

# Distilling Coarse-to-Fine Semantic Matching Knowledge for Weakly Supervised 3D Visual Grounding

Zehan Wang\* Haifeng Huang\* Yang Zhao Linjun Li Xize Cheng  
Yichen Zhu Aoxiong Yin Zhou Zhao†  
Zhejiang University

{wangzehan01, huanghaifeng, zhaozhou}@zju.edu.cn

## Abstract

3D visual grounding involves finding a target object in a 3D scene that corresponds to a given sentence query. Although many approaches have been proposed and achieved impressive performance, they all require dense object-sentence pair annotations in 3D point clouds, which are both time-consuming and expensive. To address the problem that fine-grained annotated data is difficult to obtain, we propose to leverage weakly supervised annotations to learn the 3D visual grounding model, i.e., only coarse scene-sentence correspondences are used to learn object-sentence links. To accomplish this, we design a novel semantic matching model that analyzes the semantic similarity between object proposals and sentences in a coarse-to-fine manner. Specifically, we first extract object proposals and coarsely select the top- $K$  candidates based on feature and class similarity matrices. Next, we reconstruct the masked keywords of the sentence using each candidate one by one, and the reconstructed accuracy finely reflects the semantic similarity of each candidate to the query. Additionally, we distill the coarse-to-fine semantic matching knowledge into a typical two-stage 3D visual grounding model, which reduces inference costs and improves performance by taking full advantage of the well-studied structure of the existing architectures. We conduct extensive experiments on ScanRefer, Nr3D, and Sr3D, which demonstrate the effectiveness of our proposed method.

## 1. Introduction

3D Visual grounding (3DVG) refers to the process of localizing an object in a scene based on a natural language sentence. The 3DVG task has recently gained attention due to its numerous applications. Despite the significant progress made in this area [3, 4, 39, 40, 17, 37], all these

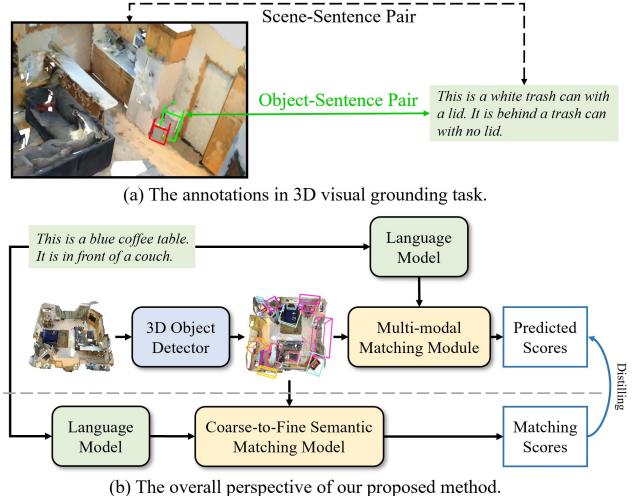


Figure 1. (a). 3D visual grounding aims to find the object-sentence pair from the whole scene. The fully supervised setting requires all the dense ground-truth object-sentence labels for training, while the weakly supervised method only needs the coarse scene-sentence annotations. (b). Coarse-to-Fine Semantic Matching Model (bottom) analyzes the matching score of each proposal to the sentence, and the semantic matching knowledge is distilled to the two-stage 3DVG architecture (upper).

approaches require bounding box annotations for each sentence query, which are laborious and expensive to obtain. For example, it takes an average of 22.3 minutes to annotate a scene in the ScanNet-v2 dataset [6]. Thus, we focus on weakly supervised training for 3DVG, which only requires scene-sentence pairs for training. This problem is meaningful and realistic since obtaining scene-level labels is much easier and can be scaled effectively.

However, weakly supervised 3DVG poses two challenges. Firstly, a 3D point cloud can contain numerous objects of various categories, and a sentence query may contain multiple objects besides the target object to aid in localization. Without knowledge of the ground-truth object-sentence pair, it is difficult to learn to link the sentence to

\*Equal contribution

†Corresponding author

its corresponding object from the enormous number of possible object-sentence pairs. Secondly, the 3DVG task often involves multiple interfering objects in the scene with the same class as the target object, and the target object must be distinguished based on its object attributes and the relations between objects described in the given sentence. As illustrated in Figure 1 (a), there are two trash cans in the scene, and the described target object can only be identified by fully comprehending the language description.

