
MLLM-For3D: Adapting Multimodal Large Language Model for 3D Reasoning Segmentation

Jiaxin Huang¹ Runnan Chen^{2,†} Ziwen Li¹ Zhengqing Gao¹ Xiao He⁴ Yandong Guo⁴
Mingming Gong^{1,3} Tongliang Liu^{1,2,†}

¹MBZUAI ²The University of Sydney ³The University of Melbourne ⁴AI2Robotic

Abstract

Reasoning segmentation aims to segment target objects in complex scenes based on human intent and spatial reasoning. While recent multimodal large language models (MLLMs) have demonstrated impressive 2D image reasoning segmentation, adapting these capabilities to 3D scenes remains underexplored. In this paper, we introduce **MLLM-For3D**, a simple yet effective framework that transfers knowledge from 2D MLLMs to 3D scene understanding. Specifically, we utilize MLLMs to generate multi-view pseudo-segmentation masks and corresponding text embeddings, then unproject 2D masks into 3D space and align them with the text embeddings. The primary challenge lies in the absence of 3D context and spatial consistency across multiple views, causing the model to hallucinate objects that do not exist and fail to target objects consistently. Training the 3D model with such irrelevant objects leads to performance degradation. To address this, we first filter irrelevant views using token attention. With these reliable pseudo-labels, we develop a token-for-Query approach for multimodal semantic alignment, enabling consistent identification of the same object across different views. Moreover, we introduce a spatial consistency strategy to enforce that segmentation masks remain coherent in the 3D space, effectively capturing the geometry of the scene. Extensive evaluations of various challenging indoor scene benchmarks demonstrate that, even without labeled 3D training data, **MLLM-For3D** outperforms existing 3D reasoning segmentation methods, effectively interpreting user intent, understanding 3D scenes, and reasoning about spatial relationships.

1 Introduction

Understanding user intent and reasoning about the 3D spatial context are crucial for real-world vision applications [38, 39, 31, 43, 58], including embodied AI, autonomous driving, and augmented/virtual reality. Recent advances in 3D scene understanding, particularly in multimodal learning [20, 69, 22, 7, 21, 46, 9, 16], have spurred the development of 3D point cloud-based Large Language Models (3D-LLMs). These methods allow systems to infer implicit goals, localize objects in complex environments, and interact seamlessly with users. Compared to conventional segmentation, reasoning segmentation poses a more complex challenge, requiring deeper levels of semantic understanding and the ability to handle underspecified or context-dependent queries.

The concept of reasoning segmentation was first introduced in the 2D domain by LISA [36], which employed the mask-as-embedding paradigm to fine-tune large language models using abundant (2D, 3D) paired data. Based on its success in 2D domains, initial attempts have been made to adapt this paradigm to 3D tasks [19, 23, 30]. However, these approaches have the prohibitive cost of generating high-quality pairs (3D, text), which often involve labor intensive manual annotation or

[†]Corresponding authors

^{*}Code available at: https://github.com/tm1lab/2025_NeurIPS_MLLM-For3D

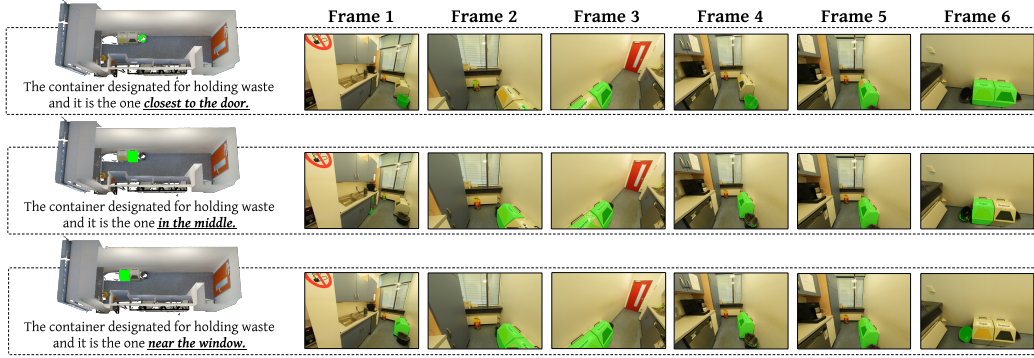


Figure 1: For the same scene, we present three different queries and display the 2D reasoning segmentation results on the same set of frames, illustrating how the model responds to varying instructions.

computationally expensive synthesizing (e.g., by GPT-4) [1]. A natural question thus arises: *Given a 3D point cloud accompanied by multiple posed RGB views, can we exploit pre-trained 2D reasoning segmentation models to approximate 3D labels?*

Our pilot study illustrates both the promise and the limitations of the idea (Figure 1). We evaluate the performance of 2D reasoning segmentation models, LISA [36] on the ScanNet++ dataset [64], by providing the model with different views corresponding to a 3D scene along with an implicit language instruction. Each frame is processed independently, generating both a reasoning response and a 2D binary segmentation mask. Ideally, the model should consistently localize the target object across all frames and infer its spatial relationships (e.g., nearest door) within the scene. However, as shown in Figure 1, there are two key limitations to the reasoning of 2D models in 3D: (i) *False positives in invisible views*: A 2D MLLM processing a single view might hallucinate or mistakenly segment objects that are described by the instruction but not visible in that particular view. Without 3D awareness, the model cannot distinguish between visible and occluded target objects, leading to incorrect mask predictions on some views. (ii) *Multi-View prediction inconsistency* occurs when the model lacks a mechanism to ensure spatial alignment of predictions across views, thereby degrading the performance when we aggregate multi-view predictions into 3D.

In this paper, we introduce **MLLM-For3D**, a simple yet effective framework that transfers 2D MLLM reasoning capabilities to 3D scene understanding. Specifically, we use a pre-trained MLLM to generate multi-view pseudo-segmentation masks and corresponding text embeddings. These masks are then unprojected into 3D space and aligned with the textual information to supervise the learning of the 3D model, eliminating the need for explicit 3D annotations. To address the critical issue of cross-view inconsistencies, we incorporate a **spatial consistency strategy** into the mask generation process, ensuring that the latent space remains coherent and mitigating object hallucinations. Specifically, we enforce latent space consistency by aggregating the per-view predictions via an attention-based fusion module. In this module, for a given 3D point visible in multiple views, the contribution of each view is weighted by its reliability and semantic similarity to a unified query derived from the [SEG] token embeddings. Furthermore, we propose a **token-for-Query** mechanism that consistently binds the same object identity across different views, enhancing the ability of the 3D model to interpret implicit user instructions and reason about spatial relationships.

MLLM-For3D is evaluated on three challenging benchmarks and shows that it achieves state-of-the-art performance in 3D reasoning segmentation tasks even without the need for any 3D annotations, which achieves about 55% higher mIoU than the previous methods. By effectively transferring 2D MLLM reasoning capabilities to 3D, our framework exhibits strong robustness to ambiguous queries, improved spatial reasoning, and superior segmentation performance.

The key contributions of our work are summarized as follows.

- We propose a simple, yet effective framework to adapt 2D MLLMs for 3D reasoning segmentation, eliminating the need for manual 3D annotations.

- We introduce a novel alignment mechanism that binds token embeddings to specific queries, ensuring consistent segmentation of the same object across views.
- We integrate a spatial consistency strategy to refine multi-view pseudo-segmentation masks, reducing the presence of hallucinated objects across frames.
- Extensive evaluations of two challenging indoor scene benchmarks demonstrate that MLLM-For3D outperforms existing 3D reasoning segmentation methods, even without any 3D labeled training data.

2 Related Works

2.1 Reasoning Segmentation

Reasoning segmentation is first introduced by LISA [36], which integrates a multimodal LLM (e.g., LLaVA [40]) with the Segmentation Anything Model (SAM) [35] to handle complex and *implicit* instructions in 2D images. PixelLM [49] builds on this paradigm by adopting a lightweight decoder and segmentation codebook for multi-object reasoning segmentation. LLM-Seg [54] uses SAM to propose candidate masks and allows the LLM to reason which mask fits the query. VISA [61] and VideoLISA [2] extend these approaches to video data, addressing temporal coherence and object tracking. FAST [51], an agent-based pipeline, further refines segmentation masks by iteratively identifying and masking key objects. These advancements in 2D [60, 67, 56, 53, 26] demonstrate the value of combining segmentation with LLM reasoning: models can interpret rich instructions and produce the corresponding mask, which is not possible with traditional segmentation alone.

