not in the image space, but in a different representation such as Bird's Eye View or the local frame. Due to this, once the model has predicted a relevant goal point on the front camera image associated with the given command, we transform it to the local frame so it can be used for downstream tasks such as path planning and obstacle avoidance. For this transformation, we have two options available which are discussed below.

Since the dataset was created in Carla, a simulated environment, the road was flat everywhere without any bumps, and the height of the camera was known and constant. This uniformity allows us to use an inverse projection technique. By using the camera matrix, we can transform the 2D image coordinates into 3D homogeneous coordinates. In this method, we obtain the exact 3D coordinate by knowing the height coordinate and scaling the other coordinates proportionately. This method is precise in controlled environments where the terrain is predictable.

However, this method won't work in real-world datasets such as NuScenes, where the road is not always guaranteed to be perfectly flat. In these scenarios, the back projection might give incorrect

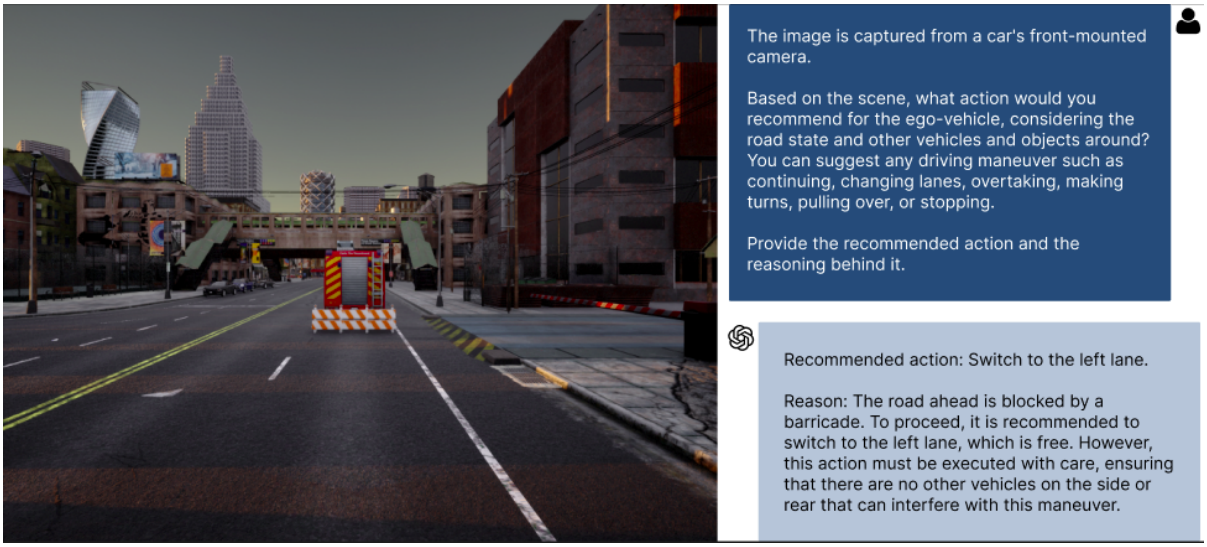

Figure 4.2 Scene Understanding: Pairing the goal-point prediction with a Large Vision-Language Model such as GPT-4V on the front camera image allows the system to operate in a self-reliant mode, eliminating the need for the user to assess and select the most suitable action. The vision-language model is capable of determining the optimal course of action from a variety of potential driving maneuvers. In the illustrated example, it accurately identifies an obstruction ahead and suggests ”switching to the left lane” to continue moving forward safely.

coordinates because the height of the road can vary from point to point. In such cases, using the inverse projection could lead to significant errors in the derived 3D coordinates.

To address this issue, we utilize the available LiDAR data in NuScenes. LiDAR provides high-resolution 3D point clouds of the environment. We can project all the LiDAR points onto the front camera image where the goal point is predicted. By identifying the LiDAR point that lies closest to the predicted goal point on the image, we can determine the goal point's 3D coordinates in the local frame. This method leverages the dense nature of LiDAR scans to ensure accuracy. Although this method is not absolutely precise, it works well in practice because the density of the LiDAR data minimizes the error when selecting the nearest point.

