V3LMA: Visual 3D-enhanced Language Model for Autonomous Driving

Jannik Lübberstedt[*1] Esteban Rivera[*12†] Nico Uhlemann[12] Markus Lienkamp[12]
[1]Technical University of Munich (TUM)
[2]Munich Institute of Robotics and Machine Intelligence (MIRMI)
firstname.lastname@tum.de

Abstract

Large Vision Language Models (LVLMs) have shown strong capabilities in understanding and analyzing visual scenes across various domains. However, in the context of autonomous driving, their limited comprehension of 3D environments restricts their effectiveness in achieving a complete and safe understanding of dynamic surroundings. To address this, we introduce V3LMA, a novel approach that enhances 3D scene understanding by integrating Large Language Models (LLMs) with LVLMs. V3LMA leverages textual descriptions generated from object detections and video inputs, significantly boosting performance without requiring fine-tuning. Through a dedicated preprocessing pipeline that extracts 3D object data, our method improves situational awareness and decision-making in complex traffic scenarios, achieving a score of 0.56 on the LingoQA benchmark. We further explore different fusion strategies and token combinations with the goal of advancing the interpretation of traffic scenes, ultimately enabling safer autonomous driving systems.

1. Introduction

For autonomous agents to navigate effectively in intricate scenarios, complete environmental understanding and interpretation are crucial. As a result, autonomous driving continues to be an unresolved research challenge, with much of the focus directed toward the fields of computer vision and natural language processing. An accurate understanding of complex scenes is essential for the development of autonomous vehicles, traffic monitoring systems, and smart city infrastructure. It involves not only detecting and interpreting objects, but also understanding the intricate relationships between them, the context, and the continuous changes in traffic dynamics.

Figure 1. 3D information is essential for accurate and safe scene understanding in autonomous driving. While LVLMs can extract rich semantic information from images, they struggle to capture 3D spatial cues. To address this limitation, we propose enhancing visual scene understanding by incorporating textual descriptions of 3D detections. These descriptions are processed by an LLM, which is better equipped to handle textual input than an LVLM.

LLMs have recently shown impressive capabilities in understanding and reasoning over textual representations of the natural world. These strengths extend to multimodal settings, where LLMs are augmented with visual inputs. Such models excel at extracting information from images when guided by textual queries and can even reason about visual scenes and predict the outcomes of actions depicted in them [22] [15]. However, these models are constrained by the data they are trained on, meaning their performance can significantly decline in specialized fields or when faced with unfamiliar data formats. To improve their reasoning abilities in a specific domain, these models are often fine-tuned using domain-specific data. Notable examples include medicine and biology, with ongoing efforts to fine-tune models to reason effectively in driving scenarios [20][8][40][33]. Yet, such fine-tuning demands considerable computational resources and access to high-quality, labeled datasets.

Recent developments in Multimodal Large Language Models (MLLMs), which integrate inputs from various sensory modalities such as images, video, text, and sensor data, have opened new opportunities for improving scene under-

[*]Equal contribution
[†]Corresponding author.

The research was partially funded by the Bavarian Research Foundation within the project "Data-Enabled Autonomous Driving", and through basic research funds from the Institute for Automotive Technology

standing in traffic environments [7][38]. These models are capable of fusing diverse input types to generate context-aware insights that can adapt to real-time, dynamic scenarios. Nevertheless, challenges remain as MLLMs often struggle to understand the relationships between heterogeneous data types and interpret ambiguous or incomplete inputs, especially in complex traffic scenes.

In this work, we explore the use of video-based LVLMs in combination with LLMs for scene understanding in autonomous driving. Our objective is to advance the capabilities of artificial intelligence (AI) systems in interpreting and responding to the diverse and unpredictable nature of traffic. By fusing visual and textual data, we aim to enhance situational awareness, support more informed decision-making, and contribute to the development of safer, more efficient transportation systems. Previous work has explored the use of textual representations alongside visual input to reduce hallucinations in LVLMs outputs, often relying on fine-tuning to ensure accurate reasoning about the textual input [33].

In contrast, we introduce Visual 3D-enhanced Language Model for Autonomous Driving (V3LMA), a novel approach that integrates textual descriptions of object detections with visual video inputs into a combined LLMs-LVLM architecture. Notably, while significantly boosting the performance of the base models, no fine-tuning is required. The textual detections are generated by a preprocessing pipeline that includes object detection and tracking modules. This pipeline is modular and easily adaptable, allowing for seamless integration of alternative models and modalities. Furthermore, by incorporating 3D detection information, V3LMA enables a deeper understanding of spatial relationships among traffic participants, contributing to a safer and more precise interpretation of complex traffic scenes. To summarize, the main contributions of this work are:

- Proposal and evaluation of V3LMA, a novel method that combines the strengths of LLMs and LVLMs to enhance 3D scene understanding in traffic scenarios—without requiring model training or fine-tuning.
- Design of a modular preprocessing pipeline that converts 3D object detections into textual input suitable for LLMs, enabling flexible integration of different detection and tracking models.
- A systematic study of fusion strategies for combining LVLMs and LLMs, analyzing the impact of fusion layers, weight distributions, and token configurations.