In 3D Domains, PARIS3D [33] and Reasoning3D [8] focus on part segmentation with explanatory capabilities for individual objects, leaving scene-level reasoning tasks relatively unexplored. More recently, SegPoint [19], Reason3D [23], and MORE3D [30] have adapted the embedding-as-mask paradigm from LISA, aiming to unify multiple 3D tasks through human-like instructions. In parallel, Point-Bind and Point-LLM [16] extend 3D understanding to the multi-modal domain by aligning point clouds with images, language, audio, and video, and further enabling 3D large language models to follow multi-modal instructions. Despite these advances, such methods typically rely on large-scale (3D, text) training data or parameter-efficient fine-tuning of LLMs, both of which are computationally costly. In contrast, our approach alleviates this problem by distilling reasoning capabilities and semantic knowledge from 2D MLLMs into a 3D model, enabling label-free 3D reasoning segmentation without any 3D supervision.

2.2 Label-Free 3D Scene Understanding

Open-Vocabulary and Zero-Shot Approaches. To alleviate the annotation burden, several label-free scene understanding methods [45, 44, 29, 25] leverage *vision foundation models* for zero-shot 3D segmentation. OpenScene [45] employs 2D open vocabulary segmentors [37, 15] to align pixel-level embeddings with 3D points, allowing object category recognition for unseen classes. CLIP2Scene [6] employs MaskCLIP [68] to obtain pixel-aligned features for annotation-free and label-efficient scene understanding. ConceptFusion [28] and CLIP-FO3D [65] further explore the acquisition of pixel-aligned knowledge through the extraction of dense region-level features using CLIP [47] and multi-view feature fusion. PLA [13] proposed a language-driven 3D scene understanding paradigm, which obtains point-language paired data through image captioning by visual-language foundation models for training 3D backbones. Similarly, RegionPLC [63] and Lowis3D [14] build point-caption pairs by projecting 2D visual-language features onto 3D geometry. Methods like OVIR3D [42] and MaskClustering [62] merge zero-shot 2D masks with 3D semantics for instance segmentation. Recent efforts [5, 29] also combine different foundation models (e.g., LLaVa-1.5 [40] and SEEM [70]) to unify zero-shot 2D embeddings and 3D point features, demonstrating strong category expansion in 3D. In parallel, LERF [34] introduces a 3D language grounding method that distills CLIP embeddings into NeRF volumes by optimizing a multi-scale language field through volume rendering and enforcing multi-view consistency.

Label-Free 3D Reasoning Segmentation. While these label-free strategies effectively handle open-vocabulary classes, their prompts remain relatively straightforward, limiting their ability to interpret more nuanced or context-heavy queries. In this work, we target *implicit* user prompts that require both semantic and spatial comprehension. Our framework, **MLLM-For3D**, inherits the reasoning

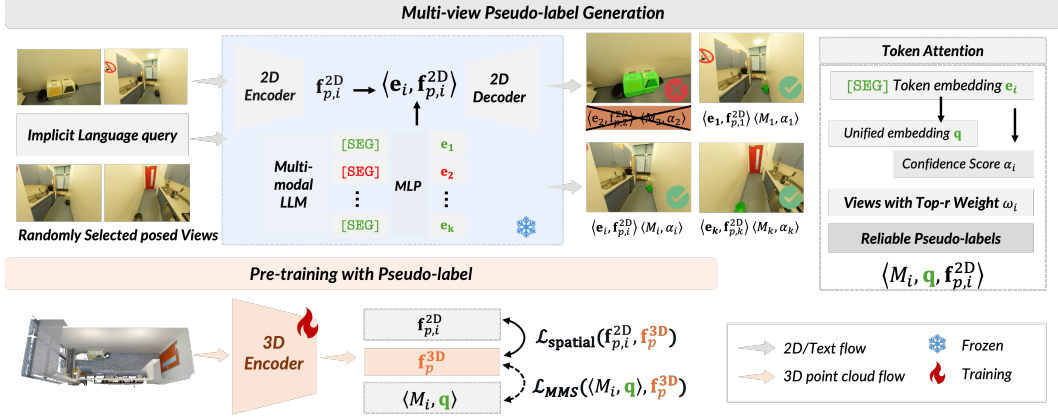


Figure 2: Overview of the proposed **MLLM-For3D** framework. We adapt multimodal large language models (MLLMs) for 3D reasoning segmentation by generating multi-view pseudo-labels and filtering irrelevant views via token attention. During the training phase, we enforce cross-view consistency via a spatial consistency strategy and align an unified embeddings \mathbf{q} with 3D per-point feature \mathbf{f}_p^{3D} via a multimodal semantic loss, enabling consistent object identity binding across views.

capabilities of 2D MLLM and applies them to 3D *without any annotations*, making it both *scalable* and *intuitive* in real-world scenarios.

3 Methodology

We propose MLLM-For3D, a label-free framework that adapts a 2D MLLM for 3D reasoning-based segmentation. As illustrated in Figure 2, a frozen MLLM and a 2D segmentation model are jointly used to generate multi-view pseudo-labels (2D binary masks M_i and associated [SEG] token embedding \mathbf{e}_i) from randomly selected posed views. However, not all views capture the queried object (discussed in Section 1). To address this, we introduce an attention-based view filtering mechanism to select reliable views. Finally, a 3D segmentation network is trained under pseudo-label supervision, incorporating both semantic alignment and spatial consistency across views. All 2D models remain frozen, and only the 3D network parameters are optimized.

3.1 Multi-View Pseudo-Label Generation

MLLM+SAM for Per-View Segmentation. For each 3D scene, we assume a set of posed RGB images covering the scene, along with a textual query implicitly describing the target object. Previous multi-view 3D understanding approaches [57, 17, 18] demonstrate that view selection and aggregation are crucial for robust 3D perception. Therefore, we randomly select k camera views and feed each image-text pair into a frozen MLLM to produce a special [SEG] token embedding \mathbf{e}_i (for the i -th view), which semantically represents the queried segment in that view. The [SEG] embedding is then passed to the integrated SAM decoder within LISA to generate a binary mask M_i and a confidence score α_i (predicted IoU). Repeating this process yields a collection of candidate 2D masks M_i , each hypothesizing the object’s location based on 2D MLLM reasoning.

View Filtering via Token Attention. Since not all selected views contain the object, some M_i may be empty or incorrect. To mitigate this, we first discard masks with very low confidence or area and then compute an attention weight ω_i to quantify each view’s reliability. Each remaining embedding \mathbf{e}_i contributes to a unified query embedding \mathbf{q} via attention-weighted fusion. The attention weight ω_i reflects both mask confidence and semantic alignment: Formally, we define an attention score using the similarity of the dot product between \mathbf{e}_i and \mathbf{q} in a shared latent space. Let $s_i = \mathbf{e}_i \cdot \mathbf{q}$ denote the semantic alignment of the prediction of view i with the unified embedding. We then set $\omega_i \propto \alpha_i \cdot \max(0, s_i)$ and normalize ω_i in all views so that $\sum_i \omega_i = 1$. Views with higher semantic consistency and clearer object visibility receive larger weights, while noisy or occluded views are effectively suppressed. This token attention mechanism thus reweights each view’s contribution during 3D fusion, ensuring that only semantically coherent masks dominate the final pseudo-label set.

During fusion, each unprojected 3D mask is scaled by its corresponding token attention weight. The final prediction is the weighted average of these masks, where weights reflect the semantic confidence of each view.

While additional input views can provide more context, we observe that naïvely increasing view count beyond 4 degrades performance due to occlusions, inconsistent reasoning, and hallucinated [SEG] tokens from partially visible objects. Our token attention mechanism mitigates this by weighting each view based on semantic alignment, effectively filtering informative views.

3.2 Training with Pseudo-Labels

With reliable multi-view pseudo-labels, we train a 3D segmentation model that learns to localize the target object in the point cloud while enforcing cross-view semantic alignment and spatial consistency. The model encodes per-point feature \mathbf{f}_p^{3D} , and only the 3D network and projection layers are trainable. All 2D components (MLLM + SAM) are frozen and serve solely as feature extractors. We reuse the SAM decoder within LISA to extract 2D binary masks while separately using LISA’s reasoning head to obtain [SEG] tokens. This modular design decouples semantic reasoning and geometric supervision, making the framework compatible with other MLLMs or mask generators.

Multi-modal Semantic Alignment. Inspired by previous works [6, 5], we develop a token-for-Query mechanism to align multimodal semantics. We first establish point–pixel correspondences such that for a 3D point p and view i , if point p projects to pixel (u, v) in the image and that pixel lies inside the mask ($M_i(u, v) = 1$), then view i votes that point p is part of the target segment. Conversely, if p is visible in view i but falls outside the mask, that view votes that p is not part of the segment. After processing all views, each point p accumulates multiple predictions from different views. We then aggregate these multi-view predictions using the attention weights. Importantly, along with these binary masks, we also associate each point with multi-modal features: for every view i that observed point p , we retrieve the vector of features of the image $\mathbf{f}_{p,i}^{2D}$ from the SAM encoder in the corresponding pixel, and we carry the unified embedding \mathbf{q} as a semantic token.

