

# Open-Vocabulary 3D Instruction Ambiguity Detection

Jiayu Ding Haoran Tang Ge Li\*

School of Electronic and Computer Engineering, Peking University

{jyding25, hrtang}@stu.pku.edu.cn, geli@pku.edu.cn

## Abstract

*In safety-critical domains, linguistic ambiguity can have severe consequences; a vague command like “Pass me the vial” in a surgical setting could lead to catastrophic errors. Yet, most embodied AI research overlooks this, assuming instructions are clear and focusing on execution rather than confirmation. To address this critical safety gap, we are the first to define Open-Vocabulary 3D Instruction Ambiguity Detection, a fundamental new task where a model must determine if a command has a single, unambiguous meaning within a given 3D scene. To support this research, we build Ambi3D, the large-scale benchmark for this task, featuring over 700 diverse 3D scenes and around 22k instructions. Our analysis reveals a surprising limitation: state-of-the-art 3D Large Language Models (LLMs) struggle to reliably determine if an instruction is ambiguous. To address this challenge, we propose AmbiVer, a two-stage framework that collects explicit visual evidence from multiple views and uses it to guide an vision-language model (VLM) in judging instruction ambiguity. Extensive experiments demonstrate the challenge of our task and the effectiveness of AmbiVer, paving the way for safer and more trustworthy embodied AI. Code and dataset available at <https://jiayuding031020.github.io/ambi3d/>.*

## 1. Introduction

The reliability of an agent’s interaction with the physical world heavily depends on the precision of its understanding of human instructions. Consider a safety-critical robotics scenario (see Figure 1): a surgeon instructs a robot, “Pass me the vial from the tray”, when a bottle of benign herbal extract and a lethal anesthetic are placed side-by-side. A system incapable of recognizing and resolving this instructional ambiguity might select an object arbitrarily, with potentially catastrophic consequences. Such safety risks, arising from linguistic ambiguity, are prevalent in numerous domains, including home services, industrial automation, and



Figure 1. This high-stakes scenario highlights a critical safety challenge where an ambiguous instruction forces a robot to choose between a harmless substance and a lethal one.

Augmented Reality (AR) assisted operations. A trustworthy intelligent system must first confirm its interpretation is unambiguous before proceeding with execution.

Despite the critical importance of this challenge, the research focus in embodied intelligence has predominantly centered on “grounding” language in vision and subsequent “execution”. While this paradigm has achieved remarkable success in many tasks, its inherent unambiguous instruction assumption introduces significant latent risks. Visual Question Answering (VQA), for instance, has significantly advanced multimodal understanding, but its models are often optimized on datasets where questions are presumed to have a single, verifiable answer. For example, in a smart home context, confronted with the question, “Was the stove in the kitchen turned off when I left?” a VQA model might observe that the main stovetop is off and answer “Yes”, overlooking a portable induction cooktop still operating in a corner. This affirmative answer creates a false sense of security, rooted in the model’s inability to recognize that the scope of “the stove in the kitchen” is itself ambiguous. Fundamentally, these models are designed to find the “right answer” to an input assumed to be valid, lacking an intrinsic mechanism to identify when the input itself is a “bad question”. This challenge is exacerbated in complex 3D environments, where referents may be partially occluded and their spatial relationships are often view-dependent. A truly reliable system must possess the ability to recognize such ambiguity

\*Corresponding author.

and proactively seek clarification, rather than blindly guessing.

This tendency toward blind guessing reflects a long-standing bias in embodied AI research: an emphasis on the correctness of instruction execution while overlooking the executability of the instruction itself. Recent studies have begun to address linguistic ambiguities arising in human–agent interaction, proposing two broad classes of resolution strategies. The first is *passive resolution*, which makes a “best guess” under uncertainty: the agent either infers the most plausible target from contextual priors [4, 15] or executes a tentative action and relies on human feedback for correction [5, 16]. The second is *active clarification* [20, 22], where the agent proactively queries the user when recognizing low confidence or uncertain actions. However, both paradigms base ambiguity judgments on the model’s *internal subjective state*. As a result, a model may express high confidence in an objectively ambiguous instruction yet hesitate over a clear one. Consequently, the agent lacks the foundational ability to first ask: “*Is this instruction objectively ambiguous given this specific 3D environment?*”

To systematically address this fundamental safety problem, we are the first to formally define the new task of **Open-Vocabulary 3D Instruction Ambiguity Detection**. The task requires a model to take a 3D scene and an open-vocabulary natural language instruction as input, and to determine whether the instruction is ambiguous. To facilitate systematic research on this new task, we construct **Ambi3D**, the large-scale benchmark featuring a meticulously human-annotated set of instructions that capture complex referential and execution ambiguities grounded in real-world scenes. However, our analysis reveals a surprising limitation in existing methods: state-of-the-art 3D LLMs struggle to reliably determine if an instruction is ambiguous. To overcome this limitation, we propose **Ambiguity Verifier** (AmBiVer), a two-stage framework that decouples scene perception from logical reasoning. Its perception stage converts the raw scene and instruction into a set of structured evidence. The reasoning stage then passes this evidence to a zero-shot VLM for logical adjudication.

In summary, our contributions are as follows:

- We define and formalize the critical safety task of Open-Vocabulary 3D Instruction Ambiguity Detection.
- We build **Ambi3D**, a large-scale benchmark for this task, featuring ~22k human-annotated instructions grounded in over 700 diverse real-world 3D scenes.
- We propose **AmBiVer**, a novel two-stage framework that decouples perception from logical reasoning, leveraging a VLM for zero-shot, evidence-based adjudication.

## 2. Related Works

**Linguistic Ambiguity** Linguistic ambiguity is a fundamental challenge in Natural Language Processing (NLP), stemming from words or phrases corresponding to multiple meanings [3, 24]. Prior work has primarily focused on ambiguity arising from language-internal factors, branching into several key directions. One direction is lexical ambiguity [1, 17, 21], which refers to a string of sounds or characters corresponding to multiple lexical or semantic interpretations. Another research avenue is syntactic ambiguity [1]. This ambiguity arises not from individual words but from the grammatical structure of a sentence or phrase, leading to multiple valid parsing interpretations for the entire sentence. Additionally, other types of ambiguity, such as semantic ambiguity [18] and phonological ambiguity [7], have also been explored. However, this work is different from our research, as it concentrates on language-internal ambiguity rooted in lexicon or syntax. In contrast, we focus on grounded instructional ambiguity, which is a property jointly determined by the instruction and the 3D scene.

