AffordBot: 3D Fine-grained Embodied Reasoning via Multimodal Large Language Models

Xinyi Wang[1,*], **Xun Yang**[1,†], **Yanlong Xu**[1], **Yuchen Wu**[2], **Zhen Li**[3], **Na Zhao**[2,†]

[1] University of Science and Technology of China [2] Singapore University of Technology and Design
[3] Chinese University of Hong Kong, Shenzhen

Figure 1: We propose fine-grained 3D embodied reasoning: given a 3D scene and a language task instruction, the agent must identify relevant affordance elements and predict a structured triplet for each: its 3D mask, motion type, and motion axis direction.

Abstract

Effective human-agent collaboration in physical environments requires understanding not only *what* to act upon, but also *where* the actionable elements are and *how* to interact with them. Existing approaches often operate at the object level or disjointedly handle fine-grained affordance reasoning, lacking coherent, instruction-driven grounding and reasoning. In this work, we introduce a new task: Fine-grained 3D Embodied Reasoning, which requires an agent to predict, for each referenced affordance element in a 3D scene, a structured triplet comprising its spatial location, motion type, and motion axis, based on a task instruction. To solve this task, we propose AffordBot, a novel framework that integrates Multimodal Large Language Models (MLLMs) with a tailored chain-of-thought (CoT) reasoning paradigm. To bridge the gap between 3D input and 2D-compatible MLLMs, we render surround-view images of the scene and project 3D element candidates into these views, forming a rich visual representation aligned with the scene geometry. Our CoT pipeline begins with an active perception stage, prompting the MLLM to select the most informative viewpoint based on the instruction, before proceeding with step-by-step reasoning to localize affordance elements and infer plausible interaction motions. Evaluated on the SceneFun3D dataset, AffordBot achieves state-of-the-art performance, demonstrating strong generalization and physically grounded reasoning with only 3D point cloud input and MLLMs. Our code is available at `https://github.com/hannahwxy/AffordBot`.

[*] This work was carried out during Xinyi's visit to the IMPL Lab at SUTD.
[†] Corresponding authors.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

1 Introduction

For intelligent agents to collaborate effectively with humans and operate autonomously in complex 3D physical worlds, they must perceive and interact with their surroundings at a fine-grained, actionable level [1, 2, 3]. This requirement aligns with the concept of "affordance", originally introduced in ecological psychology [4], which describes how elements in the environment offer possibilities for action. For example, to execute an instruction like "*unplug the Christmas tree lights*", an agent must recognize and attend to the fine details of the plug and reason about the interaction they afford, rather than merely identifying the larger object, such as the lights. Meeting this challenge requires agents to understand not only *what* elements in a scene afford interaction, but also *where* they are located and *how* to manipulate them. Such fine-grained embodied understanding is essential for grounded task execution in real-world environments [5, 6, 7, 8, 9, 10, 11, 12, 13].

While recent developments in multimodal large language models (MLLMs) [14, 15, 16, 17, 18, 19, 20, 21] and 3D scene understanding [22, 23, 24, 25, 26, 27] have advanced object-centric 3D perception, existing approaches [28, 29, 30, 31, 32, 33] stop at high-level object recognition and spatial grounding. However, they often overlook finer-grained structures required to infer how parts of objects afford specific interactions. SceneFun3D [34] makes a notable step forward by introducing benchmarks for fine-grained affordance grounding and motion estimation. However, it treats these subtasks in isolation and assumes instruction-agnostic motion prediction, requiring agents to infer motion parameters for all functional parts regardless of task context, limiting its applicability in instruction-conditioned scenarios.

To address these limitations, we propose a unified and instruction-conditioned task: **Fine-grained 3D Embodied Reasoning**, which jointly performs 3D affordance grounding and motion estimation based on a natural language instruction. Specifically, the task is formulated as a structured triplet prediction problem: for each referenced affordance element, the agent predicts a triplet comprising *affordance mask*, *motion type*, and *motion axis direction*. This formulation explicitly couples spatial grounding and interaction reasoning under natural language guidance, forming a coherent inference pipeline tailored for instruction-conditioned embodied tasks.

As a solution, we introduce **AffordBot**, a novel framework that integrates 3D geometric information with the reasoning capabilities of MLLMs. Unlike prior work [34, 35, 36, 37, 38] that relies on video-based inputs, which incur high computational overhead by processing redundant visual frames and often suffer from viewpoint limitations, AffordBot operates directly on 3D point clouds. However, MLLMs are inherently designed for 2D input and general-purpose reasoning, presenting a significant challenge when applying them to 3D spatial tasks that require physical grounding.

To bridge the modality gap between 3D input and 2D-native MLLMs, AffordBot begins by constructing a rich multimodal representation of the 3D scene. Specifically, we render surround-view images from the 3D point cloud and project structured 3D affordance candidates onto these images, establishing explicit 3D-to-2D correspondences. This enables dense and spatially aligned visual context to be provided to the MLLM, without relying on redundant video streams.

On top of this foundation, we develop a task-specific chain-of-thought (CoT) reasoning paradigm that systematically guides the MLLM through physically grounded, step-by-step logical inference. The process begins with an active perception phase, which we specifically designed to empower the MLLM to effectively interpret the task instruction and autonomously select the most informative viewpoint. This initial "*observe*" step improves reasoning focus by reducing input redundancy and emphasizing task-relevant visual cues. Subsequently, the selected viewpoint anchors the model's reasoning process. Then the MLLM is guided through two distinct reasoning stages: *affordance grounding*, where it localizes the target part in the scene, and *interaction inference*, where it predicts the motion type and axis direction based on the scene context and instruction. By conditioning every step on spatial input and task intent, our CoT paradigm enables physically plausible and semantically aligned reasoning, enhancing the agent's embodied intelligence.

We make three key contributions: (1) We introduce a new task formulation for fine-grained, task-driven embodied reasoning, 3D affordance grounding and motion estimation as structured triplet prediction from natural language. (2) We present AffordBot, a novel framework that integrates 3D perception and MLLM-based reasoning via holistic multimodal representation construction and tailored chain-of-thought process. (3) We achieve state-of-the-art results on SceneFun3D, validating the effectiveness of our approach in physically grounded, instruction-conditioned 3D reasoning.