To align each 3D point with the unified embedding \mathbf{q} , we first map it using a learned linear projection due to differences in the dimensions of the modality. Let \mathbf{t} denote the transformed unified query embeddings. Next, we compute semantic logits for each paired point taking the dot product between the transformed embedding and the 3D point features $s_p = \mathbf{f}_p^{3D} \cdot \mathbf{t}$ and the sample of corresponding predicted logits from the image-based segmentation predictions $\hat{m}_{p,i} = \hat{\mathbf{M}}_{i,u,v}$. Finally, we formulate the semantic alignment loss (\mathcal{L}_{MMS}) using binary cross-entropy (BCE) to minimize discrepancies between 3D logits and the sampled 2D predicted logits:

$$\mathcal{L}_{\text{MMS}} = \text{BCE}(s_p, \hat{m}_{p,i}) = -\frac{1}{|\mathcal{T}|} \sum_{p,i \in \mathcal{T}} [\hat{m}_{p,i} \log(\sigma(s_p)) + (1 - \hat{m}_{p,i}) \log(1 - \sigma(s_p))], \quad (1)$$

where $\sigma(\cdot)$ denotes the sigmoid activation function, and \mathcal{T} represents the set of all paired points. This token-for-query-based alignment ensures that the same object (as described by the implicit text) yields a consistent latent representation across different views. Intuitively, minimizing \mathcal{L}_{MMS} encourages the 3D model to produce consistent latent representations for the target object described implicitly by the textual query, thereby anchoring the 3D semantic features closely to the unified semantic embedding across all relevant views.

Spatial Consistency Loss. While \mathcal{L}_{MMS} binds the network to the unified semantic token, we also enforce spatial consistency between 3D and 2D modalities. For each pseudo-labeled point p , recall that we have one or more associated image feature vectors $\mathbf{f}_{p,i}^{2D}$ from the views detected p . These features capture the appearance of the object from those viewpoints. We impose the 3D point feature \mathbf{f}_p^{3D} to be close to the image features of the same point, which encourages the 3D model to agree with the 2D observations and maintain cross-view coherence. Concretely, we define a spatial consistency loss using cosine similarity. For each pair (p, i) (point p visible in view i with feature $\mathbf{f}_{p,i}^{2D}$), we maximize the cosine similarity $\cos(\mathbf{f}_p^{3D}, \mathbf{f}_{p,i}^{2D})$. Equivalently, we minimize:

$$\mathcal{L}_{\text{spatial}} = \frac{1}{|\mathcal{T}|} \sum_{(p,i) \in \mathcal{T}} \left(1 - \frac{\mathbf{f}_p^{3D} \cdot \mathbf{f}_{p,i}^{2D}}{\|\mathbf{f}_p^{3D}\| \|\mathbf{f}_{p,i}^{2D}\|} \right), \quad (2)$$

where \mathcal{T} is the set of all point–view pairs for which point p was labeled as target in view i . Minimizing $\mathcal{L}_{\text{spatial}}$ drives \mathbf{f}_p^{3D} and $\mathbf{f}_{p,i}^{2D}$ in the same direction. We combine these objectives into the overall training

loss for the 3D network:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{MMS}} + \lambda \mathcal{L}_{\text{spatial}}, \quad (3)$$

with a balancing coefficient λ .

3.3 Inference on 3D Scenes

At inference, the model takes as input a set of multi-view images of an unseen 3D scene and an implicit user instruction with the point cloud. Following the same pseudo-label generation pipeline, we compute a unified embedding \mathbf{q} from the [SEG] tokens. The 3D model produces per-point features \mathbf{f}_p^{3D} , and cosine similarity between \mathbf{f}_p^{3D} and \mathbf{t} determines each point’s probability of belonging to the queried object. Points exceeding a threshold are classified as part of the target, yielding the final 3D segmentation mask.

4 Experiments

In this section, we present the experimental results for three challenging tasks, focusing on 3D reasoning segmentation, intention grounding. For the *3D reasoning segmentation* task, we adopt Reason3D [23] (derived from ScanNet [12] and Matterport3D [3]) and Instruct3D [19] (derived from ScanNet++ v1 [64]) as benchmarks. Due to the limited open-source evaluation datasets in this area, we try to evaluate our method on similar tasks as 3D intention grounding (3D-IG) [32] and grounding without object names (VG-w/o-ON), which is an interesting benchmark introduced by [59]. These two datasets are built on ScanNet [12], taking advantage of its scene annotations for evaluation. Refer to Appendix A for more about the dataset and implementation details. Following SegPoint [19] and Reason3D [23], we evaluate in two stages: (1) Accuracy@kIoU (k=0.25/0.5) for target grounding correctness, and (2) mIoU for segmentation quality after successful grounding. This ensures that the evaluation reflects reasoning ability before mask quality.

4.1 Main Results

3D Reasoning Segmentation. As shown in Table 1, MLLM-For3D shows significant gains in 3D Reasoning Segmentation Precision over Reason3D on the Instruct3D benchmark [19]. Removing explicit instructions and annotations from Instruct3D makes the task much harder, since models can no longer rely on direct object names or step-by-step instructions and must infer the user’s intent from implicit hints. In this setting, MLLM-For3D achieves about 15% higher Acc@0.25 and 10% higher Acc@0.50 than Reason3D. It also improves the mean IoU by 10 points, indicating more precise mask predictions. These improvements highlight MLLM-For3D’s stronger reasoning capability: it can understand complex or implicit instructions to segment the correct 3D regions even when keywords are missing. In contrast, the Reason3D baseline struggles without explicit instructions, since it was designed to output masks based on more direct textual descriptions. In general, MLLM-For3D’s ability to interpret implicit instructions leads to better performance on the 3D reasoning segmentation task. The "MLLM-based Model (w/ label)" serves as a reference baseline combining multi-view pseudo-labels and available 3D ground-truth masks. Both supervise the 3D network jointly under a hybrid training regime, using MinkowskiNet14 [10] as the backbone.

3D Intention Grounding (3D-IG). 3D Intention Grounding is a novel challenging task, which requires detecting the object that fulfills an implicit human intention. This setting is quite similar to what we have defined in the previous Section 1, but still ignores the spatial relations. IntentNet, a task-specific model, achieves 41.9% AP@0.25 and 25.4% AP@0.50 on this benchmark. MLLM-For3D surpasses this with 6–7 points higher AP@0.25 and 5 points higher AP@0.50, indicating that it more reliably identifies the correct object from only the implied intent.

For further comparison, shown in Table 2, our MLLM-For3D is evaluated under two settings: (i) with labeled 3D-text data (pink rows) and (ii) without labels (yellow rows). Despite the absence of manual 3D annotations in the label-free setting, our model surpasses the specialized IntentNet[32] by a notable margin. For example, comparing the best label-free variant VideoLISA (MLLM-For3D) with IntentNet, we observe an improvement of +9.0 points in AP@0.25 (41.90% \rightarrow 50.89%) and +16.2 points in AP@0.50 (25.36% \rightarrow 41.56%). This indicates that a multimodal LLM can better interpret various intention phrases and incorporate broader world knowledge than a system trained on fixed detection templates. Meanwhile, Reason3D [23] remains slightly ahead of our approach.

Table 1: **3D Reasoning Segmentation Results.** Comparison across ScanNet, Matterport3D, and ScanNet++ datasets. The evaluation metrics include accuracy at IoU thresholds 0.25, 0.50, and mean IoU. † denotes models fine-tuned using the filtered Instruct3D training set for fair comparison. ⁰ denotes a zero-shot experiment. The color gradient indicates different MLLM backbones and training conditions (w/ label or w/o label) for MLLM-For3D models.