**Ambiguity Resolution in Embodied AI.** Although prior work in embodied AI has recognized the issue of linguistic ambiguity, research has largely focused on resolving rather than detecting it. Existing approaches generally fall into two paradigms. The first, *passive resolution*, includes both explicit and implicit forms. Explicit methods adopt a human-in-the-loop scheme [5, 16], where the agent executes a “best-guess” action and relies on human feedback for correction. Implicit methods, by contrast, bypass explicit clarification by inferring missing information from context [4], predicting the most plausible target [15], or exploiting auxiliary modalities such as gestures [23]. The second, *active clarification* [20, 22], allows the agent to proactively query the user when facing low confidence or uncertainty in its action plan, thereby seeking explicit input to resolve ambiguity. Furthermore, their associated benchmarks usually measure downstream task success (e.g., navigation) but not the correctness of the clarification decision itself. In contrast, our work formalizes the upstream task of objective ambiguity detection, enabling an agent to adjudicate ambiguity based on explicit, factual 3D scene evidence.

**Open-Vocabulary 3D Scene Understanding** The field of open-vocabulary 3D scene understanding seeks to interpret and interact with 3D environments using free-form natural language. This field encompasses several key tasks, notably 3D Visual Question Answering (VQA), which requires generating answers to questions about a scene [2, 26, 30], and 3D Referring Expression Comprehension (REC), which focuses on locating objects from textual descriptions [6, 8, 12, 13, 19]. Across these tasks, the predominant trend has been a shift from task-specific expert models to versatile, general-purpose 3D LLMs [9, 11, 27–29]. These 3D LLMs align 3D features with the LLM embedding space, unlock-

ing strong reasoning capabilities. Both expert models and advanced 3D LLMs are architecturally biased to “find the best answer” or “locate the best match”, operating on the implicit assumption that the instruction is clear and unambiguous. Our work addresses this gap by formalizing objective ambiguity detection as a crucial prerequisite for safe and reliable 3D scene interaction.

### 3. Task Formulation

In this section, we formalize our task by defining 3D instruction ambiguity from an execution-oriented perspective and characterizing its core types (§3.1). Building upon this foundation, we present the formal task definition (§3.2).

#### 3.1. Formalizing 3D Instruction Ambiguity

Diverging from traditional NLP’s focus on intrinsic textual ambiguity, our research investigates scene-grounded instructional ambiguity as it pertains to an agent’s task execution in the 3D physical world. This execution-centric focus necessitates that we define ambiguity from a pragmatic perspective: *“An instruction is ambiguous if insufficient information or vague descriptions compel an agent to rely on hazardous guesswork or request clarification to ensure safe completion”*. This allows us to focus on critical failures in human-robot interaction while filtering out acceptable vagueness. We categorize these critical issues into two primary classes: Referential Ambiguity and Execution Ambiguity.

Referential Ambiguity arises when an instruction fails to allow an agent to isolate a single, well-defined set of target objects. We identify three primary types of this ambiguity: Instance Ambiguity, Attribute Ambiguity, and Spatial Ambiguity. **1)** Instance Ambiguity occurs when an instruction uses a general class name (e.g., “the cup” or “the chair”) without any distinguishing features, but multiple objects of that class exist in the scene. **2)** Attribute Ambiguity arises from the use of subjective (e.g., “the nice book”) or relative adjectives (e.g., “the large chair”) that are not explicitly unique (e.g., “largest”). The vagueness of these terms can result in multiple objects satisfying the description. **3)** Spatial Ambiguity stems from observer-dependent spatial terms (e.g., “to the left of”), where the instruction’s correct interpretation changes based on the agent’s or user’s viewpoint.

In contrast to referential issues, Execution Ambiguity occurs when the target object is clear, but the action verb itself has multiple plausible and mutually exclusive interpretations (e.g., “deal with the cup” could mean to set it upright, move it, or discard it).

Based on this, we define an instruction as unambiguous only if it maps precisely and uniquely to a single target object or a fixed set of target objects, and its core action entails no conflicting interpretations.

### 3.2. Task Definition

We introduce the task of Open-Vocabulary 3D Instruction Ambiguity Detection. We formally define this as a binary classification problem, where a model  $\mathcal{F}$  learns the mapping  $\mathcal{F} : (S, T) \mapsto y$ . The inputs consist of a 3D scene representation  $S$  and an open-vocabulary natural language instruction  $T$ . The model must output a single binary label  $y \in \{\text{Unambiguous}, \text{Ambiguous}\}$ . The core challenge is to compel  $\mathcal{F}$  to replace the conventional forced-choice selection paradigm with a rigorous, quantity-aware perceptual and reasoning process to verify instructional uniqueness.

### 4. Benchmark

To facilitate the systematic study of Open-Vocabulary 3D Instruction Ambiguity Detection, we construct **Ambi3D**, the first benchmark for this task, built upon the diverse, real-world indoor scenes from the ScanNet dataset. In this section, we detail the dataset’s acquisition pipeline, annotation process, and statistics (§4.1) and specify the evaluation protocol (§4.2).

#### 4.1. Dataset

**Instruction Acquisition Pipeline** We designed a meticulous instruction acquisition pipeline to ensure the dataset’s authenticity, diversity, and challenge. The pipeline consists of three main components: **1)** Grounded Instructions: We leverage the high-quality, human-annotated question-answer (QA) pairs from the ScanQA dataset. We employ an LLM-based framework to automatically convert these questions (e.g., “What is the tallest object on the table?”) into semantically equivalent, executable instructions (e.g., “Please pick up the tallest object on the table”). This provides a robust, real-world semantic foundation for a subset of our data. **2)** Synthetic Ambiguous Instructions: To systematically cover the ambiguity types defined in Section 3.1, we designed a prompt-engineered LLM generation process based on detailed object metadata from ScanNet scenes. This process uses specific templates to target the generation of four ambiguity types: Instance, Attribute, Spatial, and Action Ambiguity. **3)** Hard Negative Instructions: To evaluate model robustness and prevent superficial heuristics, we also constructed a set of special unambiguous instructions. These instructions appear ambiguous on the surface (e.g., referring to a “chair” in a scene with multiple chairs) but are implicitly disambiguated by a unique qualifier (e.g., “the chair by the window”). To ensure the quality and subtlety of these samples, this subset was entirely human-authored by experts.