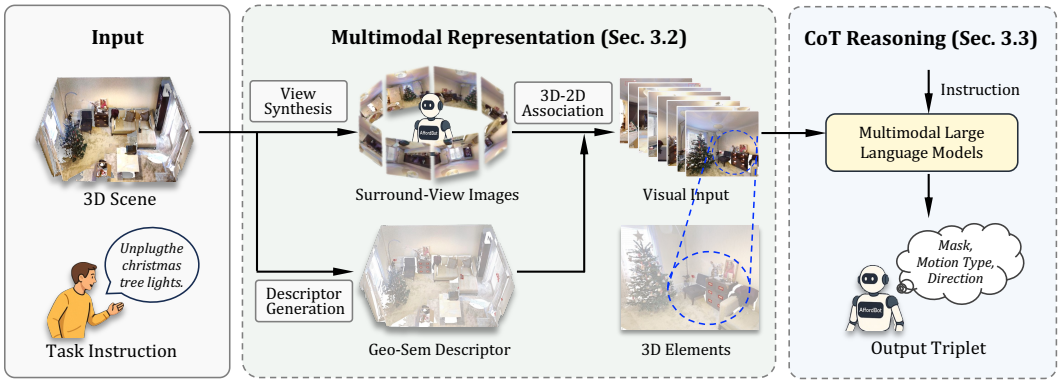


Figure 2: **AffordBot Overview**. Our method first constructs a holistic multimodal representation designed to bridge 3D scenes with 2D-native MLLMs. This process involves view synthesis, extraction of geometric-semantic descriptors, and their association. Then, our designed Chain-of-Thought (CoT) paradigm guides the MLLM to ultimately predict a structured triplet for the task.

2 Related Work

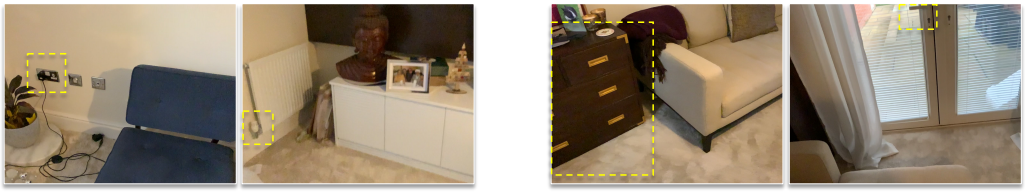
Affordance Understanding. Understanding affordances, the action possibilities offered by the environment [5, 39, 40, 41, 42, 43, 44, 45] , is crucial for robot interaction in 3D scenes. SceneFun3D [34] introduced a challenging task and dataset for grounding referred affordance elements based on task descriptions within complex 3D indoor environments, unlike earlier works focusing on simpler settings [5, 6, 7, 8, 9, 10, 12, 13]. Fun3DU [37] further leveraged VLMs [46] and a universal segmentation model [47] to parse instructions and localize the target elements in video frames.

3D Motion Estimation. 3D motion enables agents to predict and comprehend how objects move and can be manipulated [48, 49, 50, 51, 52, 53, 54, 55, 56, 57]. Earlier research efforts often focused on estimating the articulated motion and mobility of individual interactable objects with predefined structures, such as hinged parts [48, 49, 50]. These approaches typically rely on analyzing the geometric structure of those individual objects to infer their motion properties. In contrast, SceneFun3D [34] broadened this by formulating a scene-level motion estimation task across all affordance elements, providing a dataset for comprehensive evaluation.

MLLMs for 3D Understanding. Multimodal large language models (MLLMs) [14, 15, 16, 17, 18, 19] are being applied to 3D understanding through two main approaches: developing native 3D-aware models [30, 58, 59, 31, 60] for direct processing of spatial data, and adapting existing 2D VLMs [61, 62, 63, 64] by transforming 3D data into 2D representations. These efforts highlight MLLMs' potential for enhancing 3D visual understanding and semantic reasoning. Building upon these, we leverage the MLLM empowered by the tailored chain-of-thought paradigm for fine-grained 3D embodied reasoning, jointly tackling affordance grounding and motion estimation tasks.

3 Methodology

We introduce a new task termed **Fine-grained 3D Embodied Reasoning**, which aims to equip embodied agents with the ability to interpret natural linguistic instructions and reason about actionable elements in complex 3D environments. Given a 3D scene $\mathcal{S}$ represented as a point cloud and a natural language task instruction $\mathcal{T}$ describing a human-intended interaction (*e.g.*, "open the left part of the window door"), the agent is required to predict a set of **structured triplets** $\{(\mathbf{M}_i, \mathbf{t}_i, \mathbf{a}_i\}_{i=1}^{N}$, where each triplet corresponds to a referenced affordance element in the scene. Note that N=1 when only one unique element is referenced in the instruction. $\mathbf{M}_i$ indicates a 3D instance mask identifying the spatial region of the element involved in the interaction; $\mathbf{t}_i \in \mathcal{T}$ denotes the motion type (*e.g.*, "Translation"); $\mathbf{a}_i \in \mathcal{A}$ denotes the motion axis (*e.g.*, "Horizontal outwards") representing the axis along which the motion occurs. The task requires *joint perception and reasoning* over geometry, semantics, and language intent, and presents challenges in grounding ambiguous task references, understanding object affordances, and predicting physically plausible interaction cues in a 3D space.



(a) Lack of Context. (b) Incomplete target coverage.

Figure 3: **Illustrations of video-based method limitations:** (a) Instructions like *"Unplug the Christmas tree lights"* or *"Adjust the room's temperature using the radiator dial next to the curtain"* require anchors (e.g., Christmas tree, curtain) that are missing from the limited video frame. (b) Target objects or parts (*e.g.*, cabinet, door handle) in instructions like *"Open the bottom drawer of the wooden cabinet..."* or *"Open the left part of the window door"* are partially visible within the frame.

3.1 AffordBot Overview

To address the fine-grained 3D embodied reasoning task, we present AffordBot, a framework that leverages MLLMs to enable instruction-conditioned reasoning over 3D point cloud scenes. An overview of the framework is shown in Fig. 2. AffordBot integrates 3D perception with vision-language reasoning through two key components: 1) a holistic multimodal representation (Sec.3.2) that bridges the modality gap between 3D input and the 2D-native input format of MLLMs, and 2) a chain-of-thought reasoning paradigm (Sec. 3.3) that enables interpretable and accurate prediction.

For the first component, we begin by generating a set of surround-view images to effectively capture the 3D scene. Following this, we extract geometric-semantic descriptors from the 3D scene and project them onto the rendered views by an adaptive labeling strategy, establishing robust 3D-to-2D association. This representation effectively eliminates traditional video processing bottlenecks while preserving comprehensive information for downstream reasoning, as illustrated in green in Fig. 2.

Based on this constructed representation and the given instruction, the MLLM engages in a tailored chain-of-thought process to actively perceive and select the most informative view, localize the target element, and infer its required motion. By decomposing the task into a sequence of interpretable steps, our method enables the MLLM to perform robust and physically grounded inference for complex embodied tasks, depicted in blue in Fig. 2.

3.2 Holistic Multimodal Representation

In this section, we construct the holistic multimodal representation foundational for 2D MLLM reasoning. We design an enriched visual synthesis approach using dynamic surround-view generation to overcome limitations of traditional video data. Next, we describe the extraction and representation of 3D geometry and semantics via geometry-semantic descriptors. Finally, we establish the 3D-2D associations by projecting 3D information onto the generated 2D views with adaptive labeling.