To address both challenges simultaneously, we propose a coarse-to-fine semantic matching model to measure the similarity between object proposals and sentences. Specifically, our model generates object-sentence matching scores from scene-sentence annotation, guided by coarse-to-fine semantic similarity analysis. Firstly, we calculate the object category similarity and feature similarity between all the proposals and the sentence. Combining these two similarities, we roughly select  $K$  proposals with the highest similarity to the sentence, which can effectively filter out the proposals that do not belong to the target category. Secondly, we utilize NLTK [2] to conduct part-of-speech tagging on the sentences and randomly mask the more meaningful nouns and adjectives words. The selected candidates would be used to reconstruct the masked keywords of the sentence, which can help the model fully and deeply understand the whole sentence. Since the target object and the sentence query are semantically consistent, the more the candidate and the target object overlap, the more accurate its predicted keywords will be. The object-sentence matching score of each candidate can be measured by its reconstruction loss. Eventually, in order to reduce inference time and make full use of the structure of existing 3DVG models, we utilize knowledge distillation [15] to migrate the knowledge of the coarse-to-fine semantic matching model to a typical two-stage 3DVG model, where the distilled pseudo labels are generated by the object-sentence matching scores.

In summary, the key contribution is four-fold:

- To the best of our knowledge, this paper is the first work to address weakly supervised 3DVG, which eliminates the need for expensive and time-consuming dense object-sentence annotations and instead requires only scene-sentence level labels.
- We approach weakly supervised 3DVG as a coarse-to-fine semantic matching problem and propose a coarse-to-fine semantic matching model to analyze the similarity between each proposal and the sentence.
- We distill the knowledge of the coarse-to-fine semantic matching model into a two-stage 3DVG model, which fully leverages the well-studied network structure design, leading to improved performance and reduced inference costs.

- Experiments conducted on three wide-used datasets ScanRefer [4], Nr3D [1] and Sr3D [1] demonstrate the effectiveness of our method.

## 2. Related Work

**Supervised 3D Visual Grounding.** Grounding a sentence query in a 3D point cloud is a fundamental problem in vision-language tasks, with wide-ranging applications in fields like automatic robotics [35, 34, 25, 11] and AR/VR/metaverse [26, 9]. The ScanRefer [4] and Referit3D [1] datasets annotate dense object-sentence links on the widely-used 3D point cloud dataset ScanNet [6].

Most recent 3D visual grounding methods [3, 39, 40, 17, 16, 37] follow a two-stage pipeline. In the first stage, pre-trained 3D object detectors [29, 22] generate 3D object proposals. The second stage involves matching the selected object proposals with the sentence query. Existing two-stage methods improve performance by exploring the object attributes and relations between proposals in the second stage. For example, 3DVG-Transformer [40] uses a coordinate-guided contextual aggregation module to capture relations between proposals and a multiplex attention module to distinguish the target object. TransRefer3D [13] uses an entity-aware attention module and a relation-aware attention module for fine-grained cross-modal matching. 3DJCG [3] devises a joint framework for 3D visual grounding [4] and 3D dense captioning [5] tasks, and their experiments demonstrate that extra caption-level data can improve the performance of 3D visual grounding.

In contrast to these supervised methods, our approach learns to localize target objects in 3D space using only caption-level annotations.

**Weakly Supervised Image Grounding.** The image grounding task, similar to 3DVG, aims to identify objects in an image based on a sentence, and has a wide range of applications [28, 20, 8, 38, 19, 36]. Weakly supervised image grounding, which requires only images and corresponding sentences in the training phase, has gained popularity due to the low cost of annotation [12, 31, 33, 10, 7].

Weakly supervised image grounding is typically treated as a Multiple Instance Learning (MIL) problem [18, 24], where the image is represented as a bag of regions, generated by a pre-trained image object detector. Image-sentence matching scores are calculated based on region-phrase similarity scores, and ground-truth image-sentence links are used to supervise these scores. For example, ARN [21] pairs image proposals and queries based on subject, location, and context information through adaptive grounding and collaborative reconstruction. InfoGround [12] proposes a contrastive learning objective function [14] to optimize image-sentence scores. Wang et al. [33] use a pre-trained image object detector to generate pseudo category labels for all regions, achieving region-phrase alignment by distilling

