ViGiL3D: A Linguistically Diverse Dataset for 3D Visual Grounding

Austin T. Wang[1] **ZeMing Gong**[1] **Angel X. Chang**[1,2]
Simon Fraser University[1] Alberta Machine Intelligence Institute (Amii)[2]
{atw7, zmgong, angelx}@sfu.ca
https://3dlg-hcvc.github.io/vigil3d/

Abstract

3D visual grounding (3DVG) involves localizing entities in a 3D scene referred to by natural language text. Such models are useful for embodied AI and scene retrieval applications, which involve searching for objects or patterns using natural language descriptions. While recent works have focused on LLM-based scaling of 3DVG datasets, these datasets do not capture the full range of potential prompts which could be specified in the English language. To ensure that we are scaling up and testing against a useful and representative set of prompts, we propose a framework for linguistically analyzing 3DVG prompts and introduce **Vi**sual **G**rounding with **Di**verse **L**anguage in **3D** (ViGiL3D), a diagnostic dataset for evaluating visual grounding methods against a diverse set of language patterns. We evaluate existing open-vocabulary 3DVG methods to demonstrate that these methods are not yet proficient in understanding and identifying the targets of more challenging, out-of-distribution prompts, toward real-world applications.

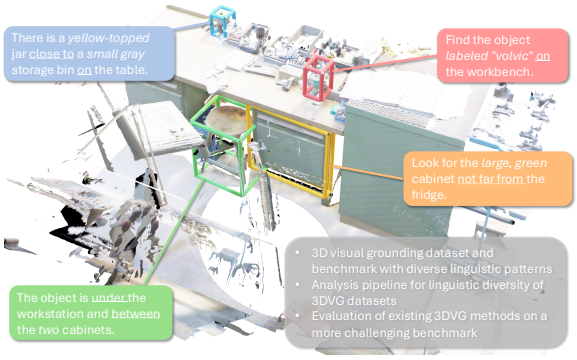

Figure 1: *Overview of ViGiL3D.* We propose a new dataset for visual grounding to better evaluate 3D visual grounding methods on the wide diversity of linguistic patterns possible to refer to objects in a scene. While existing datasets largely contain more homogenous and direct prompts, ViGiL3D includes coarse-grained object references, negation, reference resolution, and other phenomena in more varied sentence structures which allow us to more precisely and comprehensively measure state-of-the-art performance.

1 Introduction

Given a natural language description and a 3D scene, 3D visual grounding (3DVG) models localize the target entities in the scene described by the prompt. The ability to locate objects in 3D scenes based on language is useful for a variety of applications in computer graphics, robotics, and dialogue with virtual and augmented reality assistants. *Open-vocabulary* models, specifically, can generalize to novel object classes not seen during training. Such novel classes may appear in the text corpus used to pretrain the language model but are not part of the 3DVG training set. Building high performance visual grounding models enables downstream applications in embodied AI, such as robots identifying objects in an environment, and large-scale 3D scene retrieval, such as searching interior design databases for objects or attributes.

Compared to the success of recent 2D vision-language foundation models, progress in 3DVG has been slow due to the lack of large-scale 3D datasets paired with language. The most commonly used datasets—ScanRefer (Chen et al., 2020), Nr3D, and Sr3D (Achlioptas et al., 2020)—are based on just ∼700 scenes and ∼170K prompts combined. To alleviate the lack of data, recent work combined existing scene datasets and constructed LLM-based pipelines for scaling up grounding annotations (Yang et al., 2024a; Zhu et al., 2023; Jia et al., 2024). While these methods achieve reasonable performance on ScanRefer and similar datasets, these benchmarks do not fully evaluate how well models handle the diversity of language patterns and types of grounding prompts found in the English language. It is of value to ensure that we are scaling a complete set of grounding prompts and evaluating on a dataset that accurately measures understanding of language and vision and demon-

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 30453–30475

strates their viability in real-world situations.

We thus propose **Vi**sual **G**rounding with D**i**verse **L**anguage in **3D** (ViGiL3D), a diagnostic 3DVG dataset for evaluating visual grounding methods against a diverse range of language patterns, in order to determine 1) how well existing methods actually perform and 2) their specific strengths and weaknesses in grounding targets based on different linguistic phenomena. We develop an automated method to analyze 3DVG datasets, identifying a lack of linguistic diversity and several infrequently represented language patterns, such as negations. While there are methods to extract attributes and relationships from scenes and descriptions (Sun et al., 2024; Qian et al., 2024; Gu et al., 2024), ours also categorizes them and further identifies high-level linguistic patterns not captured in scene graphs. Given the limitations of existing datasets, we manually annotate ViGiL3D as a test dataset with an emphasis on prompt diversity and benchmark prior 3DVG models, demonstrating that the best models achieve an accuracy at least 20 points lower than their respective ScanRefer performances. We analyze the performance of each method on subgroups of language patterns to draw important insights about where the models are succeeding or falling short, demonstrating that further work is needed to bridge the gap of translating language understanding to 3D. Simply scaling data volume alone is insufficient for achieving good performance across the entire domain of 3DVG, but rather the right distribution must be captured. We believe that our work will contribute toward understanding the true performance and limits of the state-of-the-art and move us toward scaling the right types of prompts and building models which can tackle problems in real-world applications. In summary:

- we propose an automated pipeline for analyzing linguistic patterns in visual grounding descriptions and use it to investigate the limitations of existing 3DVG datasets;
- we construct a new dataset, ViGiL3D, for evaluating 3DVG methods against more challenging and diverse grounding descriptions than existing datasets; and
- we show that current 3DVG models perform worse on ViGiL3D than existing benchmarks, demonstrating the value of ViGiL3D for future 3DVG model development.

2 Related Work

3DVG Datasets. Existing datasets differ by scene type (indoor vs. outdoor and types of rooms), acquisition of 3D data (real-world vs. synthetic) and text (annotation process), scene size (single room vs. multiple rooms), and the scale and diversity of object or language annnotations. 3DVG research has primarily focused on providing language prompts for indoor scene datasets, with limited datasets for outdoor 3DVG (Miyanishi et al., 2023).

Early datasets obtained language prompts from crowdworkers (Chen et al., 2020; Achlioptas et al., 2020), or using simple templates (Achlioptas et al., 2020) on ScanNet (Dai et al., 2017), a dataset of real-world indoor rooms with semantically annotated objects. Later VG datasets used other sources of real-world 3D scene data (Kato et al., 2023), as well as synthetic datasets. Recent work has applied captioning models, LLMs, and scene graph generation methods to automatically generate prompts on aggregate scene datasets (Jia et al., 2024; Yang et al., 2024a; Zhu et al., 2023; Hong et al., 2023; Huang et al., 2024; Li et al., 2023; Lyu et al., 2024; Yang et al., 2024a; Zhang et al., 2024a).

Other efforts provided denser alignment (Abdelreheem et al., 2024), grounding without object names (Wu et al., 2023), and explored evaluation for grounding to multiple targets (Zhang et al., 2023), identifying regions or objects by function (Delitzas et al., 2024; He et al., 2024; Zhang et al., 2024b), or requiring reasoning to ground objects (Szymanska et al., 2024; Zhu et al., 2024a). However, both manual and LLM-scaled 3DVG datasets only capture part of the diverse language patterns in real-world applications, resulting in limitations in performance when methods are tested on out-of-distribution prompts. We propose a new diagnostic dataset covering different linguistic phenomena to study the performance of 3DVG models.

3DVG Methods. Traditional models fuse independently extracted visual and text features to identify the most likely points or regions corresponding to a target (Chen et al., 2020; Achlioptas et al., 2020; Abdelreheem et al., 2024; Wu et al., 2023; Jain et al., 2022; Cai et al., 2022; Chen et al., 2023; Jin et al., 2023; Chen et al., 2022). We focus on evaluating open-vocabulary methods, which generalize to a broader set of prompts and object classes than those used during the training or fine-tuning of grounding capabilities (Peng et al., 2023; Takmaz et al., 2023; Kerr et al., 2023; Yang et al.,

2024b; Yuan et al., 2024), enabled first by models such as CLIP (Radford et al., 2021) and later by large language models (LLMs) (Achiam et al., 2023). Recent work has focused on developing 3D foundation models for a wide variety of tasks on 3D scene data beyond visual grounding, including generic visual question-answering, captioning, segmentation, and similar tasks (Jia et al., 2024; Yang et al., 2024a; Zhu et al., 2023; Hong et al., 2023; Huang et al., 2024; Li et al., 2023; Lyu et al., 2024; He et al., 2024; Man et al., 2024; Zhu et al., 2024b). These methods are pretrained on LLM-scaled datasets to aid generalization. We show that these datasets lack crucial aspects of language, resulting in subpar performance of current foundation models on out-of-distribution 3DVG prompts.

3 Analysis of Prior Datasets

We annotate prompts from prior visual grounding datasets to identify strengths and shortcomings of each with respect to their linguistic properties, and to better understand the impact of the datasets on the methods trained and evaluated on them.

3.1 Language Patterns

We break down a grounding description into the target, anchors, attributes, and relationships. Each object is either a target (i.e. primary object of interest), or an anchor (i.e. an object or other reference region or agent used to help identify the location of the target). Attributes describe a target or anchor independent of the context, and relationships are used to compare two entities in the scene. To characterize these four aspects, we devise a set of 35 count-based or binary metrics for analyzing 1) **language diversity** (DIV), or coverage of a variety of different types and patterns; 2) **language resolution** (RES), the ability to link descriptors with their referents; and 3) **understanding attributes and relationships** (UAR), the ability to correspond each constraint to their appearance in the scene. In particular, these metrics track the reference types for targets and anchors, types and quantities of attributes and relationships, and language patterns such as negations. We also measure overall language diversity through token bigram frequency, similar to (Mensink et al., 2023). Each of the criteria is documented in Table 1.

To analyze each dataset, we devise an automated pipeline that assesses the occurrence of different language properties in each prompt (see Figure 2). For each prompt, we use GPT-4o (Achiam et al., 2023) to extract an augmented scene graph that captures the objects, attributes, and relationships in the description. We include the full prompts used for scene graph extraction in Appendix A. We also use SpaCy (Honnibal and Montani, 2017) to obtain a dependency parse of the tokens to measure diversity of bigrams in the dataset.

To validate our pipeline, we compare its output against 225 manually annotated prompts randomly sampled from all of the datasets, including at least 20 prompts from each dataset. Our pipeline achieves an average precision and recall across 24 measured binary metrics of 0.86 and 0.91, respectively. The median error and median absolute deviation for each of the counts of attributes and relationships pertaining to the target and anchor objects was 0.0 in all cases, and the mean absolute error was around 0.43. This shows the robustness of our pipeline for prompt analysis. Details for the manual validation are in Appendix A.

3.2 Datasets

We select commonly used 3DVG datasets, as well as several LLM-scaled datasets given their importance in developing 3DVG foundation models. We evaluate **ScanRefer** (Chen et al., 2020), **Nr3D/Sr3D+** (Achlioptas et al., 2020), and **Multi3DRefer** (Zhang et al., 2023). Building on the crowdsourced ScanRefer, Nr3D and Sr3D+ used manual and template-based methods, respectively, to generate prompts focused on discriminating objects of the same class. Multi3DRefer introduced zero- and multi-target grounding objectives. **Instruct3D** (He et al., 2024), another human-annotated dataset, extends the ideas of Multi3DRefer with an emphasis on reasoning about the function of objects. We further examine **ScanScribe** (Zhu et al., 2023), **3D-GRAND** (Yang et al., 2024a), and **SceneVerse** (Jia et al., 2024) as recent large-scale 3D datasets, largely leveraging template-based generation and GPT QA or rephrasing to generate prompts. Additional details for these datasets are in Appendix A.