2. Related work

2.1. Datasets

To train and evaluate scene understanding models for autonomous driving, several datasets are available. Some of them extend existent datasets with novel captions describing the scenes [6, 11, 11, 12, 14, 36]. Besides, there are datasets which present question and answer pairs closely related to the driving task; some focus on image-level granularity [2, 21, 24] or in video-level captioning [19, 32, 37]. A detailed description is given in appendix Table 11. The only available dataset with Video annotations was LingoQA[19], which we then use for our evaluation. It comprises approximately 28,000 unique short video scenarios captured in central London, accompanied by 419,000 question-answer pairs, facilitating the assessment of vision-language models in understanding and reasoning about real-world driving situations. Additionally, they introduce Lingo-Judge, a learned text classifier that serves as an evaluation metric, achieving a Spearman correlation coefficient of 0.95 with human evaluations, thereby providing a reliable measure of model performance.

2.2. Scene understanding models

The authors of LingoQA published a baseline model alongside its dataset. This model uses a similar way of incorporating visual information into the LLM as LLaVA. Instead of using a linear projection layer to project the Contrastive Language-Image Pre-Training (CLIP) features into the language feature space, it uses Querying Transformer (Q-Former) modules to retrieve a constant number of tokens per image. Similarly, the WiseAD model [39] utilizes a mobile vision-language model architecture using a frozen CLIP model, a learnable projector layer and the MobileLLaMA [4] language model. Despite its small size of around 2B parameters, it is capable of achieving a similar LingoQA score to the LingoQA baseline using less than a third of its parameters. Zhous et al. [42] approach leverages multiple visual and textual inputs to handle understanding: The fusion of these inputs is achieved through various methods: Concatenation, where the hints are combined with CLIP tokens; Self-cross-attention, which applies self-attention to CLIP tokens and cross-attention between the output and the hints; Joint cross-attention, where visual tokens act as queries, with both visual tokens and hints serving as keys and values. DriveMM [10] is a multimodal model designed to reason across different input modalities, including images and videos. Specifically, it aims to enable a LVLM to process and understand birds-eye view (BEV) images derived from LiDAR point projections and multi-view videos. The Video Token Sparsification [17] model aims to reduce computational effort, by selecting a subset of visual features based on their salience. The salience is determined using a lightweight Convolutional Neural Network (CNN) model, called proposal model, which identifies the most relevant features while discarding less informative ones. Finally, Jain et al.[12] propose an approach, where object detection and tracking is performed on LiDAR point clouds to

retrieve accurate spatial locations for objects in the environment of the ego-vehicle. This information is added to the visual question answering (VQA) prompt to reduce hallucination, however, their results show that a single LVLM cannot process a complex prompt based on the information of the detections.

In addition, some models are trained to solve the End-to-End planning task, but indirectly must also achieve certain understanding of the environment to be able to plan for it. For instance, DriveLM [30] proposes a Graph Visual Question Answering pipeline in which a BLIP-2 Visual language model (VLM) is tasked with answering questions about a scene from perception to prediction to planning to behaviour to motion. SimpleLLM4AD [41] extends the DriveLM baseline by adding a 2D object detection model and passing the bounding boxes, class and color of objects directly into the input of the baseline to extend the information available to the model. LLM-Assist [29] combines rule based planning using PDM-closed [5] with dynamic planning using a LLM for high uncertainty scenarios. PDM generates 15 trajectories at each planning step. The LLM is only used when the PDM planner has a high level of uncertainty for the generated trajectories. DriveVLM [33] fine-tunes a Qwen-VL 10B model to perform driving scene understanding and planning. It retrieves a driving condition description from the LVLM using "Describe the driving condition" and the driving video as input. The obtained driving condition description, the textual description of 2D object detection results and the driving video are then used as input to the same LVLM to obtain a scene description focused in the critical objects in the scene and their influence on the ego-vehicle.

3. Approach

A key question is how to integrate the advanced reasoning capabilities of LLMs with the visual understanding of LVLMs. One approach is to combine the textual outputs of the two individual models as input to a third model. This requires three models which leads to increasing computational demands and the data used for processing. Additionally, the output of the models have decreased representational power compared to the internal features of the models. This is a result from the textual output being a mapping from a high dimensional internal feature space to a much smaller discrete output space, allowing to represent one token per position. This process looses a large amount of information about the models internal state.

Another option is to combine the output scores for tokens from both models by merging the score distributions for each output token. This keeps the feature space much richer than relying on the textual output of the models. However, during our experiments it became clear that this is not the ideal way to merge the models as their output distributions

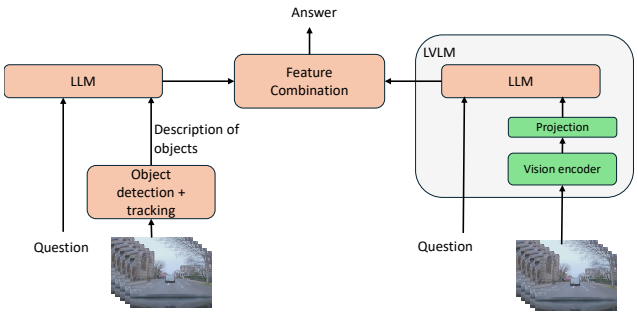

Figure 2. Overview of the approach of this work. At the left side a pre-trained base LLM is getting a textual description of surrounding objects and a question as input. On the right side, a pre-trained LVLM with the same base LLM as on the left-hand side uses the video and the question as input to reason about the visual content of the scene. The information of both models is combined to enhance the overall scene understanding.

differ significantly: LLMs typically have a broader distribution of scores across different tokens, while LVLMs have a more concentrated distribution with higher scores for fewer tokens, leading to a bias towards LVLM output in token selection.