3D Reasoning Segmentation	Venue	Modality	Reason3D (ScanNet)			Reason3D (Matterport3D)			Instruct3D (ScanNet++)		
Methods			Acc@0.25	Acc@0.50	mIoU	Acc@0.25	Acc@0.50	mIoU	Acc@0.25	Acc@0.50	mIoU
<i>non-LLM-based Model (w/ label)</i>											
TGNN [†] [24]	[AAAI'22]	3D	-	-	-	-	-	-	4.76	4.76	3.51
3D-STMN [†] [58]	[AAAI'24]	3D	25.43	17.78	18.23	20.68	10.81	13.47	-	-	-
Intent3D ⁰ [32]	[ICLR'25]	3D	29.12	19.26	-	19.70	12.83	-	9.71	3.20	-
Intent3D [†] [32]	[ICLR'25]	3D	20.57	19.46	-	13.52	9.42	-	23.30	20.41	-
<i>LLM-based Model (w/ label)</i>											
Segpoint [19]	[ECCV'24]	3D	-	-	-	-	-	-	23.7	15.6	17.2
Reason3D [23] [†]	3DV'25	3D	43.21	32.10	31.20	31.22	17.43	19.54	18.35	10.55	12.43
<i>MLLM-based Model (w/ label)</i>											
LISA-7B (MLLM-For3D)	-	3D+2D	45.53	39.54	31.94	38.31	31.46	22.56	45.50	38.70	31.90
LISA-13B (MLLM-For3D)	-	3D+2D	47.32	39.92	31.44	39.42	32.43	23.20	46.70	39.00	32.30
VideoLISA (MLLM-For3D)	-	3D+2D	48.45	41.02	39.82	39.81	29.40	24.97	48.20	40.40	34.50
<i>MLLM-based Model (w/o label)</i>											
LISA-7B (MLLM-For3D)	-	3D+2D	39.38	31.27	30.19	30.10	22.04	20.71	39.10	29.70	23.90
LISA-13B (MLLM-For3D)	-	3D+2D	40.92	33.40	32.10	31.68	23.33	20.78	40.90	30.50	26.40
VideoLISA (MLLM-For3D)	-	3D+2D	44.18	34.80	32.90	33.41	27.99	22.50	41.00	32.10	28.20

Table 2: **Evaluation of 3D Intention Grounding on the Intent3D [32] validation set.** The best results are in **bold**, and the second-best results are underlined. † indicates that we re-trained Reason3D using the Intent3D training set for a fair comparison on this benchmark. ⁰ indicates a zero-shot setting.

3D Intention Grounding Methods	Venue	Modality	Language Backbone	Intent3D (val)				
				Acc@0.25	Acc@0.50	AP@0.25	AP@0.50	mIoU
<i>non-LLM-based Model</i>								
BUTD-DETR [27]	[ECCV'22]	3D	RoBERTa [41]	47.12	24.56	31.05	13.05	-
EDA [59]	[CVPR'23]	3D	RoBERTa [41]	43.11	18.91	14.02	5.00	-
3D-VisTA [69]	[ICCV'23]	3D	-	42.76	30.37	36.10	19.93	-
IntentNet [32]	[ICLR'25]	3D	RoBERTa [41]	58.34	40.83	41.90	25.36	-
<i>LLM-based Model</i>								
Chat-3D-v2 ⁰ [21]	[NIPS'24]	3D+2D	Vicuna-7B [52]	5.86	5.24	0.15	0.13	-
Chat-Scene [21]	[NIPS'24]	3D+2D	Vicuna-7B [52]	36.71	32.78	3.23	2.58	-
Reason3D [†] [23]	[3DV'25]	3D	Flan-T5 [11]	61.71	51.68	-	-	47.30
<i>MLLM-based Model (w/ label)</i>								
LISA-7B (MLLM-For3D)	-	3D+2D	LLaVA-2 [40]	57.31	47.92	-	-	42.98
LISA-13B (MLLM-For3D)	-	3D+2D	LLaVA-2 [40]	58.40	48.75	-	-	44.13
VideoLISA (MLLM-For3D)	-	3D+2D	LLaVA-Phi-3-V [48]	<u>59.90</u>	<u>50.01</u>	-	-	<u>45.18</u>
<i>MLLM-based Model (w/o label)</i>								
LISA-7B (MLLM-For3D)	-	3D+2D	LLaVA-2 [40]	48.24	39.61	-	-	34.53
LISA-13B (MLLM-For3D)	-	3D+2D	LLaVA-2 [40]	49.92	40.10	-	-	35.75
VideoLISA (MLLM-For3D)	-	3D+2D	LLaVA-Phi-3-V [48]	50.89	41.56	-	-	36.92

We attribute this gap to Reason3D’s mask-as-embedding paradigm, which excels at implicit intent reasoning by fine-tuning LLM for direct segmentation tokens. However, MLLM-For3D still shows strong generalization in human intention reasoning, effectively combining detection and segmentation reasoning in a label-free manner. In contrast, IntentNet is primarily tailored for detection, and Reason3D focuses on search-plus-segmentation. The multimodal design of our method instead offers a more balanced approach, achieving second-best results while using no labels.

VG-w/o-ON: Visual Grounding without Object Names In the 3D visual grounding without object names task, MLLM-For3D achieves state-of-the-art results as shown in Table 3, outperforming the EDA baseline and others. VG-w/o-ON is a particularly challenging benchmark variant where the language query describes the spatial relationship of the target object without explicitly naming it. Conventional 3D referring models struggle here since they typically rely on matching object names in the query. In fact, we observe a drastic drop in baseline performance: methods such as ScanRefer [4] and TGNN [24] see their accuracy plunge to nearly chance level (e.g., 10% success) when object names are missing. Even EDA [59] reaches only 26. 5% Acc@0.25 and 21. 6% Acc@0.50 when the object names are missing, much lower than in normal queries. MLLM-For3D

Table 3: **Evaluation results on VG-w/o-ON (val) evaluated by Acc and mIoU**; * auxiliary mask head. † indicates that we re-trained Reason3D using the ScanRefer[4] training set for a fair comparison on this benchmark.

Methods	Venue	Acc@0.25 ↑	Acc@0.50 ↑	mIoU ↑
<i>LLM-based Model (w/ label)</i>				
ScanRefer [4]	[ECCV'20]	10.51	6.20	-
TGNN* [24]	[AAAI'21]	11.64	9.51	8.13
InstanceRefer [42]	[ICCV'21]	13.92	11.47	-
BUTD-DETR [27]	[ECCV'22]	11.99	8.95	-
M3DRef-CLIP [66]	[ICCV'23]	18.3	14.8	10.29
EDA [59]	[CVPR'23]	26.50	21.20	-
Reason3D [23]	[3DV'25]	17.64	13.11	13.05
IntentNet [32]	[ICLR'25]	28.12	22.63	18.92
<i>MLLM-based Model (w/ label)</i>				
LISA-7B (MLLM-For3D)†	-	31.88	29.90	28.10
LISA-13B (MLLM-For3D)	-	32.52	30.15	29.81
VideoLISA (MLLM-For3D)	-	33.12	31.21	30.45
<i>MLLM-based Model (w/o label)</i>				
LISA-7B (MLLM-For3D)	-	26.49	22.12	21.05
LISA-13B (MLLM-For3D)	-	27.31	24.49	23.93
VideoLISA (MLLM-For3D)	-	<u>29.50</u>	<u>25.61</u>	<u>24.28</u>



Figure 3: Visual comparisons of our MLLM-For3D versus a previous state-of-the-art method Reason3D on Instruct3D datasets. For each row, we show the ground-truth rendered scene (left), the baseline’s prediction, our result, and the textual query. Our method accurately interprets implicit user instructions and produces coherent 3D masks.

overcomes this limitation by using the descriptive and contextual instructions of the query to infer the target. It delivers roughly 12–13% higher Acc@0.25 and 8–9% higher Acc@0.50 than EDA on the VG-w/o-ON benchmark, along with a notable improvement in mIoU.

Such results denote that our approach is capable of spatial reasoning by connecting the spatial relationship to the correct object in space. This contextual reasoning allows it to maintain high grounding accuracy despite the missing noun, whereas methods like EDA falter because they try to decompose the sentence and end up misled or unsure without an explicit object name. Using the abundant semantic information of a 2D MLLM, our method fills the semantic gaps (the missing object names) with informed guesses and uses the 3D visual input to confirm those guesses. This results in superior grounding performance under this no-name condition.

4.2 Visual Comparisons

Figure 3 presents qualitative examples comparing our **MLLM-For3D** framework against a previous state-of-the-art baseline on 3D reasoning segmentation tasks. Each row shows the ground-truth rendered scene, the predicted mask from the baseline, our result, and the text query. In the first example (left two columns), for the instruction ‘The container designated to hold waste and it is the

Table 4: Ablation study evaluating the effectiveness of each proposed component and view configuration on the Instruct3D and VG-w/o-ON validation sets. Colored values indicate performance gain or drop compared to baseline (a).

