ObjVariantEnsemble: Advancing Point Cloud LLM Evaluation in Challenging Scenes with Subtly Distinguished Objects

Qihang Cao[1,2], **Huangxun Chen**[2*]

[1]Shanghai Jiao Tong University, Shanghai, China
[2]Hong Kong University of Science and Technology (Guangzhou), Guangzhou, China
john_cao@sjtu.edu.cn, huangxunchen@hkust-gz.edu.cn

Abstract

3D scene understanding is an important task, and there has been a recent surge of research interest in aligning 3D representations of point clouds with text to empower embodied AI. However, due to the lack of comprehensive 3D benchmarks, the capabilities of 3D models in real-world scenes, particularly those that are challenging with subtly distinguished objects, remain insufficiently investigated. To facilitate a more thorough evaluation of 3D models' capabilities, we propose a scheme, ObjVariantEnsemble, to systematically introduce more scenes with specified object classes, colors, shapes, quantities, and spatial relationships to meet model evaluation needs. More importantly, we intentionally construct scenes with similar objects to a certain degree and design an LLM-VLM-cooperated annotator to capture key distinctions as annotations. The resultant benchmark can better challenge 3D models, reveal their shortcomings in understanding, and potentially aid in the further development of 3D models.

Project Page — https://ove-benchmark.github.io/OVE/

1 Introduction

Building machine perception that can understand our 3D world has been an attractive pursuit in recent years. Prior works (Chang et al. 2015; Uy et al. 2019; Yu et al. 2022; Sun et al. 2022) have focused on learning 3D representations from point clouds, making significant progress in object classification. Due to the success of large language models (LLMs)(Touvron et al. 2023; Liu et al. 2024a), interest has expanded beyond traditional object classification tasks. Recent works, such as PointLLM(Xu et al. 2023) and 3D-LLM (Hong et al. 2023), aim to align the latent representations of textual descriptions with 3D point clouds, allowing machine perception systems to interpret and interact with the physical world more effectively through text-based instructions. For instance, as illustrated in Figure 1, if a system can accurately identify a target in a scene based on text descriptions, it could significantly enhance the intelligence of robotics applications, enabling tasks such as completing household chores through verbal instructions (Li et al. 2023; Ge et al. 2024) or improving human-robot collaboration (Xiao et al. 2022; Team et al. 2023; Huang et al. 2023c).

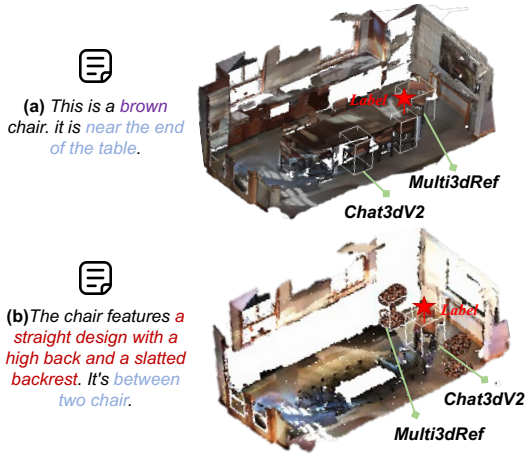


Figure 1: Comparison of 3D grounding benchmarks in challenging scenes: (a) one scene in ScanNet/ScanRef where the text is insufficient to accurately locate a chair. (b) one scene in ObjVariantEnsemble where one model accurately identifies targets with sufficient descriptions.

However, the path towards the aforementioned vision of 3D scene understanding is currently hindered by the lack of comprehensive 3D benchmarks. Existing scene-level benchmarks are primarily built on ScanNet (Dai et al. 2017) that consists of 1600+ scans of real scenarios. Subsequent works, including ScanRef (Chen, Chang, and Nießner 2020), Multi3DRef (Zhang, Gong, and Chang 2023), ScanQA (Azuma et al. 2022), *etc.* continue to refine and enhance the text annotations to ScanNet, aiming to facilitate 3D model evaluation and development.

Despite many efforts, 3D benchmarks still fall short of keeping pace with the needs of 3D model development.

(i) Small Scale. Compared to VLM works (Radford et al. 2021; Li et al. 2022; Liu et al. 2024a), which require large-scale text-image pairs, 3D benchmarks, especially those from real-world scans, are significantly smaller in scale.

(ii) Insufficient Annotation. Beyond data scale, the annotation granularity is more crucial in determining whether

*The work was done during Qihang Cao's research internship at HKUST(GZ). Corresponding author: Huangxun Chen.

3D Benchmark	#Obj	s-t pairs (wd/wod)	Anno (#Obj)
ModelNet	12k	-	Cl
ShapeNet	51k	-	
ScanObjectNN	15k	-	
OmniObject	6k	-	
ScanRefer	-	39k/12k	Cl+s/co/l (1)
Multi3DRef	-	33k/29k	Cl+s/co/l (0/1/2..)
OVE(Ours)	-	**75k/12k**	Cl+s/co/l (1 wd)

Table 1: Benchmark Comparison: "s-t pairs (wd/wod)" represent scene-text pairs with and without distractors. "Anno(#Obj)" refers to the types of annotation information available for the target object. "Cl," "s," "co," and "l" stand for Class, shape, color, and location, respectively. Numbers in parentheses indicate the number of targets the annotation can locate within the scene. Compared to other benchmarks, our dataset offers more challenging scene-text pairs.

we can objectively evaluate 3D model strengths and weaknesses for potential improvement. Figure 1(a) demonstrates a scene and its associated annotation from ScanRef. Though the predictions of two 3D models, Multi3DRef and Chat-3D-v2 (Huang et al. 2023a) do not align with ground truth label, it is hard to say the models are incapable, since the text itself contains certain ambiguities and can refer to multiple objects in the scene.