In our approach the features per token from both models are merged after individual transformer layers to enable the models to process and modify the combined features. The approach is displayed in Figure 2. This enables the models to exchange more detailed information opposed to the approaches presented above. However, it can not be expected that a combination of arbitrarily chosen LLMs and LVLMs perform good results due to their feature space being different. Here, a different feature space between the LLM and LVLM means that similar features from both models correspond to different concepts. Two aspects which ensure model's features to correspond to similar concepts are:

1. Models must have similar architectures such that at similar points inside both models, their internal representation corresponds to similar concepts. If this is not the case, features can differ significantly, compromising the feature fusion. Additionally, when the internal dimensionality differs, a mapping mechanism to match both models dimensions needs to be found, which would require additionally tuning. This can be ensured when using a LVLM and its base LLM in the combination as both have the same LLM architecture.
2. Both models should have a similar output space. In the case of LLMs this corresponds to the models sharing the vocabulary and tokens being mapped to the same index in the output. This is important as it ensures that features close to the output layer correspond to similar concepts, as a result of them being mapped to the same outputs.

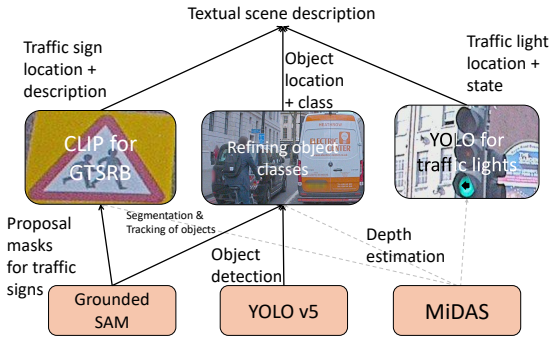

Figure 3. Pipeline for obtaining locations and descriptions for traffic signs, trafic lights and moving objects in the scene.

3.1. Models

To comply with the forementioned aspects for the feature fusion, the following combinations were tested:

1. LLaVA-Video-7B-Qwen2+Qwen2.5-7B-Instruct (V3LMA-L) [14B parameter]
2. Qwen2-VL-7B-Instruct+Qwen2.5-7B-Instruct (V3LMA-Q) [14B parameter]
3. Qwen2-VL-2B-Instruct+Qwen2.5-1.5B-Instruct (V3LMA-Q-mini) [3.5B parameter]

V3LMA-Q, V3LMA-Q-mini and V3LMA-L use the Qwen-LLM as base model, fullfillyng the first condition. Moreover, the Qwen-LLM model uses a vocabulary of size 151665, so together with the Qwen-VL they share the same vocabulary and index assignment to tokens. The V3LMA-L model has a modified vocabulary of size 151647 which uses the same vocabulary and index to token assignment as the Qwen-LLM model. However, it discards the last tokens which are used in Qwen-LLM as special tokens. This might negatively influence the performance when combining V3LMA-L with Qwen-LLM.

3.2. 3D Understanding module

The integration of LLMs and LVLMs has been well established. This raises a crucial question: what additional information could enhance the visual processing capabilities of LVLMs? For autonomous driving, 3D information may be the key to improve the reasoning capabilities.

Accurate scene understanding is essential for tasks like trajectory planning, which rely on precise 3D localization of traffic participants. Since vehicle interactions occur in 3D space, depth information is critical. However, images inherently lack depth cues, and LVLMs struggle with 3D scene comprehension as they are not explicitly provided with this data.

Therefore, we enhance the LVLM understanding of the scene through 3D information that is collected through a preprocessing pipeline based on segmentation with GroundedSam [26], detection with YOLOV5 [13] monocular depth estimation with MiDas [25]. The main structure is presented on Figure 3.

Object detection and tracking We begin by focusing on traffic participants, detecting and tracking objects in 2D using Grounded SAM [26]. To enhance segmentation, we integrate a YOLOv5 detector for specific categories of interest. In this work, key objects that significantly influence the ego-vehicle's driving behavior include trucks, motorbikes, cars, pedestrians, bicycles, traffic lights, and traffic signs.

Nonetheless, extracting 3D information from a monocular image remains essential. To achieve this, we employ the MiDAS depth estimator [25]. Depth is assigned to each object by extracting the depth value from the middle pixel of the detection. This results in a final output consisting of a list of detected objects, each labeled with its class and 3D localization relative to the ego-vehicle. A more detailed explanation of this module is provided in the Appendix Section 8.

Apart from traffic participants, we also notice that LVLMs can understand the presence of traffic lights and signs, but have issues extracting the exact meaning of some signs specific to the context and the exact state of the traffic light. To handle this, we propose a module to preprocess this information using a CLIP model and appending the output directly to our textual description of the surrounding objects. The description of those modules is shown in the appendix Section 9.