Ablation Target Setting	$+\mathcal{L}_{\text{MMS}}$	$+\mathcal{L}_{\text{spatial}}$	Instruct3D (val)			VG-w/o-ON (val)		
			Acc@0.25	Acc@0.50	mIoU	Acc@0.25	Acc@0.50	mIoU
(a) Baseline (LISA-7B)			29.25 _(ref)	25.26 _(ref)	19.75 _(ref)	20.20 _(ref)	17.24 _(ref)	11.56 _(ref)
(b) w/o Token-for-Query		✓	33.92 _(+4.67)	28.53 _(+3.27)	20.93 _(+1.18)	24.50 _(+4.30)	20.70 _(+3.46)	19.92 _(+8.36)
(c) w/o Spatial Consistency	✓		34.53 _(+5.28)	28.75 _(+3.49)	21.02 _(+1.27)	23.20 _(+3.00)	19.10 _(+1.86)	18.48 _(+6.92)
(d) #View 2	✓	✓	33.21 _(+3.96)	26.45 _(+1.19)	21.30 _(+1.55)	23.04 _(+2.84)	20.13 _(+2.89)	18.92 _(+7.36)
(e) #View 8	✓	✓	39.08 _(+9.83)	29.10 _(+3.84)	23.50 _(+3.75)	25.22 _(+5.02)	22.00 _(+4.76)	20.71 _(+9.15)
(f) Full Config (LISA-7B)	✓	✓	39.10 _(+9.85)	29.70 _(+4.44)	23.90 _(+4.15)	26.49 _(+6.29)	22.12 _(+4.88)	21.05 _(+9.49)
(g) Full Config (LISA-13B)	✓	✓	40.90 _(+11.65)	30.50 _(+5.24)	26.40 _(+6.65)	27.31 _(+7.11)	24.49 _(+7.25)	23.93 _(+12.37)

closest to the door’, the baseline incorrectly merges multiple objects or fails to capture the target, whereas **MLLM-For3D** precisely identifies only the correct container. In another example (right columns), given ‘If you want to let in natural light and fresh air once the air starts at night, which part of the room would you open?’, the baseline misses key portions under occlusion, while our approach segments the relevant object more completely. These comparisons show that our model follows high-level instructions more faithfully, resolving common failure modes by leveraging LLM-driven 3D reasoning.

4.3 Ablation Studies & Analysis

We conduct extensive experiments to verify the contribution of each component and view design. The results are shown in Table 4.

(a) 2D Projection Baseline. We first apply a projection-only baseline where 2D masks from LISA are unprojected to 3D directly, without spatial reasoning. We unproject the multi-view masks generated by LISA-7B back to the point cloud. This naive approach yields a much lower accuracy, roughly 30% lower mIoU than our full 3D method. The projected baseline often produces incomplete or misaligned segmentations, since each view sees only part of the scene without cross-view consistency enforcement. This highlights the limited spatial reasoning ability of the existing 2D reasoning segmentation model.

(b) w/o Token-for-Query. Removing the token-guided semantic alignment leads to false positives. The model cannot consistently localize the queried object, activating multiple regions per query. With the token-for-query in place, the model focuses on one target at a time, reducing false positives by 30%. This mechanism ensures that one coherent mask per query and consistent segmentation is provided, even when multiple queries are issued in one scene.

(c) w/o Spatial Consistency. Disabling this module leads to inconsistent masks and a drop of 2-4% in segmentation accuracy. Enforcing spatial consistency across views improves performance: By aligning features in 3D space, the model learns a unified segmentation that is viewpoint-invariant, resolving ambiguities from single perspectives.

(d-f) View Number Ablation: We evaluate the impact of the number of views (2, 4, and 8) used during 2D inference. As shown in Table 4, **2 views** leads to lower accuracy, as many occluded or peripheral objects are invisible from sparse views, resulting in incomplete 3D supervision. **8 views** slightly improves over 2 views but falls short of the 4-view setup. Although it offers more contextual information, it also introduces redundant or conflicting signals from occluded perspectives. In practice, increasing the number of views from 4 to 8 doubles the inference cost without a consistent performance gain. We empirically found 4 views to balance accuracy and efficiency. Repeating 5 random 4-view configurations on the Instruct3D validation set yielded stable results (Acc@0.25: 39.10±0.61, Acc@0.5: 29.70±0.48, mIoU: 23.90±0.52), confirming robustness of our token attention mechanism to stochastic view sampling.

(f,g) LISA-13B Backbone: We also test a stronger MLLM (13B vs 7B). A larger model improves reasoning ability and segmentation precision, showing that the architecture is scalable. Our best setup includes both modules, a balanced number of views, and a LISA-13B backbone.

Additional Comparison with 3D-MLLMs. To further validate the effectiveness of our label-free design, we additionally compare MLLM-For3D with representative 3D-MLLM pipelines that combine large language models with 3D segmentation backbones, including ChatScene [55, 55] and Mask3D [50]. As detailed in Appendix B, even when ChatScene is fine-tuned on Intent3D or combined with Mask3D for 3D-SAM inference, our zero-shot MLLM-For3D achieves substantially higher accuracy and mIoU without any 3D supervision. Specifically, it surpasses fine-tuned ChatScene+Mask3D by 21.7% Acc@0.25 and 14.9% Acc@0.50 on Intent3D, and outperforms the zero-shot ChatScene baseline by over 30% Acc@0.25 and 25% Acc@0.50 on Reason3D. These consistent improvements demonstrate that MLLM-For3D can transfer semantic and reasoning knowledge from 2D MLLMs more effectively than current 3D-MLLM architectures, achieving competitive or superior performance without costly fine-tuning or labeled 3D data.

5 Conclusion & Limitations

We introduce **MLLM-For3D**, a novel framework that adapts MLLM for 3D reasoning segmentation using a *label-free* paradigm. Our approach tackles challenges such as single-view hallucination and cross-view inconsistencies by employing an attention-based fusion strategy alongside a token-for-Query mechanism, enabling coherent multi-view pseudo-label generation without any annotations. Experiments on three challenging benchmarks reveal that our method achieves SOTA performance in label-free settings and demonstrates further improvements when 3D labels are made available. These results confirm the effectiveness of our adaptation strategy and open new avenues for scalable, language-guided 3D scene understanding. However, the method can be computationally demanding as it involves multiple inferences of 2D MLLMs. Future work may explore more efficient architectures, better uncertainty modeling of pseudo-labels, and broader generalization to complex, real-world 3D environments.

Acknowledgments and Disclosure of Funding

Mingming Gong was supported by ARC DP240102088 and WIS-MBZUAI 142571.

References

- [1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.
- [2] Zechen Bai, Tong He, Haiyang Mei, Pichao Wang, Ziteng Gao, Joya Chen, Zheng Zhang, and Mike Zheng Shou. One token to seg them all: Language instructed reasoning segmentation in videos. *NeurIPS*, 37:6833–6859, 2025.
- [3] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3D: Learning from RGB-D data in indoor environments. *International Conference on 3D Vision (3DV)*, 2017.
- [4] Dave Zhenyu Chen, Angel X Chang, and Matthias Nießner. Scanrefer: 3d object localization in rgb-d scans using natural language. In *ECCV*, pages 202–221. Springer, 2020.
- [5] Runnan Chen, Youquan Liu, Lingdong Kong, Nenglun Chen, Xinge Zhu, Yuexin Ma, Tongliang Liu, and Wenping Wang. Towards label-free scene understanding by vision foundation models. In *NeurIPS*, pages 75896–75910, 2023.
- [6] Runnan Chen, Youquan Liu, Lingdong Kong, Xinge Zhu, Yuexin Ma, Yikang Li, Yuenan Hou, Yu Qiao, and Wenping Wang. Clip2scene: Towards label-efficient 3d scene understanding by clip. In *CVPR*, pages 7020–7030, 2023.
- [7] Sijin Chen, Xin Chen, Chi Zhang, Mingsheng Li, Gang Yu, Hao Fei, Hongyuan Zhu, Jiayuan Fan, and Tao Chen. L3da: Visual interactive instruction tuning for omni-3d understanding reasoning and planning. In *CVPR*, pages 26428–26438, 2024.

