Multi3DRefer: Grounding Text Description to Multiple 3D Objects

Yiming Zhang[1] ZeMing Gong[1] Angel X. Chang[1,2]
Simon Fraser University[1] Alberta Machine Intelligence Institute (Amii)[2]
{yza440, zmgong, angelx}@sfu.ca
https://3dlg-hcvc.github.io/multi3drefer/

Abstract

We introduce the task of localizing a flexible number of objects in real-world 3D scenes using natural language descriptions. Existing 3D visual grounding tasks focus on localizing a unique object given a text description. However, such a strict setting is unnatural as localizing potentially multiple objects is a common need in real-world scenarios and robotic tasks (e.g., visual navigation and object rearrangement). To address this setting we propose Multi3DRefer, generalizing the ScanRefer dataset and task. Our dataset contains 61926 descriptions of 11609 objects, where zero, single or multiple target objects are referenced by each description. We also introduce a new evaluation metric and benchmark methods from prior work to enable further investigation of multi-modal 3D scene understanding. Furthermore, we develop a better baseline leveraging 2D features from CLIP by rendering object proposals online with contrastive learning, which outperforms the state of the art on the ScanRefer benchmark.

1. Introduction

There is growing interest in multi-modal methods that connect language and vision, tackling tasks such as image captioning, visual question answering, text-to-image retrieval, and generation. One fundamental task is visual grounding where natural language text queries are linked to regions of an image or 3D scene. While the problem of visual grounding has been well studied in 2D images, there are fewer datasets and methods studying the problem in 3D scenes. Being able to indicate the object that the text "the first of four stools" references in a 3D scene is useful for applications in robotics, AR/VR, and online 3D environments where we have access to not just a static image, but a 3D scene. Work on 3D datasets for visual grounding [7, 4] has spurred the development of methods for 3D visual grounding [41, 63, 60, 22, 21, 2, 58, 7] and the inverse task of 3D captioning [7], as well as unified methods that tackle both [8, 6, 9].

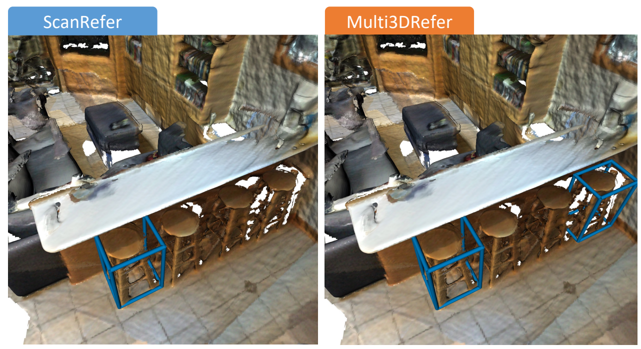

Figure 1: We introduce Multi3DRefer, a dataset and task where there are potentially multiple target objects for a given description. In ScanRefer [7] (left), the description corresponds to exactly one object (blue box), while in Multi3DRefer (right), there are multiple target objects.

However, existing datasets and tasks [7, 4] are designed with the assumption that there is a unique target object when performing visual grounding in a 3D scene. This assumption makes ambiguous descriptions that may refer to multiple objects problematic (see Fig. 1). Furthermore, this explicitly discourages visual grounding methods from demonstrating generalization to similar object instances on the basis of common features of the objects (e.g., similar size, color, texture) and spatial relations between objects (e.g., first in a row left-to-right or right-to-left).

We address these shortcomings with an enhanced dataset and task that we call Multi3DRefer where a flexible number of target objects (zero, single or multiple) in a 3D scene are localized given language descriptions. We modify and enhance language data from ScanRefer [7] and propose evaluation metrics to benchmark prior work and a CLIP-based method that we propose on the flexible number visual grounding task. In summary, we make the following contributions: 1) generalize 3D visual grounding to a flexible number of target objects given natural language descriptions. 2) create an enhanced dataset based on ScanRefer [7] with augmentations from ChatGPT[1], consisting of 61926 descriptions in 800 ScanNet V2 [15] scenes. 3) benchmark three prior 3D visual grounding approaches adapted to Multi3DRefer 4) design an end-to-end approach leveraging CLIP [47] embeddings and online rendering of object proposals with contrastive learning.

2. Related work

In this paper, we focus on description localization where a single description may describe one or more objects in a 3D scene. Below, we review work in grounding in 2D and 3D, as well as recent work leveraging pre-trained vision-language models for 3D scene understanding.

Visual grounding in 2D. A variety of datasets and methods have been proposed to investigate visual grounding tasks such as referring expressions [30, 42, 20] and phrase localization [44, 45] in 2D images. These datasets have enabled developing various visual-language grounding models [59, 61, 57, 39, 16, 37, 64]. Typically, in these datasets and tasks each phrase refers to exactly one object. A notable exception is the VGPhraseCut [54] dataset, based on Visual Genome [33] using templated phrases where each phrase is grounded to potentially multiple instance segments.

Recent work in language and vision has started to tackle more flexible grounding. Kim et al. [31] noted that not all queries can be visually grounded (i.e. it is possible to have no targets) and constructed a dataset to study grounding performance when there are unanswerable queries. Kuo et al. [35] proposed a single model for referring expression comprehension, object detection, and phrase localization. While their model can handle multiple objects, the queries for multiple objects are typically short and category-based. Recent work [29, 36] reframed the problem of object detection as phrase grounding by introducing losses to align words to regions. These methods are flexible and have been used to improve both detection and visual grounding. In our work, we construct a dataset for flexible grounding in 3D.

Visual grounding in 3D. Early work studied selecting the correct 3D shapes based on a text description in reference game setups [3, 53, 32], as well as learning joint language-3D embeddings for 3D text-to-shape retrieval [11, 51]. These works focused on descriptions of single objects in isolation. Moving beyond single 3D objects, researchers also studied grounding of language to objects in 3D scenes. At the scene level, ScanRefer [7] and ReferIt3D [4] introduce two datasets consisting of language descriptions of 3D objects from the real-world dataset ScanNet [15]. In detail, ReferIt3D [4] contains both template-based descriptions generated based on spatial relations between objects (Sr3D) and human-annotated fine-grained descriptions (Nr3D). They also propose two different grounding tasks,

[1] `https://openai.com/blog/chatgpt/`

Figure 2: Example description-scene pairs in the Multi3DRefer dataset with zero, single, or multiple target objects. Blue boxes indicate ground truth target objects.

both localizing a unique target object referred by a description. ScanRefer [7] requires both object detection and grounding, while ReferIt3D [4] focuses on discriminating a target object from multiple objects of that semantic class given ground-truth object bounding boxes.

Different approaches [7, 4] have been proposed to tackle the two tasks, with models focusing on graph representations [21, 60, 17], improved handling of relations [63, 12], neurosymbolic reasoning [19], leveraging multi-view images and 2D semantics [58, 24, 22, 5], to unified models that can address both grounding and captioning [6, 8, 23, 9]. Recently, Abdelreheem et al. [1], Wu et al. [55] showed that including training on dense annotations can improve performance. Jain et al. [25] proposed to tackle object detection and visual grounding in a unified way by aligning features for text tokens with object proposals. In this work, we compare the performance of three recent models on our task Multi3DRefer with a new CLIP-based model.