**Annotation and Quality Control** To ensure high-fidelity labels, all generated instructions underwent a multi-stage verification process conducted by 12 trained annotators, all possessing backgrounds in 3D domains. To ensure

proficiency, annotators were provided with comprehensive guidelines detailing ambiguity definitions and boundary cases, and were required to pass a qualification test before beginning. Our annotation process consisted of the following three stages: **1) Data Cleaning and Filtering:** We first applied an automated script to identify and remove instructions that were exact duplicates within the same scene. Subsequently, human annotators performed a manual review to filter out any remaining instructions that were grammatically incorrect, semantically nonsensical, or irrelevant to the 3D scene context. **2) Core Ambiguity Annotation:** Following cleaning, each valid instruction entered the core annotation stage, where it was independently assigned to three different annotators. Annotators provided two labels based on the 3D scene context: a primary binary label (Unambiguous or Ambiguous) and, if ambiguous, a secondary sub-type classification (e.g., Instance, Attribute, Action). **3) Consistency Check and Final Selection:** Finally, we employed a strict unanimous agreement protocol for final selection. To ensure maximum reliability for the primary task, we retained only those instructions where all three annotators reached a unanimous agreement on the binary label. For the retained ambiguous samples, the final sub-type label was determined by a majority vote. If at least two of the three annotators agreed on a specific sub-type, that type was assigned. However, in cases where no majority was reached (i.e., all three annotators selected a different sub-type), the sample was discarded from the final dataset. This protocol eliminates arbitrary bias by ensuring every assigned sub-type is backed by a consensus of at least two annotators.

**Dataset Statistics and Splits** The Ambi3D benchmark contains 22,081 instructions grounded in 703 unique indoor scenes from ScanNet. It features 10,480 Unambiguous instructions (47.5%) and 11,601 Ambiguous instructions (52.5%). The ambiguous instructions are categorized by type: 5,333 Instance (46.0%), 2,302 Action (19.8%), 2,216 Attribute (19.1%), and 1,750 Spatial (15.1%). Further dataset details and examples are available in the appendix.

These instructions originate from our three acquisition pipelines: 8,224 Grounded Instructions (37.2%), 7,522 Synthetic Ambiguous Instructions (34.1%), and 6,335 Hard Negative candidates (28.7%). All instructions, regardless of their pipeline source, received their final ground-truth label from the identical human annotation process.

For experiments, we split the dataset at the scene level into 649 scenes for training and 54 for testing. The training set comprises 19,950 instructions (90.3%), with 10,528 ambiguous (52.8%) and 9,422 unambiguous (47.2%) samples. The test set contains the remaining 2,131 instructions (9.7%), with 1,073 ambiguous (50.4%) and 1,058 unambiguous (49.6%), ensuring an unbiased evaluation.

Linguistically, instructions have an average length of 8.08 words. Nearly half (49.35%) fall into a medium-

complexity range (6–10 words), with the remainder balanced between simple ( $\leq 5$  words, 28.11%) and complex ( $> 10$  words, 22.53%), testing model robustness across diverse syntactic structures.

## 4.2. Evaluation Metrics

We evaluate this task as a binary classification problem. For a comprehensive assessment of overall performance, we report standard Accuracy (Acc.), Precision (Prec.), Recall (Rec.) and the macro-averaged F1-Score (F1). Furthermore, to provide a fine-grained diagnostic, we report the Accuracy breakdown for each specific category: Instance, Attribute, Spatial, Action, and Unambiguous.

## 5. Method

We propose **AmbiVer**, the first unified framework for 3D instruction ambiguity detection. In this section, we first outline the overall system architecture in §5.1. We then detail its two core components: the perception engine (§5.2) for extracting visual evidence, and the reasoning engine (§5.3) for adjudicating ambiguity based on this evidence.

### 5.1. Overall Architecture

The AmbiVer pipeline, illustrated in Figure 2, is a decoupled, two-stage framework: a perception engine (§5.2) and a reasoning engine (§5.3). The pipeline operates on an egocentric video stream  $\mathcal{V} = \{\mathcal{I}_t\}_{t=1}^N$ , corresponding camera extrinsics  $\mathcal{E} = \{\mathcal{E}_t\}_{t=1}^N$ , and a natural language instruction  $T$ . The Perception Engine is responsible for processing these raw inputs, converting the video stream and instruction into a set of structured, multimodal evidence. This evidence is then passed to the reasoning engine, which employs a zero-shot VLM to perform logical deliberation on the evidence, ultimately outputting a structured verdict that adjudicates the instruction’s ambiguity.

### 5.2. Perception Engine: From Pixels to Evidence

The perception engine must process two challenging and unstructured inputs: **1)** the free-form language instruction  $T$ , which must be parsed, and **2)** the egocentric video stream  $\mathcal{V}$ , which offers only partial and localized observability. To convert this raw data into actionable evidence, we design a multi-track pipeline. This section details the three core sub-modules: **1)** Instruction Decoupling, which parses  $T$  into a structured representation of its core components; **2)** Global Feature Acquisition, which aggregates  $\mathcal{V}$  into a unified, allocentric Bird’s-Eye View (BEV) map  $\mathcal{I}_{\text{bev}}$  for scene-level context; and **3)** Detailed Feature Acquisition, which localizes object candidates from  $\mathcal{V}$  and resolves the multi-view redundancies to identify all potential 3D instances.