- [8] Tianrun Chen, Chunan Yu, Jing Li, Jianqi Zhang, Lanyun Zhu, Deyi Ji, Yong Zhang, Ying Zang, Zejian Li, and Lingyun Sun. Reasoning3d-grounding and reasoning in 3d: Fine-grained zero-shot open-vocabulary 3d reasoning part segmentation via large vision-language models. *arXiv preprint arXiv:2405.19326*, 2024.
- [9] Yilun Chen, Shuai Yang, Haifeng Huang, Tai Wang, Runsen Xu, Ruiyuan Lyu, Dahua Lin, and Jiangmiao Pang. Grounded 3d-llm with referent tokens. *arXiv preprint arXiv: 2405.10370*, 2024.
- [10] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *CVPR*, pages 3075–3084, 2019.
- [11] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53, 2024.
- [12] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, 2017.
- [13] Runyu Ding, Jihan Yang, Chuhui Xue, Wenqing Zhang, Song Bai, and Xiaojuan Qi. Pla: Language-driven open-vocabulary 3d scene understanding. In *CVPR*, pages 7010–7019, 2023.
- [14] Runyu Ding, Jihan Yang, Chuhui Xue, Wenqing Zhang, Song Bai, and Xiaojuan Qi. Lowis3d: Language-driven open-world instance-level 3d scene understanding. *IEEE TPAMI*, 2024.
- [15] Golnaz Ghiasi, Xiuye Gu, Yin Cui, and Tsung-Yi Lin. Scaling open-vocabulary image segmentation with image-level labels. In *ECCV*, pages 540–557. Springer, 2022.
- [16] Ziyu Guo, Renrui Zhang, Xiangyang Zhu, Yiwen Tang, Xianzheng Ma, Jiaming Han, Kexin Chen, Peng Gao, Xianzhi Li, Hongsheng Li, et al. Point-bind & point-llm: Aligning point cloud with multi-modality for 3d understanding, generation, and instruction following. *arXiv preprint arXiv:2309.00615*, 2023.
- [17] Abdullah Hamdi, Silvio Giancola, and Bernard Ghanem. Mvtn: Multi-view transformation network for 3d shape recognition. In *ICCV*, pages 1–11, 2021.
- [18] Abdullah Hamdi, Silvio Giancola, and Bernard Ghanem. Voint cloud: Multi-view point cloud representation for 3d understanding. *arXiv preprint arXiv:2111.15363*, 2021.
- [19] Shuting He, Henghui Ding, Xudong Jiang, and Bihan Wen. Segpoint: Segment any point cloud via large language model. In *ECCV*, pages 349–367. Springer, 2024.
- [20] Yining Hong, Haoyu Zhen, Peihao Chen, Shuhong Zheng, Yilun Du, Zhenfang Chen, and Chuang Gan. 3d-llm: injecting the 3d world into large language models. In *NeurIPS*, pages 20482–20494, 2023.
- [21] Haifeng Huang, Yilun Chen, Zehan Wang, Rongjie Huang, Runsen Xu, Tai Wang, Luping Liu, Xize Cheng, Yang Zhao, Jiangmiao Pang, et al. Chat-scene: Bridging 3d scene and large language models with object identifiers. In *NeurIPS*, 2024.
- [22] Jiangyong Huang, Silong Yong, Xiaojian Ma, Xiongkun Linghu, Puhao Li, Yan Wang, Qing Li, Song-Chun Zhu, Baoxiong Jia, and Siyuan Huang. An embodied generalist agent in 3d world. In *ICML*, pages 20413–20451, 2024.
- [23] Kuan-Chih Huang, Xiangtai Li, Lu Qi, Shuicheng Yan, and Ming-Hsuan Yang. Reason3d: Searching and reasoning 3d segmentation via large language model. *arXiv preprint arXiv:2405.17427*, 2024.
- [24] Pin-Hao Huang, Han-Hung Lee, Hwann-Tzong Chen, and Tyng-Luh Liu. Text-guided graph neural networks for referring 3d instance segmentation. *AAAI*, 35(2):1610–1618, May 2021.
- [25] Tianyu Huang, Runnan Chen, Dongting Hu, Fengming Huang, Mingming Gong, and Tongliang Liu. Openinsgaussian: Open-vocabulary instance gaussian segmentation with context-aware cross-view fusion. In *ICCV*, pages 6341–6350, 2025.

- [26] Zhuo Huang, Chang Liu, Yinpeng Dong, Hang Su, Shibao Zheng, and Tongliang Liu. Machine vision therapy: Multimodal large language models can enhance visual robustness via denoising in-context learning. In *ICML*, 2024.
- [27] Ayush Jain, Nikolaos Gkanatsios, Ishita Mediratta, and Katerina Fragkiadaki. Bottom up top down detection transformers for language grounding in images and point clouds, 2021.
- [28] Krishna Murthy Jatavallabhula, Alihusein Kuwajerwala, Qiao Gu, Mohd Omama, Tao Chen, Alaa Maalouf, Shuang Li, Ganesh Iyer, Soroush Saryazdi, Nikhil Keetha, et al. Conceptfusion: Open-set multimodal 3d mapping. *arXiv preprint arXiv:2302.07241*, 2023.
- [29] Li Jiang, Shaoshuai Shi, and Bernt Schiele. Open-vocabulary 3d semantic segmentation with foundation models. In *CVPR*, pages 21284–21294, 2024.
- [30] Xueying Jiang, Lewei Lu, Ling Shao, and Shijian Lu. Multimodal 3d reasoning segmentation with complex scenes. *arXiv preprint arXiv: 2411.13927*, 2024.
- [31] Yili Jin, Xize Duan, Fangxin Wang, and Xue Liu. Headsetoff: Enabling photorealistic video conferencing on economical vr headsets. In *ACM MM*, pages 7928–7936, 2024.
- [32] Weitai Kang, Mengxue Qu, Jyoti Kini, Yunchao Wei, Mubarak Shah, and Yan Yan. Intent3d: 3d object detection in rgb-d scans based on human intention. *arXiv preprint arXiv:2405.18295*, 2024.
- [33] Amrin Kareem, Jean Lahoud, and Hisham Cholakkal. Paris3d: Reasoning-based 3d part segmentation using large multimodal model. In *ECCV*, pages 466–482. Springer, 2024.
- [34] Justin Kerr, Chung Min Kim, Ken Goldberg, Angjoo Kanazawa, and Matthew Tancik. Lerf: Language embedded radiance fields. *ICCV*, 2023.
- [35] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *ICCV*, pages 4015–4026, 2023.
- [36] Xin Lai, Zhuotao Tian, Yukang Chen, Yanwei Li, Yuhui Yuan, Shu Liu, and Jiaya Jia. Lisa: Reasoning segmentation via large language model. In *CVPR*, pages 9579–9589, 2024.
- [37] Boyi Li, Kilian Q Weinberger, Serge Belongie, Vladlen Koltun, and René Ranftl. Language-driven semantic segmentation. *arXiv preprint arXiv:2201.03546*, 2022.
- [38] Yanbang Li, Ziyang Gong, Haoyang Li, Xiaoqi Huang, Haolan Kang, Guangping Bai, and Xianzheng Ma. Robotic visual instruction. In *CVPR*, pages 12155–12165, 2025.
- [39] Ziwen Li, Jiaxin Huang, Runnan Chen, Yunlong Che, Yandong Guo, Tongliang Liu, Fakhri Karay, and Mingming Gong. Urbangs: Semantic-guided gaussian splatting for urban scene reconstruction. *arXiv preprint arXiv:2412.03473*, 2024.
- [40] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *CVPR*, pages 26296–26306, 2024.
- [41] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.
- [42] Shiyang Lu, Haonan Chang, Eric Pu Jing, Abdeslam Boularias, and Kostas Bekris. Ovir-3d: Open-vocabulary 3d instance retrieval without training on 3d data. In *Conference on Robot Learning*, pages 1610–1620. PMLR, 2023.
- [43] Xianzheng Ma, Yash Bhargat, Brandon Smart, Shuai Chen, Xinghui Li, Jian Ding, Jindong Gu, Dave Zhenyu Chen, Songyou Peng, Jia-Wang Bian, et al. When llms step into the 3d world: A survey and meta-analysis of 3d tasks via multi-modal large language models. *arXiv preprint arXiv:2405.10255*, 2024.

- [44] Phuc Nguyen, Tuan Duc Ngo, Evangelos Kalogerakis, Chuang Gan, Anh Tran, Cuong Pham, and Khoi Nguyen. Open3dis: Open-vocabulary 3d instance segmentation with 2d mask guidance. In *CVPR*, pages 4018–4028, 2024.
- [45] Songyou Peng, Kyle Genova, Chiyu Jiang, Andrea Tagliasacchi, Marc Pollefeys, Thomas Funkhouser, et al. Openscene: 3d scene understanding with open vocabularies. In *CVPR*, pages 815–824, 2023.
- [46] Zhangyang Qi, Zhixiong Zhang, Ye Fang, Jiaqi Wang, and Hengshuang Zhao. Gpt4scene: Understand 3d scenes from videos with vision-language models. *arXiv preprint arXiv:2501.01428*, 2025.
- [47] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, pages 8748–8763, 2021.
- [48] Hanoona Rasheed, Muhammad Maaz, Salman Khan, and Fahad S. Khan. Llava++: Extending visual capabilities with llama-3 and phi-3, 2024.
- [49] Zhongwei Ren, Zhicheng Huang, Yunchao Wei, Yao Zhao, Dongmei Fu, Jiashi Feng, and Xiaojie Jin. Pixellm: Pixel reasoning with large multimodal model. In *CVPR*, pages 26374–26383, 2024.
- [50] Jonas Schult, Francis Engelmann, Alexander Hermans, Or Litany, Siyu Tang, and Bastian Leibe. Mask3d: Mask transformer for 3d semantic instance segmentation. In *ICRA*, pages 8216–8223. IEEE, 2023.
- [51] Guangyan Sun, Mingyu Jin, Zhenting Wang, Cheng-Long Wang, Siqi Ma, Qifan Wang, Tong Geng, Ying Nian Wu, Yongfeng Zhang, and Dongfang Liu. Visual agents as fast and slow thinkers. *arXiv preprint arXiv:2408.08862*, 2024.
- [52] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv: 2307.09288*, 2023.
- [53] Zhenchen Wan, Yanwu Xu, Zhaoqing Wang, Feng Liu, Tongliang Liu, and Mingming Gong. Ted-viton: Transformer-empowered diffusion models for virtual try-on. *arXiv preprint arXiv:2411.17017*, 2024.
- [54] Junchi Wang and Lei Ke. Llm-seg: Bridging image segmentation and large language model reasoning. In *CVPR*, pages 1765–1774, 2024.
- [55] Zehan Wang, Haifeng Huang, Yang Zhao, Ziang Zhang, and Zhou Zhao. Chat-3d: Data-efficiently tuning large language model for universal dialogue of 3d scenes. *arXiv preprint arXiv:2308.08769*, 2023.
- [56] Zhaoqing Wang, Xiaobo Xia, Runnan Chen, Dongdong Yu, Changhu Wang, Mingming Gong, and Tongliang Liu. Lavin-dit: Large vision diffusion transformer. In *CVPR*, pages 20060–20070, 2025.
- [57] Xin Wei, Ruixuan Yu, and Jian Sun. View-gcn: View-based graph convolutional network for 3d shape analysis. In *CVPR*, pages 1850–1859, 2020.