3D understanding using vision-language models. Large pre-trained text-vision models such as CLIP [47] and ALIGN [27] enabled work leveraging these models for 3D scene understanding. Recent work learn joint embeddings with text-image-3D representations [56, 62], used for disambiguating referring expressions [53] or text-to-shape retrieval [51]. Incorporating pre-trained 2D visual features also enabled expansion of 3D detection and instance segmentation to a larger number of categories [50], as well as tackling open vocabulary 3D detection [40, 52], and building of 3D semantic maps [26]. In our work, we show that we can leverage CLIP for improved visual grounding.

3. Multi3DRefer dataset

To study our task, we build the Multi3DRefer dataset, a superset of the existing ScanRefer dataset [7] with language descriptions of varying granularities. We augment ScanRefer [7] to create a dataset with 3 types of description-scene pairs: a) Zero Target; b) Single Target; and c) Multiple Targets, indicating zero, single, or multiple target objects in the scene match the description (see Fig. 2). In addition, we use

Dataset	Zero Target	Single Target	Multiple Targets	**Total**
ScanRefer [7]	-	51583	-	51583
Sr3D [4]	-	83572	-	83572
Sr3D+ [4]	-	114532	-	114532
Nr3D [4]	-	41503	-	41503
Multi3DRefer	6688	42060	13178	61926

Table 1: Compared to existing 3D visual grounding datasets, our Multi3DRefer dataset contains text that describes zero, single, or multiple target objects.

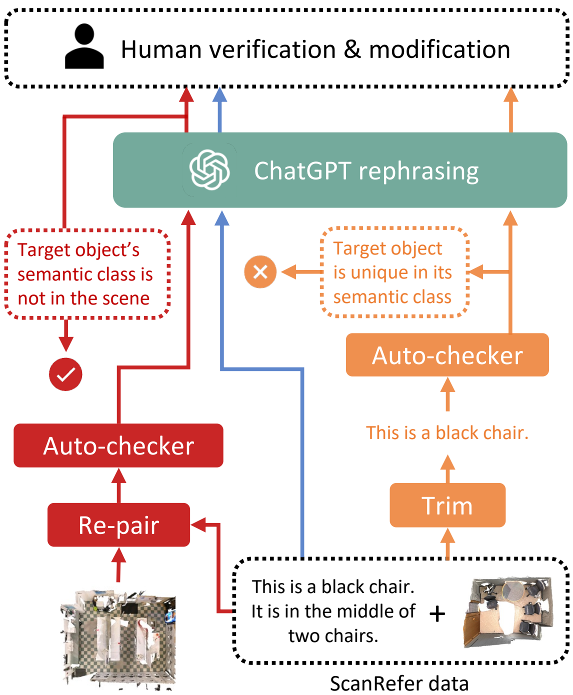

Figure 3: Multi3DRefer data construction pipeline showing the generation process of Zero Target (red), Single Target (blue) and Multiple Target (orange) data. We use ChatGPT to add diversity to the description. All scene-description pairs are manually verified and modified if needed.

ChatGPT to augment the descriptions so they are more natural and diverse (see Fig. 3 for the overall data construction pipeline). To ensure the dataset is of high quality we manually verify all generated samples. We obtain a dataset with 61926 descriptions in total (see Tabs. 1 and 3 for statistics).

3.1. Adapting text for multiple targets

We start with scene-description pairs from ScanRefer [7] and augment the data to create samples that refer to zero or more targets. For Single Target descriptions (type b), we take the released ScanRefer dataset which consists of description-scene pairs that have been initially verified to refer to unique objects [7]. We double-checked whether these descriptions indeed refer to unique objects and we found 9324 descriptions that are ambiguous. We use these ambiguous descriptions as an initial set of type (c) descriptions that can refer to multiple objects. For additional type (c) descriptions, we obtain from the ScanRefer authors 7741 ambiguous descriptions that can refer to multiple objects and annotate those descriptions with matching objects. To collect type (a) description (no target object) as well as more type (c) descriptions, we develop an efficient data collection pipeline consisting of two stages: 1) automated description generation; and 2) verification and modification. This pipeline maximizes the use of the existing ScanRefer dataset and reduces manual annotation. Below, we describe how we generate and verify additional data samples.

Zero Target. To obtain descriptions with no target objects, we establish negative pairs of scenes with existing descriptions selected randomly from other scenes. We then manually verify that descriptions do not match any objects in the new scene. To reduce the number of description-scene pairs that need to be verified, we automatically check whether the semantic class of the target object for the description appears in the scene. Only if the semantic class appears in the scene does it need to be manually verified. Out of 6688 samples, we manually verify 5630 with 1058 automatically checked to have no matching objects. For pairs that need verification, human annotators are shown the description along with an interactive view of the scene to check that there are no matching objects.

Multiple Targets. To generate descriptions with multiple targets, we start with the original description-scene pairs in the ScanRefer dataset. We then randomly select descriptions and trim the text to the first punctuation to obtain shorter, more ambiguous descriptions (e.g., *"The cabinet is white and in the back of the room. It is the one on the left."* $\rightarrow$ *"The cabinet is white and in the back of the room."*). Note that descriptions in which the semantic class of the target object only has a unique object in the scene are skipped. The trimmed text and scene pairs are sent to the annotation interface. This time, annotators are asked to select all eligible objects of a description in a 3D interactive scene mesh. Annotators are also asked to modify trimmed descriptions to fix errors and increase diversity (see Fig. 4 for examples).

3.2. Rephrasing using ChatGPT

To increase description diversity we use the ChatGPT model *text-davinci-002-render* for sentence rephrasing. We provide ChatGPT with the following prompts:

1. I will give you a sentence describing an object, please help me polish it and keep its meaning.
2. Help me reword a sentence to a different format but keep its meaning.
3. Help me reword a sentence to make it more natural.
4. Help me reword a sentence describing an object, you should describe the colors and the spatial information in a different way.
5. Help me reword a sentence to an interesting format, you should keep its meaning.

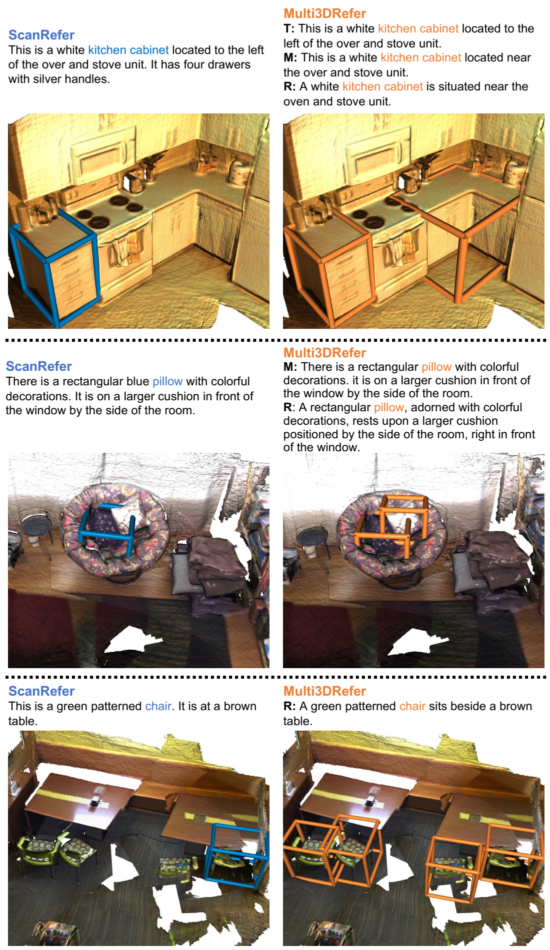

Figure 4: Examples of generated revised descriptions of multiple targets for the Multi3DRefer dataset where we trim (T) to create more ambiguous descriptions, modify (M) by human, and reword (R) using ChatGPT. All descriptions are verified by annotators (see bounding boxes for target objects). Note that the reworded (R) descriptions provide more variation in sentence structure.