**Instruction Decoupling** We employ spaCy to perform lexical analysis (tokenization and part-of-speech tagging) and syntactic analysis (dependency parsing and named entity

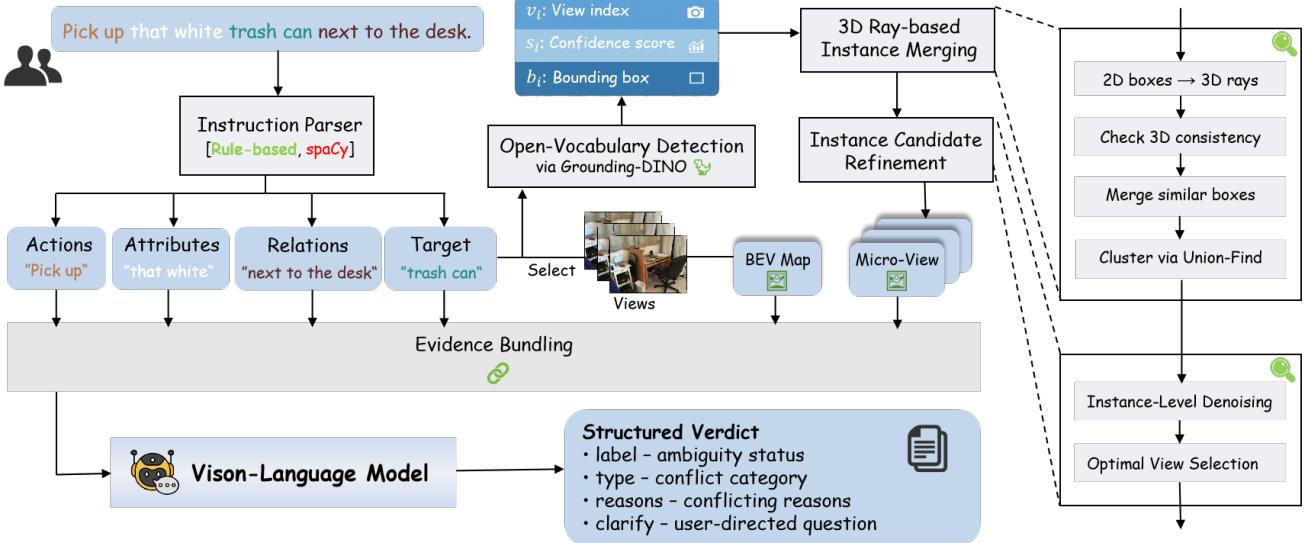


Figure 2. Overview of the **AmbiVer** framework. AmbiVer is a two-stage system composed of a perception engine and a reasoning engine. The perception stage parses an instruction into action, attribute, relation, and target components, employs an open-vocabulary grounding method to detect 2D candidates across views, and integrates them into 3D instances via ray-based fusion followed by refinement. It also generates a BEV map for scene-level context. The reasoning stage then performs multimodal evidence bundling and leverages a VLM to assess instruction ambiguity, outputting a structured verdict.

recognition) on the input instruction  $T$ . This process converts the free-form text  $T$  into a structured key-value representation containing essential elements such as the action, target (denoted as  $Q_t$ ), attributes, and relations. Once parsed, the instruction guides the perception engine along two complementary tracks: constructing a global allocentric map and identifying fine-grained local instances.

**Global Feature Acquisition** The perception engine synthesizes raw visual inputs into actionable evidence. However, the egocentric video stream  $\mathcal{V} = \{\mathcal{I}_t\}_{t=1}^N$  provides only a limited field of view, capturing partial and localized observations. To overcome this constraint and achieve a global understanding of the environment, the engine transforms the first-person views into an allocentric representation of the scene. This process aggregates multi-view observations into a unified 3D point cloud  $\mathcal{P}$ , reconstructed from the video frames  $\mathcal{V}$  and their corresponding camera poses  $\mathcal{E} = \{\mathcal{E}_t\}_{t=1}^N$  through a reconstruction pipeline  $\mathcal{R}$ :

$$\mathcal{P} = \mathcal{R}(\{\mathcal{I}_t, \mathcal{E}_t\}_{t=1}^N). \quad (1)$$

The point cloud  $\mathcal{P}$  encodes the full 3D geometry of the scene. To represent this geometry in a structured 2D form, we project  $\mathcal{P}$  into a BEV image  $\mathcal{I}_{\text{bev}}$  using a transformation  $\mathcal{T}$  defined by a fixed top-down camera extrinsic  $\mathcal{E}_{\text{top}} \in SE(3)$ :

$$\mathcal{I}_{\text{bev}} = \mathcal{T}(\mathcal{P}, \mathcal{E}_{\text{top}}). \quad (2)$$

The BEV image  $\mathcal{I}_{\text{bev}}$  provides a compact and globally consistent spatial layout of the environment, serving as the

foundation for subsequent spatial reasoning.

**Detailed Feature Acquisition** The core task of the perception engine is to identify and enumerate object instances that match the query target  $Q_t$  across multiple views, and to determine the most representative view for each instance. To this end, we design a multi-stage process that progressively refines the visual evidence.

First, processing the entire video stream  $\mathcal{V}$  is computationally prohibitive. We therefore apply an adaptive keyframe selection strategy to downsample the stream to a target frame count  $N_t$ . The algorithm iteratively scans the sequence, retaining a new keyframe only when its pose deviation from the last selected frame exceeds the translational ( $\tau_t$ ) or rotational ( $\tau_r$ ) thresholds. To converge the keyframe count  $N_c$  to the target  $N_t$ , the thresholds  $\tau_t$  and  $\tau_r$  are iteratively adjusted. In each iteration, both thresholds are multiplicatively scaled by a factor  $\alpha_i > 1$  if  $N_c > N_t$  or by  $\alpha_d < 1$  if  $N_c < N_t$ , until  $N_c$  falls within a predefined tolerance window around  $N_t$ . This procedure produces a compact yet diverse set of keyframes  $\{\mathcal{I}_v\}_{v=1}^{N_t}$  and their corresponding poses  $\{\mathcal{E}_v\}_{v=1}^{N_t}$ .