Enriched Visual Synthesis. Bridging the gap between 3D input and 2D MLLMs is nontrivial, primarily because it requires establishing accurate and robust associations between 3D structures and their corresponding 2D visual representations.

Existing methods typically rely on video sequences collected from datasets [37, 34, 59]. However, these methods face fundamental limitations: due to the limited field of view, it is often difficult to simultaneously capture the target and its associated anchors within the same frame, as shown in Fig. 3. Furthermore, the process of extracting key information from a large number of frames is both time-consuming and bottlenecks the final accuracy.

To overcome the limitations of static video frames, we propose a dynamic surround-view generation strategy. Inspired by human visual exploration of unfamiliar environments, our robot performs a 360° horizontal panoramic scan centered on the scene's central viewpoint. This scan produces a set of N candidate views $\mathcal{V} = \{V_1, \ldots, V_N\}$, where the i-th view V_i is captured at a rotation angle $\theta_i = (i-1)\frac{2\pi}{N}$.

Compared to relying on traditional video data, this method provides a more comprehensive field of view, thereby capturing more scene information. This effectively alleviates the problem of

missing information caused by a limited field of view and incomplete coverage typical of traditional video data obtained through random sampling. Furthermore, by scanning each 3D scene to acquire a corresponding set of high-quality views, we eliminate the overhead of performing keyframe extraction or detection for each task instruction, thereby completely removing the time processing overhead and accuracy bottleneck associated with analyzing redundant video frames. This ability enables agents to efficiently acquire comprehensive, high-quality visual context.

Geometry-Semantic Descriptors. For the input 3D scene $\mathcal{P}$, our method employs instance segmentation [65] to extract affordance elements, and encodes their geometric and semantic features for downstream reasoning.

During training, we optimize segmentation quality by combining Dice loss to encourage region-level alignment and cross-entropy loss for accurate point-wise classification. The overall training objective is defined as:

$$\mathcal{L}_{\text{total}} = \lambda_1 \cdot \mathcal{L}_{\text{Dice}} + \lambda_2 \cdot \mathcal{L}_{\text{CE}}, \tag{1}$$

where $\mathcal{L}_{\text{Dice}}$ denotes the Dice loss and $\mathcal{L}_{\text{CE}}$ denotes the cross-entropy loss. The weights λ_1 and λ_2 balance the trade-off between the region-level and point-level supervision.

To handle the challenge of small element segmentation, we implement the coarse-to-fine curriculum strategy from [34] with progressive ground-truth mask dilation. At curriculum stage t, each ground-truth mask $\mathcal{Q}$ is dilated within $\mathcal{P}$ according to the following:

$$\widehat{\mathcal{Q}}_{\delta_t} = \{\, x \in \mathcal{P} \mid \min_{y \in \mathcal{Q}} \|x - y\|_2 < \delta_t \}, \quad \delta_t = \delta_0 \, \beta^{\lfloor t/\tau \rfloor}. \tag{2}$$

Here δ_0 is the initial dilation radius, β is the dilation factor, and τ is the step length for the dilation factor update.

Subsequently, for each predicted affordance element j, we construct its geometry descriptor $\mathcal{G}_j$, which captures the element's spatial properties, formally defined as $\mathbf{C}_j$ and $\mathbf{\Sigma}_j$ denote the position and size, respectively. We also employ a semantic descriptor $\mathbf{S}_j$, which, in conjunction with the geometric one, represents the affordance type for element j. In summary, these descriptors together form a compact yet visually unified representation of the 3D scene:

$$\mathcal{D}(\mathcal{P}) = \{(\mathbf{C}_j \in \mathbb{R}^3, \mathbf{\Sigma}_j \in \mathbb{R}^3, S_j)\}_{j=1}^{N}, \tag{3}$$

where N is the total number of predicted affordance elements in the scene $\mathcal{P}$. These descriptors, capturing both geometric and semantic information of the affordance elements, provide a structured representation of the scene that facilitates subsequent reasoning.

3D-2D Associations. Using the generated surround-view images $\mathcal{V}$, we ground 3D affordance elements in these 2D views. For the predicted 3D elements with their descriptors $\mathcal{D}$, we project their 3D geometry onto every view $V_i \in \mathcal{V}$.

Specifically, we compute the element's 2D bounding box projection based on its 3D position and dimensions, assigning each projected box both a unique identifier linking it to the original 3D element j and its corresponding affordance type $\mathbf{S}_j$ mapped to the 2D region. This process is formalized as:

$$\hat{V}_i = \mathcal{M}_{\text{3D}\to\text{2D}}\big(\mathcal{D}(\mathcal{P}), V_i\big), \tag{4}$$

where $\mathcal{M}_{\text{3D}\to\text{2D}}$ denotes our projection operator. This process effectively transfers key 3D information onto the 2D images.

Furthermore, to ensure legible MLLM input, we introduce an adaptive-labeling refinement strategy that resolves label collisions. This involves pre-defining candidate anchor positions around each projected box. When annotating an element, our pipeline iterates through these anchors, evaluating nonoverlap with existing elements, and selects the first suitable location. Such a lightweight spatial check effectively prevents label stacking, maintains clear object visibility, and provides an uncluttered canvas for subsequent MLLM reasoning.

Together, this collaborative representation enables downstream modules to access both the fine-grained geometry of affordance elements and their visual context within the scene, laying the groundwork for subsequent reasoning.

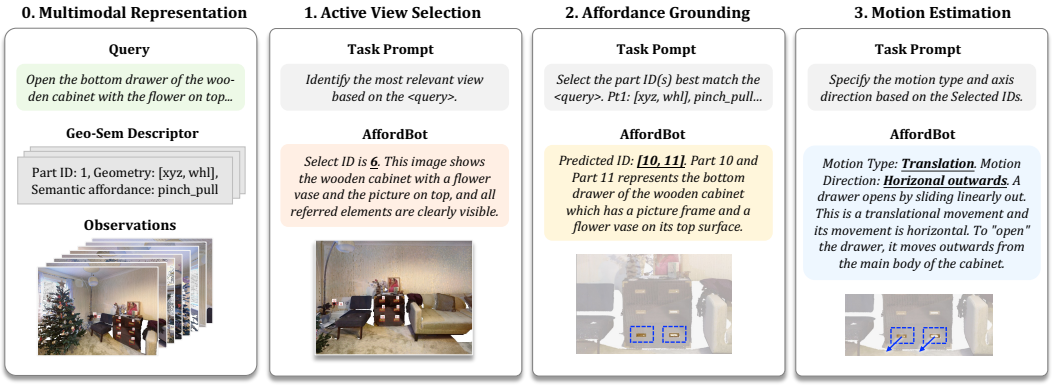


Figure 4: **AffordBot's Chain-of-Thought Pipeline for Embodied Reasoning.** This structured observe-then-infer process leverages multimodal inputs to perform: (1) Active View Selection to identify the most informative view, which may involve zooming in to better see the details of the images, followed by (2) Affordance Grounding to localize target elements, and finally (3) Motion Estimation to infer the required action details.