3.3. Merging parameters

The approach is to combine a LLM and a LVLM to obtain the reasoning capabilities on long language inputs of the LLM to support the visual reasoning of the LVLM. The features of both models will be added together at different stages inside the models and with different weightings. The summing approach keeps the size of the features constant which enables the results to being directly usable in the following model stages. The input to both models differs and therefore, typically does not have the same length, which makes it not possible to combine tokens at the same position. Therefore, only the features of the last produced token will be considered for combination with the feature tokens of the other model. The combination can either be applied pairwise between both models last token or between one models last token and all of the tokens of the other model. This might be interpreted as adding the last tokens information of one model to all the tokens information of the other model, resulting in more substantial modification of the other models features. The following components contribution to the performance of the combined model will be investigated:

Head Weights The classification heads of the LVLM and the LLM have different weights. They are each optimized to operate correctly on the features that they expect in their input as a result of their individual training. When the features of both models are merged before the classification head, the input to the classification head is modified compared to the input observed during training. This is typically referred to as a change in the feature distribution. To investigate how the weights of the head should idealy be adjusted to be able to deal with the changed input features, differently weighted sums of the head weights are considered. The different configurations are displayed in Table 1. The mathematical formulation of obtaining the combined head weight $\mathbf{W_{combined}} \in \mathbb{R}^{\text{model\_dimension} \times \text{vocabulary\_size}}$ is displayed in equation 1. p^{head}_{LVLM} corresponds to the head weighting towards the LVLM and p^{head}_{LLM} to the head weighting of the LLM.

$$\mathbf{W}^{\mathbf{head}}_{\mathbf{combined}} = p^{head}_{\text{LLM}} \cdot \mathbf{W}^{\mathbf{head}}_{\text{LLM}} + p^{head}_{\text{LVLM}} \cdot \mathbf{W}^{\mathbf{head}}_{\text{LVLM}} \quad (1)$$

Table 1. The tested head weightings for the merging

LLM Head Weight p^{head}_{LLM}	LVLM Head Weight p^{head}_{LVLM}
0.1	0.9
0.3	0.7
0.5	0.5
0.7	0.3
0.9	0.1

Weight of feature merging In our approach, features of the LLM and the LVLM will be merged by weighted summation. The weights determine the level of contribution of the different models towards the combined features. The weighting factors can be interpreted as the amount of information which is used of each model in consecutive steps. This can lead to conclusions about which models features are more important for a strong performance. The different configurations which are considered in this work are displayed in Table 2.

Table 2. Tested feature weights for the merging.

LLM Feature Weight p^{i}_{LLM}	LVLM Feature Weight p^{i}_{LVLM}
0.1	0.9
0.3	0.7
0.5	0.5
0.7	0.3
0.9	0.1

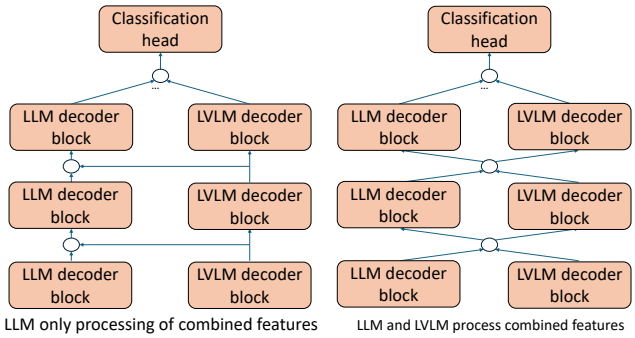

Figure 4. Two different strategies to process the combined features. On the left, the LVLM receives only its own features from the previous layers (False). The LLM processes the combined input of both models. On the right, both models receive the combined features of the previous layer as input (True).

Layers after which feature merging is applied The layer at which features are merged is crucial, as features from different transformer decoder blocks capture varying levels of abstraction. When both the LLM and the LVLM share the same output vocabulary, merging closer to the output tends to produce better alignment between the modalities. In contrast, merging features too early may impede the models' abilities to extract meaningful representations independently. In our work, features are considered for merging after each transformer decoder block. Using Qwen 2 as the base LLM, which comprises 28 transformer decoder blocks, we evaluate three different configurations: merging the features from layers 20 to 28, from layers 25 to 28, or solely from layer 28. This analysis allows us to assess the impact of merging at various depths on overall model performance.

Branches processing combined data The independent branches of the LLM and the LVLM initially process their features separately, enabling each model to extract modality-specific information aligned with its original training objectives. Replacing these distinct features with a combined representation may dilute or obscure some of this individual information. Consequently, it may be advantageous to merge the features into only one branch while allowing the other branch to continue processing its modality-specific information. In our work, we hypothesize that the LLM, given its superior capability to handle long input sequences, is better suited for reasoning about the more complex, merged features. Thus, we evaluate whether it is beneficial for the LLM to exclusively process the combined features, while the LVLM processes its visual information independently and later contributes to the combined representation that feeds into the LLM. The branch merging strategy can be seen in Figure 4.

Sum features with weight 1.0 before last layer In addition to the feature weights displayed above, it might be beneficial to sum the entire features of the two models. For the QwenLLM architecture, the normalization of feature vectors is applied in each layer after the skip connection, which leads to different magnitudes of features that are summed to the output of the self-attention compared to weights of [0.5,0.5].

We avoided adding a weight of [1.0,1.0] to the list of feature weight configurations because experiments showed it to be beneficial to weight the features after layer 28 according to the configurations in Table 2 while applying the weights of [1.0,1.0] to those before that layer. Therefore, an additional binary parameter *Sum All* is added.

3.4. Prompt design

The input prompts for the LVLM and the LLM are different. As mentioned above, LVLMs are not able to reason correctly about long and complicated textual inputs. Therefore, the prompt containing the objects in the surrounding of the ego-vehicle are only being used as input to the LLM. On the one side. LVLM receives for each task only the question and corresponding video data. On the other side, the LLM receives a prompt composed of three blocks: First, the description and structure of the object data and the general task. Second, the question for the specific task. Lastly, the prompt resulting from the concatenation of the textual information for each object in the scene. in the appendix Figure 7 the design of a prompt for a sample scene is shown.