4.6 Results

For goal-point prediction, comparing with the ground truth annotations in the test data is not meaningful. This is because L2 comparisons are not suitable in this context since there can be multiple acceptable goal points for a given command. Therefore, we evaluated the goal-point prediction results extensively through qualitative analysis.

Regarding the additional predictions, we observed an average accuracy of 91% in final velocity state prediction and 95% in identifying object references. These high accuracies are primarily due to our limited vocabulary, which allows the network to effectively identify the keywords and syntax it

encountered during training. This familiarity with the training data's vocabulary and syntax significantly contributes to the network's ability to classify these tasks accurately.

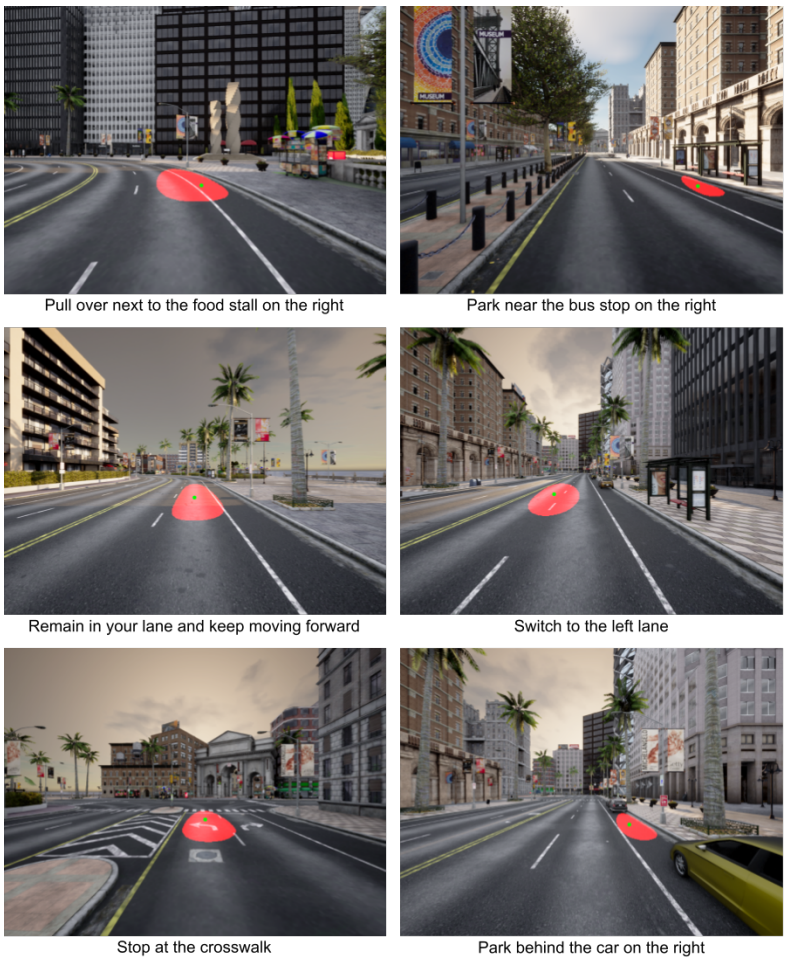

Figure 4.3 Model Predictions: The goal-point is shown in green and segmentation masks are shown in red. The model can predict reasonable outputs for different kinds of commands, identifying various lanes, other traffic actors, different objects on the roadside, and road signs.

Fig. 4.3 illustrates qualitative examples from the goal point prediction model applied to test scenes. The model demonstrates an understanding of lane concepts, including the implications of driving within the current lane or switching to the left or right lane. Additionally, it can accurately identify and respond to objects such as food stalls, bus stops, and benches, which have been annotated with similar commands in the training dataset. Moreover, the model predicts plausible positions for various driving maneuvers, such as stopping at a crosswalk or parking behind another vehicle. Given the limited vocabulary on which the model has been trained, it performs reasonably well and consistently generates collision-free goal points.