(iii) Lack of customizable challenge levels. Current 3D benchmarks are based on limited scene layouts and in-scene object combinations and placements. While they may include certain challenging scenes, they are not customizable, which can potentially lead to overfitting. It is better to have diverse scene data with customizable difficult level to facilitate sufficient evaluation of model capabilities.

Motivated by above gaps and persistent needs for more comprehensive 3D datasets, **we propose to synergize object-level and scene-level 3D point cloud resources to construct scenes with greater variety**. The community has developed comprehensive object-level 3D resources, *e.g.*, ModelNet (Sun et al. 2022), ShapeNet (Chang et al. 2015), ScanObjectNN (Uy et al. 2019), and OmniObject (Wu et al. 2023), among others as shown in Table 1. These object-level dataset generally offer broader object classes and rich variants within a class compared to scene-level ones, as detailed our project page. This makes them highly promising as a candidate pool to ensemble new scenes, especially challenging ones featuring subtly distinguished objects. For instance, Figure 1(b) showcases an ensembled scene with 3 chairs, one from ScanRef and the others retrieved from OmniObject. Besides scene construction, **we further integrate LLMs and 2D Vision-Language Models (VLMs) together to build an automated and fine-grained annotation pipeline**. Specifically, we instruct the LLM to guide the VLM in focusing on the differences between target and

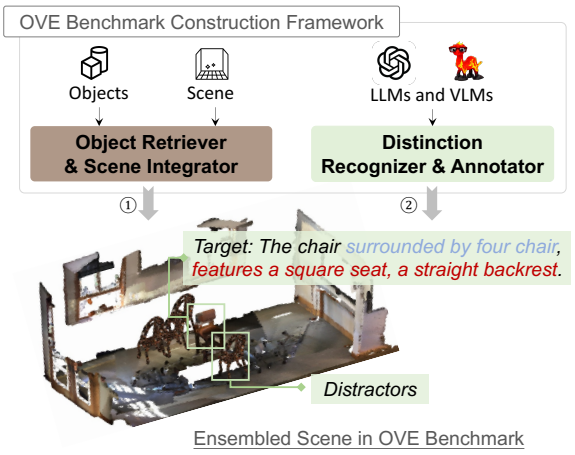


Figure 2: OVE Benchmark Construction Overview.

distractors, summarizing their distinctions in terms of multiple aspects (color/shape/location/...) as the annotation, *e.g.*, the text in Figure 1(b).

Our technical framework, **O**bj**V**ariant**E**nsemble (OVE) is shown in Figure 2. We will elaborate the technical details in §3. In summary, we make the following contributions:

- We designed an effective 3D scene construction and annotation framework to significantly expand the 3D dataset, resulting in 75k newly created scenes and associated fine-grained annotations, as summarized in Table 1.
- We developed a systematic and flexible method to introduce subtly distinguished objects adjacent to the target, along with an automated scheme to capture their key differences. This enriches the semantic complexity of the benchmark, enabling a more in-depth evaluation of 3D model.
- We evaluate state-of-the-art 3D understanding models on OVE benchmark and provide a fine-grained analysis to reveal the limitations of existing models in pure spatial relationship reasoning without visual features like shape, guiding potential directions for model improvement.

2 Related Works

In this paper, we focus on 3D understanding with point cloud data. We briefly discuss related works as follows.

3D Point Cloud Benchmarks Existing 3D point cloud datasets fall into two categories: object-level and scene-level. Object-level datasets (Chang et al. 2015; Uy et al. 2019; Sun et al. 2022; Wu et al. 2023) feature individual objects in various classes and styles, along with class-level text, to facilitate the evaluation of downstream tasks such as classification and object partial segmentation. Scene-level datasets are primarily built on ScanNet, which mainly covers indoor scenes scanned from the real world. An important subsequent work, ScanRefer, annotates objects with natural language descriptions to support the evaluation of 3D grounding tasks. Additionally, ReferIt3D (Achlioptas et al. 2020) has introduced two datasets: Sr3D, with textual annotations generated based on predefined templates, and Nr3D, with human-annotated descriptions for more

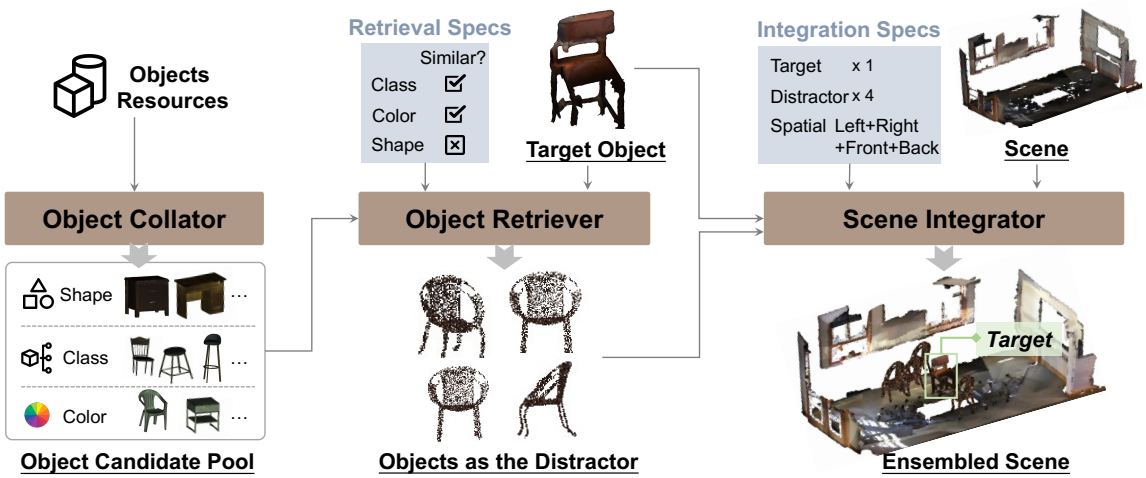


Figure 3: ObjVariantEnsemble Scene Data Generation Framework. (For clearer illustration here, we only plot the target and distractors in the scene, without other background objects.)