3.5. Experiment

We compare the different methods of combining the LLM and LVLM to each other on a subset of the LingoQA evaluation set with 200 samples. Consequently, the most promising combinations will be then evaluated in the complete dataset. Inference is performed on a single H100 GPU. In the initial experiments, a random seed of 42 is set and next tokens will be selected based on maximum score, to ensure that the experiments are reproducible.

As the approach is tested with the LingoQA dataset, their proposed metric, the Lingo Judge [19] is used as main evaluation. This metric is the output of a DeBERTa-V3 language model fine-tuned using LoRA [9] to assign a score to the quality of model answers on VQA tasks on the LingoQA dataset. For that, it uses the question, the reference answer and the models answer to output a score between 0 and 1 with 1 being a perfect answer. The LingoQA dataset provides for each question two reference answers and the maximum of the two LingoQA-Judge scores determines the score per question. In addition, the ROUGE metric is also presented, to have a result which does not depend on a extra LLM. However, their focus on recall and exact matches, respectively, could not represent correctly the performance of the models on their tasks, where several answers and formulations can be correct at the same time.

Table 3. Results of isolated models with LLM and LVLM prompts.

	LLM Prompt	LVLM Prompt
LLaVA-VL 7B	**0.419**	0.408
Qwen-VL 7B	**0.419**	0.399
Qwen-LLM 7B	0.385	0.286

4. Results

As a baseline, we present the results of using only the prepared LLM or LVLM prompt for the isolated models, as seen in Table 3. For all the models, the more complete LLM prompt provides the best scores, suggesting that the LVLM models can benefit also from longer and more complex prompts. For the Qwen-LLM the case is similar, but the performance is lower than for the Visual models, indicating that the visual information is still the key for the general scene understanding. In the following paragraphs, the results for the LVLM+LLM combination approach are presented.

The best results for the three proposed models on the selected subset of the Lingo Dataset are presented in Table 4. The pure V3LMA-Q combination achieved a 60% Lingo Score, marking a nearly 5% improvement over V3LMA-L. This difference may be attributed to the modified vocabulary used in LlaVA, as discussed in Section 3. Meanwhile, V3LMA-Q-mini achieved a 49% score. Notably, despite using only 25% of the parameters, it performed just 10% below the larger Qwen model. This result highlights the efficiency of our LVLM+LLM combination in processing information while maintaining strong comprehension performance.

Next, in Table 5 we present the results of evaluating the three architectures with the best configuration parameters in the complete Lingo Evaluation dataset, together with the current state of the art (SotA) and other models in the literature. Our approach Lingo score is 0.56. It is worth mentioning that all the models in the literature were fine tuned with the training dataset of the Lingo benchmark. We attempted to improve the performance of a given architecture without the possibility of retraining the model, to see how well was the performance of an approach which cannot have access of labeled data of the target domain source. Furthermore, comparing with the results of Table 3 there is an improvement of 15% with respect to the best isolated zero shot models, showing there is actually a large benefit of using the combined information of both architectures. Moreover, comparing with the Lingo model without fine tuning, we show an improvement of 23%, which shows that our merging architecture effectively takes advantage of general

Table 4. The best Lingo Scores for each tested model with their respective configuration evaluated on 20% of the LingoQA dataset.

Model Name	Isolate LVLM Features	Head Weights	Feature Weights	Merge Layers	Sum Weight	Rouge	Lingo Mean
V3LMA-L	False	[0.5, 0.5]	[0.9, 0.1]	[25, 26, 27, -1]	True	0.66	0.55
V3LMA-Q	True	[0.9, 0.1]	[0.9, 0.1]	[25, 26, 27, -1]	True	0.68	0.60
V3LMA-Q-mini	True	[0.1, 0.9]	[0.3, 0.7]	[25, 26, 27, -1]	False	0.56	0.49

knowledge to understand specific data distributions.

Table 5. Model performance scores on the whole LingoQA Evaluation dataset using the mean of the Lingo-Judge scores.

Name	Fine-tuned	Score
LLaVA 1.5	Yes	0.51
InternVL	Yes	0.58
LingoQA	Yes	0.61
Drive-OV	Yes	**0.70**
LingoQA	No	0.33
V3LMA-Q (Ours)	No	0.56
V3LMA-L (Ours)	No	0.50
V3LMA-Q-mini (Ours)	No	0.45

4.1. Ablation study

In our search for the optimal parameters, we evaluated all possible combinations presented in Section 3.3. To do this, we aggregated the results for each specific parameter configuration and computed the mean metrics. This approach allowed us to isolate the effects of each parameter under evaluation.

For the head weights, the results are shown in Table 6. For the bigger models, the best performance is obtained when using a balanced weighting between the LLM's and LVLM's head features. Nevertheless, the drop in performance is greater when the LLM features are weighted above the LVLM ones. On the other side, for the smaller Qwen combination the results indicate that the compressed visual features have a greater impact than the textual ones. In this case, for a heavier weighted LVLM features the difference is 21% with respect to heavier LLM features.

Results for feature weighting are shown in Table 7. It can be seen that the story is similar as for the head weights: The best performance for larger models is achieved with a balanced weighting of the features. However, in this case the influence of the feature combinations is smaller, with only 6% between the best and worse case and also decreases simetrically with respect to both LVLM and LLM weighting. Similarly to the head weights as well, for V3LMA-Q-mini a stronger priority for the LVLM weights resulted in better performance, which supports the idea that for smaller models the visual features contribute more for the performance.