	Unique words	Total words	Avg. description length
Original	5067	1016190	16.4
Rephrased	7077	936935	15.1

Table 2: Comparison of Multi3DRefer language data before and after ChatGPT rephrasing. We count the number of unique words, total words, and the average description length, excluding punctuations.

Tab. 2 shows statistics comparing the descriptions before and after rephrasing. We see a richer vocabulary but shorter descriptions after ChatGPT rephrasing. Below we provide examples of the original (O) and reworded (R) text:

O: *The table is a round table. It is located between two chairs to the right, and two chairs to the left of it.*
R: *The round table is situated between two chairs to its right and two chairs to its left.*

O: *This is a table on the wall in the room. It is next to the window and a few lined-up chairs.*
R: *A wall-mounted table resides cozily beside a window in the room, accompanied by a row of orderly chairs.*

O: *A sink on the vanity. It is to the right of the vacuum cleaners.*
R: *The sink is located on the vanity to the right of the vacuum cleaners.*

O: *This is a white kitchen cabinet located near the over and stove unit.*
R: *A white kitchen cabinet is situated near the oven and stove unit*

	Spatial	Color	Texture	Shape	**Total**
ScanRefer [7]	51117	34692	5864	17416	51583
Nr3D [4]	39711	11939	526	8568	41503
Sr3D [4]	83572	6254	0	648	83572
Sr3D+ [4]	114532	8666	0	744	114532
Multi3DRefer	60028	41307	7121	19692	61926

Table 3: Breakdown of spatial, color, texture, and shape information in object descriptions from different datasets.

The rephrased text preserves the original meaning while being more natural. In addition, ChatGPT automatically corrects typos (e.g., *over* to *oven* in the last example).

3.3. Verification

After we obtain a set of ChatGPT reworded descriptions, we manually verify the descriptions are well-written and that the object(s) matched in the scene are accurate. We create a web interface for verifiers to check whether the description matches the identified target objects (see supplement for details). The web interface shows the description together with an interactive 3D view of the scene and the target objects. The verifiers check if the description matches the target objects and only the target objects, or modify the list of target objects (by selecting appropriate objects), or improve the description to fix typos and ambiguities. The verification was performed by 5 students over a period of one month.

3.4. Dataset statistics

In total, our dataset consists of 61926 language descriptions, with 51583 directly obtained from ScanRefer [7], of which 6688 descriptions match zero-targets and 13178 match multiple. For Multiple Targets, the scenes are typically offices or meeting rooms with many chairs and tables. See Tab. 1 for a comparison of our final dataset against prior datasets. We also provide annotations for each description as to whether it refers to spatial, color, texture, or shape attributes (see Tab. 3).

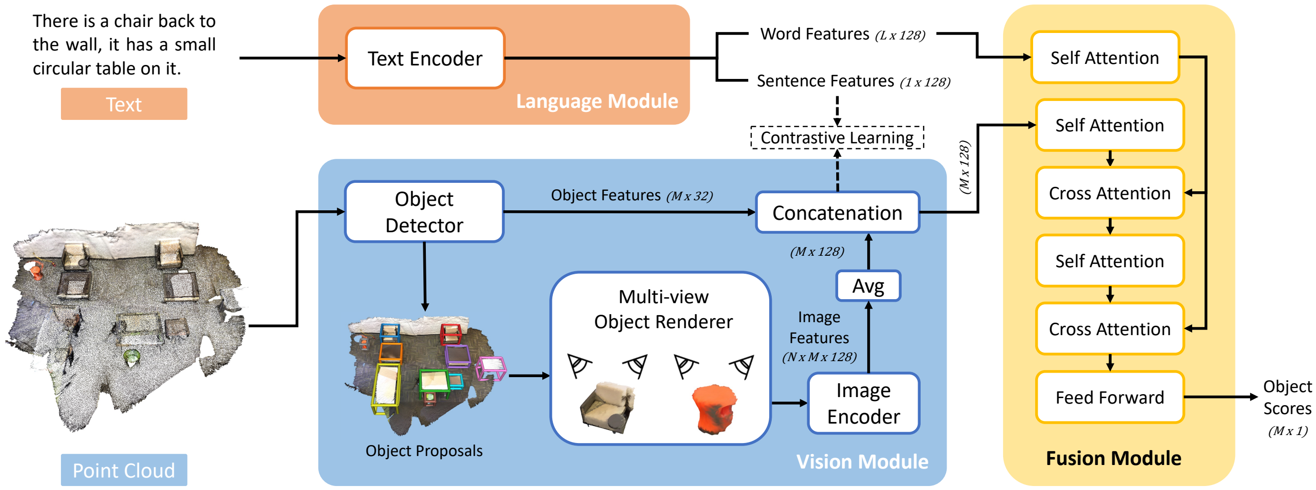

Figure 5: Our M3DRef-CLIP end-to-end architecture. Given a scene point cloud and a text description with L tokens (we pad shorter descriptions and truncate longer ones), the detector first predicts M object proposals and their 3D features. Then an online renderer renders N-view images for each proposal and feeds them into the image encoder to get 2D features. A transformer-based module then fuses both language features and 2D + 3D object features and outputs scores indicating the match of each object to the description. We use PointGroup [28] to detect and segment the objects in 3D and select CLIP [47] + MLPs as the language and image encoder. A contrastive loss is applied between sentence features and object features.

4. Task

In the Multi3DRefer task, we are given as input a 3D real-world scene represented as a point cloud $P \in \mathbb{R}^{N\times(3+C)}$ and a free-form language description with a variable number $M \in \mathbb{N}$ of referred objects, where N, C are the number of points and the number of point feature channels, respectively. The goal is to predict axis-aligned bounding boxes for all M objects that match the description. Compared to prior work, the difficulty of our task is that the number of referred objects is flexible, i.e., a description can refer to not only one or multiple target objects but also no objects. Predicting too many or too few target objects are both penalized by our evaluation metrics.

Evaluation metrics. To evaluate grounding for a flexible number of target objects, we measure the F1 score at the intersection over union (IoU) thresholds of $\tau_{\text{eval}} = 0.25$ and 0.5 (F1@0.25 and F1@0.5). To investigate model performance for different scenarios, we consider the following 5 cases: a) zero target w/o distractors of the same semantic class; b) zero target w/ distractors; c) single target w/o distractors; d) single target w/ distractors; and e) multiple targets. Note that c) and d) correspond to the "unique" and "multiple" cases in ScanRefer [7]. In addition, a) and c) are easier cases where there are zero or a unique target object of its semantic class in a scene, while b) and d) are more difficult cases containing one or multiple target objects of the same semantic class in a scene. We also report the micro-average of the 5 cases as an overall score.