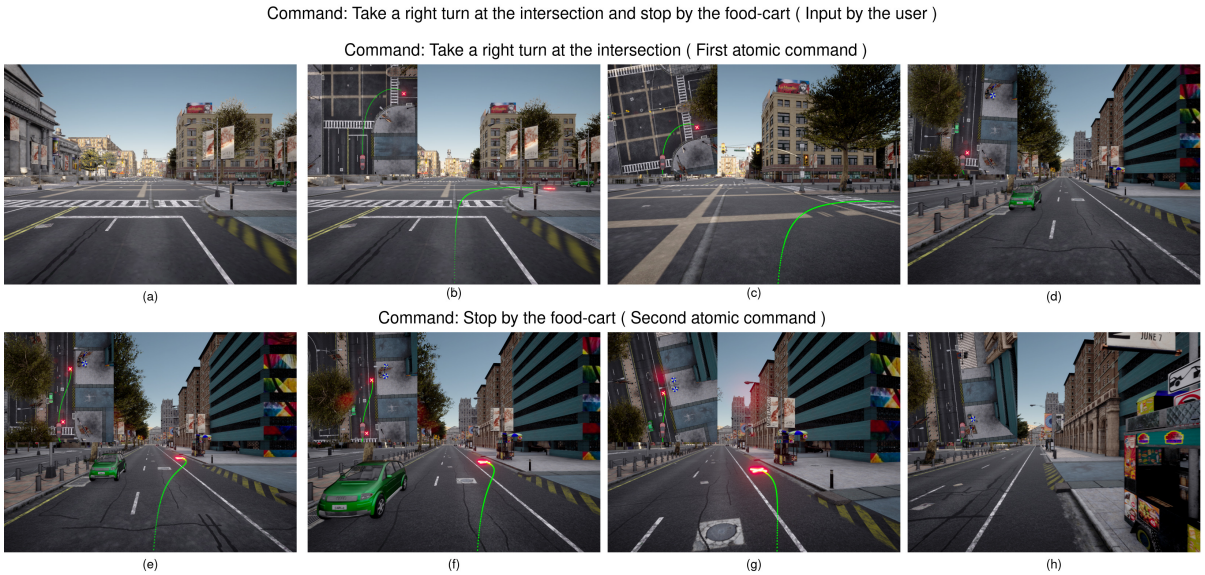

Figure 4.4 Complex command: The closed-loop simulation illustrates a compound command divided into two atomic commands, executed sequentially. Navigation initially follows the first atomic command. After reaching the first goal, the second atomic command is executed to reach the final goal.

Integration with Scene Understanding Modules: As illustrated in Fig. 4.2, the vision-language model evaluates the situation and determines the best course of action from various potential driving maneuvers. In the example, the model identifies an obstruction ahead and recommends ”switching to the left lane” to safely continue forward. This recommendation is then used by our pipeline to predict a collision-free goal point. Hence, paired with the scene understanding module, the complete pipeline is able to deduce the best action on its own and decide on the optimal goal point to move to without the human providing any help. The output ensures that the vehicle can navigate the environment effectively, avoiding obstacles and making safe driving decisions.

Handling Complex Commands: As illustrated in Fig. 4.4, we can handle complex commands step by step. Given the command, the LLM breaks it into two smaller commands. The first action is taking the right turn. We determine an appropriate goal point and then navigate to it. Once reached, the next action is stopping near the food stall. Since it is visible in the current frame, we obtain the expected goal point and plan and navigate to it successfully. Hence, we are able to navigate to the food stall even though it was not visible in the starting frame.

4.7 Chapter Summary

In this project, I worked on integrating vision and language in the context of autonomous driving systems, specifically focusing on predicting goal points for navigation tasks based on linguistic commands

and visual input from a front camera. The work builds upon the Talk2BEV pipeline, which primarily deals with scene understanding, and extends it to encompass planning aspects.

The proposed vision-language model is designed to take two inputs: a front camera image and a corresponding linguistic action command. It aims to predict a navigable segmentation mask and goal point on the image based on the provided command. To achieve this, the model architecture is detailed, involving the use of CLIP for tokenizing linguistic commands and extracting text embeddings, and a modified version of the DETR architecture to fuse image and text features for prediction tasks. Furthermore, the chapter addresses the need for additional predictions to enhance downstream tasks such as path planning. These predictions include determining whether the final state should be a rest state and whether any other vehicle is referenced in that command. By incorporating these predictions, the system can provide more detailed information to the planners, thereby reducing their solution space and improving efficiency.

