LLMI3D: MLLM-Based 3D Perception From a Single 2D Image

Fan Yang, *Graduate Student Member, IEEE*, Sicheng Zhao, *Senior Member, IEEE*, Yanhao Zhang, Hui Chen, Haonan Lu, Jungong Han, *Senior Member, IEEE*, and Guiguang Ding, *Senior Member, IEEE*

***Abstract*—Recent advancements in autonomous driving, augmented reality, robotics, and embodied intelligence have necessitated 3D perception algorithms. However, current 3D perception methods, especially specialized small models, exhibit poor generalization in open scenarios. On the other hand, multimodal large language models (MLLMs) excel in general capacity but underperform in 3D tasks, due to weak 3D local spatial object perception, poor text-based geometric numerical output, and inability to handle camera focal variations. To address these challenges, we develop LLMI3D, and propose the following solutions: Spatial-Enhanced Local Feature Mining for better 3D spatial feature extraction, 3D Query Token-Derived Info Decoding for precise geometric regression, and Geometry Projection-Based 3D Reasoning for handling camera focal length variations. We are the first to adapt an MLLM for image-based 3D perception. Additionally, we have constructed the IG3D dataset, which provides fine-grained descriptions and question-answer annotations. Extensive experiments demonstrate that our LLMI3D achieves state-of-the-art performance, outperforming other methods by a large margin.**

***Index Terms*—Multimodal large language model, 3D perception.**

I. Introduction

With the rapid development of deep learning, 2D perception tasks such as object detection, instance segmentation, and visual grounding have achieved remarkable progress [1], [2], [3], [4]. However, the real world is three-dimensional, and many practical applications, such as autonomous driving, robotics, augmented reality, and embodied intelligence, demand enhanced spatial perception. Traditional 2D methods can no longer meet these demands. Therefore, researchers have introduced the concept of 3D perception, which involves inferring the location, dimension, and pose of objects in 3D space to achieve accurate predictions of their spatial positions [5], [6], [7], [8], [9].

Many 3D perception techniques utilize LiDAR point clouds [10] or camera images. LiDAR offers excellent depth prediction but is expensive and complex. Despite lower accuracy compared to LiDAR-based methods, image-based approaches [11] are more affordable and easily integrable, making them widely used in various scenarios.

In recent years, many specialized models have been developed for image-based 3D perception. However, they face several limitations: 1. Single-modal 3D detection models lack the ability to precisely locate a specified interested object based on textual input, limiting their effectiveness in following user instructions. 2. These models possess limited logical reasoning and question-answering capabilities due to insufficient real-world knowledge and common sense. 3. They demonstrate weak generalization in open and cross-domain settings, being restricted to predefined categories and scenes within the training dataset.

Recent multimodal large language models (MLLMs) [15], [16], [17], [18], [19] demonstrate significant strength in general tasks. These pre-trained models effectively address the limitations of specialized small models with their strengths in instruction following, logical reasoning, and cross-domain tasks. However, vanilla MLLMs face issues in specific 3D perception tasks, hindering direct application.

1) *Weak 3D local spatial object perception:* MLLMs like GPT-4o [17] are primarily trained on 2D data, limiting their ability to capture 3D spatial structures. This results in poor 3D perception performance (Fig. 2(a)). Furthermore, these models struggle with capturing fine-grained details of distant and small objects, which is crucial for autonomous driving and other scenarios.
2) *Poor text-based geometric numerical output:* Existing MLLMs output numerical results in textual form (Fig. 2(b)), unsuitable for 3D values (e.g., X, Y, Z, dimensions, rotation), leading to poor precision, slow processing, and parsing difficulties. Outputting structured numerical text is error-prone, often resulting in misordering or misformatting.
3) *Inability to handle camera focal variations:* Inferring depth from 2D images is under-constrained and disturbed by camera focal length variations. As illustrated

Received 14 February 2025; revised 27 June 2025 and 4 October 2025; accepted 3 November 2025. Date of publication 16 January 2026; date of current version 22 May 2026. This work was supported in part by the National Natural Science Foundation of China under Grant 62525103, Grant 62441235, Grant 62021002, and Grant 62571294, in part by Beijing Natural Science Foundation under Grant L252009, in part by OPPO Research Fund, and in part by CCF-DiDi GAIA Collaborative Research Funds. The associate editor coordinating the review of this article and approving it for publication was Dr. Xinxin Zuo. *(Corresponding authors: Sicheng Zhao; Guiguang Ding.)*

Fan Yang and Guiguang Ding are with the School of Software, Tsinghua University, Beijing 100084, China, and also with the BNRist, Tsinghua University, Beijing 100084, China (e-mail: yfthu@outlook.com; dinggg@tsinghua.edu.cn).

Sicheng Zhao and Hui Chen are with BNRist, Tsinghua University, Beijing 100084, China (e-mail: schzhao@gmail.com; jichenhui2012@gmail.com).

Yanhao Zhang and Haonan Lu are with OPPO AI Center, Beijing 100026, China (e-mail: zhangyanhao@oppo.com; luhaonan@oppo.com).

Jungong Han is with the Department of Automation, Tsinghua University, Beijing 100084, China (e-mail: jungonghan77@gmail.com).

Digital Object Identifier 10.1109/TMM.2026.3654407

Fig. 1. Our LLMI3D endows MLLMs with 3D perception capabilities. When provided with a question or description, our LLMI3D can return the object of interest and its 3D bounding box (bbox) in 3D space. Across various datasets, our LLMI3D significantly outperforms existing methods by a large margin.

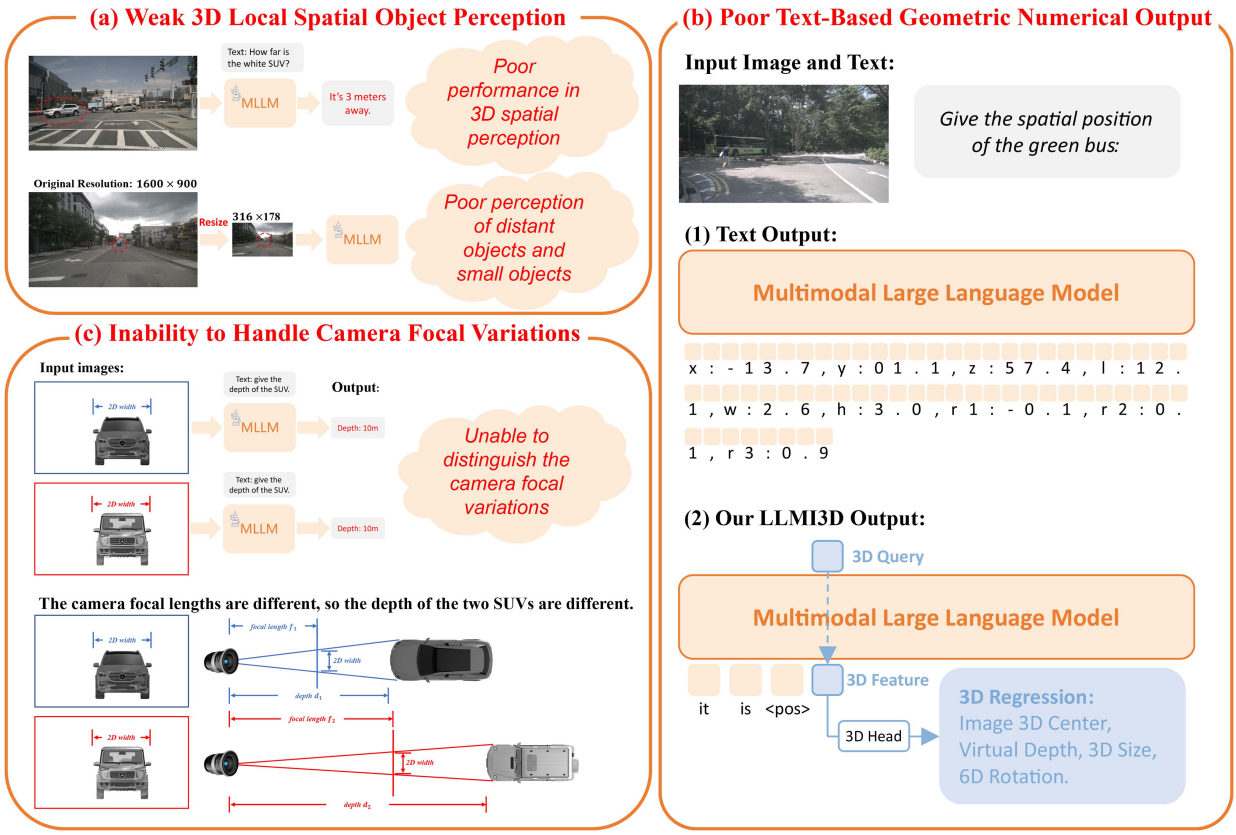

Fig. 2. Three issues of vanilla MLLMs in 3D perception tasks: (a) Weak 3D local spatial object perception: vanilla MLLMs struggle with accurate 3D object localization due to poor spatial understanding, especially for distant or small objects. (b) Poor text-based geometric numerical output: Current models output 3D values in text, which is slow and error-prone. Our approach utilizes a learnable 3D Query token with 3D heads to regress geometric values, improving accuracy significantly. (c) Inability to handle camera focal variations: Distinguishing changes in camera focal length from a single 2D image is hard. This leads to incorrect depth predictions for similarly sized objects captured at different focal lengths.

in Fig. 2(c), when two objects have the same 2D and 3D sizes, neural networks tend to predict the same 3D location. However, these two images were captured by cameras with different focal lengths, leading to substantial differences in the actual spatial positions.

To overcome the issues of specialized small models and MLLMs, we propose **LLMI3D**, an **M**LLM-based **I**mage **3D** perception model. By fine-tuning a pre-trained MLLM with LoRA [20] and proposing a 3D-friendly structure in the image encoder and token decoder, we overcome the limitations of vanilla MLLM, achieving robust 3D perception capabilities.