During evaluations, we first calculate per-pair IoUs between ground truth and predicted bounding boxes in a scene, and then apply the Hungarian algorithm [34] to get an optimal one-to-one matching between predicted and GT bounding boxes. To get the maximum matched IoU, we use the following cost function:

$$\mathrm{Cost}(i,j) = -\,\mathrm{IoU}(i,j) \text{ for } i,j \in \{1\ldots N\}$$

where $N = \max(\#\textit{Predictions}, \#\textit{GTs})$. After obtaining the optimal match, we take pairs with IoUs higher than τ_{eval} to be True Positives (TP). We treat the Zero Target (ZT) case as a special case where recall is always set to 1, and precision is set to 1 if there is no prediction or 0 otherwise.

5. Method

We propose M3DRef-CLIP, a CLIP-based approach and compare to three recent approaches: 3DVG-Transformer [63], 3DJCG [6] and D3Net [8]. We selected these methods as they were among the top performers on the ScanRefer [7] benchmark[2] with available open-source code. Note that 3DJCG [6] and D3Net [8] are unified models, which can do both grounding and captioning tasks. All four models are two-stage approaches, where a 3D object detector first identifies a set of bounding box candidates, and a disambiguation module then selects the target bounding box. In the original ScanRefer setting, the three compared models are trained using a cross-entropy loss L_{ref} where the predicted bounding box has the highest IoU ($> \tau_{\text{train}}$) with

[2]kaldir.vc.in.tum.de/scanrefer_benchmark

the GT bounding box as the target bounding box to calculate the loss. For our task, we use a binary cross-entropy loss for L_{ref} as our problem is a multi-label task (vs classification).

5.1. M3DRef-CLIP

M3DRef-CLIP follows the two-stage architecture with PointGroup [28] as the detector, CLIP [47] as the text encoder and a transformer-based fusion module (see Fig. 5). We use PointGroup [28] to obtain object proposals along with their 3D features $\mathbf{F}^{3\text{d}} \in R^{32}$. The output object proposals are fed into an online object renderer, which renders multi-view 2D images for each object proposal. We use CLIP [47] to encode the images and use an MLP to project the average of the output 2D image features to obtain $\mathbf{F}^{2\text{d}} \in \mathbb{R}^{128}$. The 2D and 3D features are concatenated to form and passed through a 1D convolution (that projects the combined 160-dim features down to 128-dim) to obtain the final visual features $\mathbf{F}^{\text{obj}} \in \mathbb{R}^{128}$ for an object proposal:

$$\mathbf{F}^{\text{obj}} = \text{conv}_{\text{1d}}([\mathbf{F}^{3\text{d}}; \mathbf{F}^{2\text{d}}]), \quad \mathbf{F}^{2\text{d}} = \frac{\text{MLP}(\sum_{i=1}^{N} \mathbf{F}_i^{2\text{d}})}{N}$$

We use the CLIP text encoder with an MLP to obtain both 128-dim token-level and sentence embeddings. We use a transformer-based fusion module to combine object features and token-level embeddings and output confidence scores for each proposal. To improve the training, we apply a contrastive loss between sentence and object features.

5.2. Loss function

We train the network end-to-end with the total loss $L = L_{\text{det}} + L_c + L_{\text{ref}}$ consisting of the detection loss L_{det}, contrastive loss L_{c}, and the reference loss L_{ref}. The detection loss L_{det} is introduced in PointGroup [28] and consists of four parts: 1) cross-entropy loss for supervising per-point semantic class prediction; 2) L_1 loss for supervising per-point offset vector towards object centers; 3)directional loss formed as a mean of minus cosine similarities for further constraining the direction of per-point offset vectors; 4) binary cross-entropy loss for supervising per-point objectness confidence score. To help learn better multi-modal embeddings, we introduce a symmetric contrastive loss L_c [47], which can handle a flexible number of target objects as shown below:

$$L_{\text{c}}^{O \rightarrow S} = -\log \frac{\exp(\cos(\bar{O}_i, S_i)/\tau)}{\sum_{i=1}^{n} \exp(\cos(\bar{O}_i, S_j)/\tau)}$$

$$L_{\text{c}}^{S \rightarrow O} = -\log \frac{\exp(\cos(S_i, \bar{O}_i)/\tau)}{\sum_{i=1}^{n} \exp(\cos(S_i, \bar{O}_j)/\tau)}$$

where S is sentence features, $\bar{O}$ is the mean of object features of all target objects paired with a description, and τ is a temperature parameter. Finally, the reference loss L_{ref} supervises the matching module to select objects satisfying the description. We use the multi-class cross-entropy loss as L_{ref} for experiments on ScanRefer [7], Nr3D [4] and Multi3DRefer (ST) where each description only refers to a single target object. For experiments on Multi3DRefer, L_{ref} is a sum of binary cross-entropy losses over detected objects.

Training. To set up positive (i.e. valid target bounding boxes) and negative instances between the GT bounding boxes and proposed bounding boxes, we use two strategies (see supplement for detailed comparisons):

All All predicted bounding boxes with IoU higher than the threshold τ_{train} with GT bounding boxes are considered target bounding boxes.

Hungarian We apply the Hungarian algorithm [34] to do bipartite matching, which ensures an optimal solution. We use the same IoU threshold τ_{train} to filter prediction-GT bounding box pairs.

Inference. At inference time, we take all bounding boxes with predicted scores above the threshold τ_{pred} as positives (predicted target bounding boxes for the description). See the supplement for comparisons of different τ_{pred}.

6. Experiments

We conduct experiments on both ScanRefer [7] and Multi3DRefer datasets and consider two setups: grounding objects with GT and predicted bounding boxes.

GT bounding boxes. For M3DRef-CLIP and D3Net [8], we input the complete scene and apply GT point masks for each object to get GT bounding boxes and masked features from the pre-trained detector (PointGroup for both D3Net and our M3DRef-CLIP). For 3DVG-Trans [63] and 3DJCG [6], we use their original GT setting following the method proposed by ReferIt3D [4]. Single input objects are first extracted from the scene using the GT masks and PointNet++ [46] is used for obtaining the object features. For all methods, we disable the L_{det} loss when using GT boxes.

Predicted bounding boxes. We use the original detector design of each method to predict bounding boxes and extract features.

6.1. Implementation details

We implement M3DRef-CLIP using PyTorch Lightning[3] and PyTorch3D [48]. We train the model end-to-end on a single NVIDIA RTX A5000 with a batch size of 4, using the AdamW optimizer [38] with a learning rate of $5e^{-4}$. We use a re-implementation of PointGroup[4] with the Minkowski Engine [14]. Following D3Net [8], we pretrain our PointGroup module on all ScanNet v2 training scans with the 18 ScanRefer categories. We take point coordinates, point normals and per-point multi-view features $P \in \mathbb{R}^{N \times (3+3+128)}$ as the input. For data augmentation, we randomly apply

[3] www.pytorchlightning.ai
[4] https://github.com/3dlg-hcvc/minsu3d

	ZT w/o Distractors				ZT w Distractors				ST w/o Distractors				ST w/ Distractors				Multiple Targets			
	Train	Val	Test	**Total**	Train	Val	Test	**Total**	Train	Val	Test	**Total**	Train	Val	Test	**Total**	Train	Val	Test	**Total**
ScanRefer [7]	-	-	-	-	-	-	-	-	8500	2297	1201	11998	28165	7211	4209	39585	-	-	-	-
Multi3DRefer	2702	528	596	3826	2160	378	324	2862	7198	2099	1106	10403	22040	5358	4259	31657	9738	2757	683	13178

Table 4: Breakdown of different datasets in 5 scenarios. ZT and ST denote Zero Target and Single Target, respectively.