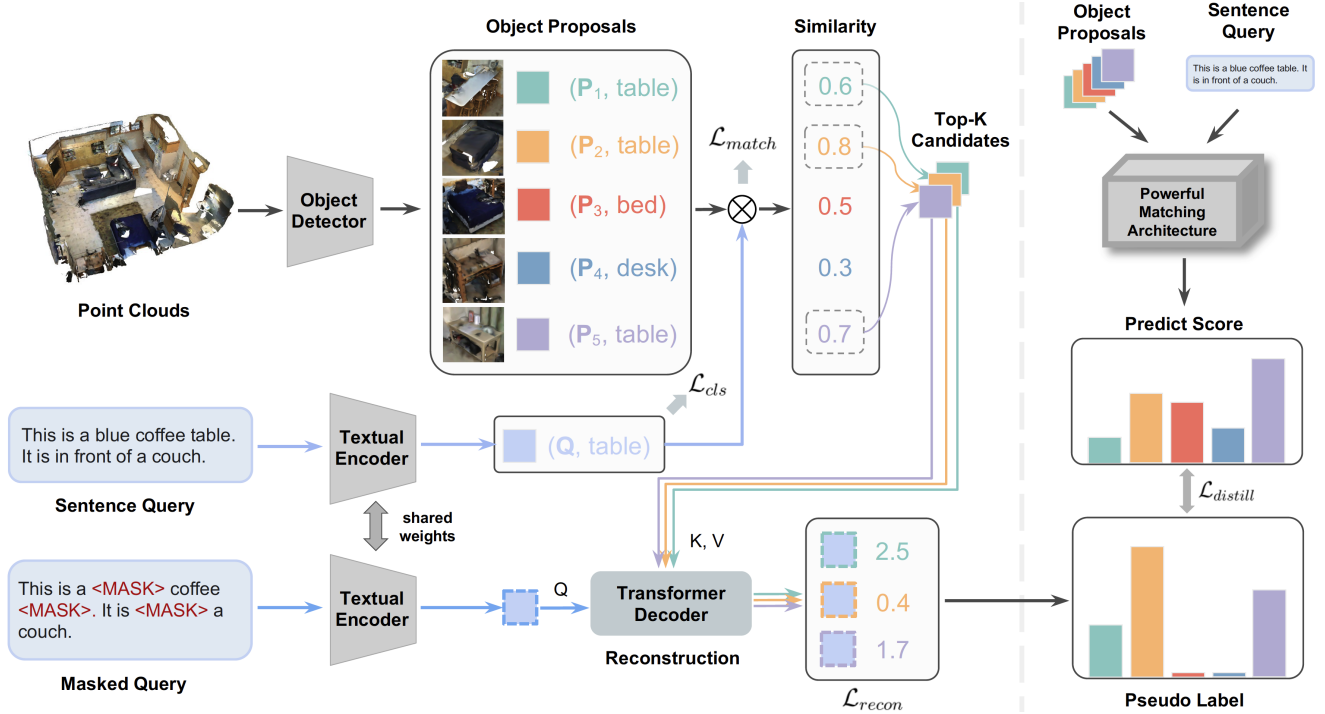


Figure 2. Overall architecture diagram of our model. The model is based on a two-stage grounding pipeline. We first extract object proposals by pre-trained object detector. Then, we propose a coarse-to-fine semantic matching process to find the matched object-query pair. Furthermore, we distill the semantic matching knowledge into an effective matching architecture to enhance the inference efficiency.

knowledge from these pseudo labels.

However, MIL-based weakly supervised image grounding methods cannot solve the weakly supervised problem in 3DVG. Firstly, the presence of numerous different objects in a single 3D scene makes it difficult to learn a stable MIL classifier. Secondly, while image grounding aims to locate objects corresponding to all phrases in the sentence, 3DVG requires the identification of a single target object, necessitating a deeper and more comprehensive understanding of the sentence’s semantic information, rather than just its phrases.

### 3. Method

#### 3.1. Problem Formulation

In this paper, we address the problem of weakly-supervised 3DVG. The input point cloud  $\mathbf{P} = \{\mathbf{p}_i\}_{i=1}^{N_p}$  contains point coordinates in 3D space, represented by  $\mathbf{p}_i \in \mathbb{R}^3$ . Correspondingly, a sentence query  $\mathbf{Q} = \{\mathbf{q}_i\}_{i=1}^{N_q}$  is given to describe the object of interest. The objective of our model is to predict a 3D bounding box  $\mathbf{B} = (\mathbf{c}, \mathbf{r})$  that encompasses the object, where  $\mathbf{c} = (c_x, c_y, c_z)$  represents the center of the box, and  $\mathbf{r} = (r_x, r_y, r_z)$  represents the dimensions of the box. The number of input points and sentence length is denoted by  $N_p$  and  $N_q$ , respectively. In the weakly-supervised setting, there are no bounding box anno-

tations available during training.