In the image encoder, to address MLLMs' weak 3D local spatial object perception, we introduce Spatial-Enhanced Local Feature Mining to extract local 3D image features while reducing the token count. For an input image, we employ a dual-branch approach consisting of a ViT branch and a CNN

TABLE I
EXISTING SPECIALIZED SMALL MODELS AND MULTIMODAL LARGE LANGUAGE MODELS HAVE VARIOUS LIMITATIONS

Methods	LLM	Instruction Understanding	Logical Reasoning	Question Answering	Open Vocabulary	Local 3D Feature Extracting	Focal Length Variation Handling	3D Box Outputting
MonoDETR [12]	✗	✗	✗	✗	✗	✓	✗	✓
Omni3D [13]	✗	✗	✗	✗	✗	✓	✓	✓
Mono3DVG [14]	✗	✓	✗	✗	✗	✓	✗	✓
DeepSeek-VL2 [15]	✓	✓	✓	✓	✓	✗	✗	✗
InternVL2.5 [16]	✓	✓	✓	✓	✓	✗	✗	✗
LLMI3D (Ours)	✓	✓	✓	✓	✓	✓	✓	✓

Only our approach, LLMI3D, exhibits comprehensive and robust 3D perception capabilities.

branch. The ViT branch processes low-resolution images, while the CNN branch handles high-resolution images. We utilize the reduced number of tokens extracted by the ViT as queries, and the high-resolution image features extracted by the CNN as keys and values in a cross-attention mechanism. This approach allows for fine-grained local feature extraction from the high-resolution images in the CNN branch, thus enhancing the ability to capture features of distant small objects. Additionally, we introduce a depth branch within the CNN, enabling it to predict depth maps. This integration of depth information into the CNN branch strengthens the spatial feature extraction capabilities of the Vision Encoder. Spatial-enhanced cross-branch attention is then employed to integrate local and global 3D features, further minimizing the token count.

In the LLM, to overcome LLM's poor text-based geometric numerical output, we propose the 3D Query Token-Derived Info Decoding method. This approach employs a learnable 3D Query Token along with 3D heads to regress 3D attributes. The input to the specialized 3D Query Token is not the output of the preceding token. Instead, it consists of learnable parameters with the same feature dimension as the hidden layers of the LLM. Through training and learning, these learnable parameters encode queries about 3D geometric clues. We then use the final hidden state of this specialized token as the 3D feature to predict 3D values, as illustrated in Fig. 2(b). Furthermore, the objective of our 3D regression is not trivial. Unlike methods that predict common 2.5D values like real depth or rotation, our approach predicts virtual depth and 6D rotation. These 2.5D values are highly beneficial for LLMs and can significantly enhance 3D perception.

For 3D box outputting, to address MLLMs' inability to handle variations in camera focal length, we introduce Geometry Projection-Based 3D Reasoning, integrating camera parameters into geometric projection. We do not solely rely on focal length-invisible neural networks. Instead, we combine uninterpretable networks with interpretable projection methods. Specifically, we utilize virtual depth instead of actual depth. Virtual depth is conceptualized assuming that all images are captured by a standardized virtual camera, independent of actual camera focal lengths. By first predicting an object's virtual depth and subsequently using the projection formula to convert it to the real depth, the depth precision can be significantly improved.

Our method empowers LLMs with the ability to perform image-based 3D spatial perception. Given an input image, our model can answer questions posed by the user as well as provide the spatial location of objects of interest. The experimental results of our method show superior accuracy compared to other approaches, as demonstrated in Fig. 1. A functional comparison of our method with existing approaches is presented in Table I.

Furthermore, an appropriate dataset is crucial for fine-tuning MLLMs. Existing image-based 3D perception datasets focus on object detection and lack fine-grained caption and question-answer data. The Mono3DRefer [14] dataset also has significant issues. It includes 3D perception results in object caption inputs, which undermines the evaluation of real 3D perception capabilities.

Therefore, we further developed **IG3D**: an **I**mage-based **3D** **G**rounding dataset. The IG3D dataset provides precise descriptions of objects within images, including detailed appearance and location information. This enables the 3D grounding task to be effectively performed. Furthermore, our IG3D dataset includes annotations for Visual Question Answering (VQA) instructions, facilitating the assessment of a model's world knowledge, common sense and logical reasoning capabilities.

In summary, our contributions are as follows:

1) We are the first to adapt an MLLM for image-based 3D perception, overcoming specialized models' limitations in instruction following, logical reasoning, and open scenario generalization. Furthermore, we identify and resolve three key issues of vanilla MLLMs for 3D perception.
2) To address the issue of weak 3D local spatial object perception, we propose a spatial-enhanced local feature mining approach. This method integrates features extracted by ViT, CNN, and depth predictor while employing the spatial-enhanced cross-branch attention to effectively capture local spatial features of objects.
3) To overcome the problem of poor text-based geometric numerical output, we propose 3D Query token-derived info decoding, which uses a single learnable 3D Query to efficiently extract 3D features within the LLM and regress the 3D values accurately.
4) To mitigate the inability of MLLMs to handle camera focal variations, we propose geometry projection-based 3D reasoning, which combines uninterpretable neural networks with interpretable projection methods. By integrating camera parameters, we reduce the significant impact of varying camera focal lengths on 3D perception.
5) We construct the IG3D, an image-based 3D perception dataset to effectively assess a model's 3D grounding and 3D VQA capabilities. Extensive experiments demonstrate that our approach achieves state-of-the-art performance on various datasets, surpassing other methods by a large margin.

II. Related Works

A. Multimodal Large Language Models

In 2023, OpenAI released GPT-4V [17], showcasing powerful vision-text multimodal capabilities. In May 2024, OpenAI introduced GPT-4o, a model capable of processing and generating any combination of text, audio, and image inputs. Some open-source multimodal models [15], [21], [22], [23] have advanced the development of the field. LLaVA [24] generated multimodal instruction data using GPT-4, fine-tuned with CLIP [25] and LLaMA [26]. VisionLLM [27] utilized large language models for visual tasks like detection and segmentation, using BERT [28] and Deformable DETR [3]. MiniGPT-v2 [29] efficiently mapped image features into language space. CogVLM [23] employed deep fusion for enhanced visual integration. PerceptionGPT [30] used single-token encoding for perceptual data, but limited to 2D.

More advanced and powerful multimodal large models [31] have also emerged. DeepSeek-VL2 [15] is an advanced mixture of experts (MoE) model that utilizes dynamic tiling strategies and a MOE language model with multi-head latent attention, enhancing the processing efficiency of high-resolution visual inputs and textual data. InternVL2.5 [16], building on the InternVL 2.0 architecture, employs a progressive scaling strategy, initially training the visual encoder with a smaller language model, and then leveraging weight-sharing to transfer this to a larger model, thereby reducing computational demands. Qwen2.5-VL [18] could understand lengthy videos and identify events at specific timestamps.

These models excel in 2D tasks but lack 3D data training, resulting in weak 3D spatial perception capabilities.

Recently, the works CubeLLM [32] and EMMA [33] have explored image-based 3D perception. They output geometric coordinates in text format, which affects accuracy and speed. Both models overlook the impact of camera focal length on 3D perception, which can degrade performance in varying focal lengths. Furthermore, CubeLLM involved substantial resources for 2D and 3D alignment, utilizing 9.6 million images and 40.9 million dialogues. The training required 64 A100 GPUs. Similarly, EMMA also employed extensive resources. In contrast, our model, LLMI3D, achieves a 3D-friendly MLLM structure using LoRA [20], requiring only two A100 GPUs. This parameter-efficient fine-tuning approach significantly reduces costs and enhances flexibility.

B. 3D Perception From a Single Image

Many studies have been conducted on 3D perception from a single image. Deep3DBox [8] introduced an approach combining 2D object detection with 3D pose estimation, marking a pivotal shift towards more accurate 3D localization. This method leveraged both appearance and geometric cues, setting the foundation for subsequent improvements. Further advancements were embodied in MonoGRNet [9] employing graph-based reasoning to capture spatial relationships within scenes, significantly enhancing the precision of 3D bounding box estimations. End-to-end learning frameworks have also shown great promise. Li et al. presented RTM3D [34], a unified model integrating detection and localization stages, yielding robust performance through sophisticated feature extraction and attention mechanisms.

In recent years, image-based 3D perception has made significant progress [12], [35], [36], [37]. MonoRCNN [38] incorporated geometric information between 2D bounding box heights and 3D object heights to estimate depth. GUPNet [39] estimated object depth through the projection of 2D and 3D heights and employed an uncertainty loss for precise scoring. Gpro3D [40] significantly enhanced the accuracy of object depth and spatial predictions by leveraging ground plane priors. In 2024, Mono3DVG [14] was introduced for 3D grounding, using BERT [28] and CNN [41] for feature extraction. However, the model can process only direct descriptions, lacking logical reasoning, common sense, and the ability to generalize to open scenarios. The Mono3DRefer dataset [14] involves 3D perception results in descriptive inputs, undermining the evaluation of image-based 3D perception.

III. Methodology

A. Overview

The architecture of our method is illustrated in Fig. 3. We fine-tune a pre-trained MLLM to empower it with image-based 3D grounding capabilities. In the image encoder, to address the weak 3D local spatial object perception issues of MLLMs, we introduce Spatial-Enhanced Local Feature Mining, enhancing the image encoder's ability to extract local 3D features and reducing the token count, as detailed in Section III-B.

In the LLM, to overcome poor text-based geometric numerical output, we propose 3D Query Token-Derived Info Decoding. This method addresses the challenges associated with slow speed, low accuracy, and parsing difficulties encountered when outputting 3D coordinates in text format. By using a single learnable 3D Query token combined with 3D heads regression, we can accurately regress the 3D attributes of objects, as elaborated in Section III-C.

To obtain the 3D bounding box (bbox) and address MLLMs' inability to handle variations in camera focal lengths, we introduce geometry projection-based 3D Reasoning. Rather than relying solely on focal length-invisible 3D reasoning methods, we appropriately utilize camera intrinsic parameters. By combining neural networks with geometric projection, our method effectively mitigates the significant errors in 3D perception caused by different camera intrinsic parameters, as detailed in Section III-D.

Additionally, in Section III-E, we introduce the IG3D dataset. The IG3D dataset provides precise descriptions of objects in images, including detailed appearance and location, facilitating the 3D grounding task. Moreover, our IG3D dataset includes annotations for Visual Question Answering (VQA) instructions, assessing the model's common sense and logical reasoningxbrk capabilities.