	Acc@0.5 on Val			Acc@0.5 on Test		
	Unique	Multiple	**All**	Unique	Multiple	**All**
3DVG-Trans+ [63]	62.0	30.3	36.4	57.9	31.0	37.0
InstanceRefer [60]	66.8	24.8	32.9	66.7	26.9	35.8
FFL-3DOG [17]	67.9	25.7	34.0	-	-	-
SAT [58]	50.8	25.2	30.1	-	-	-
3D-SPS [41]	66.7	29.8	37.0	-	-	-
MVT [22]	66.5	25.3	33.3	-	-	-
BUTD-DETR [25]	66.3	35.1	39.8	-	-	-
D3Net (G) [8]	70.4	27.1	35.6	65.8	27.3	36.0
D3Net* [8]	72.0	30.1	37.9	68.4	30.7	39.2
3DJCG (G) [6]	64.5	30.3	36.9	-	-	-
3DJCG* [6]	64.3	30.8	37.3	60.6	31.2	37.8
HAM [10]	67.9	34.0	40.6	63.7	33.2	40.1
UniT3D (G) [9]	74.8	27.6	36.5	-	-	-
UniT3D* [9]	73.1	31.1	39.1	-	-	-
M3DRef-CLIP	**77.2**	**36.8**	**44.7**	**70.9**	**38.1**	**45.5**

Table 5: For unified models [8, 6, 9], we report both the grounding-only (G) performance as well as their best performance. We use * to indicate joint grounding and captioning models trained with extra data.

	Easy	Hard	View-Dep	View-Indep	**All**
3DVG-Trans [63]	48.5	34.8	34.8	43.7	40.8
InstanceRefer [60]	46.0	31.8	34.5	41.9	38.8
FFL-3DOG [17]	48.2	35.0	37.1	44.7	41.7
TransRefer3D [18]	48.5	36.0	36.5	44.9	42.1
SAT [58]	56.3	42.4	46.9	50.4	49.2
3D-SPS [41]	58.1	45.1	48.0	53.2	51.5
MVT [22]	**61.3**	**49.1**	**54.3**	55.4	**55.4**
BUTD-DETR [25]	60.7	48.4	46.0	**58.0**	54.6
HAM [10]	54.3	41.9	41.5	51.4	48.2
LanguageRefer [49]	51.0	36.6	41.7	45.0	43.9
LAR [5]	56.1	41.8	46.7	50.2	48.9
M3DRef-CLIP	55.6	43.4	42.3	52.9	49.4

Table 6: Comparison of methods on Nr3D [4] val set with GT boxes.

	Acc on ScanRefer			Acc on Multi3DRefer (ST)		
Training Dataset	Unique	Multiple	**All**	Unique	Multiple	**All**
ScanRefer [7]	89.3	49.1	56.9	79.5	48.2	57.0
Multi3DRefer (ST)	88.5	46.9	55.0	86.3	52.9	62.3
Multi3DRefer (ST) + ScanRefer	**90.8**	**51.0**	**58.7**	**88.0**	**56.7**	**65.5**

Table 7: We compare results of training M3DRef with GT boxes on different datasets' val set. We only use the Single Target (ST) case in Multi3DRefer dataset.

coordinate jitter, x-axis flipping and rotation around the z-axis. We freeze a pre-trained CLIP with ViT-B/32 and only train additional MLPs. For encoding the text with CLIP, we follow CLIP and tokenize with a lower-cased BPE, and `[SOS]` and `[EOS]` tokens added (the output corresponding to `[EOS]` is used as the sentence representation).

Object renderer. For each object proposal, we render 3 views horizontally spaced 120 degrees apart, at a distance of 1m, with elevation angle of $45°$. We crop coordinates and colors from the input scene point cloud using predicted bounding boxes and set point radius to 2.5cm and image size to 224^2. We use CUDA to batch index scene point clouds and crop in parallel, and render the batches sparsely to avoid padding overhead. Rendering is implemented with PyTorch3D [48] and executed on the GPU.

6.2. Results

6.2.1 Performance of M3DRef-CLIP

We first validate the performance of M3DRef-CLIP on ScanRefer (see Tab. 5) and Nr3D [4] (see Tab. 6) datasets and compare it against recent models. On ScanRefer, our method outperforms all prior works including those joint models leveraging extra input data by a large margin on both val set and test set (online benchmarking). In our ablation study, we find that the use of the CLIP text encoder is a key factor to the strong performance of our model on ScanRefer. Another key factor is the use of a strong 3D object instance segmentation network (PointGroup) as our object detector. For instance, the PointGroup-based methods [8, 9] readily outperform VoteNet-based methods [63, 6] on the unique subset of ScanRefer.

For Nr3D [4], we achieve comparable but less competitive results because of our weaker ground-truth box encoder. Note that SAT [58] and MVT [22] also leverage 2D images but render them offline. Overall, using additional 2D image information is helpful for two-stage methods.

6.2.2 Multi3DRefer

We split the Multi3DRefer data into train/val/test by scene following the ScanRefer split, resulting in a rough split ratio of 7:2:1 for the scene-description pairs. See Tab. 4 for statistics, including the number of descriptions with zero, single, or multiple targets.

Usefulness of Multi3DRefer dataset. To study the usefulness of the Multi3DRefer dataset, we compare the perfor-

	F1 (GT boxes)						F1@0.5 (Pred boxes)					
	ZT w/o D	ZT w/ D	ST w/o D	ST w/ D	MT	**All**	ZT w/o D	ZT w/ D	ST w/o D	ST w/ D	MT	**All**
3DVG-Trans+ [63]	45.3	14.3	58.9	35.2	54.2	44.1	87.1	45.8	27.5	16.7	26.5	25.5
D3Net (Grounding) [8]	71.6	20.4	78.2	44.4	61.6	55.5	81.6	32.5	38.6	23.3	35.0	32.2
3DJCG (Grounding) [6]	47.9	16.4	59.1	35.5	54.2	44.6	**94.1**	**66.9**	26.0	16.7	26.2	26.6
M3DRef-CLIP	**74.2**	**29.4**	**84.1**	**52.3**	**67.2**	**62.3**	81.8	39.4	**47.8**	**30.6**	**37.9**	**38.4**

Table 8: Comparison of different methods on Multi3DRefer. Our M3DRef-CLIP outperforms prior work on most metrics.

	Acc (GT boxes)			Acc@0.5 (Pred boxes)		
Eval Dataset	Unique	Multiple	**All**	Unique	Multiple	**All**
Multi3DRefer (ST)	86.9	51.5	61.5	67.3	40.1	47.7
ScanRefer [7]	88.0	46.1	54.3	73.5	34.3	41.9

Table 9: M3DRef-CLIP evaluated on Multi3DRefer (ST) and ScanRefer with both GT and predicted bounding boxes.

mance of training M3DRef-CLIP with the original ScanRefer data and with our Multi3DRefer data for the Single Target (ST) case. We evaluate on both ScanRefer and Multi3DRefer (ST), using GT bounding boxes. Tab. 7 shows that our reworded data can improve performance on ScanRefer. Prior work has also shown that incorporating additional labeled data can help, but they typically used either a captioning model [8, 9], mixed additional annotated data [58], or used dense annotations [1]. We show that simple rewordings (without accessing the images) also help.