#### 3.2. Overview

As depicted in Figure 2, our model utilizes a two-stage grounding pipeline. In the first stage, we employ a pre-trained 3D object detector to extract  $M_p$  object proposals from the given point cloud. In the second stage, we propose a coarse-to-fine semantic matching process to evaluate the semantic similarity between each proposal and the sentence query. Specifically, the coarse-to-fine process comprises two steps. Firstly, we coarsely extract the top  $K$  object proposals, which are referred to as candidates, by computing the object-sentence similarity matrix between all proposals and the sentence query. Secondly, we generate a more accurate pseudo label by considering the semantic reconstruction result of each candidate-sentence pair. Further details will be explained in Section 3.3 and Section 3.4.

Moreover, for reducing the inference costs and further enhancing the performance, we propose to distill the semantic matching knowledge into a supervised 3DVG pipeline as elaborated in Section 3.5. Most advanced fully-supervised models typically operate using a “detection-and-matching” paradigm. This means that these powerful matching architectures can be used as plug-and-play modules to incorporate knowledge learned from weak supervision.

### 3.3. Coarse-grained Candidate Selection

**Object-Sentence Similarity.** Although we have extracted numerous high-quality object proposals from the pre-trained 3D object detector, identifying the best-matched proposal with the sentence query is still challenging. This is because a 3D scene may contain many different classes of objects, and the semantic spaces between objects and the sentence are not aligned. To overcome this challenge, we propose calculating a similarity matrix between the objects and the sentence based on both class and feature levels.

For the class level, we can obtain the object class from the pre-trained 3D object detector and the text class from a text classifier. For simplicity, we choose to train the text classifier from scratch and the classification loss  $\mathcal{L}_{cls}$  is a simple cross-entropy loss. Considering that the object detector might be pre-trained on another dataset, the object class set and the text class set may be inconsistent. Therefore, before directly comparing the object proposals and the sentence, we need to transfer the object class prediction to the target text class. To achieve this, we propose using a class transform matrix  $\mathbf{M}^c \in \mathbb{R}^{N_o^c \times N_q^c}$  for class alignment. The matrix is based on the cosine similarity between the GloVe embeddings of different class names. Here,  $N_o^c$  and  $N_q^c$  denote the number of object classes and the number of words in the sentence query, respectively.

For the feature level, we align the feature representations of the objects and the sentence query using a contrastive learning approach. Specifically, we pull the positive object-query pairs in the same scene closer and push the negative pairs further apart in the semantic space. To achieve this, all the object-query pairs in the same scene are considered as positive pairs  $\mathbb{P}$ , while those from different scenes are considered as negative pairs  $\mathbb{N}$ . The feature matching loss for object-sentence feature alignment can be computed by

$$\mathcal{L}_{match} = -\log \left( \frac{\sum_{(\mathbf{p}, \mathbf{q}) \in \mathbb{P}} e^{\phi(\mathbf{p}, \mathbf{q})}}{\sum_{(\mathbf{p}, \mathbf{q}) \in \mathbb{P}} e^{\phi(\mathbf{p}, \mathbf{q})} + \sum_{(\mathbf{p}', \mathbf{q}) \in \mathbb{N}} e^{\phi(\mathbf{p}', \mathbf{q})}} \right), \quad (1)$$

where  $\mathbf{p}$  represents an object proposal and  $\mathbf{q}$  a sentence query.  $\phi$  is the feature similarity function, which is a dot product in our practice.

We get the object-sentence similarity  $\hat{\mathbf{s}} \in \mathbb{R}^{M_p}$  by

$$\hat{\mathbf{s}} = \phi(\tilde{\mathbf{P}}^c \mathbf{M}^c, \tilde{\mathbf{Q}}^c) + \phi(\tilde{\mathbf{P}}, \tilde{\mathbf{Q}}), \quad (2)$$

where  $\tilde{\mathbf{P}} \in \mathbb{R}^{M_p \times d}$  /  $\tilde{\mathbf{Q}} \in \mathbb{R}^{N_q \times d}$  is the encoded object/sentence feature, and  $\tilde{\mathbf{P}}^c \in \mathbb{R}^{N_o^c}$  /  $\tilde{\mathbf{Q}}^c \in \mathbb{R}^{N_q^c}$  is the object/sentence class prediction.  $\phi$  is a similarity function (e.g., cosine similarity or dot product).  $M_p$  is the number of object proposals.  $d$  is the hidden dimension.

**Top-K Selection.** According to the object-sentence similarity, we coarsely select the top  $K$  candidates  $\tilde{\mathbf{C}} \in \mathbb{R}^{K \times d}$  out of the  $M_p$  proposals  $\tilde{\mathbf{P}} \in \mathbb{R}^{M_p \times d}$ , which can effectively filter out proposals that are significantly different from the semantics of the sentence.