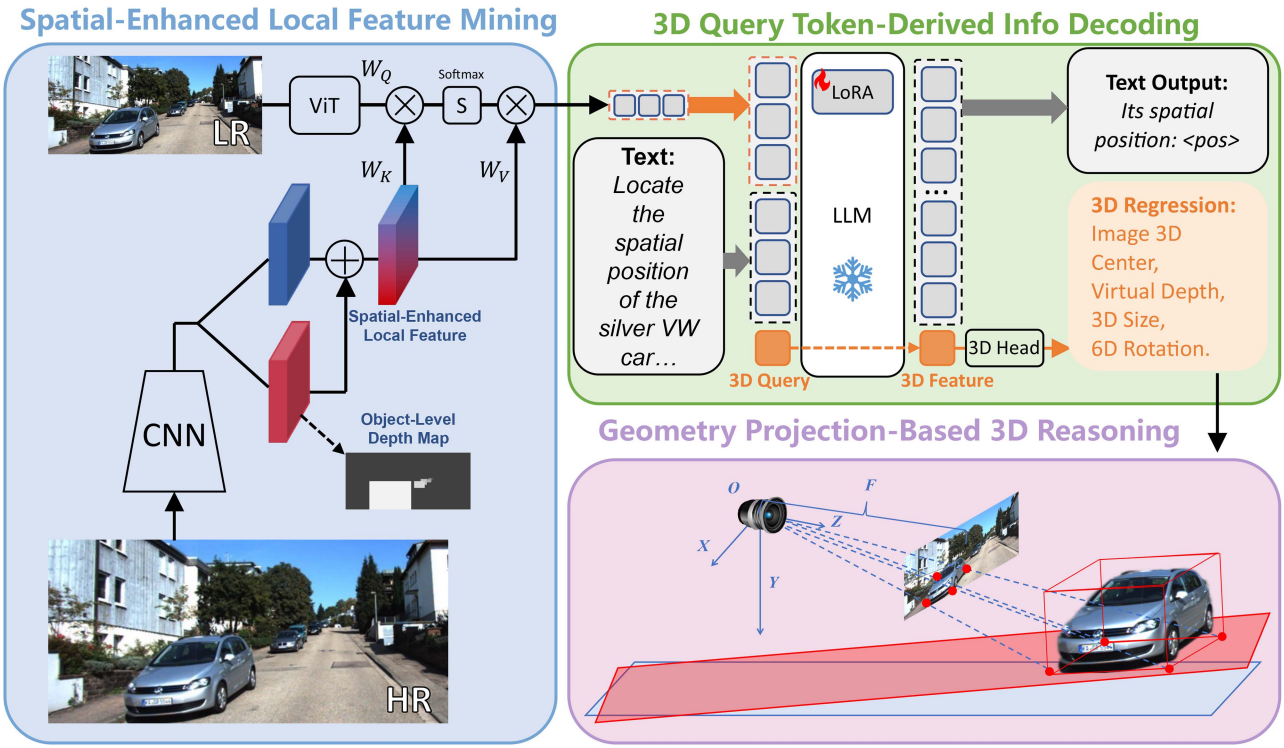

Fig. 3. Framework of the LLMI3D. (1) The image encoder utilizes Spatial-Enhanced Local Feature Mining, employing a CNN and depth predictor to extract local spatial enhanced features from high-resolution (HR) images. A ViT extracts global features with fewer tokens from low-resolution (LR) images, while spatial-enhanced cross-branch attention efficiently retrieves object spatial features and reduces the token count. (2) In the LLM, we propose 3D Query Token-Derived Info Decoding. We utilize a learnable 3D Query token to extract 3D features and employ 3D heads to regress the geometric attributes precisely. (3) To derive the 3D box of the object, we introduce Geometry Projection-Based 3D Reasoning. Rather than using focal length-invisible methods, we combine the network and geometric projection for 3D spatial reasoning, alleviating the errors introduced by varying camera focal lengths in 3D perception.

B. Spatial-Enhanced Local Feature Mining

Image encoders are crucial in multimodal large language models (MLLMs), but face challenges with spatial and local recognition: (1) MLLMs excel in semantics from 2D data but lack spatial and geometric precision needed for 3D tasks. (2) Encoders often miss local details [42], affecting critical tasks like autonomous driving.

On the other hand, inputting high-resolution images directly into encoders is impractical due to excessive tokens exceeding LLM limits and significantly hindering inference speed.

We propose the Spatial-Enhanced Local Feature Mining algorithm. CNNs and depth predictors enhance local feature extraction in high-resolution images. ViT is used in low-resolution to extract global scene information with fewer tokens. Finally, spatial-enhanced cross-branch attention integrates local and global features while reducing token count.

Unlike the self-attention [43] mechanism in ViT [44], convolutional layers excel at extracting local features and improving small object identification [45]. Local-enhanced features F_{local} from CNN [46] are split into two branches: the image RGB feature branch and the spatial depth feature branch. This yields spatial features F_{spatial} and local RGB features F_{rgb}:

$$F_{\text{spatial}} = Conv_{\text{spatial}}(F_{\text{local}}) \quad F_{\text{rgb}} = Conv_{\text{rgb}}(F_{\text{local}}) \tag{1}$$

We then predict the object-level depth map [47] using spatial features:

$$M_{\text{depth}} = Conv_{\text{depth}}(F_{\text{spatial}}) \tag{2}$$

The spatial depth branch extracts object-level depth features and strengthens the local feature extraction. We use L1 loss and object-level depth ground truth [12] for supervising the depth map.

For depth map annotation, we extract all objects' depth from the 3D detection dataset and draw the depth result onto the image based on their 2D bounding boxes. By doing so, we construct pseudo depth map ground truth. During training, these pseudo depth maps provide early supervision for the image encoder's depth branch.

Image RGB features and spatial depth features are combined to form the Spatial-Enhanced Local Feature $F_{\text{spatial-local}}$:

$$F_{\text{spatial-local}} = F_{\text{spatial}} + F_{\text{rgb}} \tag{3}$$

We use global feature tokens T_{vit} from ViT to adaptively mine the Spatial-Enhanced Local Feature $F_{\text{spatial-local}}$ from the CNN. This process ensures efficient input for the large language model with fewer yet effective tokens. And the spatial-enhanced cross-branch attention employs T_{vit} as the query and $F_{\text{spatial-local}}$ as both key and value:

$$Q = T_{\text{vit}} \times W_Q \quad K = F_{\text{spatial-local}} \times W_K \tag{4}$$

$$V = F_{\text{spatial-local}} \times W_V \quad T = Softmax(Q\,K^T/\sqrt{d_k})\,V \quad (5)$$

We use T as input for the LLM. T integrates the advantages of CNN and ViT, enhancing local spatial information extraction with a relatively small number of tokens.

C. 3D Query Token-Derived Info Decoding

When MLLMs handle visual perception tasks, they typically output coordinates in the form of text tokens [21], [23], [27], [48]. To accomplish 3D grounding tasks, a straightforward approach would be to output the object's 3D spatial coordinates in text format, including location, dimension, and rotation. However, this text-based output method has notable challenges:

1) Low Speed: LLMs like LLaMa [26] treat each digit as a separate token. For instance, 52.3 uses four tokens. 3D detection outputs (coordinates, dimensions, Euler angles) can require 40-50 tokens (Fig. 2(b)), slowing down sequential token generation.

2) Poor Accuracy: LLMs often encounter significant errors when outputting decimal coordinates as text. For example, they generally struggle to accurately interpret the mathematical significance of pitch, roll, and yaw Euler angles in object rotation. Outputting three Euler angles as text poses a significant challenge for LLMs.

3) Parsing Complexity: 3D detection includes nine degrees of freedom (location, dimensions, angles). It is challenging to force LLMs to output these nine values in a standard format. LLMs often output too many or too few numbers or do not follow the standard format, leading to frequent anomalies in parsing results from text.

Specifically, in the input tokens of the LLM, in addition to the image and text tokens, we introduce a 3D Query token. The 3D Query token is a set of learnable parameters that have the same feature dimension as the hidden layer of the LLM (we adopted 4096 as the feature dimension of the 3D Query token, the same as the hidden feature dimension of the LLM). The function of the 3D Query token is similar to the query mechanism in DETR [3]. Through adaptive learning, the 3D Query token can effectively extract image and text 3D information in the self-attention of the LLM. Only one 3D Query token is needed for this task.

We input the 3D Query token into the LLM. In LLM, the 3D Query token generates many feature layers, and its final linear layer feature is used to predict the next token for LLM. We select this final LLM layer's hidden state of the 3D Query token as the 3D feature F_{3D}.

Additionally, to determine when to use the 3D Query token, we introduce a special <pos> token. The <pos> token is placed before the 3D Query in the sequence. When the <pos> token is detected in the LLM's output, the subsequent next input token is replaced with the learnable 3D Query token.

Instead of relying on LLM's text output for spatial positioning, we use a regression head. Specifically, with the 3D Feature F_{3D}, we employ MLP to regress the object's 3D center projection on the image p_{img}, depth d_{v}, 3D size (length L, width W, height H), and rotation angles.

For the object's center, rather than the 2D box center, we use the projection point of the object's 3D center onto the image, which is more suitable for the subsequent geometric inverse projection process. We use an MLP to predicts the 3D center projection $p = (u, v)$:

$$u, v = \text{MLP}_{uv}(F_{3D}) \quad (6)$$

For the 3D size of objects in 3D space: length L, width W, and height H, and the depth d_v, we also use MLP to predict these geometry attributes:

$$L, W, H = \text{MLP}_{LWH}(F_{3D}) \quad d_v = \text{MLP}_d(F_{3D}) \quad (7)$$

Prior work [32] commonly predicts object rotations using Euler angles, which suffer from significant drawbacks: discontinuities, non-uniqueness, asymmetric parameterization, and challenging loss design. To avoid these issues, we adopt the 6D allocentric rotation representation [49], which is continuous in 6D space and more suitable to learn for neural networks.

Specifically, we use an MLP to predict the object's 6D allocentric rotation representation:

$$\text{Rot}_{6D} = \text{MLP}_{6D}(F_{3D}) \quad (8)$$

We then convert the 6D rotation representation into a 3×3 rotation matrix:

$$\text{Rot} = \text{rotation\_6d\_to\_matrix}(F_{3D}) \quad (9)$$

In this section, we employ the learnable 3D Query to extract 3D features and regress the geometric attributes from the LLM. We will derive the 3D bounding box in the following section.