3.3 Analysis

To compare datasets, we apply our automated pipeline to 1000 randomly sampled prompts from prior datasets and all 350 prompts from ViGiL3D. We show the most differentiating metrics in Table 4 and all others in Appendix A.5. We identify several shortcomings across many existing datasets.

Cat	Metric	Definition	Examples
Attribute Understanding (average number of attributes per prompts)			
UAR	Total attributes (Attr-All)	entire prompt	A *round* table sits in front of the *white* sofa.
RES	Target attributes (Attr-Tgt)	for describing the target	These are the *fancy, wooden* chairs in the room.
RES	Anchor attributes (Attr-Anc)	for describing the anchors	Where is the couch that is farthest from the *largest* bookshelf?
Relationship Understanding (average number of relationship per prompts)			
UAR	Total relationships (Rel-All)	entire prompt	Look for a personal computer *near* a simple office chair *with* arms.
RES	Target relationships (Rel-Tgt)	relationships which compare a target to an anchor	The box is *under* the table.
RES	Anchor relationships (Rel-Anc)	relationships which compare anchors to other anchors	Use the outlet under the window which is *second-to-the-right* in the room.
Target Reference (proportion of prompts)			
UAR	Generic References (Gen)	target is referred to by a generic name (e.g. "object", "thing")	The *object* is under the workstation and between the two cabinets.
UAR	Coarse-Grained References (CG)	target is referred to by a coarse category (e.g. "appliance", "device")	Locate the tallest *appliance* in the kitchen.
UAR	Fine-Grained References (FG)	target is referred to by its specific category	Identify the stainless steel *sink*.
RES	Coreferences (Cor)	coreferences is used to refer to objects	This is the rectangular whiteboard. *It* is left of the other one.
RES	Not First Noun Phrase (NFN)	target object is not the first noun phrase in the description	Facing the standalone whiteboard, grab the closest *chair* right behind you.
UAR	Negation (Neg)	negation is used	Find the food storage which does *not* have a green, rectangular object on top.
Anchor Type (proportion of prompts describing different types of anchors)			
UAR	Single-Object Anchors (Sing)	anchor references a single object	On the brown, wooden *table* is a small, rectangular projector.
UAR	Multi-Object Anchors (Mul)	anchor references multiple objects	The backpack is in between *two other backpacks*.
UAR	Non-Object Anchors (Non)	anchor references a room or region	In between the counter and table is a black trash can in the *corner*.
UAR	Agent-Based Anchors (Agt)	ences an agent or viewpoint	If *you* are sitting on the couch, this is the bag further to the right.
Diversity statistics			
DIV	Lexical bigrams (2lex)	proportion of unique bigrams of lexical tokens in descriptions	
DIV	Attribute Type	proportion of prompts with a specific attribute type describing an object	See Table 2
DIV	Relationship Type	proportion of prompts with a specific relationship type comparing objects or other entities in the scene	See Table 3

Table 1: **Dataset Criteria**. Summary of the metrics computed across each dataset, including the metric category (*Cat*), how the metric is computed, and valid example prompts from ViGiL3D.

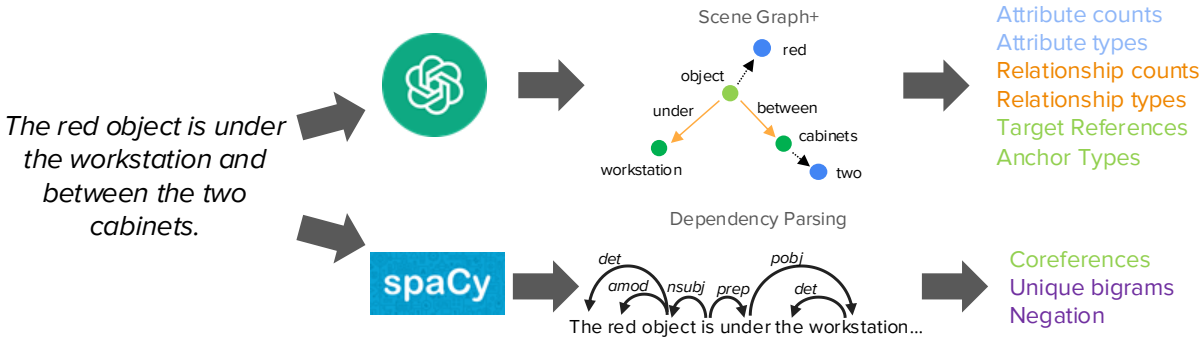

Figure 2: **Dataset Analysis Pipeline.** Using GPT-4o and SpaCy, we automatically parse each visual grounding prompt and compute aggregate statistics for a variety of linguistic patterns for each dataset. We use GPT-4o for parsing an augmented scene graph optimized for visual grounding descriptions, and SpaCy for dependency parsing.

Balanced number of descriptors. 3D-GRAND and ScanScribe employ an excessive number of attributes or relationships to describe the target object (e.g., "The chair is behind the desk, on the left side of the circular black table, to the right of the rectangular shelf, and to the left of the other chair."). While this ensures the target is uniquely defined, it also provides the models with more information than they would be given in most practical grounding prompts. 3D-GRAND and ScanScribe, for instance, have more than three times the number of attributes per prompt compared to the manually annotated datasets. Furthermore, with many attribute and relationship types, the model can over-rely on certain signals and ignore others. Sr3D and SceneVerse, on the other hand, have very few descriptors and thus may not have sufficient diversity to represent more complex prompts.

Overly specific target references. In most datasets, the target object is referenced by its explicit class. This makes it easier to identify the target when it is unique in the scene, allowing models primarily to attend to the object class name and ignore other signals. While Wu et al. (2023) evaluates this scenario, they only mask with the "object" keyword, whereas there are further gradations of detail that can be represented. Instruct3D (He et al., 2024) requires the model to identify objects based on described function and reasoning rather than semantic class. However, its scope is relatively narrow with respect to function, ignoring other potential attribute or relationship types.

Missing negations. Most prompts focus on the positive descriptors of the target object. However,

Type	Examples
Color	On top of the shelf is a *red* and *yellow* object.
Size	This is the *tallest* wooden furniture in the room.
Shape	Find the large, *rectangular* object with magnets on the front.
Number	This is the larger of the *two* toolboxes near the piano.
Material	When facing the radiator, the *metal* rail is directly on your left.
Function	Opposite the room from the left whiteboard, there is a device *for heating the room*.
Texture	These are all the long, *soft* places in the room.
Style	This is a *classy* queen-size bed, farthest from the door of the hotel room.
Text **Lab**el	Find the object *labeled "caution"*.
State	Find the *folded* chair closest to the door.

Table 2: Attribute types analyzed in each prompt.

Type	Examples
Near	Next to the table *closest* to the entrance, find all of the unfolded chairs.
Far	The table is the largest one *far* from the door.
Directional	A black bag is *in front of* a white trash can on the floor.
Vertical	*Under* the counter are two plastic bins. Find me the taller one.
Contain	*In* the center of the room is a hexagonal conference table for meetings.
Arrangement	In the *stack* of three boxes, this is the third one from the bottom.
Ordinal	In the row of chairs and tables against the wall, find the *third chair from the left*.
Comparison	This fancy rotating display is the one *nearest* to the orange carpet.

Table 3: Relationship types analyzed in each prompt.

properly eliminating objects based on any attributes that are *not* true of the target is also important, as a significant aspect of language. While ScanScribe does include some negation, it is often not useful toward identifying the target (e.g., "The description doesn't provide enough context to determine the location of the box with certainty.", commonly found in similar forms in many prompts).

Language diversity. Most of the datasets have low proportions of unique lexical bigrams, with ViGiL3D significantly outpacing other datasets at 0.52. We further observe that the target is the first noun phrase in more than 80% of prompts from all datasets except Instruct3D and ViGiL3D, further showing a lack of diversity in sentence structure in existing datasets.

Underrepresented attribute or relationship types. Some of the more challenging types are underrepresented, including numerical cues, object states, and complex spatial relationships involving long distances or multiple objects. Text labels are particularly difficult because of insufficient point cloud resolution to capture writing.

Given these shortcomings, our goal is to develop a diverse diagnostic benchmark which can adequately represent these different language phenomena for evaluation. While combining existing datasets is a viable strategy, as we saw some linguistic phenomena are absent across all datasets. Furthermore, the representations of most phenomena, while technically present, do not adequately allow us to understand how well models parse them for 3D understanding, due to strong correlations between phenomena which can confound analysis.

4 ViGiL3D

We present ViGiL3D, a new diagnostic 3DVG dataset that captures a diversity of language patterns to assess how well recent 3DVG methods perform and where they fall short.

We build our dataset on scenes from ScanNet (Dai et al., 2017) and ScanNet++ (Yeshwanth et al., 2023). We use ScanNet to assess the performance of prior works while controlling for the scene representation distribution and quality. We also annotate ScanNet++ to determine how well the model performance generalizes to new scenes and to leverage the higher quality 3D scenes, which may be critical for identifying smaller or more detailed targets.

To generate the prompts, annotators were asked to write grounding prompts for sampled objects in each scene given the RGB video stream and 3D point cloud. Targets were sampled from the ground truth annotations and could consist of zero, one, or multiple objects. Each prompt included a variety of language criteria and diverse but natural phrasing. Prompts were also designed to target a balanced level of specificity, in order to avoid both ambiguity and extraneous constraints on the target object. For zero-target prompts, annotators were instructed to craft them similarly to single-target prompts of objects in the scene but with modifications to make them unapplicable. This makes them more realistic and challenging for models compared to descriptions of absent object classes, as in Multi3DRefer (Zhang et al., 2023).

In total, we generate 350 prompts over 26 scenes for evaluation. Despite the small number, we are still able to demonstrate useful trends in the performance and believe that the principles of our dataset can be scaled up for future training and evaluation. More detailed statistics for ViGiL3D are provided in Appendix B.

We compare the linguistic diversity of ViGiL3D to previous datasets in Table 4. Although many

	Attributes						Relationships							Target Reference			Anchor Type			Language	
	All	Tgt	Anc	Num	Lab	State	All	Tgt	Anc	Far	Arr	Ord	Comp	Gen	CG	NFN	Mul	Non	Agt	Neg	2lex
ScanRefer	1.90	1.21	0.68	✓	✗	✗	2.33	1.89	0.44	✓	✓	✓	✗	✗	✗	✗	✓	✓✓	✓✓	✗	0.20
Nr3D	1.16	0.64	0.52	✓✓	✗	✗	2.22	1.63	0.59	✓	✓	✓	✓✓	✗	✗	✓✓	✓✓	✓✓	✓✓	✓	0.27
Sr3D+	0.05	0.02	0.03	✗	✗	✗	1.00	1.00	0.00	✓✓	✗	✗	✓✓	✗	✗	✗	✗	✗	✗	✗	0.02
Multi3DRefer	1.73	1.21	0.52	✓	✗	✓	2.02	1.63	0.39	✓	✓	✓	✗	✗	✗	✓✓	✓	✓✓	✓✓	✗	0.28
3D-GRAND	5.81	4.68	1.12	✓	✗	✗	2.81	2.71	0.10	✓	✓	✗	✓	✗	✗	✗	✓✓	✓✓	✓✓	✗	0.05
ScanScribe	6.04	1.15	4.89	✗	✗	✗	3.55	3.35	0.21	✗	✗	✗	✓✓	✗	✗	✗	✓✓	✓	✓✓	✗	0.10
SceneVerse	0.41	0.35	0.07	✗	✗	✗	1.33	1.30	0.03	✓	✗	✗	✗	✗	✗	✓	✓	✗	✗	✗	0.16
Instruct3D	1.40	1.30	0.09	✗	✗	✗	1.43	1.31	0.12	✗	✗	✗	✗	✓✓	✓✓	✓✓	✗	✓✓	✗	✗	0.27
ViGiL3D	1.62	1.09	0.53	✓✓	✓	✓	1.82	1.46	0.35	✓	✓	✓	✓✓	✓	✓	✓✓	✓✓	✓✓	✓✓	✓	0.45

Table 4: **Dataset Comparison**. We show here a comparison of the linguistic differences of prompts from prior visual grounding datasets. Counts of attributes and relationships are shown as average counts, while binary metrics are reported as not present (✗), some (✓), or a lot (✓✓), as thresholded at 5% and 20% of the dataset sample, respectively. While most datasets skew toward having particular types of prompts, our ViGiL3D dataset (in gray) has the best coverage and diversity for all types of prompts and includes some prompt types not well-represented in most other datasets.