Given these keyframes, the next step is to localize potential object candidates. For each  $I_v$ , we employ the pre-trained open-vocabulary detector Grounding DINO [14] with the query text  $Q_t$ . This process yields an aggregated set of 2D detections  $\mathcal{D} = \{d_i = (v_i, b_i, s_i)\}_{i=1}^M$ , where  $v_i$  is the index of the keyframe  $I_{v_i}$  where the detection was found,  $b_i$  is its 2D bounding box, and  $s_i$  is its detection confidence score. Because the same object can appear in mul-

multiple views,  $\mathcal{D}$  consequently contains redundant detections.

To unify redundant multi-view detections  $\mathcal{D}$  into consistent 3D instances, we construct a connectivity graph where each detection  $d_i$  (back-projected as ray  $\text{Ray}_i = (o_i, r_i)$ ) serves as a node. An edge is formed between two nodes  $d_i$  and  $d_j$  (from distinct views  $v_i \neq v_j$ ) if they satisfy three geometric constraints: **1)** their minimum ray distance is less than  $\epsilon_d$ ; **2)** their ray angle is within  $[\theta_{a,\min}, \theta_{a,\max}]$ ; and **3)** their bounding box area ratio,  $\frac{\min(\text{area}(b_i), \text{area}(b_j))}{\max(\text{area}(b_i), \text{area}(b_j))}$ , exceeds  $\sigma_s$ . An efficient Union-Find algorithm is applied to extract the connected components  $\{\mathcal{G}_k\}_{k=1}^{N_g}$ , each representing a unified 3D instance.

Subsequently, we assign a group-level reliability score  $S_k$  to each group  $\mathcal{G}_k$  based on the area-weighted average confidence of its constituent detections  $d_i \in \mathcal{G}_k$ :

$$S_k = \frac{\sum_{d_i \in \mathcal{G}_k} s_i \cdot \text{area}(b_i)}{\sum_{d_i \in \mathcal{G}_k} \text{area}(b_i)}. \quad (3)$$

Instances are ranked by  $S_k$ , and only the top  $K$  are retained to eliminate low-confidence or fragmented hypotheses. For each selected instance  $\mathcal{G}_k$ , a single representative detection  $d_k^* = (v_k^*, b_k^*, s_k^*)$  is chosen from its members  $d_i \in \mathcal{G}_k$  by maximizing a composite score  $f(d_i)$ . This score is defined as the product of three components:

$$f(d_i) = s_i \cdot w_{\text{vis}}(d_i) \cdot w_{\text{bnd}}(d_i) \quad (4)$$

where  $s_i$  is the raw detection confidence,  $w_{\text{vis}}(d_i) = \text{area}(b_i)/\text{area}(I_{v_i})$  measures visibility within the keyframe  $I_{v_i}$ , and  $w_{\text{bnd}}(d_i)$  is a boundary penalty set to  $\gamma$  if  $b_i$  is within  $\delta$  pixels of any image border, and 1.0 otherwise. The detection  $d_i \in \mathcal{G}_k$  that maximizes this score is selected as the representative  $d_k^*$ .

Finally, this process outputs the set of  $K$  instance candidates  $\mathcal{C}$ , defined as:

$$\mathcal{C} = \{(I_{v_k^*}, b_k^*, S_k, |\mathcal{G}_k|)\}_{k=1}^K \quad (5)$$

where each candidate  $\mathcal{C}_k$  consists of its representative image  $I_{v_k^*}$  (from the keyframe set  $\{I_v\}$ ), bounding box  $b_k^*$ , group reliability score  $S_k$ , and the group's cardinality  $|\mathcal{G}_k|$  (i.e., its cross-view detection count). This set  $\mathcal{C}$ , combined with the BEV map  $\mathcal{I}_{\text{bev}}$ , forms the complete visual evidence for the reasoning engine.

### 5.3. Reasoning Engine: From Evidence to Verdict

The reasoning engine evaluates the structured evidence from the perception pipeline against the instruction's semantic constraints. This process aggregates the evidence into a unified `Dossier` and uses a zero-shot VLM to adjudicate it, producing the final, interpretable `verdict`.

**Structured Evidence Bundling** Before reasoning, we aggregate the outputs from the perceptual pipeline into a structured evidence package, referred to as the `Dossier`. This

package integrates linguistic, geometric, and visual information into a coherent representation for the VLM. It organizes the multimodal data into three complementary components: **1)** Linguistic Context, which contains the raw instruction  $T$  and its parsed components, including the query target  $Q_t$ , attributes, and relational constraints; **2)** Global Spatial Context, represented by the top-down BEV map  $\mathcal{I}_{\text{bev}}$ , providing a holistic view of the environment; and **3)** Local Instance Evidence, corresponding to the set of  $K$  unified instance candidates  $\mathcal{C}$ . Each instance  $\mathcal{C}_k$  is associated with its representative image  $I_{v_k^*}$ , bounding box  $b_k^*$ , reliability score  $S_k$ , and the group's cardinality  $|\mathcal{G}_k|$ .

**VLM as a Zero-Shot Adjudicator** We employ a general-purpose VLM as a zero-shot logical adjudicator, tasked with evaluating whether the perceived scene satisfies the semantic constraints imposed by the instruction. To facilitate such reasoning, `Dossier` (§5.3) is encapsulated within a carefully designed multimodal prompt. This prompt guides the VLM's evidence-based verification by combining the evidence with an explicit execution-oriented ambiguity criterion, which is detailed in the Appendix. The model performs zero-shot reasoning over the full prompt to generate a structured `verdict`, consisting of four fields: a binary ambiguity label (e.g., “Ambiguous” or “Unambiguous”), a set of detected ambiguity types (e.g., “Instance”, “Spatial”), a concise textual explanation, and an optional clarification query for disambiguation. This structured formulation ensures that the results are directly parseable for downstream evaluation while maintaining interpretability.

## 6. Experiments

In this section, we first describe the implementation details (§6.1). We then evaluate our framework through quantitative (§6.2) and qualitative (§6.3) analyses, followed by ablation studies (§6.4) to validate our design.