D. Geometry Projection-Based 3D Reasoning

In order to derive the object's 3D box, we need to obtain the spatial location. However, predicting X, Y, Z coordinates directly is challenging due to varying camera parameters. Fig. 2(c) shows that when two objects have the same 2D and 3D sizes, neural networks tend to predict the same 3D location across varying focal lengths, leading to significant errors.

Moreover, predicting 3D metrics (X, Y, Z) directly is highly challenging, as each value can introduce substantial errors in spatial positioning.

Therefore, We introduce a geometry projection-based 3D reasoning approach that estimates the 2D projection of the 3D center and virtual depth instead of predicting X, Y, Z directly.

Since camera focal lengths greatly affect the generalization of depth prediction, we assume that all input images are captured by a virtual camera with unified focal length and resolution [13]. We do not regress the actual depth of the object, but instead, regress the virtual depth under the virtual camera. Subsequently, we convert this virtual depth back to the actual depth of the object. Specifically, since that image width is typically greater than height and multimodal large models usually resize images to the same resolution along the width, we improved the calculation method for virtual depth by using width as a reference.

For a space point P^i and its image projection pixel p^i, we assume that in the virtual camera, P^v projects to p^v. P^v and P^i

have the same X and Y but differ in depth Z. The pixels p^v and p^i coincide on the image.[1]

Given the real camera's intrinsic matrix $\mathbf{K^i}$ and real image width w^i, point $P^i = (X^i, Y^i, Z^i)$ is projected onto the image pixel $p^i = (x^i, y^i)$:

$$Z^i[x^i, y^i, 1]^\top = \mathbf{K^i}[X^i, Y^i, Z^i]^\top \tag{10}$$

$$\mathbf{K^i} = \begin{bmatrix} f_x^i & 0 & c_x^i \\ 0 & f_y^i & c_y^i \\ 0 & 0 & 1 \end{bmatrix} \tag{11}$$

where $f_x^i, f_y^i, c_x^i, c_y^i$ are the intrinsics of camera C^i.

From (10) and (11), we can get:

$$x^i \cdot Z^i = f_x^i \cdot X^i + c_x^i \cdot Z^i \tag{12}$$

Assuming the intrinsics of the virtual camera are $f_x^v, f_y^v, c_x^v, c_y^v$ and the virtual image width is w^v. The virtual point P^v corresponds to P^i in terms of X and Y, differing solely in the Z dimension. Therefore, $P^v = (X^i, Y^i, Z^v)$. The image projection of P^v is given by $p^v = (x^v, y^v)$. Similar to (12), we could get the projection formula in the virtual camera:

$$x^v \cdot Z^v = f_x^v \cdot X^i + c_x^v \cdot Z^v \tag{13}$$

Projection pixels $p^v = (x^v, y^v)$ and $p = (x^i, y^i)$ align in the image. And the principal point c_x^v for the virtual camera corresponds to the principal point of the real camera, so we could get:

$$x^v = \frac{x^i}{w^i} \cdot w^v, \qquad c_x^v = \frac{c_x^i}{w^i} \cdot w^v \tag{14}$$

Substituting (14) into (13) yields:

$$\frac{x}{w^i} \cdot w^v \cdot Z^v = f_x^v \cdot X + \frac{c_x^i}{w^i} \cdot w^v \cdot Z^v \tag{15}$$

Thus, we have: $x = f_x^v \cdot \frac{X}{Z^v} \cdot \frac{w^i}{w^v} + c_x$

With (12), we get:

$$\left(f_x^v \cdot \frac{X}{Z^v} \cdot \frac{w^i}{w^v} + c_x \right) \cdot Z = f_x^i \cdot X + c_x^i \cdot Z \tag{16}$$

By simplifying (16), we derive the virtual depth:

$$Z^v = \frac{f_x^v}{f_x^i} \cdot \frac{w^i}{w^v} \cdot Z \tag{17}$$

Here, f_x^i, f_x^v are the focal lengths and w^i, w^v are image widths of the real and virtual cameras. Z is the actual depth.

Virtual depth $Z^v = \frac{f_x^v}{f_x^i} \cdot \frac{w^i}{w^v} \cdot Z$ accounts for the varying focal lengths and image sizes, enabling the neural network to make 3D predictions invariant to camera intrinsic parameters.

Thus, in Section III-C, we regress the virtual depth, represented by d_v from the MLP, and convert it to actual depth. We invert (17) to convert the predicted virtual depth into the real depth:

$$Z_1 = d_v \cdot \frac{f_x}{f_x^v} \cdot \frac{w^v}{w} \tag{18}$$

[1] In this paper, we use uppercase letters to denote 3D points in space and lowercase letters for 2D pixels on images.

where f_x and w denote the focal length and width of the real camera, while f_x^v and w^v refer to those of the virtual camera.

In outdoor autonomous driving scenarios, object depths vary significantly. We adopt a geometric projection constraint in the Y direction to estimate a second independent depth. Using projection equations (10) and (11), we could get the second depth:

$$Z_2 = \frac{H}{h} \cdot f_y \tag{19}$$

where h and H are the objects' 2D and 3D height, respectively.

We then compute the average of Z_1 and Z_2 to enhance depth prediction accuracy:

$$Z = \frac{Z_1 + Z_2}{2} = \left(d_v \cdot \frac{f_x}{f_x^v} \cdot \frac{w^v}{w} + \frac{H}{h} \cdot f_y \right) /2 \tag{20}$$

Currently, we have obtained the Z-coordinate of the 3D bbox center P. Next, using the 3D center image projection point $p = (u, v)$ predicted in Section III-C, along with the depth Z, we can calculate the X and Y coordinates of P. According to the projection formula (10) and (11), we could get:

$$X = \frac{Z}{f_x} \cdot (u - c_x), \quad Y = \frac{Z}{f_y} \cdot (v - c_y) \tag{21}$$

Substituting from (20), we derive:

$$X = \left(\frac{d_v}{f_x^v} \cdot \frac{w^v}{w} + \frac{H}{h} \cdot \frac{f_y}{f_x} \right) \cdot (u - c_x)/2 \tag{22}$$

$$Y = \left(\frac{f_x}{f_y} \cdot \frac{d_v}{f_x^v} \cdot \frac{w^v}{w} + \frac{H}{h} \right) \cdot (v - c_y)/2 \tag{23}$$

$$Z = \left(d_v \cdot \frac{f_x}{f_x^v} \cdot \frac{w^v}{w} + \frac{H}{h} \cdot f_y \right) /2 \tag{24}$$

Thus, we express the 3D bbox center $P = (X, Y, Z)$, with neural network outputs and known parameters.

Additionally, in Section III-C, we obtained the dimensions of the 3D bbox: L, W, and H, and the rotation Rot. Therefore, we can obtain the final 3D bbox of the object.

E. IG3D: Image-Based 3D Grounding Dataset

3D grounding requires data with bounding boxes and object descriptions. Current datasets like KITTI [50], nuScenes [51], and Waymo [52] offer only categories. When an image contains multiple objects of the same category, it is impossible to identify the interested object based solely on the category. Furthermore, applications such as embodied intelligence and robotics necessitate that agents engage in tasks like Visual Question Answering (VQA), which require the agent to possess common sense and perform logical reasoning based on user questions to identify and respond to specific objects. Current 3D detection datasets only provide object categories and lack annotations related to 3D logical reasoning and 3D question answering. Therefore, we propose the IG3D dataset, which offers detailed descriptions or questions and supports the 3D grounding and 3D VQA tasks.

As shown in Fig. 4, we first employ visual prompting [53] to draw the 2D box to highlight the target object in the image.

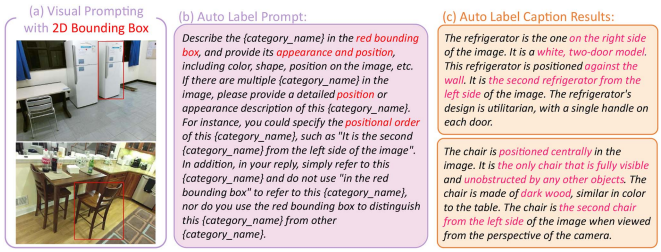

Fig. 4. The auto label process in our IG3D dataset. For each image, we use visual prompting [53] to add a 2D box around the target object. A pre-trained MLLM then generates descriptive captions. In this auto label prompt, {category_name} is replaced with the object's category name.

Then, employing a pre-trained MLLM model (Mini-Gemini-34B [54]), we produce detailed object descriptions for those objects.

To evaluate Visual Question Answering, we use GPT-4o [55] to generate specific questions. For example, GPT-4o generated the question: "I want to read a book, but the light is too dim. Which object should I use?" Then, the 3D perception model is expected to identify "the lamp" and its 3D box.

We created IG3D datasets from SUNRGBD [56], nuScenes [51], KITTI [50], and Objectron [57], termed IG3D-SUNRGBD, IG3D-nuScenes, IG3D-KITTI, and IG3D-Objectron. The IG3D-SUNRGBD-VQA dataset supports VQA tasks. For 3D box ground truth, we use the original annotations from these datasets. For the training, validation, and test splits of all these datasets, we follow Omni3D [13]. We refined the datasets by filtering erroneous annotations. Eight human experts reviewed 30% dataset annotations to ensure quality.

IV. Experiments

A. Experimental Settings

1) Datasets, Metrics, and Baselines: Experiments were performed on various datasets. SUNRGBD [56] and Objectron [57] focus on indoor scenes with more categories. NuScenes [51], KITTI [50], and Mono3DRefer [14] are outdoor autonomous driving datasets.

We follow the metrics in Mono3DVG [14]: "Acc@0.25" and "Acc@0.5" for IoU thresholds of 25% and 50%, respectively. "DepthError", "LengthError", "WidthError", and "HeightError" assess depth and dimension errors in meters.

"Text3D" refers to the results obtained from the baseline Mini-Gemini-7B [54], which outputs 3D bbox in the form of text. "TransVG+backproj" follows the baseline from Mono3DVG [14], using 2D vision grounding combined with back-projection to adapt the results to 3D. Mono3DVG is currently the SOTA image-based 3D grounding method.