fine-grained understanding. Multi3DRef generalizes ScanRef from single-target grounding to various target grounding, *i.e.*, 0, 1, 2, and so on. However, these benchmarks are inherently limited by the available scenes in ScanNet, which are fixed and limited. Our work aims to break this constraint by incorporating additional object resources to assemble more scenes, particularly challenging ones. It is worth mentioning that while Multi3DRef handles scenarios where one text corresponds to varying numbers of objects, our approach focuses on scenarios where one text corresponds to a single object with multiple distractors.

Point Cloud LLM Owing to the great success of LLMs, there has been increasing interest in aligning various modalities, including point clouds with textual data, to apply the common-sense knowledge learned by LLMs to multi-modal understanding (Han et al. 2023). Technically, this demands a robust point cloud feature extractor, which has driven many prior works on 3D representation learning (Yu et al. 2022; Huang et al. 2023b; Zeng et al. 2023; Hong et al. 2023). It is noteworthy that many of these works leverage relatively mature VLMs driven by large-scale image-text pairs to help learn representations of 3D point clouds. Recently, the release of large-scale point cloud datasets (Wu et al. 2023; Deitke et al. 2023) has enabled scalable pretraining (Xue et al. 2023; Liu et al. 2024b; Zhang et al. 2023), significantly advancing the capabilities of point cloud encoders.

However, when dealing with scene-level point clouds, object-level encoders often struggle to manage complex spatial relationships. 3D-LLM attempts to address this by projecting 3D features into 2D to integrate them into LLMs and incorporating positional embeddings and learnable position tokens. Meanwhile, Chat-3D-v2 (Huang et al. 2023a) segments scenes into object-level point clouds and integrates the features and positional information obtained from object-level 3D encoders. Additionally, Any2Point (Tang et al. 2024) has proposed positional encoding that merges spatial features across 3D, 2D, and 1D, mitigating alignment errors caused by spatial relationships. Despite these design efforts, existing models still have limited pure spatial relationship reasoning over point cloud data, as evidenced by our evaluation in § 4.

3 Benchmark Data Construction

In this section, we will provide an overview of the OVE framework and then illustrate the technical details.

3.1 OVE Overview

As shown in Figure 2, OVE consists of two main modules:
Object Retriever & Scene Integrator: This module sorts available object-level 3D resources by their features, including class, color and shape, and then assembles multiple objects with varying levels of similarity into a specific scene background to construct a new scene.
Distinction Recognizer & Annotator: Most object-level datasets only have class labels, as indicated in Table 1, which are insufficient as effective annotations in the context of our constructed scene. Therefore, this module leverages the capabilities of advanced LLMs and VLMs to extract the key characteristics that distinguish the target object from others and summarize these as annotations.

In the following, we illustrate these two modules in detail.

3.2 Object Retriever & Scene Integrator

Figure 3 shows the detailed procedure for assembling a new scene by fully leveraging object-level and scene-level 3D resources. The construction process involves three steps: object collator, object retriever and scene integrator.

Object Collator As shown in Figure 3, this step aims to prepare an object candidates pool for the subsequent retrieval stage. To achieve this, we sort the object-level resources based on class, color, and shape. Specifically, we retain the object class from its original source dataset. We then represent color using the standard 3-tuple RGB values and