Metric	ViGiL3D
Scene datasets	ScanNet, ScanNet++
# of prompts	350
# of scenes	35
Vocab size	942
# sentences per prompt	1.2
Average prompt length	14.1
# prompts with 0 targets	43
# prompts with 1 target	275
# prompts with multiple targets	32

Table 5: **Statistics of ViGiL3D.**

of the language patterns are captured by one of the prior datasets, none of them cover all of the patterns. For rare properties, such as text labels or negation, the datasets which do include them are fairly specialized or limited in other properties. Instruct3D, for example, is the only dataset with generic or coarse-grained target references but focuses primarily on grounding object functionality, at the exclusion of other attributes. Combining or subsampling all existing datasets for evaluation would still be suboptimal compared to ViGiL3D for evaluation. Combining datasets may induce correlations between patterns, such as generic target references always occurring in Instruct3D prompts with functionality-based attributes. Patterns in ViGiL3D, on the other hand, are annotated from the same distribution of prompts.

Additionally, ViGiL3D has more challenging prompts that require understanding of each phrase in the description and careful matching against objects in the scene. For instance, many of the descriptions in existing datasets are over-constrained, allowing models to ignore certain constraints (e.g. a model might only need to focus on the color of "a white chair to the left of the fridge"). Zero-target prompts are similar to valid descriptions of objects in the scene, rather than describing object classes not present in the scene. Evaluation of ViGiL3D under a reweighted pattern distribution similar to ScanRefer's can be found in Appendix C.2 to demonstrate the difficulty and value of ViGiL3D beyond its diversity.

5 Experiments

We apply recent 3DVG models on ViGiL3D and analyze their performance.

5.1 3DVG models

We focus on open-vocabulary methods that are designed to scale to new scene datasets and language descriptions not present in the 3DVG training data. We consider three groups of methods: those that use CLIP to obtain a language-aware 3D representation, zero-shot 3DVG with LLMs, and methods trained on 3DVG data. *CLIP aligned 3D representations*: We select **OpenScene** (Peng et al., 2023) that projects features directly to point clouds, and **LERF** (Kerr et al., 2023) that uses neural radiance fields. *Zero-shot with LLMs*: **ZSVG3D** (Yuan et al., 2024) and **LLM-Grounder** (Yang et al., 2024b) both use LLMs for reasoning combined with independent localization modules, the former through program synthesis and the latter directly in natural language. *Trained with 3DVG data*: **3D-VisTA** (Zhu et al., 2023) and **3D-GRAND** (Yang et al., 2024a) are both transformer architectures trained on LLM-scaled datasets, thus allowing us to study the impacts of large-scale datasets on downstream performance and generalization. Lastly, **PQ3D**

	ViGiL3D						ScanRefer	
	Acc/GT	Acc@25	Acc@50	F1/GT	F1@25	F1@50	Acc@25	Acc@50
OpenScene	2.1	1.7	1.3	2.1	1.7	1.2	13.2	6.5
LERF	2.5	2.1	2.1	2.5	2.1	2.1	4.8	0.9
ZSVG3D	18.9	8.5	5.6	12.2	6.7	5.8	36.4*	32.7*
LLM-Grounder	2.5	7.1	5.0	2.5	5.3	3.1	17.1	5.3
3D-VisTA	14.2	**15.8**	**13.3**	14.1	**15.7**	**13.2**	50.6	45.8
3D-GRAND	17.9	**15.8**	12.5	17.9	15.3	11.8	38.0	27.4
PQ3D	**26.2**	10.8	10.8	**26.8**	5.6	5.1	**57.0**	**51.2**

Table 6: Accuracy and F1 score (%) on ViGiL3D for ScanNet scenes. Each metric is computed using GT boxes or predicted boxes using IoU thresholds of 0.25 and 0.50, as is typical in the 3DVG literature. We compare against the overall ScanRefer validation set as a baseline, as reported by each method, to demonstrate the significant drop in performance on our prompts compared to existing datasets. *ZSVG3D ScanRefer results use GPT-3.5, as opposed to GPT-4o on ViGiL3D.

		Attributes			Relationships				Target Reference				Anchor Type				Lang
	Overall	Num	Lab	State	Far	Arr	Ord	Comp	Gen	CG	FG	NFN	Sing	Mul	Non	Agt	Neg
OpenScene	2.1	4.4	4.0	0.0	0.0	0.0	0.0	0.0	2.5	1.9	2.0	0.0	3.8	1.1	1.6	0.0	8.1
LERF	2.5	0.0	4.0	4.0	3.3	2.9	3.7	6.1	2.5	1.9	2.7	2.9	0.8	4.4	6.6	3.7	0.0
ZSVG3D	18.9	20.5	**12.0**	**28.0**	13.3	8.8	**19.2**	**25.0**	15.8	13.2	21.8	19.4	19.4	15.7	14.8	**23.1**	10.8
LLM-Grounder	2.5	2.2	0.0	0.0	3.3	5.7	11.1	6.1	0.0	0.0	4.1	7.2	1.5	5.5	4.9	11.1	2.7
3D-VisTA	14.2	6.7	0.0	8.0	10.0	5.7	7.4	8.2	0.0	13.2	18.4	8.8	13.7	12.1	15.0	19.2	8.1
3D-GRAND	17.9	13.3	4.0	12.0	13.3	8.6	14.8	18.4	7.5	13.2	22.4	17.4	18.3	15.4	19.7	18.5	**21.6**
PQ3D	**26.2**	**28.9**	8.0	**28.0**	**26.7**	**22.9**	7.4	24.5	**20.0**	**24.5**	**28.6**	**26.1**	**23.7**	**22.0**	**24.6**	18.5	13.5

Table 7: **Subgroup Analysis**. Breakdown of accuracy using ground truth boxes on ViGiL3D for ScanNet scenes across several subgroups of prompts. In general, we find that no one model is consistently better than another on any particular subgroup, likely suggesting that all of these models requires significant improvement to achieve any real understanding of the different linguistic phenomena and how they relate to 3D scenes.

	ScanNet		ScanNet++	
	Acc	F1	Acc	F1
ZSVG3D	**18.9**	12.2	**18.3**	**24.5**
3D-VisTA	14.2	14.1	11.8	11.1
3D-GRAND	17.9	**17.9**	9.2	9.2

Table 8: Accuracy and F1 score (%) on ViGiL3D for ScanNet++ scenes, using ground truth boxes.

(Zhu et al., 2024b) is a representative promptable query-based model trained on an aggregate of many existing 3DVG datasets. Details of the configurations of each method are in Appendix C.1.

We ran each method with both ground truth boxes and boxes predicted from Mask3D (Schult et al., 2023), following prior work. This enabled us to control for different methods of clustering points into objects and to analyze both the best-case performance as well as the realistic inference-level performance. Evaluating all methods required 22 GPU-hours on an RTX 4090 GPU.

To evaluate grounding performance, we report accuracy and F1 for ground truth and predicted boxes using Mask3D. For Mask3D predictions, we use IoU thresholds of 0.25 and 0.50 following prior work (Chen et al., 2020; Achlioptas et al., 2020; Zhang et al., 2023). For multi-target descriptions, we define accuracy as localizing all of the boxes within the specified IoU and the F1 score as the harmonic mean of precision and recall across objects.

5.2 Results

Consistently across different methods, we observe that performance on ViGiL3D is significantly lower than benchmarked grounding results for ScanRefer, even for the same scenes. Table 6 shows that even with the best model, PQ3D, the F1 is 24.4 points lower on our dataset, with similar trends for the other methods. This suggests that our prompts are likely out of distribution and harder than prior datasets. LLM-based methods and those trained on large datasets achieve significantly better per-

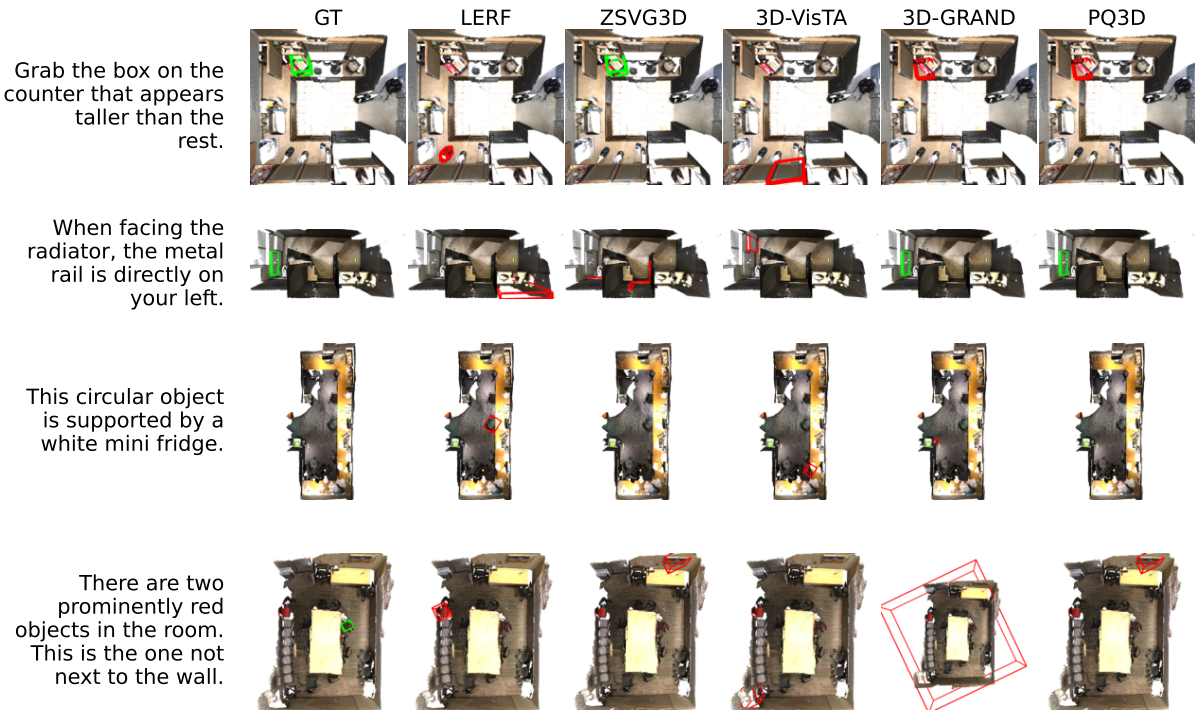

Figure 3: **Examples.** We show example predictions for each model on prompts with different linguistic patterns.

formance compared to CLIP-based methods, likely due to the inability of CLIP to parse complex language patterns (Yuksekgonul et al., 2023). Howevercorrelation with ScanRefer performance is loose, with ZSVG3D and 3D-GRAND both outperforming 3D-VisTA with GT boxes.