Regarding the feature layers which should be merged, the results are similar for the three tested models. The best performance was found by combining the last four layers of the LVLM and LLM, followed by combining only the last layer before the classification head, and in last place combining the last eight layers. In general combining too many earlier layers can hurt the performance, because those layers are not abstract enough so that the features are well aligned. For V3LMA-Q-mini, the difference is not as pronounced as for the larger models, which could mean that the abstraction capability difference between early and late features is not significant enough to be an issue for the merging.

In Table 9 can be seen that it is better to process the combined features through the LLM (Figure 4 left) and input the

Table 6. Lingo Score in the validation set for different Head weights combinations

Head weights Model	[0.1, 0.9]	[0.3, 0.7]	[0.5, 0.5]	[0.7, 0.3]	[0.9, 0.1]
V3LMA-L	0.40	0.38	0.41	0.39	0.33
V3LMA-Q	0.38	0.40	0.42	0.40	0.38
V3LMA-Q-mini	0.36	0.30	0.22	0.18	0.15

Table 7. Lingo Score in the validation for different feature weights combinations

Feature weights Model	[0.1, 0.9]	[0.3, 0.7]	[0.5, 0.5]	[0.7, 0.3]	[0.9, 0.1]
V3LMA-L	0.36	0.40	0.41	0.39	0.36
V3LMA-Q	0.38	0.40	0.41	0.40	0.38
V3LMA-Q-mini	0.33	0.24	0.24	0.22	0.22

Table 8. Lingo Score for different sets of layers's output that where fused

Layers 2 merge Model	28	[20:28]	[25:28]
V3LMA-L	0.38	0.33	0.44
V3LMA-Q	0.39	0.35	0.44
V3LMA-Q-mini	0.26	0.25	0.28

Table 9. Lingo Score for whether only the LLM process the features (False), or both LLM and LVLM process them (True)

Isolate LVLM Features Model	False	True
V3LMA-L	0.36	0.40
V3LMA-Q	0.36	0.44
V3LMA-Q-mini	0.21	0.30

Table 10. Lingo Score for whether all the layers are summed with weight [1,1] but the last one or not

Sum all Model	False	True
V3LMA-L	0.40	0.37
V3LMA-Q	0.40	0.39
V3LMA-Q-mini	0.30	0.22

LVLM layers only with the previous visual features (Figure 4 right). It is then better to not overload the LVLM with more text information than it can handle, with similar behavior for the three tested models.

The last parameter to test, is if summing all the layers with weights [1,1] improves the performance of the models. According to the results in Table 10, is not the case, but the difference is less than 1% for the bigger models, which could indicate that it may be also statistical noise in the measurement. For V3LMA-Q-mini, the difference of 8% shows a clear benefit of not combining the features with a [1,1] weighting vector.

5. Discussion

The most notable insight from our results is the apparent inconsistency between the optimal parameters of the best-performing models and the trends observed in the ablation study. For instance, for both head weights and feature weights we found in the ablation studies that a balanced weighting for LLM and LVLM should yield the best results. However, in Table 4, the top-performing models consistently favor an unbalanced weighting toward the LLM head. Moreover, in two of these models, the unbalanced setting for feature weights also outperforms the balanced configuration. A similar contradiction is observed for the "Sum All" parameter. We attribute these inconsistencies to cross-correlations between parameters, meaning that the impact of one parameter is influenced by the values of others. Such interdependencies are not captured when analyzing ablation results in isolation, as these results are marginalized over all other settings.

In contrast, for the remaining parameters, the findings are more conclusive. For the "Isolate LVLM features" parameter, we demonstrate that it is beneficial to process both LLM and LVLM features solely within the LLM decoder blocks while isolating the LVLM decoder blocks to handle only visual features. This configuration leverages the LLM's superior ability to process complex prompts enriched with 3D information, while allowing the LVLM to focus on semantic visual understanding derived from camera input, thereby complementing theLLM's reasoning with perceptual data. Furthermore, we prove that late fusion of the features enhances the alignment between the LLM and LVLM's encoder output, improving the synergy between text and visual features. Fusing only the final layer proves insufficient, indicating that the last four layers contain valuable information for the general understanding task.

Although our primary objective was to develop a zero-shot approach that eliminates the need for fine-tuning, future work could explore how retraining on the target data distribution may further enhance performance. Moreover, a deeper investigation into the cross-dependencies of fusion parameters could provide valuable insights into how specific settings influence one another—potentially enabling a more systematic and optimized fusion strategy for combining LLMs and LVLMs.

Finally, while our evaluation focused on the LingoQA dataset, our method is designed without any dataset-specific training, suggesting potential generalization capabilities. Nonetheless, validating this assumption through quantitative evaluation on other VQA datasets would be essential. Unfortunately, as detailed in Appendix Table 11, no other publicly available datasets currently offer the necessary characteristics to allow for a direct comparison.

6. Conclusion

We introduce a novel zero-shot multimodal approach for 3D scene understanding in traffic scenarios. Our method enhances the reasoning capabilities of a LLM by integrating 3D information distilled from object detectors. This enriched spatial data is fused with visual features extracted from a LVLM, enabling the model to simultaneously interpret the semantic content of images and the spatial structure of the 3D environment.

We further examine various feature fusion configurations, providing practical insights into optimizing the interplay between LLMs and LVLMs for improved scene comprehension. To the best of our knowledge, our approach achieves the highest LingoQA score for zero-shot multimodal scene understanding. Importantly, our framework generalizes well and delivers robust performance without the need for fine-tuning on the target domain, making it an adaptable solution for a wide range of real-world applications.