We also evaluate M3DRef-CLIP trained on Multi3DRefer using ScanRefer's task setting (Tab. 9) of predicted objects. We observe that the model trained on Multi3DRefer data and task achieves similar performance to the model trained on ScanRefer data and task, which illustrates the generalization of Multi3DRefer.

Evaluation on Multi3DRefer. We compare four models on the Multi3DRefer dataset using both ground-truth and predicted boxes (Tab. 8). In our experiments, we focus on two-stage methods that perform well on ScanRefer. We adapt the code of 3DVG-Trans, 3DJCG and D3Net to our task. For 3DVG-Trans, we use the enhanced version 3DVG-Trans+ provided by the authors.[5] We only train and evaluate the grounding model of 3DJCG and D3Net.

To analyze the performance of the models, we break down the zero (ZT) and single-target (ST) case to without and with distractors (D) of the same class. Overall, having distractors is more challenging. We note that M3DRef-CLIP outperforms the other methods on Multi3DRefer, and that 3DJCG is better at handling the ZT case with predicted boxes. Fig. 6 shows qualitative results.

[5] github.com/zlccccc/3DVG-Transformer

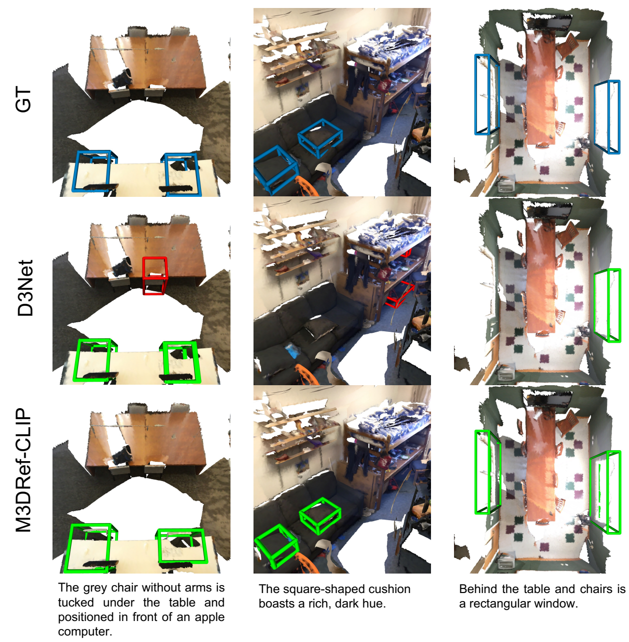

Figure 6: Qualitative results of D3Net [8] versus M3DRef-CLIP on Multi3DRefer using predicted boxes. Blue boxes indicate GT, green boxes are true positives with IoU threshold $\tau_{\text{pred}} > 0.5$. Red boxes are false positives.

6.2.3 Ablation studies

Does CLIP help? We experiment with just using pure CLIP for both text and image encoding vs combining CLIP and 3D features (see Tab. 10). We also report results using a GRU [13]-based text encoder with GloVE [43] embedding, as used in prior work [7, 63, 8]. Although our GRU and 3D variants are similar at a high level to the grounding model of D3Net (which also used GRU and PointGroup with a transformer-based fusion module), we outperform the D3Net grounding module. Tab. 10 shows that using the CLIP text encoder improves performance, and combining CLIP and 3D features yields the best performance. The CLIP image encoder by itself underperforms the 3D features, showing the usefulness of 3D information.

Does contrastive learning help? Tab. 11 reports the performance of M3DRef with and without contrastive learning on ScanRefer, Nr3D [4] and Multi3DRefer. We observe the

	Text	Vision	Acc (ScanRefer) Uniq	Mult	**All**	F1 (Multi3DRefer) ZT w/o D	ZT w/ D	ST w/o D	ST w/ D	MT	**All**
GT	GRU	3D	88.8	43.5	52.3	70.1	27.3	81.6	49.3	63.3	59.1
GT	CLIP	CLIP	78.9	42.1	49.3	58.3	27.8	74.1	44.7	61.0	54.4
GT	CLIP	3D+CLIP	**89.3**	**49.1**	**56.9**	**74.2**	**29.4**	**84.1**	**52.3**	**67.2**	**62.3**
Pred	GRU	3D	72.2	32.8	40.4	78.8	42.1	**49.4**	28.5	37.0	37.4
Pred	CLIP	CLIP	71.2	31.2	39.0	64.4	36.0	42.9	25.7	33.7	33.1
Pred	CLIP	3D+CLIP	**77.2**	**36.8**	**44.7**	**81.8**	**39.4**	47.8	**30.6**	**37.9**	**38.4**

Table 10: Ablations in training M3DRef using different feature embeddings and ground-truth boxes (top rows) and predicted boxes (bottom rows). The combination of CLIP [47] and 3D features achieves the best performance. We use PointGroup [28] for our 3D object detector and feature extractor.

	ScanRefer Unique	Multiple	**All**	Nr3D Easy	Hard	View-Dep	View-Indep	**All**	Multi3DRefer ZT w/o D	ZT w/ D	ST w/o D	ST w/ D	MT	**All**
w/ contrastive	89.3	**49.1**	**56.9**	**55.6**	**43.4**	**42.3**	**52.9**	**49.4**	**74.2**	**29.4**	**84.1**	**52.3**	**67.2**	**62.3**
w/o contrastive	**89.6**	47.2	55.4	50.9	35.3	37.6	45.6	42.9	67.4	23.8	83.4	52.0	66.5	61.1

Table 11: Ablation of M3DRef-CLIP with and without contrastive loss using GT boxes.

Text	Vision	Spatial	Color	Texture	Shape
GRU	3D	55.8	67.7	76.4	77.8
CLIP	3D+CLIP	**58.8**	**71.9**	**79.4**	**82.5**

Table 12: Breakdown of M3DRef performance (with GRU vs CLIP) on descriptions with different attributes. We report F1 scores with GT boxes on Multi3DRefer val set.

benefit of the contrastive loss for all tasks, especially Nr3D.

6.3. Analysis of M3DRef-CLIP

To better study the Multi3DRefer data and understand how M3DRef-CLIP helps address the task, we break down evaluation of Multi3DRefer based on attributes provided in the description: spatial, color, texture, and shape information. For this analysis, these four splits are mutually exclusive, i.e. we only keep descriptions that describe exactly 1 attribute from the four and discard others. We compare using M3DRef with GRU with 3D PointGroup features vs our full M3DRef-CLIP model (with CLIP image and text encoders). We use the GT boxes setting and report F1 scores in Tab. 12. We observe that adding features from CLIP [47] helps identify all these attributes. We found that descriptions with spatial terms are more challenging than descriptions with texture or shape.