### 6.1. Implementation Details.

We set the adaptive keyframe selection target to  $N_{\text{target}} = 100$ . For instance unification, we use a ray distance threshold of 0.3m, an angular limit of  $60^\circ$ , and a scale similarity of 0.2. We retain the top  $K = 6$  instances for reasoning. The representative score (Eq. 4) uses a boundary penalty  $\gamma = 0.5$  within  $\delta = 4$  pixels. The zero-shot reasoning is performed using Qwen-3-VL [25] with a temperature of 0. For the LoRA adaptation of 3D LLM baselines, we train for 3 epochs using the AdamW optimizer with a learning rate of  $2 \times 10^{-4}$ , a cosine learning rate schedule, a weight decay of 0.01, and a 3% warmup ratio. The LoRA rank is  $r = 8$ ,  $\alpha = 16$ , and dropout is 0.1. We partition the official training set into a 90% subset for training and a 10% subset for validation. During training, we select the model checkpoint that achieves the highest Acc. score on the validation set. The test set was strictly held out and used only for the final,

Table 1. Main performance comparison on the Ambi3D benchmark in the zero-shot setting. Evaluation is performed on the *entire* dataset for a comprehensive overview. **Bold** / Underline indicates the best / second-best performance per category.

Method	Venue	Overall Metrics				Acc. Breakdown by Type				
		Acc. $\uparrow$	Prec. $\uparrow$	Rec. $\uparrow$	F1 $\uparrow$	Instance	Attribute	Spatial	Action	Unamb.
3D-LLM [9]	NIPS’23	49.16	56.93	13.28	21.54	13.59	14.85	11.71	12.25	88.87
Chat-Scene [11]	NIPS’24	47.73	50.75	17.58	26.11	18.94	11.82	17.14	20.29	81.12
Video-3D LLM [27]	CVPR’25	50.10	60.65	14.31	23.16	12.94	35.79	7.03	2.35	<u>89.72</u>
LSceneLLM [28]	CVPR’25	48.78	58.04	9.07	15.70	9.36	11.01	6.06	8.86	<b>92.74</b>
LLaVA-3D [29]	ICCV’25	64.21	<u>63.93</u>	<u>73.17</u>	<u>68.24</u>	<u>59.55</u>	<b>95.62</b>	<u>62.86</u>	<b>90.96</b>	54.29
<b>AmbiVer</b>	–	<b>81.29</b>	<b>84.23</b>	<b>79.23</b>	<b>81.65</b>	<b>80.14</b>	84.03	<b>75.09</b>	75.63	83.58

one-time evaluation.

## 6.2. Quantitative Analysis

We systematically evaluate 3D LLMs and our proposed AmbiVer framework on the Ambi3D benchmark, with the results detailed in Table 1. As existing baselines are not natively designed for this binary classification task, we adapt them by prompting models to output a ‘0’ (unambiguous) or ‘1’ (ambiguous) token. Since models may generate free-text responses, we employ a robust, rule-based parsing algorithm to extract the final judgment.

Strikingly, as shown in Table 1, all zero-shot baselines perform poorly on Ambi3D, revealing two contrary biases. Most baselines [9, 11, 27, 28] exhibit a strong unambiguous bias, yielding minimal recall. This failure to identify ambiguous cases causes their accuracy on all subtypes to collapse, while the unambiguous category shows an inflated accuracy. Conversely, LLaVA-3D [29] shows an opposing ambiguous bias. While effective on certain subtypes, its performance on unambiguous cases plummets to 54.29%, bordering on random chance. In contrast, AmbiVer uniquely balances conflicts, delivering substantial and uniform gains across all categories.

We compare AmbiVer with 3D LLM baselines fine-tuned parameter-efficiently (LoRA [10]) on Ambi3D, representing a strong adaptation of current paradigms, as shown in Table 2. LoRA notably improves all baselines, showing that the ambiguity task can be learned through supervised training. Models that showed strong unambiguous bias in the zero-shot setting gain an absolute increase of 52 to 63 percentage points in F1 score after fine-tuning, indicating that LoRA effectively corrects their recall failures by teaching them to recognize ambiguous cases. However, despite these data-driven improvements, the fine-tuned baselines still fall short of AmbiVer’s zero-shot performance. This highlights a deeper limitation: while LoRA enforces statistical pattern learning, it fails to provide generalizable reasoning about ambiguity. We argue that the baseline architectures are inherently unsuited for the complex trade-offs and judgment required. In contrast, AmbiVer’s two-

Table 2. Comparison against LoRA fine-tuned baselines on the Ambi3D test set. Our method remains zero-shot.

Model	Acc. $\uparrow$	F1 $\uparrow$
3D-LLM [9] <sup>†</sup>	78.67 $\uparrow$ 27.29	78.54 $\uparrow$ 52.75
Chat-Scene [11] <sup>†</sup>	78.74 $\uparrow$ 29.75	79.17 $\uparrow$ 52.57
Video-3D LLM [27] <sup>†</sup>	79.63 $\uparrow$ 27.26	<u>81.35</u> $\uparrow$ 55.06
LSceneLLM [28] <sup>†</sup>	77.24 $\uparrow$ 26.84	77.05 $\uparrow$ 63.48
LLaVA-3D [29] <sup>†</sup>	80.95 $\uparrow$ 17.32	80.61 $\uparrow$ 14.03
<b>AmbiVer<sup>‡</sup></b>	<b>83.29</b>	<b>83.25</b>

<sup>†</sup> denotes LoRA fine-tuning; <sup>‡</sup> denotes our zero-shot method.

stage architecture achieves state-of-the-art zero-shot performance, surpassing all supervised baselines.

## 6.3. Qualitative Analysis

We present qualitative examples in Figure 3 to illustrate our framework’s efficacy. The figure highlights four representative cases: **1**) an unambiguous instruction (“Pick up the backpack”) where the model correctly identifies the unique referent; **2**) Referential Ambiguity (“...the trash can”) where the model detects multiple candidates and requests clarification; **3**) Action Ambiguity (“...handle the bicycle”) where the target is clear but the verb is flagged as vague; and **4**) Mixed Ambiguity (“...adjust the pillow...”) where our framework identifies both instance and action ambiguities, generating a comprehensive clarification question. These results confirm the efficacy of our decoupled approach.