3.3 Chain-of-thought Reasoning

This section presents our method for enabling MLLM to perform embodied reasoning. Specifically, we design a tailored chain-of-thought paradigm that follows a structured observe-then-infer pipeline, as shown in Fig. 4. Grounded in the real physical world, our pipeline leverages visual observations to guide the MLLM through a sequence of inference steps: first actively selecting the most informative viewpoint (*observe*), followed by identifying targets and their required interactions in context (*infer*).

Step 1: Active View Selection. The goal of this step is to select the most informative view from the generated surround-view images $\hat{\mathcal{V}}$. Unlike prior methods that rely on instruction parsing and heuristic filtering over pre-processed features, we leverage the multimodal reasoning capabilities of an MLLM to guide view selection directly, enhancing both flexibility and accuracy.

Given the set of annotated views $\{\hat{V}_1, \ldots, \hat{V}_N\}$, where each includes projected 3D elements with unique IDs and affordance types, along with the instruction $\mathcal{T}$, the MLLM receives these as inputs. It is then tasked with selecting the view in which all referenced elements are visible and their identifiers are clearly shown, providing the most relevant visual content. The model directs attention to the selected view $\hat{V}_{\text{selected}}$, which serves as the semantically grounded visual input for subsequent reasoning.

Step 2: Affordance Grounding. This step aims to localize the specific affordance elements referenced by the linguistic instruction within the scene. Using the selected view $\hat{V}_{\text{selected}}$ from the previous step, the MLLM receives this annotated view, the original instruction $\mathcal{T}$, and detailed descriptors of the 3D affordance elements, including their unique IDs and spatial attributes. The MLLM interprets the instruction and visual cues to identify the region that best matches the described task. This process yields unique IDs of the localized target elements, which serve as a critical intermediate decision for the subsequent motion inference stage.

Step 3: Motion Estimation. This final step focuses on inferring the motion details of the elements localized in Step 2. The MLLM is provided with the original instruction $\mathcal{T}$, the image $\hat{V}_{\text{selected}}$, and its localized information from the previous step. Using these inputs, the MLLM deduces the intended action for the target element, including both the motion type and the direction of the motion axis.

To enable compatibility between continuous direction vectors and MLLM outputs, we discretize the former into interpretable categories. These categories broadly distinguish motion directions as horizontal or vertical, with specific refinements for translational movements (*e.g.*, inward/outward relative to the object's centroid) and rotational axes. This structured discretization ensures physical plausibility while remaining interpretable by the MLLM. The final output is an affordance-motion tuple that integrates the discretized motion representation with the model's language-driven reasoning.

Table 1: **Quantitative comparisons of our fine-grained embodied reasoning task.** We report the quantitative results of affordance grounding and motion estimation task on SceneFun3D [34] dataset.

Task		Grounding				Motion	
Method	Raw 2D Input	mIoU	AP	AP_{50}	AP_{25}	+T	+TD
OpenMask3D [36]	✓	-	-	0.0	0.0	-	-
LERF [35]	✓	-	-	4.9	11.3	-	-
OpenMask3D-F [34]	✓	-	-	8.0	17.5	-	-
OpenIns3D [66]	✗	0.0	0.0	0.0	0.0	-	-
Fun3DU [37]	✓	11.5	6.1	12.6	23.1	-	-
Fun3DU (+motion)	✓	10.0	4.6	9.9	18.7	11.5	4.0
AffordBot	✗	**14.0**	**15.5**	**20.0**	**23.3**	**18.3**	**10.8**

4 Experiments

This section presents a comprehensive experimental evaluation of our proposed fine-grained embodied reasoning framework, validating its effectiveness in joint affordance grounding and motion estimation. In addition, we conduct an in-depth analysis to assess how key module optimizations impact the system's overall accuracy and robustness.

4.1 Experimental Setup

Dataset. We conduct experiments on SceneFun3D [34], currently the only dataset that provides comprehensive annotations for fine-grained affordance grounding and motion estimation in 3D indoor scenes. It comprises a total of 230 richly annotated scenes, including 200 scenes for training, 30 for validation. Each scene provides dense point clouds annotated with element-level affordance masks, motion types, and motion axis directions. To facilitate instruction-oriented embodied reasoning for our task, we curate the annotation with task-specific annotation triples.

Evaluation Metrics. We adopt standard evaluation metrics [34] to assess performance on 3D affordance grounding and motion estimation. Specifically, we report mean Intersection-over-Union (mIoU), mean average precision (mAP), and average precision at IoU thresholds (AP, AP_{25}, AP_{50}), between predicted and ground-truth masks. To incorporate motion parameter accuracy, we adapt the AP_{25} metric as proposed in [34, 51, 50], extending it with additional constraints on motion type and direction. Specifically, we further constrain mask prediction based on whether the model correctly predicts the motion type (+T), and both the motion type and motion axis direction (+TD).

Implementation Details. For visual-language reasoning, we employ Qwen2.5-VL-72B [15] locally deployed on four NVIDIA A800 GPUs. To construct the geometry descriptors and get segmented elements masks, we fine-tune Mask3D [65] from a pretrained checkpoint on ScanNet200 [67]. We train for 1,000 epochs on an NVIDIA A800 with the learning rate of 0.0001, a batch size of 2, and 2cm voxelization to preserve spatial detail.

4.2 Quantitative Results

The quantitative results for the fine-grained embodied reasoning on SceneFun3D dataset, as presented in Table 1, demonstrate the effectiveness of our AffordBot approach. By encoding 3D scenes into structured representations and processing them with task instructions via the MLLM, AffordBot outperforms existing methods including OpenMask3D [36, 34], LERF [35], OpenIns3D [66], and Fun3DU [37], as well as our enhanced Fun3DU (+motion) baseline. As shown in the table, AffordBot reports higher scores in both affordance grounding and motion estimation.

Notably, the results of our reproduced Fun3DU (+motion) baseline (second to last row) highlight the impact of incorporating a motion estimation branch into their original affordance grounding framework. For this baseline, we prompted Molmo [46] to infer motion parameters based on 2D segmentation results of affordance elements. The significant outperformance of AffordBot across all reported metrics underscores the advantage of our approach in accurately identifying and understanding the potential motions associated with affordance elements in 3D scenes. Specifically, our higher AP score, which is the average precision over IoU thresholds ranging from 0.5 to 0.95,

Table 2: **Ablation on key components of our AffordBot.** ALR denotes Adaptive Label Refinement, EVS denotes Enriched Visual Synthesis, and AVS denotes Active View Selection. Each variant incrementally incorporates one module, and finally Ex4 corresponds to our AffordBot.