7. Conclusion

We present Multi3DRefer, a more realistic task of grounding a flexible number of objects in real-world 3D scenes using natural language descriptions. We designed a simple and efficient data generation pipeline to create data with less human effort, by leveraging existing language data and ChatGPT. With this pipeline, we created a more diverse dataset consisting of more natural descriptions of varying granularity. We also explored an end-to-end baseline method for solving the new task, which enables the online rendering of proposal objects to generate 2D cues, it also demonstrated the usefulness of CLIP [47] and the multi-modal contrastive loss. We believe Multi3DRefer will bring more challenges and practical value in the direction of bridging 3D vision and language, especially for robotics and embodied AI tasks.

Future Work. Our current design relies on features from the 3D object detector to capture the global context, and the 2D image encoder to capture per-object attributes. Using positional encoding could improve the ability of the model to handle spatial relations. Investigating whether positional encoding improves the model, and what kind of positional encoding works best is a great avenue for future work.

Acknowledgements

This work was funded in part by a Canada CIFAR AI Chair and NSERC Discovery Grant, and enabled in part by support from the Digital Research Alliance of Canada. We thank Zhenyu (Dave) Chen for providing us with the ambiguous descriptions initially collected for ScanRefer, and Manolis Savva for proofreading and editing suggestions. We also thank Archita Srivastava, Cody Ning, and Austin T. Wang for helping to verify the Multi3DRefer dataset.

References

[1] Ahmed Abdelreheem, Kyle Olszewski, Hsin-Ying Lee, Peter Wonka, and Panos Achlioptas. ScanEnts3D: Exploiting phrase-to-3D-object correspondences for improved visio-linguistic models in 3D scenes. *arXiv preprint arXiv:2212.06250*, 2022.

[2] Ahmed Abdelreheem, Ujjwal Upadhyay, Ivan Skorokhodov, Rawan Al Yahya, Jun Chen, and Mohamed Elhoseiny. 3DRefTransformer: Fine-grained object identification in real-world scenes using natural language. In *Proc. of the Winter Conference on Applications of Computer Vision (WACV)*, pages 3941–3950, 2022.

[3] Panos Achlioptas, Judy Fan, Robert Hawkins, Noah Goodman, and Leonidas J Guibas. ShapeGlot: Learning language for shape differentiation. In *Proc. of the International Conference on Computer Vision (ICCV)*, pages 8938–8947, 2019.

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

[5] Eslam Mohamed Bakr, Yasmeen Alsaedy, and Mohamed Elhoseiny. Look around and refer: 2D synthetic semantics knowledge distillation for 3D visual grounding. *Advances in neural information processing systems*, 2022.

[6] Daigang Cai, Lichen Zhao, Jing Zhang, Lu Sheng, and Dong Xu. 3DJCG: A unified framework for joint dense captioning and visual grounding on 3D point clouds. In *Proc. of the Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16464–16473, 2022.

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

[8] Dave Zhenyu Chen, Qirui Wu, Matthias Nießner, and Angel X. Chang. D3Net: A unified speaker-listener architecture for 3D dense captioning and visual grounding. In *Proc. of the European Conference on Computer Vision (ECCV)*, 2022.

[9] Dave Zhenyu Chen, Ronghang Hu, Xinlei Chen, Matthias Nießner, and Angel X Chang. UniT3D: A unified transformer for 3D dense captioning and visual grounding. In *Proc. of the International Conference on Computer Vision (ICCV)*, 2023.

[10] Jiaming Chen, Weixin Luo, Xiaolin Wei, Lin Ma, and Wei Zhang. HAM: Hierarchical attention model with high performance for 3D visual grounding. *arXiv preprint arXiv:2210.12513*, 2022.

[11] Kevin Chen, Christopher B Choy, Manolis Savva, Angel X Chang, Thomas Funkhouser, and Silvio Savarese. Text2shape: Generating shapes from natural language by learning joint embeddings. In *Asian conference on computer vision*, pages 100–116. Springer, 2018.

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

[13] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[14] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4D spatio-temporal convnets: Minkowski convolutional neural networks. In *Proc. of the Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3075–3084, 2019.

[15] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. ScanNet: Richly-annotated 3D reconstructions of indoor scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017.

[16] Jiajun Deng, Zhengyuan Yang, Tianlang Chen, Wengang Zhou, and Houqiang Li. TransVG: End-to-end visual grounding with transformers. In *Proc. of the International Conference on Computer Vision (ICCV)*, pages 1769–1779, 2021.

[17] Mingtao Feng, Zhen Li, Qi Li, Liang Zhang, XiangDong Zhang, Guangming Zhu, Hui Zhang, Yaonan Wang, and Ajmal Mian. Free-form description guided 3D visual graph network for object grounding in point cloud. In *Proc. of the International Conference on Computer Vision (ICCV)*, pages 3722–3731, 2021.

[18] Dailan He, Yusheng Zhao, Junyu Luo, Tianrui Hui, Shaofei Huang, Aixi Zhang, and Si Liu. TransRefer3D: Entity-and-relation aware transformer for fine-grained 3D visual grounding. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 2344–2352, 2021.

[19] Joy Hsu, Jiayuan Mao, and Jiajun Wu. NS3D: Neuro-symbolic grounding of 3D objects and relations. In *Proc. of the Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2614–2623, 2023.

[20] Ronghang Hu, Huazhe Xu, Marcus Rohrbach, Jiashi Feng, Kate Saenko, and Trevor Darrell. Natural language object retrieval. In *Proc. of the Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4555–4564, 2016.

[21] Pin-Hao Huang, Han-Hung Lee, Hwann-Tzong Chen, and Tyng-Luh Liu. Text-guided graph neural networks for referring 3D instance segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021.

[22] Shijia Huang, Yilun Chen, Jiaya Jia, and Liwei Wang. Multi-view transformer for 3D visual grounding. In *Proc. of the Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15524–15533, 2022.

[23] Shijia Huang, Feng Li, Hao Zhang, Shilong Liu, Lei Zhang, and Liwei Wang. A unified mutual supervision framework for referring expression segmentation and generation. *arXiv preprint arXiv:2211.07919*, 2022.

[24] Ayush Jain, Nikolaos Gkanatsios, Ishita Mediratta, and Katerina Fragkiadaki. Looking outside the box to ground language in 3D scenes. *arXiv preprint arXiv:2112.08879*, 2021.

[25] Ayush Jain, Nikolaos Gkanatsios, Ishita Mediratta, and Katerina Fragkiadaki. Bottom up top down detection transformers for language grounding in images and point clouds. In *Proc. of the European Conference on Computer Vision (ECCV)*, 2022.

[26] Krishna Murthy Jatavallabhula, Alihusein Kuwajerwala, Qiao Gu, Mohd Omama, Tao Chen, Shuang Li, Ganesh Iyer, Soroush Saryazdi, Nikhil Keetha, Ayush Tewari, et al. ConceptFusion: Open-set multimodal 3D mapping. *arXiv preprint arXiv:2302.07241*, 2023.

[27] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International Conference on Machine Learning*, pages 4904–4916. PMLR, 2021.

[28] Li Jiang, Hengshuang Zhao, Shaoshuai Shi, Shu Liu, Chi-Wing Fu, and Jiaya Jia. PointGroup: Dual-set point grouping for 3D instance segmentation. In *Proc. of the Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4867–4876, 2020.

[29] Aishwarya Kamath, Mannat Singh, Yann LeCun, Gabriel Synnaeve, Ishan Misra, and Nicolas Carion. MDETR-modulated detection for end-to-end multi-modal understanding. In *Proc. of the International Conference on Computer Vision (ICCV)*, pages 1780–1790, 2021.