Complex commands, where the final goal location may not be visible in the current frame, are handled by decomposing them into a sequence of atomic commands. This decomposition, facilitated by a few-shot learning technique and a large vision-language model like GPT-4V, enables the system to iteratively execute atomic commands, utilizing predicted goal points as intermediate waypoints to reach the final destination.

Chapter 5

Conclusion

For my thesis work at the Robotics Research Center, I primarily focused on two submissions.

In the first project, we introduced the Talk2BEV pipeline, a language interface for Bird's Eye View (BEV) maps used in autonomous driving systems. Leveraging recent advances in Large Language Models (LLMs) and Large Vision-Language Models (LVLMs), Talk2BEV supports a variety of autonomous driving tasks. These tasks include visual and spatial reasoning, predicting unsafe traffic interactions, and plotting alternative routes. This approach promises to advance the real-world applicability of autonomous driving systems. Talk2BEV not only expands the range of scenarios an autonomous driving system can handle but also bridges the gap between traditional autonomous driving models and the extensive capabilities of pre-trained image-language models. As we integrate large pre-trained models into autonomous driving systems, we emphasize the necessity of safety and alignment research before deploying these models in safety-critical environments. Additionally, we introduced Talk2BEV-Bench, a benchmark designed to evaluate future work in LVLMs for autonomous driving applications. This benchmark provides a foundational step for rigorously assessing the role and effectiveness of image-language models in autonomous driving. Our findings suggest that the future of autonomous driving systems is leaning towards more integrated, adaptable, and intelligent models.

In the second project, we addressed the Visual Language Navigation task of grounding navigable goal points based on linguistic commands to guide autonomous vehicles. We proposed a novel transformer-based model and conducted comprehensive experiments to demonstrate the effectiveness of our approach. This model can predict reasonable goal points along with other relevant predictions for downstream tasks. We demonstrated how the goal points predicted from front camera images are suitable for autonomous driving applications. In a separate effort, my teammate also showed that end-to-end training with a planner in the pipeline enhances the goal point network's ability to predict more reachable goal points. In this work, we focused on grounding within single frames; however, future research could explore grounding at the video level. This would be more realistic for commands with temporal constraints and could illustrate the evolution of goal points based on the presence of dynamic objects in the scene.

As a combined effort from both pipelines, future work could focus on extracting image features from peripheral camera images and embedding them into the BEV maps to provide rich context from cameras, rather than just semantic class information. This can be achieved by projecting image features, such as those from LSeg, onto the BEV. However, a challenge arises in that such context-rich BEV might result in a very large input size. For instance, if the BEV is 200×200 pixels and the image feature vectors have a length of 512, the resulting context-rich BEV would be $200 \times 200 \times 512$, which is a large input for any deep learning model. This issue might be addressed by filtering out embeddings from non-driving objects and passing the feature vectors of driving-related objects through another encoder to reduce the size. Such a perception-rich BEV could be used directly for tasks like scene understanding and goal point prediction, which are crucial for autonomous driving.

Related Publications

1. **Talk2BEV: Language-enhanced Bird's-eye View Maps for Autonomous Driving**

   Tushar Choudhary*, Vikrant Dewangan*, Shivam Chandhok*, Shubham Priyadarshan, Anushka Jain, Arun K. Singh, Siddharth Srivastava, Krishna Murthy Jatavallabhula, K. Madhava Krishna

   In proceedings of IEEE International Conference on Robotics and Automation (ICRA) 2024

2. **LeGo-Drive: Language-enhanced Goal-oriented Closed-Loop End-to-End Autonomous Driving**

   Pranjal Paul, Anant Garg*, **Tushar Choudhary***, Arun Kumar Singh, K Madhava Krishna

   Submitted to IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) 2024

Other Publications

1. **UAP-BEV: Uncertainty Aware Planning using Bird's Eye View generated from Surround Monocular Images**

   Vikrant Dewangan, Basant Sharma, **Tushar Choudhary**, Sarthak Sharma, Aakash Aanegola, Arun K Singh, K Madhava Krishna

   In proceedings of IEEE International Conference on Automation Science and Engineering (CASE) 2023

Bibliography

[1] Jialian Wu, Jianfeng Wang, Zhengyuan Yang, Zhe Gan, Zicheng Liu, Junsong Yuan, and Lijuan Wang. Grit: A generative region-to-text transformer for object understanding, 2022.