GT vs. predicted boxes. While better performance with ground truth information is expected, LLM-Grounder and 3D-VisTA actually score better with Mask3D predictions, and the performance of PQ3D drops precipitously compared to 3D-VisTA and 3D-GRAND, both of which achieve 5.0% improved Acc@25. It is likely that the ground truth information is not complete, and thus for some methods, the additional information afforded by Mask3D predictions can provide greater signal. Furthermore, there may be a significant degree of sensitivity of these methods to the objects presented to the models.

ScanNet vs. ScanNet++. We report results on ScanNet++ annotations in Table 8. ZSVG3D achieved as good or better performance on ScanNet++, in contrast with 3D-VisTA and 3D-GRAND. While the additional point cloud resolution may be useful for certain prompts, in practice the scenes are out of distribution and much larger, in terms of floorplan size, object counts, and semantic classes. While 3D-GRAND maxes out its input token limit, ZSVG3D scales at a slower rate to larger scenes and can still process all of the objects, albeit more slowly. Overall, grounding targets evidently becomes more difficult with many potential objects, necessitating future work to improve performance in challenging scenes.

Training dataset vs. performance. All of the methods trained on LLM-scaled datasets, notably 3D-VisTA and 3D-GRAND, significantly outperformed the CLIP-based methods, which have difficulty parsing complex language relations (Yuksekgonul et al., 2023). However, PQ3D, trained on an aggregate dataset of majorly manually annotated prompts, outperformed the other methods, suggesting that simply scaling the volume of 3D-language pair data may not be a guarantee of better performance. Future work should explore these differences further to identify the effects of data vs. architecture on performance.

Subgroup analysis. A detailed breakdown of key results is in Table 7. On ground truth boxes, PQ3D achieves the best performance in nearly all categories, with ZSVG3D scoring better on prompts with text labels, ordinal relationships, comparisons, and agent-based anchors. However, these trends are volatile depending on whether GT or predicted boxes are provided (see Table 14 for Mask3D subgroup performance). We highlight the following deficiencies in existing models:

Generic and coarse-grained target references. Most models have lower performance on prompts without a fine-grained object class. PQ3D achieved

comparable performance across all types of target references for both GT and Mask3D boxes.

Challenging attribute types. Text labels were challenging for all models, due to insufficient point cloud resolution and inability of most methods to ingest RGB-D image data directly. Aside from ZSVG3D and PQ3D, models performed worse on number and state attributes as well.

Challenging relationship types. "Far" and "arrangement" relationships were challenging for all models except PQ3D, while ordinal relationships were challenging for all models except ZSVG3D.

Negation. Prompts with negation to describe attributes or relationships in most cases led to worse performance compared to those without. 3D-GRAND uniquely achieved relatively strong performance on negative prompts, with both ground truth and Mask3D predictions.

The consistent drop in performance on ViGiL3D across all models compared to the existing benchmarks demonstrates a need for further improvement of models to achieve strong performance across a more diverse range of 3DVG prompts. While we do find that at least one model performs comparably on every subgroup compared to the control, no model consistently outperforms the others on all categories. Future work could bridge the gap in certain language phenomena for specific models or combine learnings across all models.

6 Conclusion

ViGiL3D demonstrates the need for incorporating greater linguistic diversity when training and evaluating 3D visual grounding. We provide a framework and automated pipeline for analyzing language patterns and ViGiL3D to evaluate the successful parsing of different language patterns. Our analysis shows the need for further establishment of a comprehensive benchmark prompt and a need for better 3DVG performance on several subgroups of prompts.

Scaling up ViGiL3D would be valuable in future development for large-scale training and evaluation, with an emphasis on more precise conditioning for language generation. While we have expanded the domain of visual grounding within language patterns, further work is also required to fully capture the complete space of potential prompts in the visual domain as well, toward ultimately utilizing learnings from 3DVG in general visual question-answering, embodied AI, and other applications.

7 Limitations

We provide detailed and high quality annotations for evaluating visual grounding on 3D scenes. However, our dataset is relatively small, which may affect the power of conclusions one can draw from it. Vision-language models (VLMs) are a natural solution for scaling, given their alleged flexibility in language and ability to parse visual inputs directly, but current state-of-the-art VLMs suffer from several key limitations.

Firstly, VLMs lack comprehensive 3D understanding. While they can identify basic spatial relationships, mapping objects across frames and extrapolating different viewpoints is challenging. Furthermore, identifying multi-object relationships, such as ordinal positions of chairs in a lecture hall, is likewise difficult. When generating grounding descriptions from images and extracted captions, VLMs did not consistently reconcile information correctly across views or captions. Furthermore, VLMs rarely generated correct viewpoint-dependent prompts, suggesting an inability to ground objects from particular perspectives without significant guidance.

Secondly, VLMs do not inherently generate diverse descriptions, relying on in-context examples to condition the distribution (Chang et al., 2024). Without a wide distribution of examples, VLM-generated descriptions tend to overfit to the provided examples. Even with stricter prompting and many in-context examples, VLMs in our trials did not necessarily capture the full distribution, for instance neglecting certain patterns or lacking variance in sentence length. Furthermore, prompting for increased diversity risked generating descriptions that are incorrect.

Lastly, they do not reliably reason about all of the objects in a scene—while they can caption single objects, grounding requires *contrasting* objects and thus identifying whether attributes are also true of all other objects in the scene. We found that grounding descriptions generated directly from scene graphs using VLMs were usually true of the targets but failed to consistently differentiate them from other objects in the scene, even with the full scene context. We believe that even at the modest scale of ViGiL3D, our evaluation, framework, and insights are still beneficial for the language and vision communities. Future work to address the limitations of VLMs will be key for scaling up the dataset in future work.

While our prompts are focused on describing the contents of 3D scenes, potentially offensive language could be present. We focused primarily on physical descriptions of objects and their everyday use cases, and we reviewed our prompts to ensure that no such language was used.

We focus on grounding objects in English. We acknowledge that different cultures may have different types of indoor scenes from those of ScanNet and ScanNet++ or different ways of describing objects. Future work should explore these differences further.

Acknowledgement

This work was funded in part by a CIFAR AI Chair and the NSERC Discovery Grant. We thank Yiming Zhang, Hou In Ivan Tam, Hou In Derek Pun, Xingguang Yan, and Karen Yeh for helpful discussions and feedback.

References

Ahmed Abdelreheem, Kyle Olszewski, Hsin-Ying Lee, Peter Wonka, and Panos Achlioptas. 2024. ScanEnts3D: Exploiting phrase-to-3D-object correspondences for improved visio-linguistic models in 3D scenes. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3524–3534.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*.

Panos Achlioptas, Ahmed Abdelreheem, Fei Xia, Mohamed Elhoseiny, and Leonidas Guibas. 2020. ReferIt3D: Neural listeners for fine-grained 3D object identification in real-world scenes. In *Proc. of European Conference on Computer Vision (ECCV)*, pages 422–440. Springer.

Daigang Cai, Lichen Zhao, Jing Zhang, Lu Sheng, and Dong Xu. 2022. 3DJCG: A unified framework for joint dense captioning and visual grounding on 3D point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16464–16473.

Matthew Chang, Gunjan Chhablani, Alexander Clegg, Mikael Dallaire Cote, Ruta Desai, Michal Hlavac, Vladimir Karashchuk, Jacob Krantz, Roozbeh Mottaghi, Priyam Parashar, et al. 2024. Partnr: A benchmark for planning and reasoning in embodied multi-agent tasks. *arXiv preprint arXiv:2411.00081*.

Dave Zhenyu Chen, Angel X Chang, and Matthias Nießner. 2020. ScanRefer: 3D object localization in RGB-D scans using natural language. *Proc. of European Conference on Computer Vision (ECCV)*.

Shizhe Chen, Pierre-Louis Guhur, Makarand Tapaswi, Cordelia Schmid, and Ivan Laptev. 2022. Language conditioned spatial relation reasoning for 3D object grounding. *Advances in neural information processing systems*, 35:20522–20535.

Zhenyu Chen, Ronghang Hu, Xinlei Chen, Matthias Nießner, and Angel X Chang. 2023. Unit3D: A unified transformer for 3D dense captioning and visual grounding. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 18109–18119.

Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. 2017. ScanNet: Richly-annotated 3D reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839.

Alexandros Delitzas, Ayca Takmaz, Federico Tombari, Robert Sumner, Marc Pollefeys, and Francis Engelmann. 2024. 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.

Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Knowledge Discovery and Data Mining (KDD)*, volume 96, pages 226–231.

Qiao Gu, Ali Kuwajerwala, Sacha Morin, Krishna Murthy Jatavallabhula, Bipasha Sen, Aditya Agarwal, Corban Rivera, William Paul, Kirsty Ellis, Rama Chellappa, et al. 2024. Conceptgraphs: Open-vocabulary 3d scene graphs for perception and planning. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5021–5028. IEEE.

Shuting He, Henghui Ding, Xudong Jiang, and Bihan Wen. 2024. SegPoint: Segment any point cloud via large language model. In *Proc. of European Conference on Computer Vision (ECCV)*.

Yining Hong, Haoyu Zhen, Peihao Chen, Shuhong Zheng, Yilun Du, Zhenfang Chen, and Chuang Gan. 2023. 3D-LLM: Injecting the 3D world into large language models. *Advances in Neural Information Processing Systems*, 36:20482–20494.

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing.

Jiangyong Huang, Silong Yong, Xiaojian Ma, Xiongkun Linghu, Puhao Li, Yan Wang, Qing Li, Song-Chun Zhu, Baoxiong Jia, and Siyuan Huang. 2024. An embodied generalist agent in 3D world. In *International Conference on Machine Learning (ICML)*.

Ayush Jain, Nikolaos Gkanatsios, Ishita Mediratta, and Katerina Fragkiadaki. 2022. Bottom up top down detection transformers for language grounding in images and point clouds. In *European Conference on Computer Vision*, pages 417–433. Springer.

Baoxiong Jia, Yixin Chen, Huangyue Yu, Yan Wang, Xuesong Niu, Tengyu Liu, Qing Li, and Siyuan Huang. 2024. SceneVerse: Scaling 3D vision-language learning for grounded scene understanding. In *Proc. of European Conference on Computer Vision (ECCV)*.

Zhao Jin, Munawar Hayat, Yuwei Yang, Yulan Guo, and Yinjie Lei. 2023. Context-aware alignment and mutual masking for 3D-language pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10984–10994.

Shunya Kato, Shuhei Kurita, Chenhui Chu, and Sadao Kurohashi. 2023. ArkitSceneRefer: Text-based localization of small objects in diverse real-world 3D indoor scenes. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 784–799.

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

Mingsheng Li, Xin Chen, Chi Zhang, Sijin Chen, Hongyuan Zhu, Fukun Yin, Gang Yu, and Tao Chen. 2023. M3DBench: Let's instruct large models with multi-modal 3D prompts. *arXiv preprint arXiv:2312.10763*.

Ruiyuan Lyu, Tai Wang, Jingli Lin, Shuai Yang, Xiaohan Mao, Yilun Chen, Runsen Xu, Haifeng Huang, Chenming Zhu, Dahua Lin, et al. 2024. MMScan: A multi-modal 3D scene dataset with hierarchical grounded language annotations. *arXiv preprint arXiv:2406.09401*.