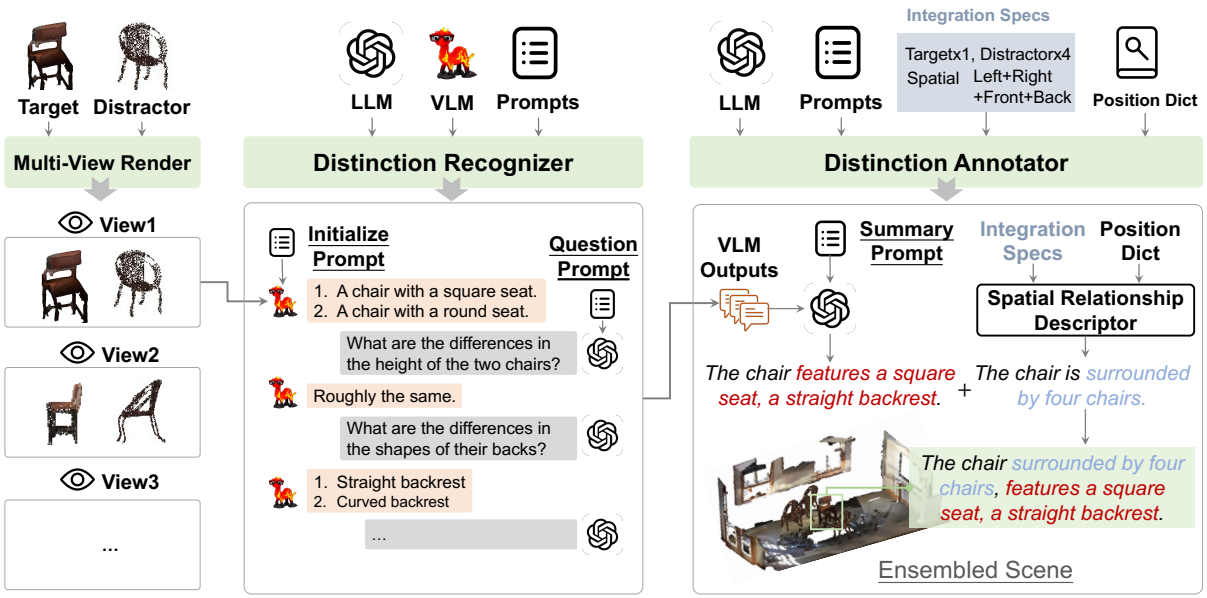


Figure 4: Process for Capturing Annotations with Key Distinguishing Information. We render multi-view images and use LLAVA(Liu et al. 2024a) to extract differences from various perspectives. A LLM is then employed to generate questions based on previous Q&A interactions. Finally, we use LLM to summarize the key differences from all descriptions.

use the L2-norm to quantify color similarity. Shape is harder to quantify, so to reduce potential bias, we currently focus on standard shapes, mainly including cuboids, L-shapes and spheres that are prevalent in our candidates to facilitate subsequent scene construction with similar-shaped objects.

Object Retriever This step aims to retrieve objects based on retrieval specifications. A retrieval specification consists of: i) a target object, where the retrieved objects are expected to preserve certain similarities with the target one, and ii) similarity dimension, where the distinction could occur in class, color, shape, or a combination of these factors. The example in Figure 3 shows a case of finding an object with the same class and color but a different shape compared to the target, a brown chair. The object retriever searches over the object candidate pool and identifies another brown chair with a different shape to serve as the distractor. When choosing distractors, we prioritize real-scanned objects (OmniObject /ScanObjectNN). If no suitable objects found, we then consider CAD data (ModelNet/ShapeNet).

Scene Integrator This step involves seamlessly integrating the retrieved distractors and the target object into a real-world scanned background extracted from the scene-level dataset. As shown in Figure 3, the integration specifications determine the number of distractors, which adjusts the challenge level of the assembled scene for model understanding. With more distractors, it generally becomes more difficult for the model to locate the correct target object.

Additionally, we specify the spatial relationship between the target and distractors. To better cover various possible object placements, we use six spatial primitives: `left`, `right`, `front`, `back`, `up` and `down`. Building upon these primitives and their combinations, we define a total of 13 spatial predicates (detailed in our project page), including terms such as `lower left`, `between`, `surrounding` for scenarios with multiple distractors.

It is worth mentioning that we carefully ensure the coherence of the ensembled scene. First, the target object is segmented from the scene-level dataset, and we use its original scene background as the base for constructing the new scene. Second, for the retrieved distractor objects, we properly resample, rescale, and reorient them to ensure coherence with the target and the scene, effectively mitigating scale mismatches between their original datasets and the target's ones. Third, after enforcing spatial relationship between distractor and target object, we calculate their bounding boxes to only exclude overlapping background objects to ensure scene reasonability. We manually screen assembled scenes and have not identified noticeably odd aspects.

The output of Object Retriever&Scene Integrator is newly ensembled scenes, each with a target object and a few distractor objects that share certain similarities with the target, as illustrated in the lower right of Figure 3.

3.3 Distinction Recognizer & Annotator

3D scenes alone cannot serve to evaluate the model without associated annotations. Figure 4 shows the detailed procedure to derive the key distinctions between the target object and the distractors to obtain an accurate annotation. This process involves two main steps: distinction recognizer and distinction annotator.

Distinction Recognizer This step aims to capture as many distinctions as possible, even subtle ones, between the target and its distractors. We employ two methods to achieve this goal: i) multi-view recognizer. To avoid missing critical dis-

Algorithm 1: Target Annotation Process

for each (tgt, distr) in pairs **do**
  for each rnd in rounds **do**
    for each v in views **do**
      tgt_desc.append(v : **LLaVA**(tgt_img))
      distr_desc.append(v : **LLaVA**(distr_img))
      img $\leftarrow$ **Concat**(tgt_img, distr_img)
      cap $\leftarrow$ **IterCap**(img, iter_rounds)
      cap_all.append(v : cap)
    end for
    sum $\leftarrow$ **GPT**(cap_all, SUM_P_2)
    tgt_sum $\leftarrow$ **GPT**(tgt_desc, SUM_P_2)
    distr_sum $\leftarrow$ **GPT**(distr_desc, SUM_P_2)
  end for
  desc $\leftarrow$ **GPT**(tgt_sum, distr_sum, sum, SUM_P_3)
end for

tinctions, we render the target-distractor pair from multiple perspectives and then let VLMs recognize the differences. ii) iterative difference capturing: we design an iterative QA process between VLM and LLM to enhance the comprehensiveness of distinction recognition. Specifically, after initializing the VLM to recognize differences, we prompt a LLM to continuously ask VLM about new potential distinction dimensions, forcing the VLM to capture more differences, as illustrated in Figure 4. This QA process would be repeated multiple rounds (6-7 in our case) to ensure sufficient and high-quality distinctions are captured. The detailed algorithm is provided in Algorithm 1, and the involved prompts are detailed in our project page.