2) More Implement Details: We perform parameter-efficient fine-tuning using LoRA [20] to fine-tune Mini-Gemini-7B [54]. The rank of LoRA is set to 64, and alpha is set to 16. For the LLM, we choose the Vicuna-7B version [58]. We use the L1 loss for the 3D regression heads. We follow the fine-tuning hyper-parameters set by Mini-Gemini, using the AdamW optimizer and the learning rate 2e-5. The batch size is 4, and gradient accumulation steps are set to 4. The fine-tuning process is conducted with two NVIDIA A100 GPUs.

B. Comparison With Other Methods

1) 3D Grounding and 3D VQA Results: As shown in Table II, our method achieves SOTA performance. The precision of "TransVG+backproj" is very low, highlighting the difficulty of 3D grounding, as it requires depth and size estimation for bounding boxes, which is challenging from single images. "Text3D" underperforms due to directly outputting the 3D numerical value in text form. Mono3DVG [14] performs well in outdoor scenarios but struggles in indoor scenes like SUNRGBD. Our method exceeds Mono3DVG in various datasets, overcoming its yaw angle prediction limitation through 6D allocentric rotation.

The IG3D-SUNRGBD-VQA dataset involves question-answering and complex 3D reasoning tasks, evaluating the general abilities of models, which are crucial in fields like robotics and embodied intelligence. TransVG+backproj and Mono3DVG are small specialized models and cannot perform reasoning. Our method, utilizing the general capabilities of LLMs, shows strong performance. Our LLMI3D achieves state-of-the-art performance in all datasets, surpassing other methods by a significant margin.

2) Open Vocabulary 3D Grounding: The real world comprises an infinite number of categories, including numerous rare sample classes and corner cases. Existing works and methods predominantly evaluate the accuracy of models on known categories encountered during training without testing their performance on open-world categories.

To evaluate the models' capability in open-world scenarios, we tested their zero-shot open-vocabulary 3D grounding performance. Specifically, we divided the categories in datasets such as SUNRGBD, nuScenes, KITTI, and Objectron into 80% base classes and 20% novel classes. We trained all models on the base classes and assessed their 3D grounding performance on the novel classes, as illustrated in Table III.

In Table III, our model demonstrates significantly higher accuracy than existing works. Models like Mono3DVG, when trained on base classes, perform well on these classes but exhibit poor performance on novel classes. This indicates the inability of specialized small models like Mono3DVG to handle open-world scenarios and address the vast array of object categories in real-world settings.

MLLMs inherently possess significant advantages over specialized small models in open vocabulary tasks. When faced with new categories, small models struggle to ascertain the 3D spatial sizes of new classes, failing to accurately determine their dimensions. In contrast, large models are endowed with world knowledge and common sense, enabling them to understand the actual sizes of novel categories in the physical world and infer their physical positions from the environment. Thus, large models outperform small models significantly in open vocabulary tasks.

TABLE II
COMPARISON OF OUR LLMI3D WITH OTHER METHODS ON THE IG3D-SUNRGBD, IG3D-SUNRGBD-VQA, IG3D-NUSCENES, IG3D-KITTI, AND IG3D-OBJECTRON DATASETS

Dataset	Method	Acc@0.25↑	Acc@0.5↑	DepthError↓	LengthError↓	WidthError↓	HeightError↓
IG3D-SUNRGBD	TransVG + backproj	5.6	0.4	0.88	0.57	0.84	0.59
	Text3D	11.5	1.7	0.45	0.20	0.34	0.21
	Mono3DVG	25.2	6.8	0.53	0.14	0.26	0.16
	LLMI3D	**42.3**	**11.8**	**0.32**	**0.12**	**0.21**	**0.12**
IG3D-SUNRGBD-VQA	TransVG + backproj	2.2	0.2	0.97	0.71	0.93	0.77
	Text3D	7.8	1.0	0.56	0.24	0.45	0.29
	Mono3DVG	9.8	1.4	0.63	0.21	0.41	0.27
	LLMI3D	**35.1**	**8.6**	**0.36**	**0.16**	**0.28**	**0.17**
IG3D-nuScenes	TransVG + backproj	8.6	3.5	7.51	2.28	0.77	0.82
	Text3D	13.7	5.2	4.25	1.75	0.28	0.27
	Mono3DVG	27.5	9.8	2.80	0.55	0.19	0.21
	LLMI3D	**31.6**	**13.2**	**2.19**	**0.50**	**0.16**	**0.17**
IG3D-KITTI	TransVG + backproj	2.9	0.3	8.42	1.39	0.31	0.35
	Text3D	5.4	0.7	4.15	0.70	0.16	0.17
	Mono3DVG	27.7	7.7	2.08	0.44	0.13	0.14
	LLMI3D	**32.4**	**10.3**	**1.56**	**0.34**	**0.11**	**0.11**
IG3D-Objectron	TransVG + backproj	23.0	6.7	0.14	0.05	0.03	0.05
	Text3D	35.4	10.7	0.08	**0.03**	**0.02**	0.04
	Mono3DVG	45.5	12.4	0.09	**0.03**	**0.02**	**0.03**
	LLMI3D	**55.6**	**18.7**	**0.05**	**0.03**	**0.02**	**0.03**

TABLE III
IN THE OPEN VOCABULARY 3D GROUNDING TASK, RESULTS COMPARISON OF OUR LLMI3D WITH OTHER METHODS ON THE IG3D-SUNRGBD, IG3D-NUSCENES, IG3D-KITTI, AND IG3D-OBJECTRON DATASETS

Dataset	Method	Acc@0.25↑	Acc@0.5↑	DepthError↓	LengthError↓	WidthError↓	HeightError↓
IG3D-SUNRGBD	Text3D	8.6	1.5	0.52	0.31	0.47	0.24
	Mono3DVG	19.4	2.4	0.54	0.21	0.40	0.18
	LLMI3D	**40.1**	**4.4**	**0.38**	**0.19**	**0.29**	**0.11**
IG3D-nuScenes	Text3D	8.2	1.9	5.50	4.81	0.47	0.75
	Mono3DVG	10.5	2.2	5.42	3.45	0.58	0.93
	LLMI3D	**30.5**	**7.5**	**3.11**	**2.56**	**0.36**	**0.57**
IG3D-KITTI	Text3D	3.1	0.3	5.30	3.88	0.37	0.62
	Mono3DVG	2.2	0.2	5.97	1.88	0.46	0.51
	LLMI3D	**30.1**	**8.1**	**1.84**	**0.84**	**0.17**	**0.26**
IG3D-Objectron	Text3D	20.5	6.0	0.08	0.04	**0.03**	**0.06**
	Mono3DVG	6.4	0.1	0.35	0.07	0.15	0.16
	LLMI3D	**32.8**	**7.9**	**0.07**	**0.03**	**0.03**	**0.06**

TABLE IV
IN THE DOMAIN GENERALIZATION FOR THE 3D GROUNDING TASK, RESULTS COMPARISON OF OUR LLMI3D WITH OTHER METHODS

Generalization Setting	Method	Acc@0.25↑	DepthError↓	LengthError↓	WidthError↓	HeightError↓
nuScenes → KITTI	Text3D	5.0	4.98	0.67	0.37	0.20
	Mono3DVG	16.5	2.24	0.84	0.29	0.16
	LLMI3D	**23.1**	**2.11**	**0.59**	**0.20**	**0.13**
KITTI → nuScenes	Text3D	0.9	8.47	0.72	0.65	0.62
	Mono3DVG	4.3	5.30	0.62	0.35	0.24
	LLMI3D	**20.5**	**3.32**	**0.54**	**0.29**	**0.22**

nuScenes→KITTI refers to models trained on the nuScenes dataset and tested on the KITTI dataset, while KITTI→nuScenes represents models trained on the KITTI dataset and tested on the nuScenes dataset.

3) Domain Generalization for the 3D Grounding: In real-world applications, we encounter not only novel categories but also domain gaps. The 3D domain gap is more complex than 2D due to variations in focal lengths and resolutions across datasets, such as lower resolutions in older datasets like KITTI compared to newer ones like nuScenes, significantly affecting 3D localization [59].

In Table IV, we present experiments conducted under different domain generalization settings. The nuScenes → KITTI setting involves training on the nuScenes dataset and testing

TABLE V
THE RESULTS ON MONO3DREFER DATASETS

Method	Type	Unique		Multiple		Overall	
		Acc@0.25↑	Acc@0.5↑	Acc@0.25↑	Acc@0.5↑	Acc@0.25↑	Acc@0.5↑
ZSGNet + backproj	One-Stage	9.02	0.29	16.56	2.23	15.14	1.87
FAOA + backproj	One-Stage	11.96	2.06	13.79	2.12	13.44	2.11
ReSC + backproj	One-Stage	11.96	0.49	23.69	3.94	21.48	3.29
TransVG + backproj	Tran.-based	15.78	4.02	21.84	4.16	20.70	4.14
Mono3DVG-TR	Tran.-based	57.65	33.04	65.92	46.85	64.36	44.25
LLMI3D	Tran.-based	**60.14**	**35.91**	**69.19**	**49.20**	**67.48**	**46.70**

TABLE VI
WHEN THE INPUT PROMPT IS CHANGED TO CAPTION+2D BOX, COMPARISON OF OUR LLMI3D WITH OTHER METHODS ON THE IG3D-SUNRGBD, IG3D-NUSCENES, IG3D-KITTI, AND IG3D-OBJECTRON DATASETS

Dataset	Method	Acc@0.25↑	Acc@0.5↑	DepthError↓	LengthError↓	WidthError↓	HeightError↓
IG3D-SUNRGBD	Text3D	21.3	4.7	0.39	0.17	0.28	0.16
	Mono3DVG	47.8	18.1	0.35	0.13	0.20	0.12
	LLMI3D	**56.6**	**18.6**	**0.25**	**0.11**	**0.18**	**0.11**
IG3D-nuScenes	Text3D	22.4	8.9	2.16	1.19	0.26	0.19
	Mono3DVG	34.1	11.5	2.31	0.46	0.17	0.15
	LLMI3D	**42.4**	**17.6**	**1.45**	**0.43**	**0.15**	**0.14**
IG3D-KITTI	Text3D	10.2	2.8	2.75	0.77	0.17	0.17
	Mono3DVG	43.7	13.4	0.88	0.38	0.12	**0.10**
	LLMI3D	**50.1**	**17.5**	**0.77**	**0.34**	**0.11**	**0.10**
IG3D-Objectron	Text3D	36.7	11.5	0.08	**0.03**	**0.02**	0.04
	Mono3DVG	53.4	16.4	0.05	**0.03**	**0.02**	**0.03**
	LLMI3D	**64.4**	**21.9**	**0.04**	**0.03**	**0.02**	**0.03**

on the KITTI dataset, while the KITTI → nuScenes setting involves training on the KITTI dataset and testing on the nuScenes dataset.