[2] Yuning Du, Chenxia Li, Ruoyu Guo, Xiaoting Yin, Weiwei Liu, Jun Zhou, Yifan Bai, Zilin Yu, Yehua Yang, Qingqing Dang, and Haoshuang Wang. Pp-ocr: A practical ultra lightweight ocr system, 2020.

[3] OpenAI. Gpt-4 technical report, 2023.

[4] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving, 2020.

[5] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. Scaling instruction-finetuned language models, 2022.

[6] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric. P Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena, 2023.

[7] OpenAI. Chatgpt, 2021. Accessed: yyyy-mm-dd.

[8] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models, 2023.

[9] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher,

Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023.

[10] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models, 2023.

[11] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models, 2023.

[12] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning, 2023.

[13] Jonah Philion and Sanja Fidler. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d, 2020.

[14] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Qiao Yu, and Jifeng Dai. Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers, 2022.

[15] Anthony Hu, Zak Murez, Nikhil Mohan, Sofía Dudas, Jeffrey Hawke, Vijay Badrinarayanan, Roberto Cipolla, and Alex Kendall. Fiery: Future instance prediction in bird's-eye view from surround monocular cameras, 2021.

[16] Shengchao Hu, Li Chen, Penghao Wu, Hongyang Li, Junchi Yan, and Dacheng Tao. St-p3: End-to-end vision-based autonomous driving via spatial-temporal feature learning, 2022.

[17] Stefano V. Albrecht, Cillian Brewitt, John Wilhelm, Balint Gyevnar, Francisco Eiras, Mihai Dobre, and Subramanian Ramamoorthy. Interpretable goal-based prediction and planning for autonomous driving, 2021.

[18] Amina Ghoul, Itheri Yahiaoui, Anne Verroust-Blondet, and Fawzi Nashashibi. Interpretable goal-based model for vehicle trajectory prediction in interactive scenarios, 2023.

[19] Junru Gu, Chen Sun, and Hang Zhao. Densetnt: End-to-end trajectory prediction from dense goal sets, 2021.

[20] Nivedita Rufus, Kanishk Jain, Unni Krishnan R Nair, Vineet Gandhi, and K Madhava Krishna. Grounding linguistic commands to navigable regions. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, September 2021.

[21] Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Zhenyu Qiu, Wei Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, and Rongrong Ji. Mme: A comprehensive evaluation benchmark for multimodal large language models, 2023.

[22] Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, Kai Chen, and Dahua Lin. Mmbench: Is your multi-modal model an all-around player?, 2023.

[23] Peng Xu, Wenqi Shao, Kaipeng Zhang, Peng Gao, Shuo Liu, Meng Lei, Fanqing Meng, Siyuan Huang, Yu Qiao, and Ping Luo. Lvlm-ehub: A comprehensive evaluation benchmark for large vision-language models, 2023.

[24] Bohao Li, Rui Wang, Guangzhi Wang, Yuying Ge, Yixiao Ge, and Ying Shan. Seed-bench: Benchmarking multimodal llms with generative comprehension, 2023.

[25] Panos Achlioptas, Ahmed Abdelreheem, F. Xia, Mohamed Elhoseiny, and Leonidas J. Guibas. Referit3d: Neural listeners for fine-grained 3d object identification in real-world scenes. In *European Conference on Computer Vision*, 2020.

[26] Pin-Hao Huang, Han-Hung Lee, Hwann-Tzong Chen, and Tyng-Luh Liu. Text-guided graph neural networks for referring 3d instance segmentation. In *AAAI Conference on Artificial Intelligence*, 2021.

[27] Mingtao Feng, Zhen Li, Qi Li, Liang Zhang, XiangDong Zhang, Guangming Zhu, Hui Zhang, Yaonan Wang, and Ajmal Mian. Free-form description guided 3d visual graph network for object grounding in point cloud, 2021.

[28] Dave Zhenyu Chen, Angel X. Chang, and Matthias Nießner. Scanrefer: 3d object localization in rgb-d scans using natural language, 2020.