References

[1] Hidehisa Arai, Keita Miwa, Kento Sasaki, Yu Yamaguchi, Kohei Watanabe, Shunsuke Aoki, and Issei Yamamoto. Covla: Comprehensive vision-language-action dataset for autonomous driving, 2024. 2

[2] Xu Cao, Tong Zhou, Yunsheng Ma, Wenqian Ye, Can Cui, Kun Tang, Zhipeng Cao, Kaizhao Liang, Ziran Wang, James Rehg, and Chao Zheng. Maplm: A real-world large-scale vision-language dataset for map and traffic scene understanding. https://github.com/LLVM-AD/MAPLM, 2023. 2

[3] Tushar Choudhary, Vikrant Dewangan, Shivam Chandhok, Shubham Priyadarshan, Anushka Jain, Arun K. Singh, Siddharth Srivastava, Krishna Murthy Jatavallabhula, and K. Madhava Krishna. Talk2bev: Language-enhanced bird's-eye view maps for autonomous driving, 2023. 2

[4] Xiangxiang Chu, Limeng Qiao, Xinyu Zhang, Shuang Xu, Fei Wei, Yang Yang, Xiaofei Sun, Yiming Hu, Xinyang Lin, Bo Zhang, and Chunhua Shen. Mobilevlm v2: Faster and stronger baseline for vision language model, 2024. 2

[5] Daniel Dauner, Marcel Hallgarten, Andreas Geiger, and Kashyap Chitta. Parting with misconceptions about learning-based vehicle motion planning, 2023. 3

[6] Thierry Deruyttere, Simon Vandenhende, Dusan Grujicic, Luc Van Gool, and Marie-Francine Moens. Talk2car: Taking control of your self-driving car. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics, 2019. 2

[7] Ziyu Guo, Renrui Zhang, Xiangyang Zhu, Yiwen Tang, Xianzheng Ma, Jiaming Han, Kexin Chen, Peng Gao, Xianzhi Li, Hongsheng Li, and Pheng-Ann Heng. Point-bind and point-llm: Aligning point cloud with multi-modality for 3d understanding, generation, and instruction following, 2023. 2

[8] Iryna Hartsock and Ghulam Rasool. Vision-language models for medical report generation and visual question answering: a review. *Frontiers in Artificial Intelligence*, 7, 2024. 1

[9] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021. 6

[10] Zhijian Huang, Chengjian Feng, Feng Yan, Baihui Xiao, Zequn Jie, Yujie Zhong, Xiaodan Liang, and Lin Ma. Drivemm: All-in-one large multimodal model for autonomous driving, 2024. 2

[11] Yuichi Inoue, Yuki Yada, Kotaro Tanahashi, and Yu Yamaguchi. Nuscenes-mqa: Integrated evaluation of captions and qa for autonomous driving datasets using markup annotations, 2023. 2

[12] Sandesh Jain, Surendrabikram Thapa, Kuan-Ting Chen, A. Lynn Abbott, and Abhijit Sarkar. Semantic understanding of traffic scenes with large vision language models. In *2024 IEEE Intelligent Vehicles Symposium (IV)*, pages 1580–1587, 2024. 2

[13] Glenn Jocher, Alex Stoken, Jirka Borovec, NanoCode012, ChristopherSTAN, Liu Changyu, Laughing, tkianai, Adam Hogan, lorenzomammana, yxNONG, AlexWang1900, Laurentiu Diaconu, Marc, wanghaoyang0106, ml5ah, Doug, Francisco Ingham, Frederik, Guilhen, Hatovix, Jake Poznanski, Jiacong Fang, Lijun Yu, changyu98, Mingyu Wang, Naman Gupta, Osama Akhtar, PetrDvoracek, and Prashant Rai. ultralytics/yolov5: v3.1 - bug fixes and performance improvements, 2020. 4, 1

[14] Jinkyu Kim, Anna Rohrbach, Trevor Darrell, John Canny, and Zeynep Akata. Textual explanations for self-driving vehicles, 2018. 2

[15] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning, 2023. 1

[16] Yuhang Lu, Yichen Yao, Jiadong Tu, Jiangnan Shao, Yuexin Ma, and Xinge Zhu. Can lvlms obtain a driver's license? a benchmark towards reliable agi for autonomous driving, 2024. 2

[17] Yunsheng Ma, Amr Abdelraouf, Rohit Gupta, Ziran Wang, and Kyungtae Han. Video token sparsification for efficient multimodal llms in autonomous driving, 2024. 2

[18] Srikanth Malla, Chiho Choi, Isht Dwivedi, Joon Hee Choi, and Jiachen Li. Drama: Joint risk localization and captioning in driving, 2022. 2

[19] Ana-Maria Marcu, Long Chen, Jan Hünermann, Alice Karnsund, Benoit Hanotte, Prajwal Chidananda, Saurabh Nair, Vijay Badrinarayanan, Alex Kendall, Jamie Shotton, Elahe Arani, and Oleg Sinavski. Lingoqa: Visual question answering for autonomous driving, 2024. 2, 6

[20] Michael Moor, Qian Huang, Shirley Wu, Michihiro Yasunaga, Cyril Zakka, Yash Dalmia, Eduardo Pontes Reis, Pranav Rajpurkar, and Jure Leskovec. Med-flamingo: a multimodal medical few-shot learner, 2023. 1