Yunze Man, Shuhong Zheng, Zhipeng Bao, Martial Hebert, Liang-Yan Gui, and Yu-Xiong Wang. 2024. Lexicon3D: Probing visual encoding models for complex 3D scene understanding. *arXiv preprint arXiv:2409.03757*.

Thomas Mensink, Jasper Uijlings, Lluis Castrejon, Arushi Goel, Felipe Cadar, Howard Zhou, Fei Sha, André Araujo, and Vittorio Ferrari. 2023. Encyclopedic VQA: Visual questions about detailed properties of fine-grained categories. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3113–3124.

Taiki Miyanishi, Fumiya Kitamori, Shuhei Kurita, Jungdae Lee, Motoaki Kawanabe, and Nakamasa Inoue. 2023. CityRefer: geography-aware 3D visual grounding dataset on city-scale point cloud data. *arXiv preprint arXiv:2310.18773*.

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

Tianwen Qian, Jingjing Chen, Linhai Zhuo, Yang Jiao, and Yu-Gang Jiang. 2024. Nuscenes-qa: A multi-modal visual question answering benchmark for autonomous driving scenario. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 4542–4550.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR.

Jonas Schult, Francis Engelmann, Alexander Hermans, Or Litany, Siyu Tang, and Bastian Leibe. 2023. Mask3D: Mask transformer for 3D semantic instance segmentation. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 8216–8223. IEEE.

Penglei Sun, Yaoxian Song, Xiang Liu, Xiaofei Yang, Qiang Wang, Tiefeng Li, Yang Yang, and Xiaowen Chu. 2024. 3d question answering for city scene understanding. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 2156–2165.

Emilia Szymanska, Mihai Dusmanu, Jan-Willem Buurlage, Mahdi Rad, and Marc Pollefeys. 2024. Space3D-Bench: Spatial 3D question answering benchmark. *arXiv preprint arXiv:2408.16662*.

Ayça Takmaz, Elisabetta Fedele, Robert W Sumner, Marc Pollefeys, Federico Tombari, and Francis Engelmann. 2023. OpenMask3D: Open-vocabulary 3D instance segmentation. *Advances in neural information processing systems*.

Matthew Tancik, Ethan Weber, Evonne Ng, Ruilong Li, Brent Yi, Justin Kerr, Terrance Wang, Alexander Kristoffersen, Jake Austin, Kamyar Salahi, Abhik Ahuja, David McAllister, and Angjoo Kanazawa. 2023. NeRFStudio: A modular framework for neural radiance field development. In *ACM SIGGRAPH Conference Proceedings*.

Yanmin Wu, Xinhua Cheng, Renrui Zhang, Zesen Cheng, and Jian Zhang. 2023. EDA: Explicit text-decoupling and dense alignment for 3D visual grounding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19231–19242.

Jianing Yang, Xuweiyi Chen, Nikhil Madaan, Madhavan Iyengar, Shengyi Qian, David F Fouhey, and Joyce Chai. 2024a. 3D-GRAND: A million-scale dataset for 3D-LLMs with better grounding and less hallucination. *arXiv preprint arXiv:2406.05132*.

Jianing Yang, Xuweiyi Chen, Shengyi Qian, Nikhil Madaan, Madhavan Iyengar, David F Fouhey, and Joyce Chai. 2024b. LLM-grounder: Open-vocabulary 3D visual grounding with large language model as an agent. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7694–7701. IEEE.

Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. 2023. ScanNet++: A high-fidelity dataset of 3D indoor scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12–22.

Zhihao Yuan, Jinke Ren, Chun-Mei Feng, Hengshuang Zhao, Shuguang Cui, and Zhen Li. 2024. Visual programming for zero-shot open-vocabulary 3D visual grounding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20623–20633.

Mert Yuksekgonul, Federico Bianchi, Pratyusha Kalluri, Dan Jurafsky, and James Zou. 2023. When and why vision-language models behave like bags-of-words, and what to do about it? In *Proc. of International Conference on Learning Representations (ICLR)*.

Haochen Zhang, Nader Zantout, Pujith Kachana, Zongyuan Wu, Ji Zhang, and Wenshan Wang. 2024a. VLA-3D: A dataset for 3D semantic scene understanding and navigation. *arXiv preprint arXiv:2403.09631*.

Yiming Zhang, ZeMing Gong, and Angel X Chang. 2023. Multi3Drefer: Grounding text description to multiple 3D objects. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15225–15236.

Zhuofan Zhang, Ziyu Zhu, Pengxiang Li, Tengyu Liu, Xiaojian Ma, Yixin Chen, Baoxiong Jia, Siyuan Huang, and Qing Li. 2024b. Task-oriented sequential grounding in 3D scenes. *arXiv preprint arXiv:2408.04034*.

Chenming Zhu, Tai Wang, Wenwei Zhang, Kai Chen, and Xihui Liu. 2024a. ScanReason: Empowering 3D visual grounding with reasoning capabilities. In *Proc. of European Conference on Computer Vision (ECCV)*.

Ziyu Zhu, Xiaojian Ma, Yixin Chen, Zhidong Deng, Siyuan Huang, and Qing Li. 2023. 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.

Ziyu Zhu, Zhuofan Zhang, Xiaojian Ma, Xuesong Niu, Yixin Chen, Baoxiong Jia, Zhidong Deng, Siyuan Huang, and Qing Li. 2024b. Unifying 3D vision-language understanding via promptable queries. In *Proc. of European Conference on Computer Vision (ECCV)*.

Appendices

We provide additional details about the analysis of prior datasets (Appendix A, construction of ViGiL3D (Appendix B), implementation details for methods we compared (Appendix C.1), and additional experiment results (Appendix C.2).

A Dataset Analysis

In this work, we focused on analyzing 3DVG datasets in English. Table 5 provides further details about each of the dataset we analyzed, including the statistics on the text description (e.g. prompts), the annotation method, and the source scene datasets.

We document below the process for creating the language criteria used to evaluate datasets and model predictions as well as further details concerning the manual validation.

A.1 Criteria Selection

Semantic scene graphs, which can be used to represent the key objects in an image or scene and their attributes and relationships to one another, comprised the initial basis for criteria selection. While they are typically used to describe an entire scene, the concept also applies usefully to visual grounding, in which there is a special object or set of objects (*target*) to identify. Thus the high-level categories for characterization included 1) how objects, especially the target, were specified; 2) the types of attributes; 3) the types of relationships; and 4) the grammatical structure that could be used to translate a scene graph to natural language.

Given this framework, the specific criteria used to analyze existing 3DVG datasets and construct subgroups for analysis were selected based on qualitative observation of a sample of prompts from existing datasets. Based on observed similarities and notable gaps, we constructed a set of criteria (see Table 1) that could be quantitatively measured and thus reasonably executed by LLMs and natural language libraries.

While there may be other, more comprehensive taxonomies by which to characterize grounding prompts, we believe that our system is 1) sufficiently detailed to identify the categories of patterns which could affect a model's ability to ground objects and 2) useful for characterizing the vast majority of 3DVG prompts in existing datasets and usefully highlights language patterns which are still under-represented.

Dataset	# Prompts	$\|V\|$	Gen Method	Scene Datasets	Example
ScanRefer	52K	6,919	Human	ScanNet	There is a black counter top to the left of the fridge. It has a stainless steel sink on it.
Nr3D	42K	6,951	Human	ScanNet	While at the sink, it is the third option on the top.
Sr3D+	115K	200	Template	ScanNet	The whiteboard that is close to the couch
Multi3DRefer	62K	9,645	Human, LLM	ScanNet	The black frame houses the picture, and it hangs above the toilet.
3D-GRAND	6.2M	8,279	Template, LLM	3D-FRONT, Structured3D	This refrigerator is a muted silver, presenting a sleek and modern look with its brushed metal finish. The refrigerator is positioned close to one of the dining chairs, near to another dining chair, and far from the loveseat sofa.
ScanScribe	278K	2,881	Human, Template, LLM	ScanNet, 3RScan	The chair is behind the desk, on the left side of the circular black table, to the right of the rectangular shelf, and to the left of the other chair, bag, trash can, and backpack.
SceneVerse	2.5M	18,427	Template, LLM	ProcTHOR, Structured3D, ARKitScenes, HM3D, ScanNet, 3RScan, MultiScan	The couch is situated to the right of the computer tower.
Instruct3D	2.6K	1,787	Human	ScanNet++	Where are the window coverings used to control light and privacy? It is the one faces two doors and black chair.

Table 9: **Dataset Overview**. Overview of dataset sizes and example prompts from each prior dataset. We include the size of each dataset as represented by the number of 3DVG prompts and vocabulary size ($|V|$), as well as the underlying scene datasets. The sizes are calculated by counting distinct words across the dataset prompts used for visual grounding, which may be a subset of the total number of prompts provided in each dataset (particularly for 3D-GRAND, ScanScribe, and SceneVerse).

A.2 Analysis Pipeline

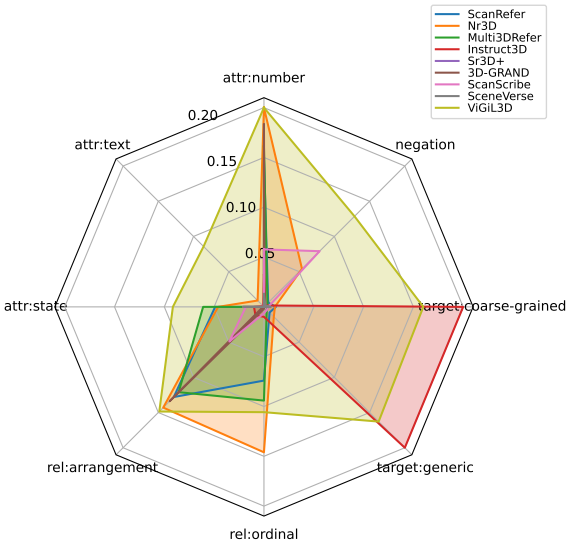

Figure 4: **Dataset Comparison.** We visualize the proportion of prompts representing different metrics up to 20% of the dataset size.

In our pipeline, we use `gpt-4o-2024-08-06` (Achiam et al., 2023) as our LLM to parse the grounding prompt into an augmented scene graph-like representation, as well as to extract certain linguistic properties of the prompt. We use the `en_core_web_md` spaCy pipeline (Honnibal and Montani, 2017) for token and PoS parsing, including identifying negation and computing the unique bigram frequency.

We include the three prompts used to automate the analysis of objects, relationships, and attributes in the prompts in Listings 1, 2, and 3, respectively. The script, including OpenAI API calls, evaluates a single prompt in around 9 seconds per iteration, thus requiring a total of 21 compute-hours to generate the full analysis across all datasets, at a cost of around 23 USD per 1000 prompts. The burden of computation was primarily on GPT-4o (Achiam et al., 2023), so only CPU resources were required here.

A.3 Threshold Selection

We use thresholds of 5% and 20% to measure whether a particular prompt characteristic was sufficiently reflected in the aggregate performance, while accounting for the fact that not every prompt should reflect every characteristic. While many language patterns can co-occur, too much co-occurrence would prevent us from being able to test a model's ability to parse particular patterns and allow models to over-attend to one phenomena at the exclusion of others. This is precisely what we observe in 3D-GRAND and ScanScribe, which use excessive attributes and relationships per prompt. Thus, there is a limit to the value of having a high proportion of prompts with a particular characteristic.