The results presented in Table IV reveal that the Text3D and the Mono3DVG suffer significant accuracy declines under domain generalization conditions. When the camera parameters change, these models struggle to identify variations in focal length and continue to use the old focal lengths to infer object spatial positions. They fail to adapt to focal length variations for accurate spatial position inference. In contrast, under domain generalization settings, our method leverages virtual depth and geometry projection-based 3D reasoning. Consequently, our approach significantly surpasses the baseline models.

4) Results on Mono3DRefer Dataset: As shown in Table V, to demonstrate the robustness of our model, we evaluated it on the Mono3DRefer dataset [14]. We adopt the evaluation metrics from Mono3DVG, where "unique" refers to images with a single object of a category, and "multiple" indicates images containing more than one object of the same category. We follow Mono3DVG's baseline methods. These 2D grounding baselines, combined with back projection, underperformed on Mono3DRefer, highlighting the complexity of 3D tasks. Our LLMI3D outperformed all current methods, achieving state-of-the-art accuracy across all metrics, demonstrating our model's strong generalization on diverse datasets.

5) Various Types of Input Prompts: Table VI shows results when the prompt includes the caption and the 2D box (as shown in the upper part of Fig. 7). During input, we draw a 2D box on the image as a visual prompting [53] to assist the model in performing 3D grounding. Adding the 2D box to prompt inputs improves accuracy in all datasets compared to caption-only prompts, highlighting the challenge of 3D grounding with only textual cues. When images have multiple overlapping objects within the same category, distinguishing them using text alone is difficult. Utilizing both caption and 2D box as input significantly enhances localization accuracy.

Table VII reports results with prompts including text caption and a 2D point (as shown in the lower part of Fig. 7). We draw a 2D point on the input image as a visual prompting [53] to assist the model in performing 3D grounding. Including a 2D point in the input prompt helps with grounding the object. In robotics and augmented reality, users may employ various inputs and cues to locate the object of interest, such as a 2D point and textual description. Our method effectively adapts to these user input types, demonstrating robust and versatile capabilities.

6) Performance With Extra Depth Prediction or Segmentation Models: Tables VIII and IX illustrate our experiments integrating depth maps predicted by models like UniDepth [60], Depth Anything [61], and Depth Pro [62], and segmentation results from SAM [63], as additional inputs to the MLLM. The inputs to the MLLM not only include image features but also the depth maps or segmentation results predicted by these models.

As shown in Tables VIII and IX, the integration of additional depth prediction modules does not substantially enhance performance. These depth predictors primarily focus on pixel-level depth prediction for the entire image scene, whereas our method concentrates on the depth estimation of specific objects. Furthermore, these external depth prediction models differ fundamentally from our approach. Our approach enhances the spatial and depth perception capabilities of the image encoder by introducing depth supervision within the encoder, with minimal computational overhead. In contrast, models like Depth

Fig. 5. Examples of the 3D VQA of our LLMI3D in the IG3D-SUNRGBD-VQA dataset. Our LLMI3D is capable of understanding user-input various questions, leveraging common knowledge and logical reasoning to identify objects of interest, and returning the corresponding 3D bboxes.

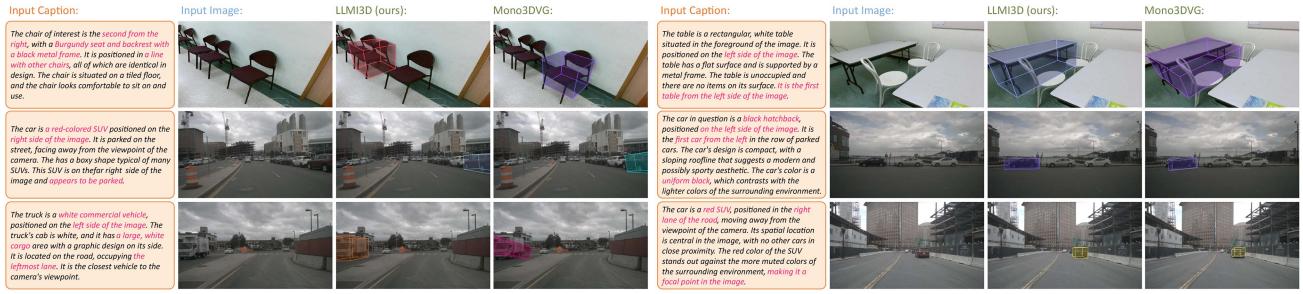

Fig. 6. The 3D grounding visualizations of our LLMI3D and Mono3DVG [14] in the SUNRGBD and nuScenes dataset. When users provide an image and a caption describing the object of interest, the Mono3DVG method lacks sufficient understanding and logical reasoning abilities for natural language, often misidentifying the object and producing inaccurate 3D bounding boxes. Our approach, however, generates precise 3D bounding boxes for the specified objects.

TABLE VII
WHEN THE INPUT PROMPT IS CHANGED TO CAPTION+2D POINT, COMPARISON OF OUR LLMI3D WITH OTHER METHODS ON THE IG3D-SUNRGBD, IG3D-NUSCENES, IG3D-KITTI, AND IG3D-OBJECTRON DATASETS

Dataset	Method	Acc@0.25↑	Acc@0.5↑	DepthError↓	LengthError↓	WidthError↓	HeightError↓
IG3D-SUNRGBD	Text3D	15.7	2.5	0.40	0.19	0.33	0.18
	Mono3DVG	42.6	11.4	0.37	0.13	0.23	0.13
	LLMI3D	**53.7**	**15.8**	**0.25**	**0.12**	**0.20**	**0.12**
IG3D-nuScenes	Text3D	19.7	7.6	2.20	1.23	0.29	0.25
	Mono3DVG	30.1	10.4	2.41	0.53	0.18	0.18
	LLMI3D	**37.4**	**17.4**	**1.74**	**0.46**	**0.17**	**0.16**
IG3D-KITTI	Text3D	8.3	2.1	3.19	0.78	0.16	0.17
	Mono3DVG	31.4	9.5	1.72	0.38	0.13	0.12
	LLMI3D	**39.0**	**14.3**	**1.05**	**0.34**	**0.11**	**0.10**
IG3D-Objectron	Text3D	37.9	12.6	0.07	**0.03**	**0.02**	0.04
	Mono3DVG	49.6	15.1	0.06	**0.03**	**0.02**	**0.03**
	LLMI3D	**62.8**	**22.9**	**0.05**	**0.03**	**0.02**	**0.03**

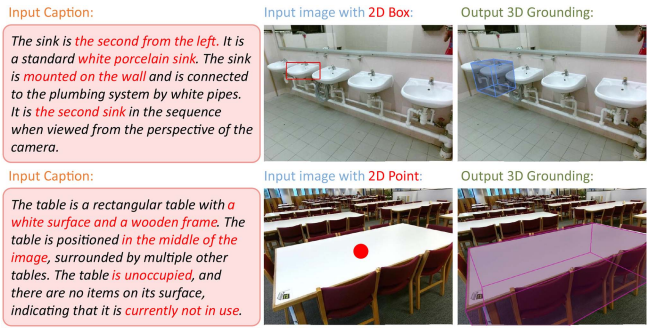

Fig. 7. Our LLMI3D can accept various input prompts, such as a caption with a 2D box or a caption with a 2D point, and output the corresponding 3D box, thereby accommodating various user input forms.

TABLE VIII
EXPERIMENTAL RESULTS ON THE IG3D-SUNRGBD DATASET WHEN ADDING EXTRA DEPTH PREDICTION MODEL OR SEGMENTATION MODEL

Exp.	Setting	Acc@0.25↑	Acc@0.5↑	DepthError↓
(a)	SAM +LLMI3D	41.7	11.0	0.35
(b)	Depth Anything +LLMI3D	41.9	11.3	0.33
(c)	UniDepth +LLMI3D	41.9	11.3	0.31
(d)	Depth Pro +LLMI3D	42.0	11.6	**0.30**
(e)	LLMI3D	**42.3**	**11.8**	0.32

TABLE IX
EXPERIMENTAL RESULTS ON THE IG3D-NUSCENES DATASET WHEN ADDING EXTRA DEPTH PREDICTION MODEL OR SEGMENTATION MODEL

Exp.	Setting	Acc@0.25↑	Acc@0.5↑	DepthError↓
(a)	SAM+LLMI3D	30.8	12.6	2.27
(b)	UniDepth +LLMI3D	31.0	12.8	2.23
(c)	Depth Anything+LLMI3D	31.1	12.8	2.19
(d)	Depth Pro +LLMI3D	31.3	12.9	**2.15**
(e)	LLMI3D	**31.6**	**13.2**	2.19

TABLE X
ABLATION STUDY ON THE SPATIAL-ENHANCED LOCAL FEATURE MINING METHOD USING THE IG3D-SUNRGBD DATASET

Exp.	HR Branch	Depth Branch	SECBA	Acc@0.25↑	Acc@0.5↑	DepthError↓
(a)				30.7	6.1	0.73
(b)	✓			35.4	8.6	0.58
(c)	✓	✓		37.4	9.3	0.42
(d)	✓	✓	✓	**42.3**	**11.8**	**0.32**

"SECBA" stands for Spatial-Enhanced Cross-Branch Attention.

Anything operate as independent entities, thereby failing to enhance the inherent spatial awareness of the image encoder. And these models significantly increase the computational cost.

C. Ablation Study

1) Ablation Study on the Spatial-Enhanced Local Feature Mining: As shown in Table X, the "HR branch" uses a CNN for high-resolution feature extraction. The "depth branch" estimates object-level depth to enhance spatial features. The "Spatial-Enhanced Cross-Branch Attention" (SECBA) mechanism leverages spatially-enhanced local features from the CNN and global tokens from the ViT to perform spatially-enhanced cross-branch attention.