	ALR	EVS	AVS	AP	AP_{50}	AP_{25}
Ex1				9.7	12.8	15.7
Ex2	✓			9.7	13.0	16.1
Ex3	✓	✓		14.8	19.4	22.1
Ex4	✓	✓	✓	**15.5**	**20.0**	**23.3**

Table 3: **Ablation on viewpoint selection.** 'BEV' projects bird-eye view; 'Video Frame' uniformly samples frames from dataset, while 'Query-Aligned' picks the query-matching view; 'Ours' renders surround views for MLLM selection.

Method	AP	AP_{50}	AP_{25}
BEV	6.1	9.1	12.7
Video Frame	9.4	11.4	15.6
Query-Aligned	9.7	13.0	16.1
Ours	**15.5**	**20.0**	**23.3**

indicates that AffordBot not only performs well in rough localization but also maintains high precision under stricter localization requirements (*i.e.*, higher IoU thresholds). This is crucial for robotic manipulation tasks, where precise segmentation is essential for accurate grasping and manipulation. Furthermore, the results suggest the importance of grounding accuracy for subsequent tasks and indicate the enhanced spatial awareness provided by 3D-based motion reasoning.

4.3 Ablation Studies

To quantify the contribution of AffordBot, we conduct ablation studies focusing solely on the affordance-grounding task, which is the critical prerequisite for motion estimation and downstream execution. This targeted approach is justified because the investigated modules (representation design and the MLLM-driven view-selection mechanism) operate entirely upstream of motion estimation. Once a target is accurately grounded, motion prediction relies solely on that grounded object and a fixed MLLM prompt. Consequently, any modifications to these upstream components propagate through the entire pipeline and are comprehensively reflected in grounding performance metrics.

Ablation on Key Components. Through systematic component-wise analysis, we demonstrate how progressive module integration contributes to the performance, as shown in Tab. 2. Baseline Ex1, adapted from [68], initially employs the MLLM to parse instructions and identify target affordance types. It then renders all matching segmented elements from the scene's center view, annotating each with unique identifiers at their 2D centroids. Finally, the MLLM processes these rendered views to localize the correct element. Adding Adaptive Label Refinement (ALR, Ex2) identifier labels to avoid occlusion, our method yields a modest but consistent lift of $+0.4\%$ AP_{25}. The major improvement comes from enriched visual synthesis (EVS, Ex3). The model gains much richer context, pushing AP_{25} from 16.1% to 22.1% ($+6.0\%$). This significant improvement demonstrates that global, information-dense observation, as provided by EVS, is much more valuable than the single frame used in [68]. Finally, Ex4 employs the active view selector (AVS). This *focus-then-infer* pipeline both trims redundant visual information and exploits the best evidence, raising AP_{25} to 23.3% and achieving the highest overall accuracy.

Ablation on Viewpoint selection. To probe how viewpoint choice affects subsequent inference, we evaluate this in Tab. 3. While Bird's-Eye View (BEV) representations provide scene overview, they prove ineffective for our task as affordance elements typically require fine-grained appearance details due to their small size. Sampling images directly from the video stream (*i.e.* Video Frame baseline) also yields limited effectiveness, outperforming the previous results. Query-aligned baseline retrieves a single pre-tagged frame whose affordance class matches the query.

Our Dynamic strategy takes a different route: it first synthesises a dense 360° sweep of surround views, then asks the MLLM to select the frame that best matches the instruction. This active "observe-then-infer" routine supplies rich global context while keeping the final input compact, boosting AP_{25} to 23.3%, an absolute gain of 7.6% over the query-aligned method, as shown in Tab. 3.

Probing the Primary Bottleneck of AffordBot. To investigate the primary bottlenecks constraining upstream segmentation and downstream active view selection performance, we progressively replace Mask3D's predicted masks with ground-truth masks (GT proposals) and provide an ideal front-view perspective (GT proposals + views), as summarized in Table 4. The first row shows the baseline configuration (Mask3D proposals), which corresponds to our AffordBot. Replacing the predicted

Table 4: **Probing the bottleneck of our method**. 'Mask3D proposals' refers to our Affordbot, which uses predicted proposals. 'GT proposals' denotes the use of ground-truth masks, while 'GT proposals & views' additionally adopt ground-truth views.

Method	AP	AP_{50}	AP_{25}
Mask3D proposals	15.5	20.0	23.3
GT proposals	35.7	39.4	45.4
GT proposals & views	38.3	42.3	47.4

Table 5: **Comparison of different MLLMs.** Deployable models (LLaVA-v1.6-34B, Qwen2.5-VL-72B) and commercial GPT APIs show consistent trends, with larger models yielding stronger performance.

Method	AP	AP_{50}	AP_{25}
LLaVA-v1.6-34B	10.6	14.2	16.9
Qwen2.5-VL-72B	15.5	20.0	23.3
GPT-4o	16.5	22.1	28.9
GPT-o1	24.8	30.3	33.4

Table 6: **Performance variation across affordance types**. 'Segment' measures upstream segmentation accuracy, while 'Reason' reports final grounding.

Type	rotate	key_press	tip_push	hook_pull	pinch_pull	hook_turn	foot_push	plug_in	unplug
Segment	0.0	11.3	5.3	27.5	27.1	67.8	100.0	11.1	15.3
Reason	2.5	30.4	5.1	18.0	23.5	45.1	100.0	8.3	16.7

Table 7: **Performance variation with different numbers of target elements**. We compare tasks involving a single ground-truth element ('Unique') *versus* multiple elements ('Multiple').

Method	mIoU	AP	AP_{50}	AP_{25}	+T	+TD
Unique	13.2	13.8	18.2	21.4	16.5	9.9
Multiple	19.1	27.2	32.1	35.8	30.2	17.0
Overall	14.0	15.5	20.0	23.3	18.3	10.8

masks with GT proposals results in a 22.1% boost in AP25, highlighting that *instance segmentation noise is the primary limiting factor*. With perfect segmentation, adding the optimal viewpoint further improves performance by +2.0% AP25, suggesting that while active perception can still be optimized, it is not the dominant bottleneck.

Comparison of Different MLLMs. We replace the default Qwen2.5-VL-72B with several representative alternatives (LLaVA-v1.6-34B, GPT-4o, and GPT-o1), as reported in Tab. 5. While the commonly used Qwen achieves 23.3% AP_{25}, adopting the more advanced GPT-o1, which features superior reasoning and visual understanding, further boosts performance to 33.4% AP_{25}. This demonstrates that leveraging stronger MLLMs can unlock even greater potential within our framework.