We found during ViGiL3D annotation that 20% was a natural threshold, given that increasing the proportion of one pattern tends to reduce the proportions of others. Because most papers still ultimately compare performance on aggregate statistics, we opt to use target percentages over absolute thresholds on prompt counts per language pattern.

```
"""
You are given a description of an object that someone is supposed to find in a scene. Similar to the Visual Genome dataset, we
      would like to identify the objects, attributes, and relationships in the following text:
"{}"

Please first return a list of the objects in the scene in a JSON format:
{{
  "success": boolean,
  "objects": [
    {{
      "id": id of object,
      "name": name of object as string,
    }}
  ],
  "target": list of ids of object,
  "target_reference_type": "generic", "categorical", or "fine-grained",
  "first_noun": boolean
}}

The object IDs should start with 0 and increment.

The name of the object should be lowercase (except for proper nouns) and sufficient to define the object class, if specified.
      Attributes should not be included in the class name. For instance, "a big, red apple" has the object name of "apple",
      and a "rectangular washing machine" has the object name of "washing machine". Parts of objects should be included as
      separate objects. For instance, "a chair with four legs" has objects "chair" and "legs".

If the target object is implied in the text but not explicitly named, such as in, "Where can I keep food cold?" the target
      object name should just be specified as "object".

The target ID should be the list of IDs of the objects that the user is supposed to find. For example, if the prompt is, "This
       is the toolbox on the shelf", the target ID should be the ID of the toolbox.

target_reference_type should be specified as the most specific term used to name the object:
* generic - object is referred to by a general term such as "object", "thing", or "item", or if the object is implied but not
      explicitly named.
* categorical - object is referred to by a category which is more specific than a generic reference but not specific to a
      particular object class. This includes references such as "appliance", "seat", "container", "display", "machine", or "
      device".
* fine-grained - object is referred to by a specific object class, from which it should be easy to infer what the object is
      and how it should be used.

If the prompt is not a description of an object in a scene, set "success" to False and ignore the rest of the output.
      Otherwise, set it to True.

If the target object is the first noun phrase mentioned in the description, set "first_noun" to True. Otherwise, set it to
      False.
"""
```

Listing 1: **Object Prompt for Analysis.** We feed this prompt first to GPT-4o to obtain information about the objects in the description and their respective types.

A.4 Manual Validation

To manually validate the pipeline, a random sample of 20 prompts were annotated by the authors from each of the prior datasets as well as 100 prompts from ViGiL3D (total of 225 prompts). The quantitative design of the criteria made it straightforward to annotate each count or binary flag, and a similar prompt of examples were used as a guideline for each metric. For each of the 24 binary metrics, we computed a precision, recall, and F1 score, and for each of the counts (e.g., number of attributes per prompt), we measured the deviation using the median error, median absolute deviation, and mean absolute error. While a couple metrics with lower representation, such as the style attribute, were harder to predict correctly by the model, most of the criteria were predicted correctly by the model with above 80% precision and recall. For most prompts, examples where the model predicted "incorrectly" were gray-area cases, suggesting that the model largely picked up on the right signals.

A.5 Additional Analysis

We include the full analysis of linguistic properties in tables 10 to 12. Table 10 includes all of the attribute types analyzed by the pipeline, and Table 11 includes all of the relationship types. Table 12 contains all other analyzed metrics, including those for target reference, anchor type, and diversity. We see that unlike other datasets, ViGiL3D includes all of the linguistic patterns, with the highest prevalence in most categories. ViGiL3D furthermore is more balanced across linguistic properties compared to existing datasets.

While most datasets have parseable grounding prompts, some are invalid or confusing. 3D-VisTA has the largest number of parsing failures, including 6% of prompts which could not be interpreted by the analysis pipeline. This is largely due to a lack of data cleaning, as a significant portion of the LLM-generated responses involve the LLM expressing inability to respond to the query (e.g., "I'm sorry, I don't have enough information to answer that question...").

	All	Tgt	Anc	Col	Siz	Sha	Num	Mat	Fun	Tex	Sty	Lab	Sta
ScanRefer	1.90	1.21	0.68	✓✓	✓	✓	✓	✓	✗	✗	✗	✗	✗
Nr3D	1.16	0.64	0.52	✓✓	✓	✗	✓✓	✗	✗	✗	✗	✗	✗
Sr3D+	0.05	0.02	0.03	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
Multi3DRefer	1.73	1.21	0.52	✓✓	✓	✓✓	✓	✓	✗	✗	✗	✗	✓
3D-GRAND	5.81	4.68	1.12	✓✓	✓	✓✓	✓	✓✓	✓✓	✓✓	✓✓	✗	✗
ScanScribe	6.04	1.15	4.89	✓✓	✗	✓✓	✗	✓	✗	✗	✗	✗	✗
SceneVerse	0.41	0.35	0.07	✗	✓	✗	✗	✗	✗	✓	✓	✗	✗
Instruct3D	1.40	1.30	0.09	✗	✗	✗	✗	✗	✗	✓✓	✗	✗	✗
ViGiL3D	1.62	1.09	0.53	✓✓	✓	✓	✓✓	✓	✓	✓	✓	✓	✓

Table 10: **Dataset Attributes Comparison**. We show here a comparison of visual grounding datasets by attributes. The descriptions of each attribute type are provided in Table 2. Binary metrics for attribute types are reported as not present (✗), some (✓), or a lot (✓✓), as thresholded at 5% and 20% of the dataset sample.

	All	Tgt	Anc	Near	Far	Dir	Ver	Cont	Arr	Ord	Comp
ScanRefer	2.33	1.89	0.44	✓✓	✓	✓✓	✓✓	✓✓	✓	✓	✗
Nr3D	2.22	1.63	0.59	✓✓	✓	✓✓	✓✓	✓	✓	✓	✓✓
Sr3D+	1.00	1.00	0.00	✓✓	✓✓	✗	✗	✗	✗	✗	✓✓
Multi3DRefer	2.02	1.63	0.39	✓✓	✓	✓✓	✓✓	✓✓	✓	✓	✗
3D-GRAND	2.81	2.71	0.10	✓✓	✓	✓✓	✓✓	✓✓	✓	✗	✓
ScanScribe	3.55	3.35	0.21	✓✓	✗	✓✓	✓✓	✓	✗	✗	✓✓
SceneVerse	1.33	1.30	0.03	✓✓	✓	✓✓	✓✓	✓	✗	✗	✗
Instruct3D	1.43	1.31	0.12	✓✓	✗	✗	✓✓	✓✓	✗	✗	✗
ViGiL3D	1.82	1.46	0.35	✓✓	✓	✓✓	✓✓	✓✓	✓	✓	✓✓

Table 11: **Dataset Relationships Comparison**. We show here a comparison of visual grounding datasets by relationships. The descriptions of each relationship type are provided in Table 3. Binary metrics for relationships types are reported as not present (✗), some (✓), or a lot (✓✓), as thresholded at 5% and 20% of the dataset sample.

Additionally, some attributes were reasonably identified by the pipeline as present in certain datasets despite not contributing meaningfully to grounding. For instance, SceneVerse has many prompts with embellishments which the pipeline interprets as "state" attributes (e.g., "The bag rests gracefully on the floor's surface."). However, while interesting in their own right and arguably not an error in parsing, it does not describe a useful state for grounding. Clear cases like these, when egregious, were manually corrected in postprocessing.

	Gen	CG	FG	NFN	Cor	Sing	Mul	Non	Agt	Neg	2lex
ScanRefer	✗	✗	✓✓	✗	✓✓	✓✓	✓	✓✓	✓✓	✗	0.20
Nr3D	✗	✗	✓✓	✓✓	✓	✓✓	✓✓	✓✓	✓✓	✓	0.27
Sr3D+	✗	✗	✓✓	✗	✗	✓✓	✗	✗	✗	✗	0.02
Multi3DRefer	✗	✗	✓✓	✓✓	✓	✓✓	✓	✓✓	✓✓	✗	0.28
3D-GRAND	✗	✗	✓✓	✗	✓✓	✓✓	✓✓	✓✓	✓✓	✗	0.05
ScanScribe	✗	✗	✓✓	✗	✓✓	✓✓	✓✓	✓	✓✓	✗	0.10
SceneVerse	✗	✗	✓✓	✓	✓	✓✓	✓	✗	✗	✗	0.16
Instruct3D	✓✓	✓✓	✓	✓✓	✗	✓✓	✗	✓✓	✗	✗	0.27
ViGiL3D	✓	✓	✓✓	✓✓	✓✓	✓✓	✓✓	✓✓	✓✓	✓	0.45

Table 12: **Additional Dataset Comparison**. We show here a comparison of visual grounding datasets by metrics for target reference, anchor type, and diversity. The descriptions of each metric are provided in Table 1. Binary metrics are reported as not present (✗), some (✓), or a lot (✓✓), as thresholded at 5% and 20% of the dataset sample.

```
"""
You are given a description of an object that someone is supposed to find in a scene. Your goal is to analyze the prompt for
      information about the relationships used to describe objects in the description:
"{}"
The parsed list of objects is as follows:
{}
Return the output in a JSON format according to the following format:
{{
  "relationships": [
    {{
        "name": name of relationship as string as it appears in the prompt,
        "subject_id": list of ids of objects which are the subject of the relationship,
        "recipient_id": list of ids of objects which is the recipient of the relationship
    }}
  ],
  "num_relationship_type": {{
    "near": {{
      "exists": True if relationship is found in prompt or False otherwise,
      "explanation": list of relationships identified, or empty if none
    }},
    "far": {{
      "exists": True if relationship is found in prompt or False otherwise,
      "explanation": list of relationships identified, or empty if none
    }},
    "viewpoint_dependent": {{
      "exists": True if relationship is found in prompt or False otherwise,
      "explanation": list of relationships identified, or empty if none
    }},
    "vertical": {{
      "exists": True if relationship is found in prompt or False otherwise,
      "explanation": list of relationships identified, or empty if none
    }},
    "contain": {{
      "exists": True if relationship is found in prompt or False otherwise,
      "explanation": list of relationships identified, or empty if none
    }},
    "arrangement": {{
      "exists": True if relationship is found in prompt or False otherwise,
      "explanation": list of relationships identified, or empty if none
    }},
    "ordinal": {{
      "exists": True if relationship is found in prompt or False otherwise,
      "explanation": list of relationships identified, or empty if none
    }},
    "comparison": {{
      "exists": True if relationship is found in prompt or False otherwise,
      "explanation": list of relationships identified, or empty if none
    }}
  }},
  "anchors": {{
    "single": True if at least one of the anchor objects is a single object otherwise False,
    "multiple": True if at least one of the anchor objects represents multiple objects otherwise False,
    "non_object": True if at least one of the anchor objects represents a region or room otherwise False
    "viewpoint": True if one of the relationships requires a specific viewpoint otherwise False
  }}
}}
A relationship compares an object(s) or region to another object(s) or region. Relationships should capture objects in the
      scene and not hypothetical objects. The name of the relationship should be the word or phrase used in the description to
       describe the relationship. If a noun in the description is a part of an object rather than a distinct object, then it
      should be counted as a part rather than an object. If a recipient is relative to the speaker of the description, use the
       ID of the "<speaker>" object.
Each relationship type is defined as follows:
near: Any relationship which describes one object in proximity of another. Examples include near, next to, nearby, adjacent,
      close to, proximate, amidst, among, covered, or contact relationships (against, leaning on, on, hanging on, supported by
      , attached to).
far: Any relationship which describes one object far away from another. Examples include far from, opposite, across from, and
      distant from.
viewpoint_dependent: Any relationship which can only be identified based on a canonical reference frame of the object or the
      viewpoint of the speaker. This includes left, right, in front of, facing, behind, or any cardinal direction.
vertical: Any relationship which describes one object above or below another. Examples include above, below, on top of, under,
       underneath, or vertical support relationships (e.g., an object on another).
contain: Any relationship which describes one object contained within another or some part that belongs to another. Examples
      include in, inside, within, with, has, or have.
arrangement: Any relationship which describes one object as part of an ordered arrangement. Examples include "between", "
      surrounded by", "row of", "column of", "stack of", or "pile of" other objects. You should exclude "amidst", "among", "
      nearby" or other non-structured relationships.
ordinal: Any relationship which describes the numerical position of an object in a spatial order or array. Examples include
      first, 2nd, middle, last. You should exclude cases of an object being the closest, leftmost, rightmost, or equivalent.
comparison: Any relationship which compares properties of different objects and requires identifying which one is more or less
      , or the most or least, of something. Examples include taller, tallest, shorter, greenest, closest, furthest, or same as
      .
In the explanation, each relationship should be given as a list of [subject, relationship, recipient].
Lastly, indicate the following:
1. If any of the subjects or recipients of a relationship, excluding the target, is a single object, set "single" to True
2. If any of the subjects or recipients of a relationship, excluding the target, represents multiple objects, set "multiple"
      to True. Examples include "the table is surrounded by six chairs", "the car is in between the shovel and the desk", "the
       book is the third one on the shelf", or "the chair is the one closest to the door".
3. If any of the subjects or recipients of a relationship is a region or room, set "non_object" to True. Examples include "the
       shelf in the center of the room", "the microwave in the kitchen", or "the books in the area around the couch".
4. If finding the target is dependent on a specific viewpoint in the scene, set "viewpoint" to True. Examples include "the
      leftmost wall" or "the window on your right."
"""
```