TABLE XI
ABLATION STUDY ON THE IG3D-SUNRGBD DATASET TO EVALUATE THE DECODING METHODS IN THE LLM PART

Exp.	Setting	Acc@0.25↑	Acc@0.5↑	DepthError↓
(a)	Text Output	11.5	1.7	0.45
(b)	Text Feature	21.6	4.7	0.42
(c)	Position Token	40.5	10.4	0.37
(d)	3D Query Decoder	**42.3**	**11.8**	**0.32**

TABLE XII
ABLATION STUDY ON THE IG3D-NUSCENES DATASET TO EVALUATE THE GEOMETRY PROJECTION-BASED 3D REASONING METHOD

Exp.	Back Projection	Height Depth	Virtual Depth	Acc@0.25↑	Acc@0.5↑	DepthError↓
(a)				19.8	8.6	3.46
(b)	✓			26.9	10.4	2.94
(c)	✓	✓		28.8	12.1	2.73
(d)	✓		✓	29.4	12.5	2.57
(e)	✓	✓	✓	**31.6**	**13.2**	**2.19**

Experiment (a) does not incorporate the high-resolution (HR) branch and relies solely on low-resolution ViT tokens, which limits spatial feature extraction and results in increased DepthError and reduced accuracy. Experiment (b) adds a CNN HR branch to enhance the identification of small and distant objects using high-resolution imagery. Experiment (c) introduces a depth branch in the CNN to improve spatial extraction through depth supervision. Experiments (b) and (c) naively apply max pooling to combine feature maps extracted by the CNN with ViT tokens before inputting them into the LLM. Experiment (d) utilizes SECBA to effectively integrate CNN local spatial features with ViT tokens, achieving the highest accuracy.

2) Ablation Study on the 3D Query Token-Derived Info Decoding: As shown in Table XI, in Experiment (a), the 3D grounding results are output in text form, which complicates parsing and reduces accuracy. Experiment (b) enhances accuracy by using hidden layer features of text tokens with an MLP to regress 3D values, demonstrating the effectiveness of regression compared to text-based outputs. In Experiment (c), <pos> token features are used with an MLP, which proves effective but less accurate than the 3D Query method. Experiment (d) details our comprehensive approach, introducing a learnable 3D Query token within the LLM, enabling precise 3D feature extraction using a single 3D Query token.

3) Ablation Study on the Geometry Projection-Based 3D Reasoning: As detailed in Table XII, we present the results of our ablation study on Geometry Projection-Based 3D Reasoning using the IG3D-nuScenes dataset. The nuScenes dataset consists of images captured by six different cameras with varying focal lengths. The term "back projection" refers to whether back projection is performed using the predicted image projection of the 3D center and depth. The "2D-3D height depth" denotes the depth value derived using the height of the object's 2D and 3D bounding box, as explained in (19): $Z_2 = \frac{H}{h} \cdot f_y$. The "Virtual depth" is calculated using (18): $Z_1 = d_v \cdot \frac{f_x}{f_x^v} \cdot \frac{w^v}{w}$.

In Table XII, Experiment (a) does not perform back projection. It does not predict the image projection or the depth value of the object's center. Instead, Experiment (a) directly regresses the spatial coordinates x, y, and z of the object. This approach

TABLE XIII
EXPERIMENTS ON THE IG3D-SUNRGBD DATASET TO EVALUATE THE ROTATION ANGLE PREDICTION METHODS

Exp.	Rotation Prediction	Acc@0.25↑	Acc@0.5↑
(a)	Euler Angle	37.1	8.8
(b)	6D Allocentric Rotation	**42.3**	**11.8**

faces significant challenges: x, y, and z are all absolute spatial coordinates, making them difficult to infer from the image. Prediction errors in any of these coordinates result in considerable inaccuracies in the 3D box's position. Consequently, the accuracy of Experiment (a) is appreciably lower compared to other experiments.

Experiment (b) employs the 3D center and depth, combined with back projection, to determine the spatial position of the 3D bounding box. This method markedly enhances accuracy over Experiment (a). Experiment (c) builds on Experiment (b) by incorporating 2D-3D height depth. Deriving depth values in outdoor scenes using the equation $Z_2 = \frac{H}{h} \cdot f_y$ provides a valuable reference, leading to improved accuracy over Experiment (b).

Experiment (d) integrates virtual depth. Virtual depth is assumed to be generated under a consistent virtual camera. When working with images of different focal lengths, regressing a unified virtual depth via the neural network offers distinct advantages. As the nuScenes dataset comprises images captured by six cameras with varying focal lengths, employing virtual depth addresses focal length discrepancies and enhances 3D localization performance on this dataset.

Experiment (e) represents our complete version of LLMI3D, which combines 2D-3D height and virtual depth to determine the final depth, as outlined in (20). Moreover, it utilizes the predicted image projection of the object's 3D center to derive the 3D bounding box. This approach mitigates the limitations of MLLM in handling camera focal variations, thereby achieving the highest experimental accuracy observed.

By incorporating camera intrinsics and geometric projections, the performance of 3D perception can be significantly improved. However, when the camera intrinsics are perturbed, the performance of geometric projection will be affected. In these cases, MLLMs, endowed with world knowledge, commonsense reasoning, and logical inference, demonstrate greater robustness to such camera intrinsic perturbations. For instance, they can exploit relative positional relationships among surrounding objects to mitigate the impact of inaccurate intrinsics.

In scenarios where camera intrinsics are unknown, geometry projection-based approaches become inapplicable. In such cases, one possible solution is to adopt methods similar to VGGT [64], which estimate camera intrinsics directly from images. Moreover, since we have constructed a 3D dataset, we can train the models to predict 3D spatial positions directly. Although this approach underperforms geometric projection methods, it can still improve 3D perception compared to general MLLMs owing to task-specific post-training.

4) Experiments on the Rotation Angle Prediction: As presented in Table XIII, Experiment (a), which uses Euler angles, demonstrates significant errors. In contrast, Experiment (b) employs 6D allocentric rotation, a continuous representation that is better suited for neural networks.

D. Visualization Results

Fig. 5 showcases the model's performance on the IG3D-SUNRGBD-VQA dataset. Our model effectively answers the questions and locates objects in 3D.

Fig. 6 illustrates 3D grounding on the indoor and outdoor autonomous driving scenarios of our LLMI3D and Mono3DVG [14]. With image and caption inputs, Mono3DVG [14] often struggles with natural language comprehension and reasoning, leading to confusion between objects and incorrect spatial localization. In contrast, our LLMI3D accurately identifies and localizes the object of interest across diverse environments.

Fig. 7 illustrates our LLMI3D 's capability to accept various input prompts, including captions, 2D boxes, and 2D points, demonstrating its flexibility.

V. CONCLUSION

The demand for 3D perception such as autonomous driving, augmented reality, and robotics is growing rapidly. In this paper, we identified and tackled three major issues faced by MLLMs in 3D perception: (1) Weak 3D local spatial object perception, (2) Poor text-based geometric numerical output, and (3) Inability to handle camera focal variations.

To address these challenges, we proposed LLMI3D, a 3D-friendly MLLMs architecture. For the image encoder, we introduced Spatial-Enhanced Local Feature Mining. In the LLM part, we proposed 3D Query Token-Derived Info Decoding. To obtain the 3D bounding boxes of objects, we proposed Geometry Projection-Based 3D Reasoning. Furthermore, we introduce the IG3D dataset, which assesses fine-grained grounding, logical reasoning, and question-answering in 3D perception models.

Extensive experiments on various settings, including 3D grounding and 3D VQA, open-vocabulary 3D grounding, and domain generalization confirm that our method achieves state-of-the-art results, surpassing other methods by a significant margin. However, our work still has limitations. While the MLLM-based 3D perception framework demonstrates strong general capabilities, its inference latency is larger than that of specialized small models. In the future, we aim to improve the reasoning efficiency of the MLLM-based framework.

REFERENCES

[1] Y. Li et al., "LMEye: An interactive perception network for large language models," *IEEE Trans. Multimedia*, vol. 26, pp. 1095–10964, 2024.
[2] H. Zhang, J. Wang, J. Zhang, T. Zhang, and B. Zhong, "One-stream vision-language memory network for object tracking," *IEEE Trans. Multimedia*, vol. 26, pp. 1720–1730, 2023.
[3] N. Carion et al., "End-to-end object detection with transformers," in *Proc. Eur. Conf. Comput. Vis.*, vol. 12346, 2020, pp. 213–229.
[4] S. Zhao, H. Yao, C. Lin, Y. Gao, and G. Ding, "Multi-source-free domain adaptive object detection," *Int. J. Comput. Vis.*, vol. 52, pp. 5950–5982, 2024.
[5] P. An et al., "SP-Det: Leveraging saliency prediction for voxel-based 3D object detection in sparse point cloud," *IEEE Trans. Multimedia*, vol. 26, pp. 2795–2808, 2024.

[6] T. Shen, D. Li, F.-Y. Wang, and H. Huang, "Depth-aware multi-person 3D pose estimation with multi-scale waterfall representations," *IEEE Trans. Multimedia*, vol. 25, pp. 1439–1451, 2022.

[7] G. Hua et al., "Weakly-supervised 3D human pose estimation with cross-view U-shaped graph convolutional network," *IEEE Trans. Multimedia*, vol. 25, pp. 1832–1843, 2022.

[8] A. Mousavian, D. Anguelov, J. Flynn, and J. Kosecka, "3D bounding box estimation using deep learning and geometry," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 5632–5640.

[9] Z. Qin, J. Wang, and Y. Lu, "MonoGRNet: A geometric reasoning network for monocular 3D object localization," in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 8851–8858.

[10] A. H. Lang et al., "PointPillars: Fast encoders for object detection from point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 1697–12705.

[11] Y. Sun et al., "Learning monocular regression of 3D people in crowds via scene-aware blending and de-occlusion," *IEEE Trans. Multimedia*, vol. 26, pp. 2289–2302, 2024.

[12] R. Zhang et al., "MonoDETR: Depth-guided transformer for monocular 3D object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2023, pp. 9121–9132.