Performance Variation Analysis. We conduct a detailed performance analysis to uncover variations across different affordance types and target element counts. Tab. 6 highlights significant disparities in AP_{50} across different affordance types, partly reflecting dataset class imbalance and a strong dependence on initial segmentation quality. Performance ranges widely (*e.g.*, 100% for "foot_push" vs. 0% for "rotate"]), limiting grounding accuracy. The MLLM improves performance for some categories using linguistic cues (*e.g.*, "key_press"), but challenges remain in aligning visual and language cues for others (*e.g.*, "tip_push" degradation).

Further analysis of task subsets (Tab. 7) reveals better performance for "Multiple" target elements than "Unique" ones. This difference is primarily due to Mask3D struggling with the small, weakly-textured objects typical of "Unique" instances, resulting in noisier descriptors that hinder subsequent affordance grounding and motion estimation.

4.4 Qualitative Results

Fig. 5 provides detailed qualitative results of our fine-grained reasoning task. AffordBot generally shows more accurate and consistent grounding of target elements, particularly in complex scenarios with multiple or small targets. Overall, qualitative evidence further suggests that AffordBot achieves significantly improved affordance understanding compared to the prior SOTA method [37].

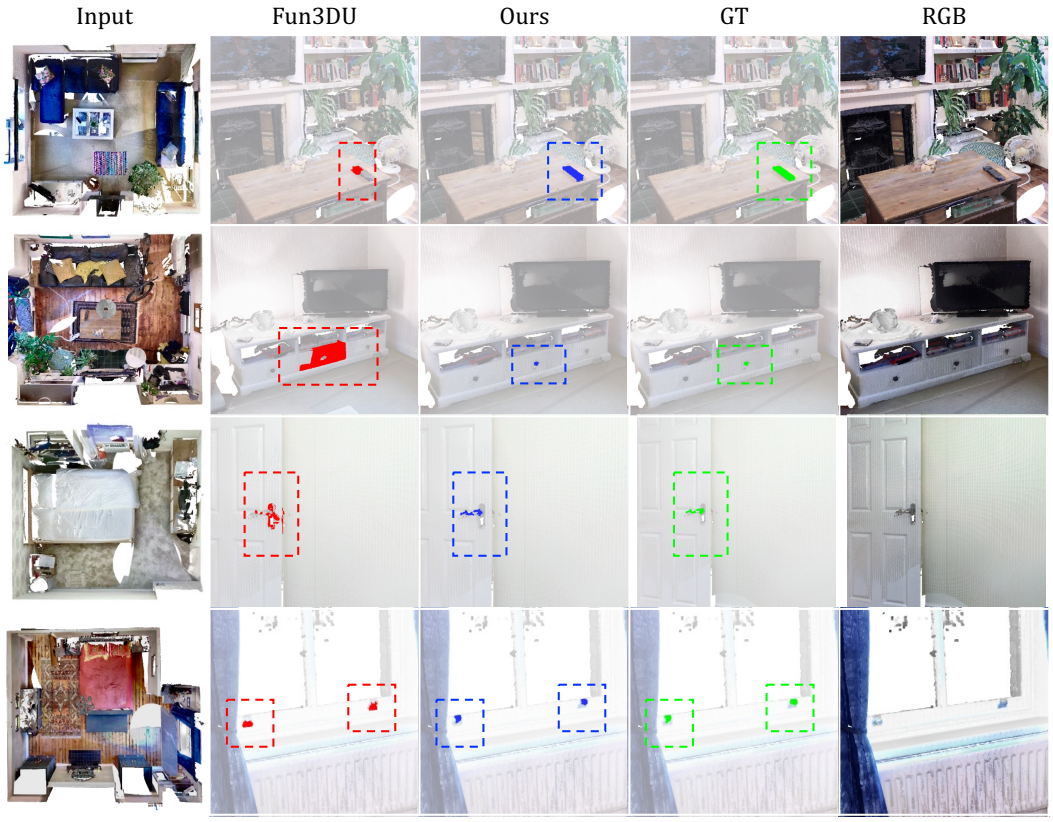


Figure 5: **Qualitative Results.** The figure showcases visual examples of AffordBot performing fine-grained grounding. The illustrated examples include: (1) *"Turn on the TV using the remote control on the table."* (2) *"Open the middle drawer of the TV stand."* (3) *"Close the bedroom door."* (4) *"Open the window above the radiator"*. Please zoom in digitally to view more details.

5 Conclusion

Fine-grained embodied reasoning in 3D worlds serves as a crucial bridge from perception to action, making it pivotal for agents to perform sophisticated tasks. While prior work has explored affordance grounding and motion estimation in isolation, our work unifies these tasks under a structured reasoning framework, AffordBot, bridging perception and action through instruction-aware triplet prediction. By leveraging MLLM with a tailored chain-of-thought paradigm, our method ensures physically grounded reasoning that advances from scene perception to affordance localization and motion synthesis. Through extensive experiments, AffordBot not only advances state-of-the-art performance but also demonstrates the feasibility of efficient, task-coherent embodied reasoning, paving the way for more intuitive human-agent interaction in complex 3D spaces. In the future, we will empower AffordBot with more advanced multimodal understanding ability [69].

Acknowledgments

This work was supported by the National Natural Science Foundation of China (NSFC) under Grant U22A2094, and also supported by the Ministry of Education, Singapore, under its MOE Academic Research Fund Tier 2 (MOE-T2EP20124-0013). We also acknowledge the support of the Supercomputing Center of USTC for providing advanced computing resources and of the NSFC with Grant No. 62573371.

References

[1] Mohammed Hassanin, Salman Khan, and Murat Tahtali. Visual affordance and function understanding: A survey. *ACM Computing Surveys (CSUR)*, 54(3):1–35, 2021.

[2] Paola Ardón, Eric Pairet, Katrin S Lohan, Subramanian Ramamoorthy, and Ronald Petrick. Building affordance relations for robotic agents-a review. *arXiv preprint arXiv:2105.06706*, 2021.

[3] Dongpan Chen, Dehui Kong, Jinghua Li, Shaofan Wang, and Baocai Yin. A survey of visual affordance recognition based on deep learning. *IEEE Transactions on Big Data*, 9(6):1458–1476, 2023.

[4] James J Gibson. The theory of affordances:(1979). In *The people, place, and space reader*, pages 56–60. Routledge, 2014.

[5] Shengheng Deng, Xun Xu, Chaozheng Wu, Ke Chen, and Kui Jia. 3d affordancenet: A benchmark for visual object affordance understanding. In *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1778–1787, 2021.