Listing 2: **Relationships Prompt for Analysis.** We then feed this prompt to GPT-4o, using the object information, to obtain information about the relationships between the inferred objects in the prompt.

```
"""
You are given a description of an object that someone is supposed to find in a scene. Your goal is to calculate statistics
      about the attributes used to describe objects in the description:
"{}"
The parsed list of objects is as follows:
{}
The parsed list of relationships is as follows:
{}
Return the output in a JSON format according to the following format:
{{
  "num_attribute_type": {{
    "color": {{
        "exists": True if attribute is found in prompt or False otherwise,
        "explanation": list of attributes identified, or empty if none
    }},
    "size": {{
        "exists": True if attribute is found in prompt or False otherwise,
        "explanation": list of attributes identified, or empty if none
    }},
    "shape": {{
        "exists": True if attribute is found in prompt or False otherwise,
        "explanation": list of attributes identified, or empty if none
    }},
    "number": {{
        "exists": True if attribute is found in prompt or False otherwise,
        "explanation": list of attributes identified, or empty if none
    }},
    "material": {{
        "exists": True if attribute is found in prompt or False otherwise,
        "explanation": list of attributes identified, or empty if none
    }},
    "texture": {{
        "exists": True if attribute is found in prompt or False otherwise,
        "explanation": list of attributes identified, or empty if none
    }},
    "function": {{
        "exists": True if attribute is found in prompt or False otherwise,
        "explanation": list of attributes identified, or empty if none
    }},
    "style": {{
        "exists": True if attribute is found in prompt or False otherwise,
        "explanation": list of attributes identified, or empty if none
    }},
    "text_label": {{
        "exists": True if attribute is found in prompt or False otherwise,
        "explanation": list of attributes identified, or empty if none
    }},
    "state": {{
        "exists": True if attribute is found in prompt or False otherwise,
        "explanation": list of attributes identified, or empty if none
    }}
  }},
  "attributes": [
    {{
      "object_id": id of object,
      "attributes": list of attributes
    }}
  ]
}}
Attributes are any descriptors that help distinguish an object from others. The name of an object does NOT count as an
      attribute.
Each attribute type is defined as follows:
color: Any attribute which describes the color properties of an object. Examples include red, blue, black, light, dark,
      monocolor, or colorful.
size: Any attribute which describes the size of an object. Examples include big, small, large, larger, tall, long, short, or
      medium. You should exclude cases where the height of an object is described to capture vertical position rather than
      size.
shape: Any attribute which describes the shape or form of an object. Examples include round, square, rectangular, or circular.
number: Any attribute which describes the quantity of an object. Examples include "two chairs". This does not include cases
      where the number is used to describe the relative order of the object, or cases where "one" is used as a pronoun to
      refer to the object.
material: Any attribute which describes what an object is made of. Examples include wood, metal, plastic, or glass. If the
      attribute describes the texture but not what the object is actually made of, e.g. metallic, then it should count as a
      texture attribute rather than a material.
texture: Any attribute which describes the texture of an object. Examples include smooth, rough, soft, metallic, or comfy.
function: Any attribute which describes what an object can be used for or the function it performs in a space. Examples
      include a chair for sitting or a lamp that makes the space warm and welcoming. The name of the object does not count as
      a function.
style: Any attribute which describes the style of an object or the effect of its presence in the space. Examples include
      modern, vintage, antique, futuristic, luxurious, or industrial, or describing its prominent or subtle presence in a room
      .
text_label: Any attribute which describes text that can be found on an object. Examples include "fragile" on a box or "exit"
      on a door.
state: Any attribute which describes the state of an object, which can be changed. Examples include "unopened" to describe a
      jar, "broken", or "drying" to describe clothes hanging on a rack.
You should also include a list of each of the attributes for each object in the scene.
"""
```

Listing 3: **Attributes Prompt for Analysis.** Finally, we feed this prompt to GPT-4o, substituting in the object and relationship information, to obtain information about the attributes in the prompt describing the inferred objects and their respective types. Including the relationship information helps the model to not double-count relationships as attributes.

B ViGiL3D Dataset

We provide further details here for the process to develop ViGiL3D.

B.1 Grounding Annotation

Prompts for ViGiL3D were annotated internally by the authors. The annotator demographic was primarily Asian and included a native English speaker. Scenes were sampled from ScanNet (Dai et al., 2017) and ScanNet++ (Yeshwanth et al., 2023). ScanNet has 1513 scenes with RGB-D video and point cloud representations, reconstructed from real indoor scenes. ScanNet++ has 460 reconstructed scenes with similar RGB-D streams as well as DSLR images and laser scan data. While having fewer scenes, ScanNet++ has higher quality reconstructions and larger scenes overall. in order, primarily from the validation splits of ScanNet and ScanNet++, in order to minimize the possibility of other models having trained on the scenes we used. Both datasets redact any potentially identifying information in the videos and scenes, and our text prompts focus only on describing the contents of the scenes themselves.

Annotators were provided a 3D point cloud view of each scene with access to a ground truth instance segmentation of the scene and an RGB video of an agent navigating the scene. For each prompt, they were instructed to select an object and then to craft a description identifying that object using a combination of the class name, attributes, or relationships to other entities in the scenes. In order to achieve linguistic diversity, annotators were instructed to use different sampled linguistic patterns for each prompt, in order to represent each phenomena in the dataset well. Each prompt was further annotated with metadata concerning the linguistic patterns present. Annotations were manually validated to ensure correctness of the target objects with respect to the descriptions and metadata.

B.2 Prompt Validation

We ran a study with human evaluators, presenting them with the 3D point cloud and corresponding RGB-D video of each scene and asking them to identify the target objects (0, 1, or multiple) of each grounding prompt. When presented with the same ground truth object segmentations as the models, we found that they achieved an overall accuracy of 84% on the ScanNet prompts, significantly exceeding model performance. The main challenge encountered was searching through large scenes and videos, especially given their low resolution. This demonstrates that the prompts are sufficiently solvable and that existing models are not yet attaining human performance.

C Evaluation

C.1 Implementation Details

We describe the details of reproducing each method below, toward faithfully representing each model while also ensuring a fair comparison between them:

OpenScene (Peng et al., 2023) is a simple contrastive learning-based approach which aligns 3D point features to projected 2D segmentation features in the CLIP space. We use the pretrained OpenSeg version of the model with `MinkUNet18A` for 3D encoding and evaluate using only the 3D features. To adapt OpenScene with provided boxes (from ground truth or Mask3D), we compute the mean cosine similarity of all embeddings for points contained within each box. We select the target object which has the highest mean similarity score.

LERF (Kerr et al., 2023) is a model which augments neural radiance fields by learning a CLIP feature for each point. We use the LERF model based on ViT-B/16, using the `Nerfstudio` package (Tancik et al., 2023). We optimize a LERF model on each scene using every 20th frame from the RGB-D videos, at a frame resolution of 320×240. We sample 10×6 points per scene for inference and compute the target object similarly to OpenScene. Note that as LERF requires optimizing the 3D representation for each scene, it is the most computationally expensive during inference.

ZSVG3D (Yuan et al., 2024) is an LLM-based method which leverages visual program synthesis for reasoning. We use GPT-4o (Achiam et al., 2023) as the LLM to bring the method closer in line with the other methods, which use GPT-4 or variants. Furthermore, we achieve better results than GPT-3.5 as used in the original implementation, with only a 16.2% accuracy on ground truth boxes with GPT-3.5 compared to 22.9% with GPT-4o. The LOC module applies CLIP (ViT-B/16) to the ground truth or predicted labels of each box to compute alignment against the object of interest.

LLM-Grounder (Yang et al., 2024b) is a zero-shot method using CLIP-based methods to detect objects and an LLM (GPT-4, as in the original implementation) to plan and reason to identify the

	# Parameters	Visual Encoder	Text Encoder	Training Strategy	Training Dataset	Inference Time
OpenScene	100M	MinkUNet18A	CLIP text encoder	CLIP aligned	ScanNet200	0.12
LERF	237M	CLIP-ViT	CLIP text encoder	CLIP aligned	ScanNet	23m
ZSVG3D	149M*	CLIP-ViT	GPT-4o	Zero-shot	N/A	3.81
LLM-Grounder	100M*	CLIP-ViT	GPT-4	Zero-shot	ScanNet	40.70
3D-VisTA	138M	PointNet++	BERT	3DVG data	ScanScribe	0.02
3D-GRAND	6.74B	Mask3D	Llama-2	3DVG data	3D-GRAND	2.31
PQ3D	248M	CLIP-ViT,PointNet++	CLIP text encoder	3DVG data	Aggregate	0.31

Table 13: **Method Comparison**. Overview of the 3D visual grounding methods. We present the visual and text encoder they use, along with their training strategies and datasets. PQ3D is trained on an aggregate dataset including ScanRefer, Nr3D, Sr3D, Multi3DRefer, and other segmentation, QA, and captioning datasets. Inference times are reported in seconds, unless otherwise specified, on ScanNet with a batch size of 1. *The number of parameters of GPT is not included, since it has not been publicly reported.

target objects based on attributes and anchors. We use OpenScene as the visual grounder in our experiments. We observe that the performance was significantly decreased when using ground truth or Mask3D boxes compared to the original method of clustering using DBSCAN (Ester et al., 1996). This is likely caused by the fact that the clustering is based on only those points whose OpenScene embeddings have a high cosine similarity with the text embeddings, as opposed to purely geometric clustering. Per the original implementation, the box dimensions provided to the LLM are based on the clustered points in the original method, causing the LLM to receive different box dimensions than those of the actual object. Interestingly, we find that modifying the implementation to serve the original box dimensions to GPT instead of using the cluster-extract ones yielded slightly worse performance.

3D-VisTA (Yang et al., 2024a) is pretrained on the ScanScribe dataset and uses a simplified dual-encoder transformer architecture to perform a variety of downstream 3D tasks. We use the pretrained checkpoint from the authors fine-tuned on ScanRefer to optimize performance for visual grounding on ScanNet-like scenes. In order to support larger scenes, the point cloud sequence length was extended from 80 to 250.