## 6.4. Ablation Study

**Ablation Study on the Perception Engine** To evaluate the contribution of the perception engine, we design four ablation variants by modifying key components of the pipeline: **1**) Case #1 bypasses instruction preprocessing by directly passing the raw instruction  $T$  to the grounding model without extracting the target query  $Q_t$ . **2**) Case #2 replaces adaptive keyframe selection with uniform temporal sampling of  $N_t$  frames. **3**) Case #3 removes ray-based 3D geometric fusion. The top- $K$  2D detections  $\mathcal{D}$  are selected by confi-

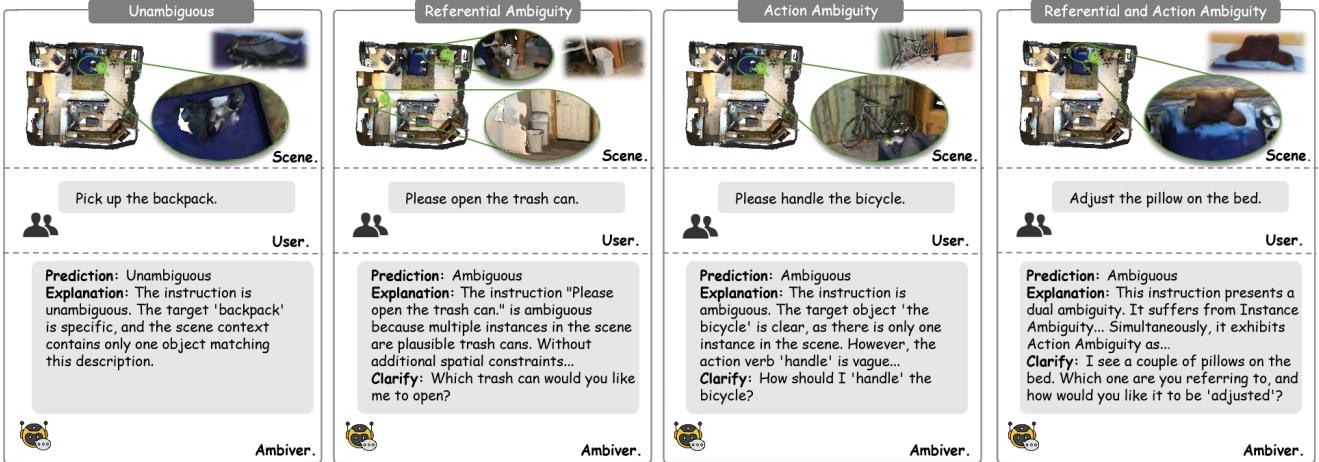


Figure 3. Qualitative results of our AmbiVer framework on the Ambi3D benchmark.

Table 3. Ablation study of the perception engine.

Case	Method	Acc. $\uparrow$	F1 $\uparrow$
#1	w/o Instruction Decoupling	62.06	65.62
#2	w/o Adaptive Keyframe Selection	65.04	54.56
#3	w/o 3D Fusion	58.59	38.42
#4	w/o Refinement Weights	79.93	80.31
#5	Full Perception	<b>81.29</b>	<b>81.65</b>

dence and used as local evidence  $\mathcal{C}$  without spatial redundancy filtering. **4)** Case #4 keeps 3D fusion but simplifies the refinement step. The representative view  $d_k^*$  is chosen solely based on detection confidence, ignoring visibility scores ( $w_{\text{vis}}$ ) and boundary penalties ( $w_{\text{bnd}}$ ). **5)** Case #5 uses the full pipeline.

The ablation results are shown in Table 3. Case #1 results in ambiguity because the full instruction  $T$  is grounded without isolating  $Q_t$ , causing incorrect and irrelevant candidate selection. Case #2 causes significant omission of scene information owing to uniform keyframe sampling. Case #3 shows difficulty in capturing instruction-relevant visual details, often misclassifying ambiguous instructions that lack sufficient visual evidence. Case #4 produces occluded or marginal instances with reduced clarity because the representative views are selected without considering visibility or boundary constraints. Each component, as verified by case #5, contributes to producing complete, clean, and query-relevant 3D evidence.

**Ablation Study on the Reasoning Engine** To evaluate the contribution of each modality in the reasoning pipeline, we ablate different input components in the Dossier received by the VLM during decision-making: **1)** Case #1 removes the BEV map  $\mathcal{I}_{\text{bev}}$ . **2)** Case #2 removes all object crops  $\mathcal{C}$ , leaving the BEV map and language input. **3)** Case #3 removes both  $\mathcal{I}_{\text{bev}}$  and  $\mathcal{C}$ , providing only the raw instruction  $T$  and

Table 4. Ablation on the reasoning engine.

Case	Method	Acc. $\uparrow$	F1 $\uparrow$
#1	w/o Global Context	56.95	70.57
#2	w/o Local Evidence	60.71	53.56
#3	w/o Visual Information	53.99	56.91
#4	Full Reasoning	<b>81.29</b>	<b>81.65</b>

clarification prompt. **4)** Case #4 serves as the baseline and uses all inputs: the BEV map, local object crops, and the full instruction. These ablations isolate the roles of global layout, object-level features, and visual grounding.

The ablation results are shown in Table 4. In case #1, accuracy collapses while F1 remains high. This is caused by a massive increase in false positives, as the model, lacking the BEV map, loses the ability to confirm uniqueness and flags unambiguous instructions as ambiguous. Case #2 causes the most significant performance collapse, proving that the local candidates  $\mathcal{C}$  are critical for identifying fine-grained ambiguities. The failure of Case #3 confirms that visual grounding is essential for this task. Case #4, which integrates all modalities, achieves the best overall accuracy and balance between precision and recall, demonstrating the complementarity of global layout and local evidence.

## 7. Conclusion

We introduce Open-Vocabulary 3D Instruction Ambiguity Detection, a new task for identifying ambiguous instructions in 3D environments. To support this research, we present Ambi3D, a large-scale benchmark for this task, and propose AmbiVer, a two-stage framework that constructs grounded 3D visual evidence and then performs zero-shot ambiguity adjudication using a VLM. Experiments on Ambi3D demonstrate the value of our proposed

task and benchmark, and show that AmbiVer enables more reliable ambiguity detection.