- [58] Changli Wu, Yiwei Ma, Qi Chen, Haowei Wang, Gen Luo, Jiayi Ji, and Xiaoshuai Sun. 3d-stmn: Dependency-driven superpoint-text matching network for end-to-end 3d referring expression segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 5940–5948, 2024.
- [59] Yanmin Wu, Xinhua Cheng, Renrui Zhang, Zesen Cheng, and Jian Zhang. Eda: Explicit text-decoupling and dense alignment for 3d visual grounding. In *CVPR*, pages 19231–19242, 2023.
- [60] Shaoan Xie, Lingjing Lingjing, Yujia Zheng, Yu Yao, Zeyu Tang, Eric P Xing, Guangyi Chen, and Kun Zhang. Smartclip: Modular vision-language alignment with identification guarantees. In *CVPR*, pages 29780–29790, 2025.
- [61] Cilin Yan, Haochen Wang, Shilin Yan, Xiaolong Jiang, Yao Hu, Guoliang Kang, Weidi Xie, and Efstratios Gavves. Visa: Reasoning video object segmentation via large language models. In *ECCV*, pages 98–115. Springer, 2024.
- [62] Mi Yan, Jiazhao Zhang, Yan Zhu, and He Wang. Maskclustering: View consensus based mask graph clustering for open-vocabulary 3d instance segmentation. In *CVPR*, pages 28274–28284, 2024.
- [63] Jihan Yang, Runyu Ding, Weipeng Deng, Zhe Wang, and Xiaojuan Qi. Regionplc: Regional point-language contrastive learning for open-world 3d scene understanding. In *CVPR*, pages 19823–19832, 2024.
- [64] Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. Scannet++: A high-fidelity dataset of 3d indoor scenes. In *ICCV*, 2023.
- [65] Junbo Zhang, Runpei Dong, and Kaisheng Ma. Clip-fo3d: Learning free open-world 3d scene representations from 2d dense clip. In *ICCV*, pages 2048–2059, 2023.
- [66] Yiming Zhang, ZeMing Gong, and Angel X Chang. Multi3drefer: Grounding text description to multiple 3d objects. In *ICCV*, pages 15225–15236, 2023.
- [67] Jiyang Zheng, Jialiang Shen, Yu Yao, Min Wang, Yang Yang, Dadong Wang, and Tongliang Liu. Chain-of-focus prompting: Leveraging sequential visual cues to prompt large autoregressive vision models, 2025.
- [68] Chong Zhou, Chen Change Loy, and Bo Dai. Extract free dense labels from clip. In *ECCV*, pages 696–712. Springer, 2022.
- [69] Ziyu Zhu, Xiaojian Ma, Yixin Chen, Zhidong Deng, Siyuan Huang, and Qing Li. 3d-vista: Pre-trained transformer for 3d vision and text alignment. In *ICCV*, pages 2911–2921, 2023.
- [70] Xueyan Zou, Jianwei Yang, Hao Zhang, Feng Li, Linjie Li, Jianfeng Wang, Lijuan Wang, Jianfeng Gao, and Yong Jae Lee. Segment everything everywhere all at once. *NeurIPS*, 36:19769–19782, 2023.

A Technical Appendices and Supplementary Material

A.1 Datasets Descriptions

We base our 3D reasoning-segmentation experiments on existing indoor scene benchmarks with generated language queries. The Reason3D dataset [23] provides query-conditioned object masks on ScanNetV2 [12] and Matterport3D [3]. Each sample in Reason3D is a single-room point cloud paired with a natural-language query and a binary mask of the target object. We follow the official splits and statistics: Matterport3D contributes 934 training and 837 validation samples, while ScanNetV2 contributes 405 training and 308 validation samples. These datasets supply (implicit) query-object-ID supervision for reasoning segmentation.

We use Instruct3D [19], a harder benchmark derived from the high-fidelity ScanNet++ v1 [64] dataset. Instruct3D omits explicit object names or step-by-step instructions, requiring models to infer both

user intent and spatial relationships from context. The filtered Instruct3D split contains 136 training scenes and 45 validation scenes, yielding 1,034 and 321 query-answer (QA) pairs, respectively. These QA pairs consist of implicit queries with spatial relationships and segmentation masks of the referred objects. Removing direct mentions of object names and requiring spatial cues makes Instruct3D significantly more challenging.

Finally, we evaluate grounding on two 3D spatial-reasoning datasets built on ScanNet scenes: 3D-IG (Intent3D) [32] and VG-w/o-ON [59]. The Intent3D dataset (3D intention grounding) contains free-form human-intent descriptions paired with target object detections, while VG-w/o-ON ("visual grounding without object names") contains spatial queries that deliberately omit object names. Both are constructed atop ScanNet annotations and test the model’s ability to localize objects from descriptive, context-dependent language.

A.2 Implementation Details

Our 3D segmentation network is implemented using MinkowskiNet14 as the backbone, built on the PyTorch framework. Throughout training, both the multimodal large language model (MLLM, specifically LISA-7B) and the Segment-Anything Model (SAM) remain frozen to leverage pre-trained 2D multimodal knowledge, while only the 3D projection model is optimized. Training is performed on four NVIDIA A100 GPUs (40 GB each).

We summarize key training hyperparameters and configurations used across the ScanNetV2, Matterport3D, and ScanNet++ v1 datasets in Table 5. For optimization, we employ stochastic gradient descent (SGD) with momentum set to 0.9 and a weight decay of 1×10^{-4} . Data augmentations such as random rotation around the upright axis, random flips on point clouds, and random horizontal flips and resized crops on images were consistently applied to enhance model generalization.

Table 5: Training configurations across different datasets.

Dataset	Backbone	Batch Size	LR	Epochs	GPUs	Voxel Size	Max Sweeps
ScanNetV2	MinkowskiNet14	8	0.10	40	4	0.05 m	1
Matterport3D	MinkowskiNet14	4	0.10	40	4	0.05 m	1
ScanNet++ (Instruct3D)	MinkowskiNet14	4	0.01	40	4	0.05 m	1

Training times vary depending on the complexity and size of the data set, especially the language number. ScanNetV2 and Matterport3D (Reason3D) typically converges within approximately 20 hours, whereas ScanNet++ v1 (Instruct3D) datasets require roughly 30–40 hours. These configurations were empirically chosen to ensure robust convergence across all datasets. Batch size and GPUs are configured based on a PyTorch Lightning DataModule, with batch size set by dividing the total batch size by the number of GPUs (i.e., `self.batch_size = config["batch_size"] // config["num_gpus"]`)

A.3 Model Architecture

Our model comprises two fused branches: a 3D sparse-CNN branch for processing voxelized scene data, and a 2D vision-language branch for language-conditioned image segmentation. The 3D branch is implemented as a UNet-style sparse convolutional network (MinkowskiNet14). All 3D convolutions use a $3 \times 3 \times 3$ kernel and produce 512-dimensional features per occupied voxel. Batch-normalization layers (with momentum 0.05) follow each convolution. The network follows an encoder-decoder ("U-Net") design with skip connections between corresponding scales. During training, only this 3D branch is updated; the 2D branch parameters remain fixed.