[29] Dave Zhenyu Chen, Ali Gholami, Matthias Nießner, and Angel X. Chang. Scan2cap: Context-aware dense captioning in rgb-d scans, 2020.

[30] Daichi Azuma, Taiki Miyanishi, Shuhei Kurita, and Motoaki Kawanabe. Scanqa: 3d question answering for spatial scene understanding, 2022.

[31] Shih-Han Chou, Wei-Lun Chao, Wei-Sheng Lai, Min Sun, and Ming-Hsuan Yang. Visual question answering on 360-degree images, 2020.

[32] Erik Wijmans, Samyak Datta, Oleksandr Maksymets, Abhishek Das, Georgia Gkioxari, Stefan Lee, Irfan Essa, Devi Parikh, and Dhruv Batra. Embodied question answering in photorealistic environments with point cloud perception, 2019.

[33] Xu Yan, Zhihao Yuan, Yuhao Du, Yinghong Liao, Yao Guo, Zhen Li, and Shuguang Cui. Comprehensive visual question answering on point clouds through compositional scene manipulation. *arXiv preprint arXiv:2112.11691*, 2021.

[34] Yining Hong, Haoyu Zhen, Peihao Chen, Shuhong Zheng, Yilun Du, Zhenfang Chen, and Chuang Gan. 3d-llm: Injecting the 3d world into large language models, 2023.

[35] Runsen Xu, Xiaolong Wang, Tai Wang, Yilun Chen, Jiangmiao Pang, and Dahua Lin. Pointllm: Empowering large language models to understand point clouds, 2023.

[36] Arun Balajee Vasudevan, Dengxin Dai, and Luc Van Gool. Object referring in videos with language and human gaze, 2018.

[37] Thierry Deruyttere, Simon Vandenhende, Dusan Grujicic, Luc Van Gool, and Marie-Francine Moens. Talk2car: Taking control of your self-driving car. In *Conference on Empirical Methods in Natural Language Processing*, 2019.

[38] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding, 2016.

[39] Dongming Wu, Wencheng Han, Tiancai Wang, Xingping Dong, Xiangyu Zhang, and Jianbing Shen. Referring multi-object tracking, 2023.

[40] Dongming Wu, Wencheng Han, Tiancai Wang, Yingfei Liu, Xiangyu Zhang, and Jianbing Shen. Language prompt for autonomous driving, 2023.

[41] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019.

[42] Licheng Yu, Zhe Lin, Xiaohui Shen, Jimei Yang, Xin Lu, Mohit Bansal, and Tamara L. Berg. Mattnet: Modular attention network for referring expression comprehension, 2018.

[43] Long Chen, Wenbo Ma, Jun Xiao, Hanwang Zhang, and Shih-Fu Chang. Ref-nms: Breaking proposal bottlenecks in two-stage referring expression grounding, 2021.

[44] Richang Hong, Daqing Liu, Xiaoyu Mo, Xiangnan He, and Hanwang Zhang. Learning to compose and reason with language tree structures for visual grounding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(2), February 2022.

[45] Yongfei Liu, Bo Wan, Xiaodan Zhu, and Xuming He. Learning cross-modal context graph for visual grounding, 2019.

[46] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks, 2016.

[47] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn, 2018.

[48] Yue Liao, Si Liu, Guanbin Li, Fei Wang, Yanjie Chen, Chen Qian, and Bo Li. A real-time cross-modality correlation filtering method for referring expression comprehension, 2020.

[49] Zhengyuan Yang, Tianlang Chen, Liwei Wang, and Jiebo Luo. Improving one-stage visual grounding by recursive sub-query construction, 2020.

[50] Jiajun Deng, Zhengyuan Yang, Tianlang Chen, Wengang Zhou, and Houqiang Li. Transvg: End-to-end visual grounding with transformers, 2022.

[51] Yiyi Zhou, Rongrong Ji, Gen Luo, Xiaoshuai Sun, Jinsong Su, Xinghao Ding, Chia wen Lin, and Qi Tian. A real-time global inference network for one-stage referring expression comprehension, 2019.

[52] Gen Luo, Yiyi Zhou, Xiaoshuai Sun, Liujuan Cao, Chenglin Wu, Cheng Deng, and Rongrong Ji. Multi-task collaborative network for joint referring expression comprehension and segmentation, 2020.