Distinction Annotator All answers from VLM above are then compiled and fed into a summarization LLM to distill the most essential distinction information, as shown in Figure 4. It is worth mentioning that during the distinction recognition stage, VLMs are primarily used to identify distinctions in terms of detailed shape and color. For a complete annotation, we further enhance the spatial location information based on the integration specifications used for scene construction. With this, we can further describe the target's location in the scene and combine this with others to create a fine-grained annotation, as illustrated in Figure 4.

OVE ensures high-quality annotations through: (i) Basic attribute annotations (e.g., class, color, position) are guaranteed during scene construction. (ii) LLM-VLM collaboration helps capture high-quality visual difference annotations. Multi-round QA is used to reduce hallucinations. In each round, the LLM prompts the VLM to focus on a single attribute (e.g., height). The LLM asks the same question multiple times and discards overly verbose answers. The summary prompt explicitly instructs the LLM to exclude unspecified aspects, such as texture. (iii) We continuously sample annotations for manual verification to ensure quality.

3.4 OVE Benchmark Summary

In a nutshell, we construct comprehensive scenes with various challenge levels and distinction dimensions. The resultant benchmark is summarized in Figure 5. We support four different distinction types between the target object and distractors: `location`, `location+shape`, `location+color`, and `location+class`. Compared to existing scene-level 3D datasets, our work substantially expands the semantic richness of 3D benchmarks. The newly constructed scenes, with customizable challenge levels and fine-grained distinction annotations, can be used to better evaluate and develop 3D models. Moreover, our highly customizable pipeline can be seamlessly extended to construct more tasks during scene creation, including 3D object counting, captioning, QA, *etc.*.

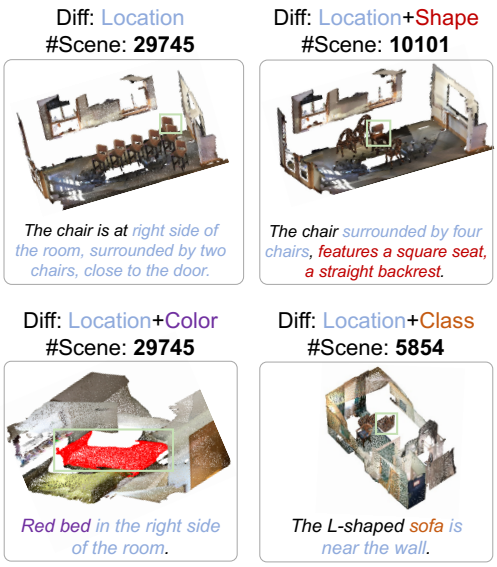


Figure 5: OVE Benchmark Summary

4 Benchmark on ObjVariantEnsemble

In this section, we evaluate the performance of state-of-the-art models on our benchmark.

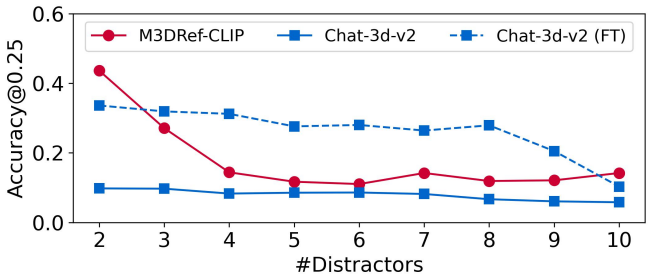
4.1 Evaluation Setting

Evaluation Task: We adopt 3D grounding as our evaluation task, because it is a fundamental perception task to support higher-level tasks like reasoning (Hong et al. 2023) and planning (Bai et al. 2024). Specifically, we feed the point cloud scene S_p and its associated textual description T from the OVE benchmark into a model $\mathcal{M}$. The model is then required to predict the target object's bounding box B_t in the scene. Formally, this can be described as follows:

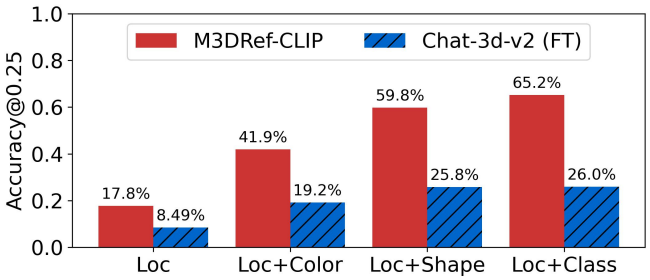
$$B_t = \mathcal{M}(\mathcal{S}_p, T) \tag{1}$$

Evaluation Metric: We adopt Acc@0.25 and Acc@0.5 as evaluation metrics, *i.e.*, prediction accuracy when the intersection over union (IoU) between the predicted and actual bounding boxes reaches 0.25 and 0.5, respectively.

Evaluation Models: We chose to evaluate 3D understanding models that rely solely on 3D scene information and 1D textual information to complete 3D grounding tasks. Chat-3D-v2 represents the state-of-the-art performance in



(a) Performance(ACC@0.25) of 3D grounding models on OVE across different numbers of distractors, considering only location.



(b) Performance(ACC@0.25) of 3D grounding models on OVE across various distinction types.

Figure 6: Experiments conducted on the OVE benchmark.