[6] Kaichun Mo, Yuzhe Qin, Fanbo Xiang, Hao Su, and Leonidas Guibas. O2o-afford: Annotation-free large-scale object-object affordance learning. In *Conference on robot learning*, pages 1666–1677. PMLR, 2022.

[7] Yuhang Yang, Wei Zhai, Hongchen Luo, Yang Cao, Jiebo Luo, and Zheng-Jun Zha. Grounding 3d object affordance from 2d interactions in images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10905–10915, 2023.

[8] Tushar Nagarajan and Kristen Grauman. Learning affordance landscapes for interaction exploration in 3d environments. *Advances in Neural Information Processing Systems*, 33:2005–2015, 2020.

[9] Chao Xu, Yixin Chen, He Wang, Song-Chun Zhu, Yixin Zhu, and Siyuan Huang. Partafford: Part-level affordance discovery from 3d objects. *arXiv preprint arXiv:2202.13519*, 2022.

[10] Ruihai Wu, Kai Cheng, Yan Zhao, Chuanruo Ning, Guanqi Zhan, and Hao Dong. Learning environment-aware affordance for 3d articulated object manipulation under occlusions. *Advances in Neural Information Processing Systems*, 36, 2024.

[11] Songyou Peng, Kyle Genova, Chiyu Jiang, Andrea Tagliasacchi, Marc Pollefeys, Thomas Funkhouser, et al. Openscene: 3d scene understanding with open vocabularies. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 815–824, 2023.

[12] Kaichun Mo, Leonidas J Guibas, Mustafa Mukadam, Abhinav Gupta, and Shubham Tulsiani. Where2act: From pixels to actions for articulated 3d objects. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6813–6823, 2021.

[13] Chuanruo Ning, Ruihai Wu, Haoran Lu, Kaichun Mo, and Hao Dong. Where2explore: Few-shot affordance learning for unseen novel categories of articulated objects. *Advances in Neural Information Processing Systems*, 36, 2024.

[14] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.

[15] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

[16] Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024.

[17] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR, 2023.

[18] Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24185–24198, 2024.

[19] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023.

[20] Zhen Zeng, Leijiang Gu, Xun Yang, Zhangling Duan, Zenglin Shi, and Meng Wang. Visual-oriented fine-grained knowledge editing for multimodal large language models. In *ICCV*, 2025.

[21] Haowen Pan, Yixin Cao, Xiaozhi Wang, Xun Yang, and Meng Wang. Finding and editing multi-modal neurons in pre-trained transformers. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 1012–1037, 2024.

[22] Yucheng Han, Na Zhao, Weiling Chen, Keng Teck Ma, and Hanwang Zhang. Dual-perspective knowledge enrichment for semi-supervised 3d object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 2049–2057, 2024.

[23] Pengkun Jiao, Na Zhao, Jingjing Chen, and Yu-Gang Jiang. Unlocking textual and visual wisdom: Open-vocabulary 3d object detection enhanced by comprehensive guidance from text and image. In *European Conference on Computer Vision*, pages 376–392. Springer, 2024.

[24] Linfeng Li and Na Zhao. End-to-end semi-supervised 3d instance segmentation with pcteacher. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5352–5358. IEEE, 2024.

[25] Jiangyi Wang and Na Zhao. Uncertainty meets diversity: A comprehensive active learning framework for indoor 3d object detection. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 20329–20339, 2025.

[26] Na Zhao, Tat-Seng Chua, and Gim Hee Lee. Few-shot 3d point cloud semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8873–8882, 2021.

[27] Chaofan Luo, Donglin Di, Xun Yang, Yongjia Ma, Zhou Xue, Chen Wei, and Yebin Liu. Trame: Trajectory-anchored multi-view editing for text-guided 3d gaussian splatting manipulation. *IEEE Transactions on Multimedia*, 2025.

[28] Dave Zhenyu Chen, Angel X Chang, and Matthias Nießner. Scanrefer: 3d object localization in rgb-d scans using natural language. In *European conference on computer vision*, pages 202–221. Springer, 2020.

[29] Xinyi Wang, Na Zhao, Zhiyuan Han, Dan Guo, and Xun Yang. Augrefer: Advancing 3d visual grounding via cross-modal augmentation and spatial relation-based referring. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 8006–8014, 2025.

[30] 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. *Advances in Neural Information Processing Systems*, 36:20482–20494, 2023.

[31] 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. *arXiv preprint arXiv:2311.12871*, 2023.

[32] Sijin Chen, Xin Chen, Chi Zhang, Mingsheng Li, Gang Yu, Hao Fei, Hongyuan Zhu, Jiayuan Fan, and Tao Chen. Ll3da: Visual interactive instruction tuning for omni-3d understanding reasoning and planning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26428–26438, 2024.

[33] Kuan-Chih Huang, Xiangtai Li, Lu Qi, Shuicheng Yan, and Ming-Hsuan Yang. Reason3d: Searching and reasoning 3d segmentation via large language model. In *International Conference on 3D Vision 2025*, 2025.

[34] Alexandros Delitzas, Ayca Takmaz, Federico Tombari, Robert Sumner, Marc Pollefeys, and Francis Engelmann. Scenefun3d: Fine-grained functionality and affordance understanding in 3d scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14531–14542, 2024.

[35] Justin Kerr, Chung Min Kim, Ken Goldberg, Angjoo Kanazawa, and Matthew Tancik. Lerf: Language embedded radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19729–19739, 2023.

[36] Ayça Takmaz, Elisabetta Fedele, Robert W Sumner, Marc Pollefeys, Federico Tombari, and Francis Engelmann. Openmask3d: Open-vocabulary 3d instance segmentation. *arXiv preprint arXiv:2306.13631*, 2023.

[37] Jaime Corsetti, Francesco Giuliari, Alice Fasoli, Davide Boscaini, and Fabio Poiesi. Functionality understanding and segmentation in 3d scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

[38] Zhihao Yuan, Shuyi Jiang, Chun-Mei Feng, Yaolun Zhang, Shuguang Cui, Zhen Li, and Na Zhao. Scene-r1: Video-grounded large language models for 3d scene reasoning without 3d annotations. *arXiv preprint arXiv:2506.17545*, 2025.

[39] Thanh-Toan Do, Anh Nguyen, and Ian Reid. Affordancenet: An end-to-end deep learning approach for object affordance detection. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 5882–5889. IEEE, 2018.

[40] Yicong Li, Na Zhao, Junbin Xiao, Chun Feng, Xiang Wang, and Tat-seng Chua. Laso: Language-guided affordance segmentation on 3d object. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14251–14260, 2024.

[41] Timo Luddecke and Florentin Worgotter. Learning to segment affordances. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 769–776, 2017.

[42] Xue Zhao, Yang Cao, and Yu Kang. Object affordance detection with relationship-aware network. *Neural Computing and Applications*, 32(18):14321–14333, 2020.