3D-GRAND (Yang et al., 2024a) is trained on the dataset of its namesake, representing 3D-LLM performance on a significantly larger dataset of synthetic scenes. We use the `merged_weights_grounded_obj_ref` model checkpoint based on Llama-2-7b. Due to the large number of Mask3D predictions and 2048 input sequence token limit, we truncate the input object tokens accordingly.

PQ3D (Zhu et al., 2024b) is a promptable query transformer-based model for 3D, unifying different 3D representations, modality prompts, and output forms through prompting. The model is pretrained on 8 different 3D datasets, including notably ScanRefer, Nr3D, Sr3D, and Multi3DRefer. We use the provided checkpoint for the unified 3D model, after the second stage of training. The pretrained model based on CLIP (ViT-L/14) and uses PointNet++ for point cloud encoding.

In general, all LLM-based methods were executed multiple times in cases where failures occurred for specific prompts during inference, largely due to syntactic errors in parsing the LLM output. ZSVG3D encountered the most errors due to unparseable outputs, such as calling functions in its generated programs which were not supported by the domain language, and we were unable to generate a valid output for around 2% of prompts after 5 attempts.

A summary of the models evaluated can be found in Table 13.

C.2 Additional Results

We report the subgroup analysis based on Mask3D boxes in Table 14. As expected, the models that have the best aggregate performance, 3D-VisTA and 3D-GRAND, with Mask3D predictions achieve the best performance across most subgroups, while ZSVG3D and PQ3D are largely outperformed in most categories except prompts with *state* attributes and *generic* target references, respectively.

We also provide additional qualitative examples of model performance on ViGiL3D with ScanNet and ScanNet++ scenes in Figures 6 and 7, respectively.

Statistical Analysis. We find that, despite the dataset size, many of the results we observe in our

		Attributes			Relationships				Target Reference				Anchor Type				Lang
	Overall	Num	Lab	State	Far	Arr	Ord	Comp	Gen	Cat	FG	NFN	Sing	Mul	Non	Agt	Neg
OpenScene	1.7	2.2	0.0	4.0	0.0	0.0	0.0	0.0	2.5	1.9	1.4	0.0	3.1	2.2	1.6	0.0	8.1
LERF	2.1	4.4	0.0	0.0	3.3	2.9	0.0	0.0	2.5	0.0	2.7	1.4	2.3	1.1	1.6	3.7	0.0
ZSVG3D	8.5	12.8	4.0	**12.5**	3.3	11.4	3.7	9.1	2.6	7.8	10.4	9.0	11.5	7.1	5.1	11.5	11.8
LLM-Grounder	7.1	8.9	4.0	8.0	6.7	8.6	3.7	4.1	7.5	3.8	8.2	11.6	3.8	6.6	11.5	3.7	5.4
3D-VisTA	**15.8**	13.3	4.0	12.0	13.3	8.6	3.7	16.3	7.5	**9.4**	20.4	14.5	16.0	**15.4**	**19.7**	**14.8**	8.1
3D-GRAND	**15.8**	**26.7**	**8.0**	12.0	**16.7**	**14.3**	**11.1**	**22.4**	5.0	**9.4**	**21.1**	**17.4**	**16.8**	**15.4**	16.4	11.1	**18.9**
PQ3D	10.8	2.2	4.0	8.0	3.3	8.6	3.7	4.1	**10.0**	**9.4**	11.6	7.2	10.7	5.5	8.2	11.1	8.1

Table 14: **Subgroup Analysis using Mask3D**. Breakdown of accuracy at 0.25 IoU using Mask3D box predictions on ViGiL3D for ScanNet scenes across several subgroups of prompts.

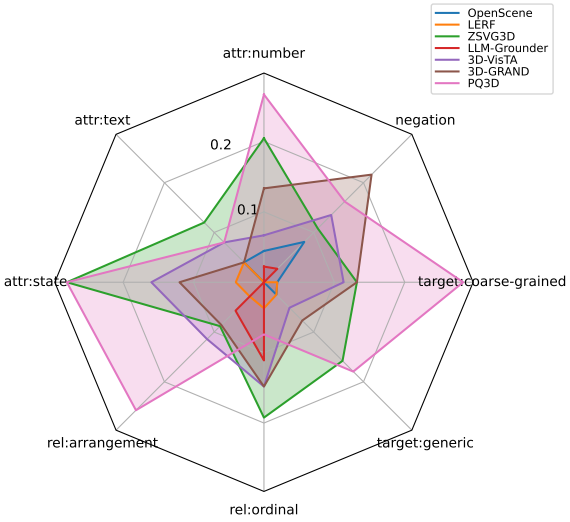

Figure 5: **Model Comparison on ViGiL3D.** We compare the performances of models on various subgroups on the ScanNet scenes of ViGiL3D using ground truth bounding boxes.

subgroup analysis are statistically significant, such as the performance of ground truth vs. predicted boxes for PQ3D and ZSVG3D and the performance on generic target specification for 3D-VisTA. We use the 2-tailed two proportion z-test to compare the accuracies of the higher performing models on each subgroup compared to the performance on the rest of the ViGiL3D prompts for ScanNet scenes.

Reweighted evaluation. We analyze performance when reweighting the distribution of prompts to more closely match that of linguistic patterns in ScanRefer. This was to assess whether the drop in performance on ViGiL3D compared to ScanRefer is caused more by the presence or absence of specific patterns or another factor. We find that the model performance only improves marginally, such as for 3D-GRAND from 0.18 to 0.19 when reweighted to its training dataset. This is likely caused in part by the difficulty in estimating the frequency of every co-occurrence of 25 language patterns, in which a simplified model causes a regression toward the unweighted mean. We also hypothesize that our dataset is more challenging, even in the language patterns represented more prominently in other datasets, thus causing the increase to be fairly marginal: 1) the targets can be 0, 1, or multiple objects, whereas most datasets specify exactly one object; 2) ViGiL3D prompts are designed to specify attributes or relationships that must each be parsed successfully to identify the possible targets, whereas most other VG prompts refer to unique object classes or overspecify constraints; and 3) within each language pattern, there is still potential for large variation, as evidenced by ViGiL3D having the highest frequency of unique bigrams. Even if the model has seen a particular pattern before, its performance may still be poor as a result of these additional challenges, and we observe both cases where the model performed better and worse on subgroups which were frequent in their training set.

D Scientific Artifacts

The licenses used in this paper include the following: ScanNet (terms of use[1]), ScanNet++ (terms of use[2]) ScanRefer (CC BY-NC-SA 3.0), Nr3D/Sr3D+ (MIT), Multi3DRefer (MIT), 3D-GRAND (CC BY 4.0), 3D-VisTA and ScanScribe (MIT), SceneVerse (terms of use [3]), Instruct3D (CC BY-NC-SA 4.0), OpenScene (apache-2.0), LERF (MIT), ZSVG3D (N/A), LLM-Grounder (apache-2.0), PQ3D (MIT), OpenAI (terms of use[4]), and spaCy (MIT). We follow the intended use of all of the licenses in the paper and reported our intended

[1] https://kaldir.vc.in.tum.de/scannet/ScanNet_TOS.pdf

[2] https://kaldir.vc.in.tum.de/scannetpp/static/scannetpp-terms-of-use.pdf

[3] https://drive.google.com/file/d/14Ji7PLOKsAxrXpxV6EWLsQGjzcEuk35N/view

[4] https://openai.com/policies/terms-of-use/

Figure 6: **Examples.** We provide additional examples for prompts from ViGiL3D on ScanNet scenes.

usage in the terms as appropriate.

LLMs, including ChatGPT, were used in the analysis pipeline and in some of the baseline methods. We also used them as assistive tools for generating code and researching methodologies.

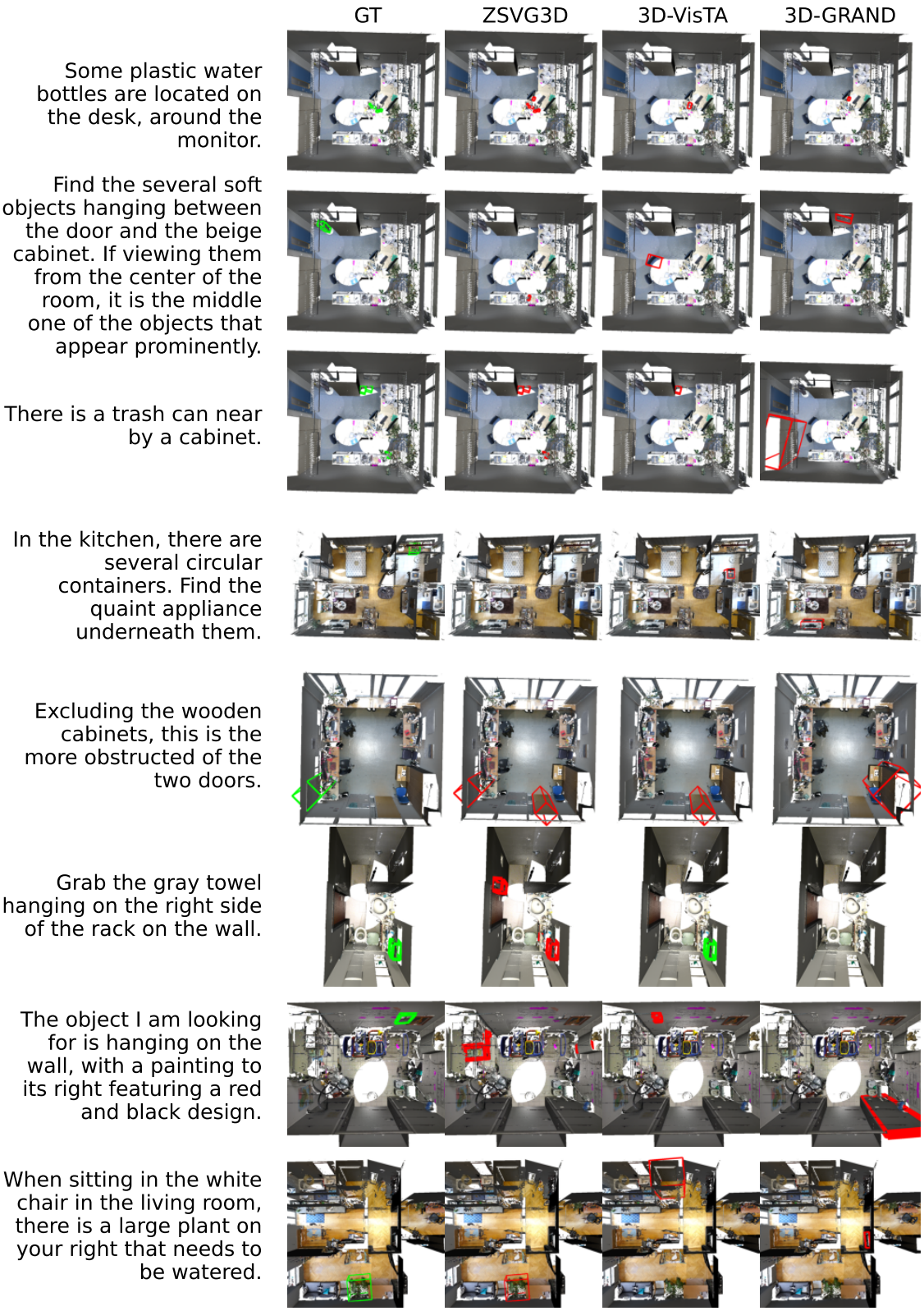

Figure 7: **Examples.** We provide additional examples for prompts from ViGiL3D on ScanNet++ scenes.