[13] G. Brazil et al., "Omni3D: A large benchmark and model for 3D object detection in the wild," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 13154–13164.

[14] Y. Zhan, Y. Yuan, and Z. Xiong, "Mono3DVG: 3D visual grounding in monocular images," in *Proc. AAAI Conf. Artif. Intell.*, 2024, pp. 6988–6996.

[15] Z. Wu et al., "DeepSeek-Vl2: Mixture-of-experts vision-language models for advanced multimodal understanding," 2024, *arXiv:2412.10302*.

[16] Z. Chen et al., "Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling," 2024, *arXiv:2412.05271*.

[17] OpenAI, "GPT-4 technical report," 2023, *arXiv:2303.08774*.

[18] Q. Team, "Qwen2.5-Vl," Jan. 2025. [Online]. Available: https://qwenlm.github.io/blog/qwen2.5-vl/

[19] W. Zhang et al., "Unleash the power of vision-language models by visual attention prompt and multi-modal interaction," *IEEE Trans. Multimedia*,, vol. 27, pp. 2399–2411, 2024.

[20] E. J. Hu et al., "LoRa: Low-rank adaptation of large language models," in *Proc. Int. Conf. Learn. Representations*, 2022, pp. 1–20.

[21] P. Wang et al., "Qwen2-Vl: Enhancing vision-language model's perception of the world at any resolution," 2024, *arXiv:2409.12191*.

[22] Z. Chen et al., "InternVL: Scaling up vision foundation models and aligning for generic visual-linguistic tasks," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2024, pp. 24185–24198.

[23] W. Wang et al., "COGVLM: Visual expert for pretrained language models," in *Proc. Adv. Neural Inf. Process. Syst.*, 2024, pp. 121475–121499.

[24] H. Liu, C. Li, Q. Wu, and Y. J. Lee, "Visual instruction tuning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2023, pp. 1–25.

[25] A. Radford et al., "Learning transferable visual models from natural language supervision," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 8748–8763.

[26] H. Touvron et al., "LLaMA: Open and efficient foundation language models," 2023, *arXiv:2302.13971*.

[27] W. Wang et al., "VisionLLM: Large language model is also an open-ended decoder for vision-centric tasks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2023, pp. 61501–61513.

[28] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Hum. Lang. Technol.*, 2019, pp. 4171–4186.

[29] J. Chen et al., "MiniGPT-V2: Large language model as a unified interface for vision-language multi-task learning," 2023, *arXiv:2310.09478*.

[30] R. Pi, L. Yao, J. Gao, J. Zhang, and T. Zhang, "PerceptionGPT: Effectively fusing visual perception into LLM," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2024, pp. 27124–27133.

[31] F. Yang et al., "HEIE: MLLM-based hierarchical explainable AIGC image implausibility evaluator," in *Proc. Comput. Vis. Pattern Recognit. Conf.*, 2025, pp. 3856–3866.

[32] J. H. Cho et al., "Language-image models with 3D understanding," 2024, *arXiv:2405.03685*.

[33] J.-J. Hwang et al., "EMMA: End-to-end multimodal model for autonomous driving," 2024, *arXiv:2410.23262*.

[34] P. Li, H. Zhao, P. Liu, and F. Cao, "RTM3D: Real-time monocular 3D detection from object keypoints for autonomous driving," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 644–660.

[35] Z. Wu, Y. Gan, L. Wang, G. Chen, and J. Pu, "MonoPGC: Monocular 3D object detection with pixel geometry contexts," in *Proc. IEEE/CVF Int. Conf. Robot. Autom.*, 2023, pp. 4842–4849.

[36] F. Yang et al., "Ground plane matters: Picking up ground plane prior in monocular 3D object detection," 2022, *arXiv:2211.01556*.

[37] L. Yan, P. Yan, S. Xiong, X. Xiang, and Y. Tan, "MonoCD: Monocular 3D object detection with complementary depths," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2024, pp. 10248–10257.

[38] X. Shi et al., "Geometry-based distance decomposition for monocular 3D object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 15152–15161.

[39] Y. Lu et al., "Geometry uncertainty projection network for monocular 3D object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 3091–3101.

[40] F. Yang et al., "GPRO3D: Deriving 3D BBOX from ground plane in monocular 3D object detection," *Neurocomputing*, vol. 562, 2023, Art. no. 126894.

[41] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[42] A. Zhang, W. Ji, and T. Chua, "Next-Chat: An LMM for Chat, Detection and Segmentation," 2023, *arXiv:2311.04498*.

[43] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 6000–6010.

[44] A. Dosovitskiy et al., "An image is worth 16x16 words: Transformers for image recognition at scale," in *Proc. Int. Conf. Learn. Representations*, 2021, pp. 1–22.

[45] Y. Li, K. Zhang, J. Cao, R. Timofte, and L. Van Gool, "Localvit: Bringing locality to vision transformers," 2021, *arXiv:2104.05707*.

[46] Z. Liu et al., "A convnet for the 2020 s," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 11966–11976.

[47] K.-C. Huang, T.-H. Wu, H.-T. Su, and W. H. Hsu, "MonoDTR: Monocular 3D object detection with depth-aware transformer," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 4002–4011.

[48] J. Bai et al., "Qwen-Vl: A frontier large vision-language model with versatile abilities," 2023, *arXiv:2308.12966*.

[49] Y. Zhou, C. Barnes, J. Lu, J. Yang, and H. Li, "On the continuity of rotation representations in neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 5745–5753.

[50] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 3354–3361.

[51] H. Caesar et al., "nuScenes: A multimodal dataset for autonomous driving," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 11618–11628.

[52] S. Ettinger et al., "Large scale interactive motion forecasting for autonomous driving: The waymo open motion dataset," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 9690–9699.

[53] J. Chen, H. Guo, K. Yi, B. Li, and M. Elhoseiny, "VisualGPT: Data-efficient adaptation of pretrained language models for image captioning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 18030–18040.

[54] Y. Li et al., "Mini-Gemini: Mining the potential of multi-modality vision language models," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2025, pp. 1–14.

[55] OpenAI, "GPT-4o: The cutting-edge advancement in multimodal LLM," 2024. [Online]. Available: https://openai.com/index/hello-gpt-4o/

[56] S. Song, S. P. Lichtenberg, and J. Xiao, "SUN RGB-D: A RGB-D scene understanding benchmark suite," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 567–576.

[57] A. Ahmadyan, L. Zhang, A. Ablavatski, J. Wei, and M. Grundmann, "Objectron: A large scale dataset of object-centric videos in the wild with pose annotations," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 7822–7831.

[58] W.-L. Chiang et al., "Vicuna: An open-source chatbot impressing GPT-4 with 90% * ChatGPT quality," Accessed: Apr. 14, 2023. [Online]. Available: https://vicuna.lmsys.org

[59] F. Yang et al., "Geometry-guided domain generalization for monocular 3D object detection," in *Proc. AAAI Conf. Artif. Intell.*, 2024, pp. 6467–6476.

[60] L. Piccinelli et al., "UniDepth: Universal monocular metric depth estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2024, pp. 10106–10116.

[61] L. Yang et al., "Depth anything: Unleashing the power of large-scale unlabeled data," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2024, pp. 10371–10381.

[62] A. Bochkovskii et al., "Depth Pro: Sharp monocular metric depth in less than a second," 2024, *arXiv:2410.02073*.
[63] A. Kirillov et al., "Segment anything," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2023, pp. 4015–4026.
[64] J. Wang et al., "VGGT: Visual geometry grounded transformer," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2025, pp. 5294–5306.

Fan Yang (Graduate Student Member, IEEE) received the B.E. degree in 2021 from the School of Software, Tsinghua University, Beijing, China, where he is currently working toward the Ph.D. degree with the School of Software. His research interests include multimodal large language models, reinforcement learning, and 3D perception.

Sicheng Zhao (Senior Member, IEEE) received the Ph.D. degree from the Harbin Institute of Technology, Harbin, China, in 2016. From 2013 to 2014, he was a Visiting Scholar with the National University of Singapore, Singapore, Postdoctoral Research Fellow with the University of California at Berkeley, Berkeley, CA, USA, from 2017 to 2020, and Postdoctoral Research Scientist with Columbia University, New York, NY, USA, from 2020 to 2022. He is currently a Research Associate Professor with Tsinghua University. His research interests include affective computing, multimedia, and computer vision.

Yanhao Zhang received the Ph.D. degree from the Harbin Institute of Technology, Harbin, China, in 2017. He is currently an Expert Researcher with the OPPO AI Center. Prior to joining OPPO, in 2022, he was a Senior Researcher with Alibaba Damo Academy, Beijing, China. His research interests include AIGC, multimodal large language model, multimodal understanding, and generation.

Hui Chen is currently an Assistant Researcher with the Beijing National Research Center for Information Science and Technology, Tsinghua University, Beijing, China. He has authored or coauthored more than 15 peer-reviewed top conference and journal papers, including CVPR, ICCV, and ICLR. His research interests include efficient and effective multi-modal perception and learning.

Haonan Lu received the Ph.D. degree from The Chinese University of Hong Kong, Hong Kong, in 2017. From 2014 to 2017, he was a Visiting Scholar with The European Organization for Nuclear Research, Switzerland, and a Senior Research Engineer with Huawei, from 2018 to 2021. He is currently the Head with Multimodal Algorithm Team, OPPO's AI Center. His research interests include natural language processing, multimodal, and graph representation learning.

Jungong Han (Senior Member, IEEE) is currently a Professor with the Department of Automation, Tsinghua University, Beijing, China. He also holds an Honorary Professorship with the University of Warwick, Coventry, U.K. His research interests include computer vision, artificial intelligence, and machine learning. He is a Fellow of IAPR and a Fellow of AAIA.

Guiguang Ding (Senior Member, IEEE) is currently a Tenured Professor with the School of Software, Tsinghua University, Beijing, China. He has more than 30 papers published in top-tier journals including IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, *IEEE Signal Processing Magazine*, and IEEE TRANSACTIONS ON IMAGE PROCESSING. His research interests include model architecture design and compression, visual semantic recognition and description, transfer learning, and few-shot learning. He has presented more than 70 papers at top-tier international conferences, such as CVPR, NeurIPS, and ICML.