[53] Hengcan Shi, Hongliang Li, Fanman Meng, and Qingbo Wu. Key-word-aware network for referring expression image segmentation. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision – ECCV 2018*, 2018.

[54] Linwei Ye, Mrigank Rochan, Zhi Liu, and Yang Wang. Cross-modal self-attention network for referring image segmentation, 2019.

[55] Shaofei Huang, Tianrui Hui, Si Liu, Guanbin Li, Yunchao Wei, Jizhong Han, Luoqi Liu, and Bo Li. Referring image segmentation via cross-modal progressive comprehension, 2020.

[56] Kanishk Jain and Vineet Gandhi. Comprehensive multi-modal interactions for referring image segmentation. In *Findings of the Association for Computational Linguistics: ACL 2022*. Association for Computational Linguistics, 2022.

[57] 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, 2023.

[58] Wayve. Lingo-1: Exploring natural language for autonomous driving. https://wayve.ai/thinking/lingo-natural-language-autonomous-driving/, Year. Accessed: 2 October 2023.

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

[60] Hao Sha, Yao Mu, Yuxuan Jiang, Li Chen, Chenfeng Xu, Ping Luo, Shengbo Eben Li, Masayoshi Tomizuka, Wei Zhan, and Mingyu Ding. Languagempc: Large language models as decision makers for autonomous driving. *arXiv preprint arXiv:2310.03026*, 2023.

[61] Zhenhua Xu, Yujia Zhang, Enze Xie, Zhen Zhao, Yong Guo, Kenneth KY Wong, Zhenguo Li, and Hengshuang Zhao. Drivegpt4: Interpretable end-to-end autonomous driving via large language model. *arXiv preprint arXiv:2310.01412*, 2023.

[62] Chenbin Pan, Burhaneddin Yaman, Tommaso Nesti, Abhirup Mallik, Alessandro G Allievi, Senem Velipasalar, and Liu Ren. Vlp: Vision language planning for autonomous driving, 2024.

[63] Long Chen, Oleg Sinavski, Jan Hünermann, Alice Karnsund, Andrew James Willmott, Danny Birch, Daniel Maund, and Jamie Shotton. Driving with llms: Fusing object-level vector modality for explainable autonomous driving. *arXiv preprint arXiv:2310.01957*, 2023.

[64] Hao Shao, Yuxuan Hu, Letian Wang, Steven L Waslander, Yu Liu, and Hongsheng Li. Lmdrive: Closed-loop end-to-end driving with large language models. *arXiv preprint arXiv:2312.07488*, 2023.

[65] Chonghao Sima, Katrin Renz, Kashyap Chitta, Li Chen, Hanxue Zhang, Chengen Xie, Ping Luo, Andreas Geiger, and Hongyang Li. Drivelm: Driving with graph visual question answering. *arXiv preprint arXiv:2312.14150*, 2023.

[66] Kaustubh Mani, Swapnil Daga, Shubhika Garg, Sai Shankar, Krishna Murthy Jatavallabhula, and Madhava Krishna K. Monolayout: Amodal scene layout from a single image. In *WACV*, 2020.

[67] Kaustubh Mani, Sai Shankar, Krishna Murthy Jatavallabhula, and Madhava Krishna K. Autolay: Benchmarking monocular layout estimation. In *IROS*, 2020.

[68] Xu Zhao, Wenchao Ding, Yongqi An, Yinglong Du, Tao Yu, Min Li, Ming Tang, and Jinqiao Wang. Fast segment anything, 2023.

[69] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models, 2023.

[70] Krishna Murthy Jatavallabhula, Alihusein Kuwajerwala, Qiao Gu, Mohd Omama, Tao Chen, Shuang Li, Ganesh Iyer, Soroush Saryazdi, Nikhil Keetha, Ayush Tewari, Joshua B. Tenenbaum, Celso Miguel de Melo, Madhava Krishna, Liam Paull, Florian Shkurti, and Antonio Torralba. Conceptfusion: Open-set multimodal 3d mapping, 2023.

[71] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. Scaling instruction-finetuned language models, 2022.

[72] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, March 2023.

[73] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning, 2023.

[74] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation, 2018.