## References

- [1] Miuru Abeysiriwardana and Deshan Sumanathilaka. A survey on lexical ambiguity detection and word sense disambiguation. *arXiv preprint arXiv:2403.16129*, 2024. [2](#)
- [2] Daichi Azuma, Taiki Miyanishi, Shuhei Kurita, and Motoaki Kawanabe. Scanqa: 3d question answering for spatial scene understanding. In *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 19129–19139, 2022. [2](#)
- [3] Daniel M Berry and Erik Kamsties. Ambiguity in requirements specification. In *Perspectives on software requirements*, pages 7–44. Springer, 2004. [2](#)
- [4] Haonan Chen, Hao Tan, Alan Kuntz, Mohit Bansal, and Ron Alterovitz. Enabling robots to understand incomplete natural language instructions using commonsense reasoning. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1963–1969. IEEE, 2020. [2](#)
- [5] Yinpei Dai, Run Peng, Sikai Li, and Joyce Chai. Think, act, and ask: Open-world interactive personalized robot navigation. In *2024 IEEE international conference on robotics and automation (ICRA)*, pages 3296–3303. IEEE, 2024. [2](#)
- [6] Jiayu Ding, Xinpeng Liu, Zhiyi Pan, Shiqiang Long, and Ge Li. Polysemous language gaussian splatting via matching-based mask lifting. *arXiv preprint arXiv:2509.22225*, 2025. [2](#)
- [7] Ram Frost, Laurie B Feldman, and Leonard Katz. Phonological ambiguity and lexical ambiguity: Effects on visual and auditory word recognition. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 16(4):569, 1990. [2](#)
- [8] Shuteng He, Guangquan Jie, Changshuo Wang, Yun Zhou, Shuming Hu, Guanbin Li, and Henghui Ding. ReferSplat: Referring segmentation in 3d gaussian splatting. In *International Conference on Machine Learning*. [2](#)
- [9] 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. [2, 7](#)
- [10] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022. [7](#)
- [11] 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. *Advances in Neural Information Processing Systems*, 37: 113991–114017, 2024. [2, 7](#)
- [12] Aishwarya Kamath, Mannat Singh, Yann LeCun, Gabriel Synnaeve, Ishan Misra, and Nicolas Carion. Mdetr-modulated detection for end-to-end multi-modal understanding. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1780–1790, 2021. [2](#)
- [13] Jingyu Liu, Liang Wang, and Ming-Hsuan Yang. Referring expression generation and comprehension via attributes. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4856–4864, 2017. [2](#)
- [14] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *European conference on computer vision*, pages 38–55. Springer, 2024. [5](#)
- [15] Aly Magassouba, Komei Sugiura, and Hisashi Kawai. A multimodal classifier generative adversarial network for carry and place tasks from ambiguous language instructions. *IEEE Robotics and Automation Letters*, 3(4):3113–3120, 2018. [2](#)
- [16] Arjun Majumdar, Fei Xia, Dhruv Batra, Leonidas Guibas, et al. Findthis: Language-driven object disambiguation in indoor environments. In *7th Annual Conference on Robot Learning*, 2023. [2](#)
- [17] Gergely Pethö. What is polysemy? a survey of current research and results. *Pragmatics and the flexibility of word meaning*, pages 175–224, 2001. [2](#)
- [18] Massimo Poesio. Semantic ambiguity and perceived ambiguity. *arXiv preprint cmp-lg/9505034*, 1995. [2](#)
- [19] Yanyuan Qiao, Chaorui Deng, and Qi Wu. Referring expression comprehension: A survey of methods and datasets. *IEEE Transactions on Multimedia*, 23:4426–4440, 2020. [2](#)
- [20] Allen Z Ren, Anushri Dixit, Alexandra Bodrova, Sumeet Singh, Stephen Tu, Noah Brown, Peng Xu, Leila Takayama, Fei Xia, Jake Varley, et al. Robots that ask for help: Uncertainty alignment for large language model planners. *arXiv preprint arXiv:2307.01928*, 2023. [2](#)
- [21] Vera Dmitrieva Tabanakova et al. Term “homonymy” as a semantic category. *European Proceedings of Social and Behavioural Sciences*, 2021. [2](#)
- [22] Francesco Taioli, Edoardo Zorzi, Gianni Franchi, Alberto Castellini, Alessandro Farinelli, Marco Cristani, and Yiming Wang. Collaborative instance object navigation: Leveraging uncertainty-awareness to minimize human-agent dialogues. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18781–18792, 2025. [2](#)
- [23] Dulanga Weerakoon, Vigneshwaran Subbaraju, Nipuni Karumpulli, Tuan Tran, Qianli Xu, U-Xuan Tan, Joo Hwee Lim, and Archan Misra. Gesture enhanced comprehension of ambiguous human-to-robot instructions. In *Proceedings of the 2020 International Conference on Multimodal Interaction*, pages 251–259, 2020. [2](#)
- [24] Apurwa Yadav, Aarshil Patel, and Manan Shah. A comprehensive review on resolving ambiguities in natural language processing. *AI Open*, 2:85–92, 2021. [2](#)
- [25] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. [6](#)
- [26] Yiming Zhang, ZeMing Gong, and Angel X Chang. Multi3drefer: Grounding text description to multiple 3d objects. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15225–15236, 2023. [2](#)

- [27] Duo Zheng, Shijia Huang, and Liwei Wang. Video-3d llm: Learning position-aware video representation for 3d scene understanding. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 8995–9006, 2025. [2](#), [7](#)
- [28] Hongyan Zhi, Peihao Chen, Junyan Li, Shuailei Ma, Xinyu Sun, Tianhang Xiang, Yinjie Lei, Mingkui Tan, and Chuang Gan. Lscenellm: Enhancing large 3d scene understanding using adaptive visual preferences. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 3761–3771, 2025. [7](#)
- [29] Chenming Zhu, Tai Wang, Wenwei Zhang, Jiangmiao Pang, and Xihui Liu. Llava-3d: A simple yet effective pathway to empowering lmms with 3d capabilities. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4295–4305, 2025. [2](#), [7](#)
- [30] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2911–2921, 2023. [2](#)