[30] Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. ReferItGame: Referring to objects in photographs of natural scenes. In *Proceedings of the conference on empirical methods in natural language processing (EMNLP)*, pages 787–798, 2014.

[31] Yongmin Kim, Chenhui Chu, and Sadao Kurohashi. Flexible visual grounding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 285–299, 2022.

[32] Juil Koo, Ian Huang, Panos Achlioptas, Leonidas J Guibas, and Minhyuk Sung. PartGlot: Learning shape part segmentation from language reference games. In *Proc. of the Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16505–16514, 2022.

[33] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123:32–73, 2017.

[34] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

[35] Weicheng Kuo, Fred Bertsch, Wei Li, AJ Piergiovanni, Mohammad Saffar, and Anelia Angelova. FindIt: Generalized localization with natural language queries. In *Proc. of the European Conference on Computer Vision (ECCV)*, 2022.

[36] Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, et al. Grounded language-image pre-training. In *Proc. of the Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10965–10975, 2022.

[37] Muchen Li and Leonid Sigal. Referring transformer: A one-step approach to multi-task visual grounding. *Advances in neural information processing systems*, 34:19652–19664, 2021.

[38] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[39] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. ViLBERT: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *Advances in neural information processing systems*, 32, 2019.

[40] Yuheng Lu, Chenfeng Xu, Xiaobao Wei, Xiaodong Xie, Masayoshi Tomizuka, Kurt Keutzer, and Shanghang Zhang. Open-vocabulary 3D detection via image-level class and debiased cross-modal contrastive learning. *arXiv preprint arXiv:2207.01987*, 2022.

[41] Junyu Luo, Jiahui Fu, Xianghao Kong, Chen Gao, Haibing Ren, Hao Shen, Huaxia Xia, and Si Liu. 3D-SPS: Single-stage 3D visual grounding via referred point progressive selection. In *Proc. of the Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16454–16463, 2022.

[42] Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L Yuille, and Kevin Murphy. Generation and comprehension of unambiguous object descriptions. In *Proc. of the Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11–20, 2016.

[43] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[44] Bryan A Plummer, Liwei Wang, Chris M Cervantes, Juan C Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *Proc. of the International Conference on Computer Vision (ICCV)*, pages 2641–2649, 2015.

[45] Bryan A Plummer, Arun Mallya, Christopher M Cervantes, Julia Hockenmaier, and Svetlana Lazebnik. Phrase localization and visual relationship detection with comprehensive image-language cues. In *Proc. of the International Conference on Computer Vision (ICCV)*, pages 1928–1937, 2017.

[46] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. PointNet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017.

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

[48] Nikhila Ravi, Jeremy Reizenstein, David Novotny, Taylor Gordon, Wan-Yen Lo, Justin Johnson, and Georgia Gkioxari. Accelerating 3D deep learning with PyTorch3D. *arXiv:2007.08501*, 2020.

[49] Junha Roh, Karthik Desingh, Ali Farhadi, and Dieter Fox. Languagerefer: Spatial-language model for 3d visual grounding. In *Conference on Robot Learning*, pages 1046–1056. PMLR, 2022.

[50] David Rozenberszki, Or Litany, and Angela Dai. Language-grounded indoor 3D semantic segmentation in the wild. In *Proc. of the European Conference on Computer Vision*

(ECCV), pages 125–141. Springer, 2022.

[51] Yue Ruan, Han-Hung Lee, Ke Zhang, and Angel X Chang. Tricolo: Trimodal contrastive loss for fine-grained text to shape retrieval. *arXiv preprint arXiv:2201.07366*, 2022.

[52] Nur Muhammad Mahi Shafiullah, Chris Paxton, Lerrel Pinto, Soumith Chintala, and Arthur Szlam. CLIP-fields: Weakly supervised semantic fields for robotic memory. *arXiv preprint arXiv:2210.05663*, 2022.

[53] Jesse Thomason, Mohit Shridhar, Yonatan Bisk, Chris Paxton, and Luke Zettlemoyer. Language grounding with 3D objects. In *Conference on Robot Learning*, pages 1691–1701. PMLR, 2022.

[54] Chenyun Wu, Zhe Lin, Scott Cohen, Trung Bui, and Subhransu Maji. PhraseCut: Language-based image segmentation in the wild. In *Proc. of the Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10216–10225, 2020.

[55] Yanmin Wu, Xinhua Cheng, Renrui Zhang, Zesen Cheng, and Jian Zhang. EDA: Explicit text-decoupling and dense alignment for 3D visual grounding. In *Proc. of the Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19231–19242, 2023.

[56] Le Xue, Mingfei Gao, Chen Xing, Roberto Martín-Martín, Jiajun Wu, Caiming Xiong, Ran Xu, Juan Carlos Niebles, and Silvio Savarese. ULIP: Learning unified representation of language, image and point cloud for 3D understanding. *arXiv preprint arXiv:2212.05171*, 2022.

[57] Zhengyuan Yang, Boqing Gong, Liwei Wang, Wenbing Huang, Dong Yu, and Jiebo Luo. A fast and accurate one-stage approach to visual grounding. In *Proc. of the International Conference on Computer Vision (ICCV)*, pages 4683–4693, 2019.

[58] Zhengyuan Yang, Songyang Zhang, Liwei Wang, and Jiebo Luo. SAT: 2D semantics assisted training for 3D visual grounding. In *Proc. of the International Conference on Computer Vision (ICCV)*, pages 1856–1866, 2021.

[59] Licheng Yu, Zhe Lin, Xiaohui Shen, Jimei Yang, Xin Lu, Mohit Bansal, and Tamara L Berg. MAttNet: Modular attention network for referring expression comprehension. In *Proc. of the Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1307–1315, 2018.

[60] Zhihao Yuan, Xu Yan, Yinghong Liao, Ruimao Zhang, Zhen Li, and Shuguang Cui. InstanceRefer: Cooperative holistic understanding for visual grounding on point clouds through instance multi-level contextual referring. In *Proc. of the International Conference on Computer Vision (ICCV)*, pages 1791–1800, 2021.

[61] Hanwang Zhang, Yulei Niu, and Shih-Fu Chang. Grounding referring expressions in images by variational context. In *Proc. of the Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4158–4166, 2018.

[62] Renrui Zhang, Ziyu Guo, Wei Zhang, Kunchang Li, Xupeng Miao, Bin Cui, Yu Qiao, Peng Gao, and Hongsheng Li. PointCLIP: Point cloud understanding by CLIP. In *Proc. of the Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8552–8562, 2022.

[63] Lichen Zhao, Daigang Cai, Lu Sheng, and Dong Xu. 3DVG-Transformer: Relation modeling for visual grounding on point clouds. In *Proc. of the International Conference on Computer Vision (ICCV)*, pages 2928–2937, 2021.

[64] Chaoyang Zhu, Yiyi Zhou, Yunhang Shen, Gen Luo, Xingjia Pan, Mingbao Lin, Chao Chen, Liujuan Cao, Xiaoshuai Sun, and Rongrong Ji. SeqTR: A simple yet universal network for visual grounding. In *Proc. of the European Conference on Computer Vision (ECCV)*, pages 598–615. Springer, 2022.