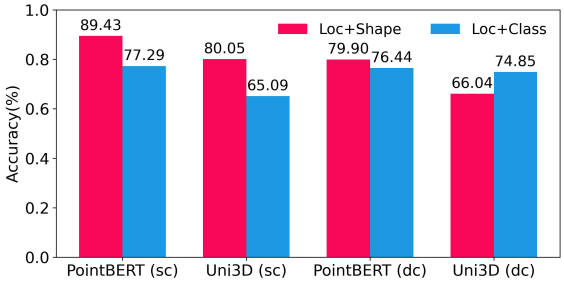


Figure 7: Performance(mIoU$_I$(%)) of 3D representation learning models on segmentation tasks using data resampled by OVE framework.

3D grounding, while Multi3DRef takes the initiative in addressing multi-object grounding. Both models utilize segmentation as an auxiliary task to extract scene features and then employ a fusion module to align 3D features with 1D text. Chat-3D-V2 uses Mask3D (Schult et al. 2023) for segmentation and Uni3D (Zhang et al. 2023) for encoding to realize 3D object identification and localization. We also incorporated a fine-tuned version of Chat-3D-v2 for additional evaluation using Vicuna1.5 (Zheng et al. 2024) and LORA (Hu et al. 2021). The training was conducted using a one-stage joint method, with a batch size of 32. Multi3DRef employs PointGroup (Jiang et al. 2020) as its detector and a CLIP-based (Radford et al. 2021) encoder to realize effective multi-object localization in complex scenes. We selected a configured version of M3dRefCLIP that utilizes only the `use_color` and `use_normal` settings. All experiments were performed on 4 NVIDIA A6000 cards.

4.2 Evaluation Results

Impact of #Distractors Here, we aim to evaluate whether existing 3D understanding models can maintain their performance, *i.e.*, predict the correct object bounding box, across various challenge levels, specifically with different numbers of distractors. To do this, we selected objects from the validation set of ScanRefer as the target and assembled scenes with the target and varying numbers of distractors, ranging from 2 to 10, to observe the model's performance. The evaluation results are shown in Figure 6a. It is noted that as the number of distractors increases, the grounding performance of the evaluated models declines to some extent. M3DRef-CLIP achieves the average performance reported in ScanRefer (Chen, Chang, and Nießner 2020) with 2 distractors but shows a decline as the number of distracting objects increases. The original Chat-3D-v2 performs rather poorly on the OVE benchmark, significantly less than reported in ScanRefer, potentially due to its overfitting to ScanRefer dataset. However, the fine-tuned version of Chat-3D-v2, using a portion of our dataset for fine-tuning, demonstrates enhanced performance. Compared to ScanRefer, the OVE benchmark supports a more fine-grained model evaluation.

Impact of Distinction Type We further investigate which aspects of existing 3D models' capabilities have the most weaknesses. We used four distinction types with a well-balanced distribution of distractor numbers, as shown in Figure 5 to conduct a more fine-grained evaluation on M3DRef-CLIP and the fine-tuned version of Chat-3D-v2. The evaluation results are presented in Figure 6b. The results indicate that the current models' ability to align textual information with pure location information is far inferior to their ability to correspond to visual features like shape, color, and class. Even though Chat-3D-v2 includes an additional encoding scheme for location information, its performance does not surpass that of Multi3DRef in challenging scenarios. This suggests that the spatial reasoning capabilities of current 3D understanding models are still ineffective.

Key Takeaways Through the lens of OVE, we reexamine the grounding capability of state-of-the-art 3D understanding models. We highlight how different types of information help distinguish multiple similar objects, revealing the models' strengths and weaknesses. From Figure 6b, we observe that shape and class contribute significantly more than color or location, with location playing the least role in object grounding. This raises our natural questions about the effectiveness of model designs related to position embedding. We may need more effort to better include spatial information in 3D representations.

4.3 Reexamining 3D Representation Learning

The performance gain from class/shape information shown in Figure 6b suggests a hypothesis that the 3D encoders un-

Figure 8: Visualizations of Segmentation Results on ShapeNet(Chang et al. 2015) Resampled: row (a) represents Uni3d, row (b) represents PointBERT, and row (c) represents the ground truth labels.

derlying 3D models could be robust, potentially owing to large-scale pre-training on object-level point clouds.

To test this hypothesis, we constructed scenes with multiple objects using the same concatenation method as in OVE scene construction framework. We created two cases, as illustrated in Figure 8: i) Loc+Shape: objects in the scene belong to the same class but differ in shape and location; ii) Loc+Class: objects differ in class and location but share similar shapes. The evaluation task is object partial segmentation, *i.e.*, correctly segmenting parts of objects. The evaluation metric is $\text{mIoU}_\text{I}(\%)$, the mean IoU across all instances. 3D grounding evaluates model capacity for text-visual alignment (§4.2), while 3D segmentation assesses visual modal understanding (§4.3). We evaluate both tasks to study how shape and class differences affect model performance.

We selected Uni3D (Zhang et al. 2023) and PointBERT (Yu et al. 2022) for evaluation, as they represent state-of-the-art performance. Considering the data adaptation issue, where models may require fine-tuning to perform well on new tasks, we added extra fully connected layers to fine-tune existing models on the tasks we constructed. We used two types of data for fine-tuning: 'sc' refers to scenes containing objects from the same class, while 'dc' refers to scenes with objects from different classes.

The experiment results are shown in Figure 7. Compared to their performance on single objects, which is 85.6% for PointBERT and 78.2% for Uni3D, we observe that the presence of distractors slightly hinders object partial segmentation. Figure 8 visualizes some segmentation results, highlighting some errors these models make when predicting the IoU of targets. For example, when locating tables that are distracted by skateboards, both PointBERT and Uni3D incorrectly identify the skateboard as part of the bottom of the table. This suggests that while these 3D encoders can distinguish between similar objects, there is still room for improvement for more complex scenarios.