[43] Gen Li, Deqing Sun, Laura Sevilla-Lara, and Varun Jampani. One-shot open affordance learning with foundation models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3086–3096, 2024.

[44] Edmond Tong, Anthony Opipari, Stanley Lewis, Zhen Zeng, and Odest Chadwicke Jenkins. Oval-prompt: Open-vocabulary affordance localization for robot manipulation through llm affordance-grounding. *arXiv preprint arXiv:2404.11000*, 2024.

[45] Claudia Cuttano, Gabriele Rosi, Gabriele Trivigno, and Giuseppe Averta. What does clip know about peeling a banana? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2238–2247, 2024.

[46] Matt Deitke, Christopher Clark, Sangho Lee, Rohun Tripathi, Yue Yang, Jae Sung Park, Mohammadreza Salehi, Niklas Muennighoff, Kyle Lo, Luca Soldaini, et al. Molmo and pixmo: Open weights and open data for state-of-the-art multimodal models. *arXiv preprint arXiv:2409.17146*, 2024.

[47] 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 *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4015–4026, 2023.

[48] Hanxiao Jiang, Yongsen Mao, Manolis Savva, and Angel X Chang. Opd: Single-view 3d openable part detection. In *European Conference on Computer Vision*, pages 410–426. Springer, 2022.

[49] Cheng-Chun Hsu, Zhenyu Jiang, and Yuke Zhu. Ditto in the house: Building articulation models of indoor scenes through interactive perception. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3933–3939. IEEE, 2023.

[50] Xiaohao Sun, Hanxiao Jiang, Manolis Savva, and Angel Chang. Opdmulti: Openable part detection for multiple objects. In *2024 International Conference on 3D Vision (3DV)*, pages 169–178. IEEE, 2024.

[51] Yongsen Mao, Yiming Zhang, Hanxiao Jiang, Angel Chang, and Manolis Savva. Multiscan: Scalable rgbd scanning for 3d environments with articulated objects. *Advances in neural information processing systems*, 35:9058–9071, 2022.

[52] Yuki Kawana, Yusuke Mukuta, and Tatsuya Harada. Unsupervised pose-aware part decomposition for man-made articulated objects. In *European Conference on Computer Vision*, pages 558–575. Springer, 2022.

[53] Xiaogang Wang, Bin Zhou, Yahao Shi, Xiaowu Chen, Qinping Zhao, and Kai Xu. Shape2motion: Joint analysis of motion parts and attributes from 3d shapes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8876–8884, 2019.

[54] Xiaolong Li, He Wang, Li Yi, Leonidas J Guibas, A Lynn Abbott, and Shuran Song. Category-level articulated object pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3706–3715, 2020.

[55] Xianghao Xu, Yifan Ruan, Srinath Sridhar, and Daniel Ritchie. Unsupervised kinematic motion detection for part-segmented 3d shape collections. In *ACM SIGGRAPH 2022 Conference Proceedings*, pages 1–9, 2022.

[56] Liu Liu, Jianming Du, Hao Wu, Xun Yang, Zhenguang Liu, Richang Hong, and Meng Wang. Category-level articulated object 9d pose estimation via reinforcement learning. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 728–736, 2023.

[57] Liu Liu, Anran Huang, Qi Wu, Dan Guo, Xun Yang, and Meng Wang. Kpa-tracker: Towards robust and real-time category-level articulated object 6d pose tracking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 3684–3692, 2024.

[58] 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.

[59] 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.

[60] Donglin Di, Jiahui Yang, Chaofan Luo, Zhou Xue, Wei Chen, Xun Yang, and Yue Gao. Hyper-3dg: Text-to-3d gaussian generation via hypergraph. *International Journal of Computer Vision*, 133(5):2886–2909, 2025.

[61] Rao Fu, Jingyu Liu, Xilun Chen, Yixin Nie, and Wenhan Xiong. Scene-llm: Extending language model for 3d visual understanding and reasoning. *arXiv preprint arXiv:2403.11401*, 2024.

[62] Youquan Liu, Lingdong Kong, Jun Cen, Runnan Chen, Wenwei Zhang, Liang Pan, Kai Chen, and Ziwei Liu. Segment any point cloud sequences by distilling vision foundation models. *Advances in Neural Information Processing Systems*, 36:37193–37229, 2023.

[63] Baoxiong Jia, Yixin Chen, Huangyue Yu, Yan Wang, Xuesong Niu, Tengyu Liu, Qing Li, and Siyuan Huang. Sceneverse: Scaling 3d vision-language learning for grounded scene understanding. In *European Conference on Computer Vision*, pages 289–310. Springer, 2024.

[64] 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 *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[65] Jonas Schult, Francis Engelmann, Alexander Hermans, Or Litany, Siyu Tang, and Bastian Leibe. Mask3d: Mask transformer for 3d semantic instance segmentation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8216–8223. IEEE, 2023.

[66] Zhening Huang, Xiaoyang Wu, Xi Chen, Hengshuang Zhao, Lei Zhu, and Joan Lasenby. Openins3d: Snap and lookup for 3d open-vocabulary instance segmentation. *European Conference on Computer Vision*, 2024.

[67] David Rozenberszki, Or Litany, and Angela Dai. Language-grounded indoor 3d semantic segmentation in the wild. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022.

[68] Rong Li, Shijie Li, Lingdong Kong, Xulei Yang, and Junwei Liang. Seeground: See and ground for zero-shot open-vocabulary 3d visual grounding. *arXiv preprint arXiv:2412.04383*, 2024.

[69] Zhiyuan Han, Beier Zhu, Yanlong Xu, Peipei Song, and Xun Yang. Benchmarking and bridging emotion conflicts for multimodal emotion reasoning. In *ACM Multimedia*, 2025.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

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

   Answer: [Yes]

   Justification: We prove our claim through quantitative and qualitative experiments.

   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: Yes, in Appendix.

   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: There is no theoretical results.

   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: Yes, we will release the inference code on GitHub.

   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: No, We are committed to reproducible research and plan to publicly release our source code upon acceptance of the paper.

   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: Yes, see the Experiments section and Appendix for details.

   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: No, error bars are not available.

   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: Yes, in the Experiments section.

   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: Yes, the research in the paper conform, in every respect, with the NeurIPS Code of Ethics.

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: [Yes]

    Justification: Yes, in Appendix.

    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 paper poses no such risks.

    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: Yes, all existing assets used in this paper are properly credited with citations. And the license of assets are listed in Appendix.

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: No new assets.

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 paper does not use crowdsourcing or conduct research with human subjects.

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: [Yes]

    Justification: In the study described in this paper, no potential risks were identified for participants. The research design was carefully crafted to ensure the safety and well-being of all individuals involved.

    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 rigorousness, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: Yes, see the Method and Experiment section for details.

    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.