[21] Ming Nie, Renyuan Peng, Chunwei Wang, Xinyue Cai, Jianhua Han, Hang Xu, and Li Zhang. Reason2drive: Towards interpretable and chain-based reasoning for autonomous driving, 2024. 2

[22] OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, and Ilge Akkaya and. Gpt-4 technical report, 2024. 1

[23] SungYeon Park, MinJae Lee, JiHyuk Kang, Hahyeon Choi, Yoonah Park, Juhwan Cho, Adam Lee, and DongKyu Kim. Vlaad: Vision and language assistant for autonomous driving. In *2024 IEEE/CVF Winter Conference on Applications of Computer Vision Workshops (WACVW)*, pages 980–987, 2024. 2

[24] 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, 2024. 2

[25] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer, 2020. 4, 1

[26] Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kunchang Li, He Cao, Jiayu Chen, Xinyu Huang, Yukang Chen, Feng Yan, Zhaoyang Zeng, Hao Zhang, Feng Li, Jie Yang,

Hongyang Li, Qing Jiang, and Lei Zhang. Grounded sam: Assembling open-world models for diverse visual tasks, 2024. 4, 1

[27] Enna Sachdeva, Nakul Agarwal, Suhas Chundi, Sean Roelofs, Jiachen Li, Mykel Kochenderfer, Chiho Choi, and Behzad Dariush. Rank2tell: A multimodal driving dataset for joint importance ranking and reasoning, 2023. 2

[28] 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, 2023. 2

[29] S P Sharan, Francesco Pittaluga, Vijay Kumar B G, and Manmohan Chandraker. Llm-assist: Enhancing closed-loop planning with language-based reasoning, 2023. 3

[30] Chonghao Sima, Katrin Renz, Kashyap Chitta, Li Chen, Hanxue Zhang, Chengen Xie, Jens Beißwenger, Ping Luo, Andreas Geiger, and Hongyang Li. Drivelm: Driving with graph visual question answering, 2025. 3, 2

[31] Johannes Stallkamp, Marc Schlipsing, Jan Salmen, and Christian Igel. The german traffic sign recognition benchmark: A multi-class classification competition. In *The 2011 International Joint Conference on Neural Networks*, pages 1453–1460, 2011. 1

[32] SungYeon Park, MinJae Lee, JiHyuk Kang, Hahyeon Choi, Yoonah Park, Juhwan Cho, Adam Lee, and DongKyu Kim. Vlaad: Vision and language assistant for autonomous driving. 2

[33] Xiaoyu Tian, Junru Gu, Bailin Li, Yicheng Liu, Yang Wang, Zhiyong Zhao, Kun Zhan, Peng Jia, Xianpeng Lang, and Hang Zhao. Drivevlm: The convergence of autonomous driving and large vision-language models, 2024. 1, 2, 3

[34] Shihao Wang, Zhiding Yu, Xiaohui Jiang, Shiyi Lan, Min Shi, Nadine Chang, Jan Kautz, Ying Li, and Jose M. Alvarez. Omnidrive: A holistic llm-agent framework for autonomous driving with 3d perception, reasoning and planning, 2024. 2

[35] Dongming Wu, Wencheng Han, Tiancai Wang, Xingping Dong, Xiangyu Zhang, and Jianbing Shen. Referring multi-object tracking, 2023. 2

[36] Dongming Wu, Wencheng Han, Tiancai Wang, Yingfei Liu, Xiangyu Zhang, and Jianbing Shen. Language prompt for autonomous driving, 2023. 2

[37] Li Xu, He Huang, and Jun Liu. Sutd-trafficqa: A question answering benchmark and an efficient network for video reasoning over traffic events, 2021. 2

[38] Runsen Xu, Xiaolong Wang, Tai Wang, Yilun Chen, Jiangmiao Pang, and Dahua Lin. Pointllm: Empowering large language models to understand point clouds. 2

[39] Songyan Zhang, Wenhui Huang, Zihui Gao, Hao Chen, and Chen Lv. Wisead: Knowledge augmented end-to-end autonomous driving with vision-language model, 2024. 2

[40] Sheng Zhang, Yanbo Xu, Naoto Usuyama, Hanwen Xu, Jaspreet Bagga, Robert Tinn, Sam Preston, Rajesh Rao, Mu Wei, Naveen Valluri, Cliff Wong, Andrea Tupini, Yu Wang, Matt Mazzola, Swadheen Shukla, Lars Liden, Jianfeng Gao, Angela Crabtree, Brian Piening, Carlo Bifulco, Matthew P. Lungren, Tristan Naumann, Sheng Wang, and Hoifung Poon. Biomedclip: a multimodal biomedical foundation model pretrained from fifteen million scientific image-text pairs, 2025. 1

[41] Peiru Zheng, Yun Zhao, Zhan Gong, Hong Zhu, and Shaohua Wu. Simplellm4ad: An end-to-end vision-language model with graph visual question answering for autonomous driving, 2024. 3

[42] Hao Zhou, Zhanning Gao, Maosheng Ye, Zhili Chen, Qifeng Chen, Tongyi Cao, and Honggang Qi. Hints of prompt: Enhancing visual representation for multimodal llms in autonomous driving, 2024. 2

[43] Xingcheng Zhou, Konstantinos Larintzakis, Hao Guo, Walter Zimmer, Mingyu Liu, Hu Cao, Jiajie Zhang, Venkatnarayanan Lakshminarasimhan, Leah Strand, and Alois C. Knoll. Tumtraffic-videoqa: A benchmark for unified spatio-temporal video understanding in traffic scenes, 2025. 2