Comparing the performance across different datasets ('sc' v.s. 'dc'), we found that models trained on scenes containing objects from the same class performed better than those fine-tuned on data with different classes. This insight suggests that incorporating more distractor cases in the training data may be beneficial for enhancing 3D grounding capabilities. More details on segmentation results across different categories and shapes can be found in our project page.

5 Conclusion and Future Work

OVE advances point cloud LLM evaluation. It incorporates objects with controlled variations in properties (e.g., class, color, shape) and spatial relationships into real-world scanned scenes, enabling scene-level challenge customization. Besides, we develop an LLM-VLM-cooperated annotator to obtain fine-grained annotations for 3D grounding. Previously, evaluating which aspects a 3D model relies on for object differentiation was challenging; with OVE, such fine-grained evaluations are now more accessible. Our evaluation shows existing 3D models are limited in pure spatial reasoning when visual features (*e.g.*, shape) are absent, which offer insights for improving 3D models, *e.g.*, rethink position encoding effectiveness.

In the future, we plan to apply OVE to a broader range of scenes, including synthetic ones (Jia et al. 2024; Yang et al. 2024), and expand our spatial predicate set to capture more spatial relationships. Besides, most object-level datasets only retain point information and lack mesh data, which means OVE cannot render 2D images with rich texture details, unlike real-scanned scenes in ScanNet. This prevents us from objectively evaluating 3D models that heavily rely on 2D features (Chen et al. 2024). We plan to incorporate generative models to generate mesh data from point information to address this issue.

Acknowledgments

We sincerely thank the anonymous reviewers for their valuable comments and suggestions, which have been instrumental in improving and refining this paper. We are also deeply grateful to our friends and families for their continuous encouragement throughout this research.

This work is supported by the Guangdong Provincial Key Lab of Integrated Communication, Sensing and Computation for Ubiquitous Internet of Things (No.2023B1212010007).

References

Achlioptas, P.; Abdelreheem, A.; Xia, F.; Elhoseiny, M.; and Guibas, L. 2020. Referit3d: Neural listeners for fine-grained 3d object identification in real-world scenes. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*, 422–440. Springer.

Azuma, D.; Miyanishi, T.; Kurita, S.; and Kawanabe, M. 2022. Scanqa: 3d question answering for spatial scene understanding. In *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 19129–19139.

Bai, F.; Du, Y.; Huang, T.; Meng, M. Q.-H.; and Zhao, B. 2024. M3d: Advancing 3d medical image analysis with multi-modal large language models. *arXiv preprint arXiv:2404.00578*.

Chang, A. X.; Funkhouser, T.; Guibas, L.; Hanrahan, P.; Huang, Q.; Li, Z.; Savarese, S.; Savva, M.; Song, S.; Su, H.; et al. 2015. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*.

Chen, D. Z.; Chang, A. X.; and Nießner, M. 2020. Scanrefer: 3d object localization in rgb-d scans using natural language. In *European conference on computer vision*, 202–221. Springer.

Chen, S.; Chen, X.; Zhang, C.; Li, M.; Yu, G.; Fei, H.; Zhu, H.; Fan, J.; and Chen, T. 2024. 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*, 26428–26438.

Dai, A.; Chang, A. X.; Savva, M.; Halber, M.; Funkhouser, T.; and Nießner, M. 2017. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 5828–5839.

Deitke, M.; Schwenk, D.; Salvador, J.; Weihs, L.; Michel, O.; VanderBilt, E.; Schmidt, L.; Ehsani, K.; Kembhavi, A.; and Farhadi, A. 2023. Objaverse: A universe of annotated 3d objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 13142–13153.

Ge, Y.; Tang, Y.; Xu, J.; Gokmen, C.; Li, C.; Ai, W.; Martinez, B. J.; Aydin, A.; Anvari, M.; Chakravarthy, A. K.; et al. 2024. BEHAVIOR Vision Suite: Customizable Dataset Generation via Simulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 22401–22412.

Han, J.; Zhang, R.; Shao, W.; Gao, P.; Xu, P.; Xiao, H.; Zhang, K.; Liu, C.; Wen, S.; Guo, Z.; et al. 2023. Imagebind-llm: Multi-modality instruction tuning. *arXiv preprint arXiv:2309.03905*.

Hong, Y.; Zhen, H.; Chen, P.; Zheng, S.; Du, Y.; Chen, Z.; and Gan, C. 2023. 3d-llm: Injecting the 3d world into large language models. *Advances in Neural Information Processing Systems*, 36: 20482–20494.

Hu, E. J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; and Chen, W. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Huang, H.; Wang, Z.; Huang, R.; Liu, L.; Cheng, X.; Zhao, Y.; Jin, T.; and Zhao, Z. 2023a. Chat-3D v2: Bridging 3D Scene and Large Language Models with Object Identifiers. *arXiv preprint arXiv:2312.08168*.

Huang, T.; Dong, B.; Yang, Y.; Huang, X.; Lau, R. W.; Ouyang, W.; and Zuo, W. 2023b. Clip2point: Transfer clip to point cloud classification with image-depth pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 22157–22167.

Huang, W.; Wang, C.; Zhang, R.; Li, Y.; Wu, J.; and Fei-Fei, L. 2023c. Voxposer: Composable 3d value maps for robotic manipulation with language models. *arXiv preprint arXiv:2307.05973*.