3D Sparse Convolutional Branch. We adopt the Minkowski Engine’s 3D U-Net backbone ("MinkowskiNet14"). Following prior practice in high-dimensional CNNs, all convolutions use kernel size $3 \times 3 \times 3$, with hyper-cross patterns ("+" indicates 1-D in the third dimension). The network has multiple down- and up-sampling stages (forming a U-shape) with symmetric skip connections. Each sparse conv layer produces 512 features per voxel (after the final encoder stage), and we attach a BatchNorm (momentum 0.05) and ReLU nonlinearity to each layer. The output of the 3D branch is a set of per-voxel features over the scene. These features will be aligned (during training) to

the projected 2D segmentations. All weights in this 3D branch are learnable, while the 2D branch is kept frozen.

2D Vision-Language Segmentation Branch. The 2D branch leverages pretrained LISA multimodal LLMs [36, 2] to perform language-guided image segmentation. We use the publicly released LISA-7B, LISA-13B, and VideoLISA models as frozen multimodal feature extractors. In all cases, we operate in single-frame inference mode (treating VideoLISA as an image model), with no temporal aggregation. The LISA models consist of a CLIP ViT-L/14 vision encoder and a LLaVA-based language model. The vision encoder extracts dense image features from each input frame, and the decoder (a lightweight dilated convolutional network) upsamples the mask logits to the original image resolution. The entire LISA model, including the SAM-style segmentation decoder, remains frozen during training. To guide the segmentation process, we use LISA’s vocabulary, which includes a special [SEG] token that acts as a semantic anchor. Given a natural language instruction, we prepend the [SEG] token to the input prompt. After multimodal processing by the LLM, the model generates an embedding corresponding to [SEG], denoted $\tilde{\mathbf{h}}_{\text{seg}}$. This embedding is projected via a learned linear transformation to match the dimensionality required by the SAM decoder:

$$\mathbf{h}_{\text{seg}} = W\tilde{\mathbf{h}}_{\text{seg}}, \quad (4)$$

where W is a learnable projection matrix.

SAM Decoder. The projected token embedding [SEG] is used to prompt the frozen SAM decoder. The decoder processes the image features and the token prompt to produce a coarse segmentation mask. This mask is subsequently refined and upsampled by a lightweight dilated convolutional decoder, producing the final high-resolution binary segmentation mask.

Frozen Inference We do not modify the architecture or parameters of LISA or SAM. All 2D weights are frozen, and only the 3D segmentation model is updated during training. This design enables effective language-conditioned segmentation with no additional finetuning of the 2D backbone, allowing the 3D model to inherit multimodal reasoning capabilities from LISA through differentiable pseudo-label projection.

B Additional Comparisons with 3D-MLLM Baselines

In this section, we provide additional quantitative comparisons with existing 3D-MLLM pipelines that combine large language models with 3D segmentation backbones, such as ChatScene [55, 21] and Mask3D [50].

B.1 Evaluation on Intent3D and Reason3D

We evaluate both **3D Intention Grounding** and **3D Reasoning Segmentation** tasks using the public validation sets of Intent3D [32] and Reason3D [23]. For fair comparison, we selected Mask3D [50] as the 3D-SAM baseline, as it provides a pre-trained checkpoint on the ScanNet validation set and avoids dataset discrepancies between benchmarks.

For Intent3D, we follow the pipeline of ChatScene [21], where object ID strings are generated via a fine-tuned Vicuna-7B model and aligned with Mask3D object proposals to obtain the final 3D masks. For Reason3D, we evaluate zero-shot reasoning segmentation performance under the same experimental setup.

1. 3D Intention Grounding on Intent3D

Model	Acc@0.25	Acc@0.5	mIoU
ChatScene (FT) + Mask3D	36.71	21.50	12.08
Ours (Zero-Shot)	58.40	41.00	26.15

2. 3D Reasoning Segmentation on Reason3D

Model	Acc@0.25	Acc@0.5	mIoU
ChatScene (Zero-Shot) + Mask3D	9.23	8.89	1.01
Ours (Zero-Shot)	44.18	34.80	32.90

Overall, our zero-shot pipeline consistently surpasses both fine-tuned 3D-MLLM + 3D-SAM frameworks on Intent3D and zero-shot settings on Reason3D, demonstrating strong generalization without any 3D supervision. The inferior performance of existing 3D-MLLMs mainly stems from two factors: (1) the scarcity of 3D training data for aligning visual features within the LLM embedding space, and (2) hallucinations and false positives produced by LLM responses. In contrast, **MLLM-For3D** leverages frozen 2D MLLMs and semantic alignment to achieve superior reasoning-based segmentation without costly fine-tuning or labeled 3D annotations.

NeurIPS Paper Checklist

1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [\[Yes\]](#)

Justification: The abstract and introduction explicitly state three core contributions: (i) a label-free framework that transfers 2D MLLM reasoning to 3D, (ii) an attention-based spatial-consistency strategy that filters noisy multi-view masks, and (iii) a token-for-query mechanism that aligns language, image, and point-cloud features. All three claims are developed in the technical sections, formalized in the methodology, and validated experimentally on Instruct3D, Intent3D, and VG-w/o-ON.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [\[Yes\]](#)

Justification: A dedicated *Limitations* paragraph is provided in the last section. It explicitly acknowledges the computational overhead introduced by repeatedly invoking a frozen 2D MLLM+SAM pipeline for every training epoch,

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. Theory assumptions and proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: The paper is primarily empirical and algorithmic; it does not present formal theorems or proofs.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. Experimental result reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The paper details *all* components required for reproduction.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
 - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
 - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
 - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
 - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: While the code is not publicly released at this stage, the appendix provides detailed descriptions of the experimental settings, model architecture, training procedure, and evaluation metrics. These details allow readers to understand and potentially reproduce the key results. The authors indicate the intention to release code in the future.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental setting/details

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: The paper provides comprehensive experimental details, including dataset splits, training hyperparameters, optimizer settings, and evaluation metrics. While some configurations are deferred to the appendix, the descriptions are sufficiently detailed to understand and interpret the reported results. For example, the appendix specifies batch size, learning rate schedule, optimizer type, and training epochs.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. Experiment statistical significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: paper does not report error bars, but follows standard evaluation protocols on fixed dataset splits with consistent performance gains across multiple baselines. While not ideal, this is common practice in the 3D scene understanding literature.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments compute resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: The paper specifies the compute resources used for the experiments, including GPU type (NVIDIA A100), memory configuration, batch size, and training time per epoch. These details are provided in the appendix and allow for a reasonable estimation of the total compute required to reproduce the experiments.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. Code of ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines?>

Answer: [Yes]

Justification: The study relies solely on publicly available indoor-scene datasets (ScanNet, ScanNet++, Intent3D), each of which was released with explicit consent for academic use; no personally identifiable information or sensitive biometric data are involved. All pretrained models used (LLaVA, SAM, MinkowskiNet) are distributed under permissive research licenses, and we do not reverse-engineer or expose proprietary weights. The method does not generate or manipulate human likenesses, nor does it facilitate disallowed surveillance or discriminatory profiling. We report computational cost and energy usage (four A100-40 GB GPUs) to promote transparency around environmental impact, and we release configuration files to encourage efficient replication.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. Broader impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [NA]

Justification: The current manuscript does not contain an explicit “Broader Impact” or “Societal Impact” discussion. While Section *Limitations* addresses computational cost and dataset scope, it does not articulate possible positive applications (e.g., safer robot navigation, assistive VR scene understanding) or negative consequences (e.g., privacy risks in indoor mapping, misuse for unauthorized surveillance).

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: The work does not introduce new large-scale generative models, language models, or scraped datasets. It relies exclusively on publicly available indoor-scene datasets (ScanNet, ScanNet++, Intent3D) and pretrained open-source models (LLaVA, SAM, MinkowskiNet) released under permissive research licenses.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.

- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [\[Yes\]](#)

Justification: All external assets employed in this work are properly cited.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. New assets

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [\[NA\]](#)

Justification: The work does not release any new datasets, pretrained models, or codebases beyond minor training scripts that wrap existing open-source libraries.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. Crowdsourcing and research with human subjects

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [\[NA\]](#)

Justification: The study does not involve any crowdsourced data collection, user studies, or other experiments with human subjects. All datasets employed are publicly released indoor-scene RGB-D datasets collected in prior work under their own ethics approvals.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. Institutional review board (IRB) approvals or equivalent for research with human subjects

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: No experiments with human subjects were conducted. The work exclusively uses publicly available indoor-scene datasets collected in prior studies under their own ethical approvals.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. Declaration of LLM usage

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: paper incorporates Multi-modal large language models (MLLMs) as a core component of the proposed method—for example, by leveraging a pre-trained MLLMs to interpret textual instructions and generate segmentation-relevant tokens (e.g., [SEG]) to perform reasoning over 3D scenes. These usages are clearly described in both methodology section and appendix.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.