Jia, B.; Chen, Y.; Yu, H.; Wang, Y.; Niu, X.; Liu, T.; Li, Q.; and Huang, S. 2024. Sceneverse: Scaling 3d vision-language learning for grounded scene understanding. In *European Conference on Computer Vision*, 289–310. Springer.

Jiang, L.; Zhao, H.; Shi, S.; Liu, S.; Fu, C.-W.; and Jia, J. 2020. Pointgroup: Dual-set point grouping for 3d instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and Pattern recognition*, 4867–4876.

Li, C.; Zhang, R.; Wong, J.; Gokmen, C.; Srivastava, S.; Martín-Martín, R.; Wang, C.; Levine, G.; Lingelbach, M.; Sun, J.; et al. 2023. Behavior-1k: A benchmark for embodied ai with 1,000 everyday activities and realistic simulation. In *Conference on Robot Learning*, 80–93. PMLR.

Li, J.; Li, D.; Xiong, C.; and Hoi, S. 2022. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International conference on machine learning*, 12888–12900. PMLR.

Liu, H.; Li, C.; Wu, Q.; and Lee, Y. J. 2024a. Visual instruction tuning. *Advances in neural information processing systems*, 36.

Liu, M.; Shi, R.; Kuang, K.; Zhu, Y.; Li, X.; Han, S.; Cai, H.; Porikli, F.; and Su, H. 2024b. Openshape: Scaling up 3d shape representation towards open-world understanding. *Advances in Neural Information Processing Systems*, 36.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, 8748–8763. PMLR.

Schult, J.; Engelmann, F.; Hermans, A.; Litany, O.; Tang, S.; and Leibe, B. 2023. Mask3d: Mask transformer for

3d semantic instance segmentation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 8216–8223. IEEE.

Sun, J.; Zhang, Q.; Kailkhura, B.; Yu, Z.; Xiao, C.; and Mao, Z. M. 2022. Benchmarking robustness of 3d point cloud recognition against common corruptions. *arXiv preprint arXiv:2201.12296*.

Tang, Y.; Liu, J.; Wang, D.; Wang, Z.; Zhang, S.; Zhao, B.; and Li, X. 2024. Any2Point: Empowering Any-modality Large Models for Efficient 3D Understanding. *arXiv preprint arXiv:2404.07989*.

Team, A. A.; Bauer, J.; Baumli, K.; Baveja, S.; Behbahani, F.; Bhoopchand, A.; Bradley-Schmieg, N.; Chang, M.; Clay, N.; Collister, A.; et al. 2023. Human-timescale adaptation in an open-ended task space. *arXiv preprint arXiv:2301.07608*.

Touvron, H.; Martin, L.; Stone, K.; Albert, P.; Almahairi, A.; Babaei, Y.; Bashlykov, N.; Batra, S.; Bhargava, P.; Bhosale, S.; et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Uy, M. A.; Pham, Q.-H.; Hua, B.-S.; Nguyen, T.; and Yeung, S.-K. 2019. Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data. In *Proceedings of the IEEE/CVF international conference on computer vision*, 1588–1597.

Wu, T.; Zhang, J.; Fu, X.; Wang, Y.; Ren, J.; Pan, L.; Wu, W.; Yang, L.; Wang, J.; Qian, C.; et al. 2023. Omniobject3d: Large-vocabulary 3d object dataset for realistic perception, reconstruction and generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 803–814.

Xiao, T.; Chan, H.; Sermanet, P.; Wahid, A.; Brohan, A.; Hausman, K.; Levine, S.; and Tompson, J. 2022. Robotic skill acquisition via instruction augmentation with vision-language models. *arXiv preprint arXiv:2211.11736*.

Xu, R.; Wang, X.; Wang, T.; Chen, Y.; Pang, J.; and Lin, D. 2023. Pointllm: Empowering large language models to understand point clouds. *arXiv preprint arXiv:2308.16911*.

Xue, L.; Yu, N.; Zhang, S.; Panagopoulou, A.; Li, J.; Martín-Martín, R.; Wu, J.; Xiong, C.; Xu, R.; Niebles, J. C.; et al. 2023. Ulip-2: Towards scalable multimodal pre-training for 3d understanding. *arXiv preprint arXiv:2305.08275*.

Yang, J.; Chen, X.; Madaan, N.; Iyengar, M.; Qian, S.; Fouhey, D. F.; and Chai, J. 2024. 3D-GRAND: A Million-Scale Dataset for 3D-LLMs with Better Grounding and Less Hallucination. *arXiv preprint arXiv:2406.05132*.

Yu, X.; Tang, L.; Rao, Y.; Huang, T.; Zhou, J.; and Lu, J. 2022. Point-bert: Pre-training 3d point cloud transformers with masked point modeling. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 19313–19322.

Zeng, Y.; Jiang, C.; Mao, J.; Han, J.; Ye, C.; Huang, Q.; Yeung, D.-Y.; Yang, Z.; Liang, X.; and Xu, H. 2023. Clip2: Contrastive language-image-point pretraining from real-world point cloud data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 15244–15253.

Zhang, B.; Yuan, J.; Shi, B.; Chen, T.; Li, Y.; and Qiao, Y. 2023. Uni3d: A unified baseline for multi-dataset 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9253–9262.

Zhang, Y.; Gong, Z.; and Chang, A. X. 2023. Multi3drefer: Grounding text description to multiple 3d objects. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 15225–15236.

Zheng, L.; Chiang, W.-L.; Sheng, Y.; Zhuang, S.; Wu, Z.; Zhuang, Y.; Lin, Z.; Li, Z.; Li, D.; Xing, E.; et al. 2024. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36.