### 3.4. Fine-grained Semantic Matching

Given the  $K$  object candidates, we propose a semantic reconstruction module to measure fine-grained semantic similarity between the objects and the sentence query.

As depicted in Figure 2, we mask important words in the sentence query, such as the target object (*table*), its attribute (*blue*), and its relation to other objects (*in front of*) in the scene. We reconstruct the masked words with the assistance of each candidate, respectively. The candidate that provides the most useful semantic information to predict the keywords and contains the least amount of noise is expected to be the best match.

We encode the masked sentence query using a textual encoder, denoted as  $\tilde{\mathbf{Q}}^m \in \mathbb{R}^{N_q \times d}$ . For the  $k$ -th candidate  $\tilde{\mathbf{c}}^k \in \mathbb{R}^d$ , we obtain the cross-modal semantic representation  $\mathbf{f}^k = \{\mathbf{f}_i^k\}_{i=1}^{N_q} \in \mathbb{R}^{N_q \times d}$  by a transformer decoder

$$\mathbf{f}^k = \text{Dec}(\tilde{\mathbf{Q}}^m, \tilde{\mathbf{c}}^k). \quad (3)$$

To predict the masked words, we compute the energy distribution  $\mathbf{e}^k = \{\mathbf{e}_i^k\}_{i=1}^{N_q} \in \mathbb{R}^{N_q \times d}$  over the vocabulary by

$$\mathbf{e}_i^k = \mathbf{W} \mathbf{f}_i^k + \mathbf{b}, \quad (4)$$

where  $\mathbf{e}_i^k \in \mathbb{R}^{N_v}$  represents the energy distribution of the  $i$ -th predicted word, and  $N_v$  is the number of words in the vocabulary.  $\mathbf{W} \in \mathbb{R}^{N_v \times d}$  and  $\mathbf{b} \in \mathbb{R}^{N_v}$  are learnable parameters of a fully-connected layer.

Then, we use a reconstruction loss to train the semantic reconstruction module to effectively learn key information from the object context and predict the masked words. Specifically, the reconstruction can be computed as

$$\mathcal{L}_{recon}^k = - \sum_{i \in N_{mask}} \log p(\mathbf{q}_i | \mathbf{e}_i^k), \quad (5)$$

where  $N_{mask}$  represents positions of masked words in the query and  $\mathcal{L}_{recon}^k$  is the reconstruction loss for the  $k$ -th candidate  $\tilde{\mathbf{c}}^k$ . Then the total loss for all the  $K$  candidates is  $\mathcal{L}_{recon} = \sum_{k=1}^K \mathcal{L}_{recon}^k$ .

### 3.5. Knowledge Distillation

As mentioned earlier, a lower reconstruction loss indicates that the object candidate provides more consistent semantic information. A direct approach for object prediction is to select the candidate with the lowest reconstruction loss, as it is likely to be the best match. However, this coarse-to-fine matching process is computationally expensive during



Table 1. Performance comparison on ScanRefer. “SUN” and “SCAN” denotes that the 3D object detector is pretrained on SUN RGB-D[32] or ScanNet[6], respectively. For the “ $R@n, IoU@m$ ” metric,  $n \in \{1, 3\}$  and  $m \in \{0.25, 0.5\}$ .

	Method	R@3						R@1	
		Unique		Multiple		Overall		Overall	
		$m=0.25$	$m=0.5$	$m=0.25$	$m=0.5$	$m=0.25$	$m=0.5$	$m=0.25$	$m=0.5$
SUN	Upper Bound	57.07	35.28	55.30	35.29	55.65	35.29	-	-
	Random	15.88	6.99	7.38	3.28	9.03	3.96	3.66	1.37
	MIL-Margin [10]	19.94	10.51	10.18	3.60	12.07	4.94	6.80	2.37
	MIL-NCE [12]	19.13	10.95	7.57	3.56	9.81	5.00	5.64	2.69
	<b>Ours</b>	<b>24.07</b>	<b>18.05</b>	<b>12.54</b>	<b>7.50</b>	<b>14.78</b>	<b>9.55</b>	<b>10.43</b>	<b>6.37</b>
SCAN	Upper Bound	93.82	77.02	72.61	58.01	76.72	61.70	-	-
	Random	21.36	14.25	10.10	7.15	12.28	8.53	4.74	3.32
	MIL-Margin [10]	29.54	22.49	11.48	8.04	14.99	10.84	8.16	5.66
	MIL-NCE [12]	48.94	40.76	17.41	13.73	23.53	18.97	18.95	14.06
	<b>Ours</b>	<b>70.84</b>	<b>58.21</b>	<b>25.28</b>	<b>20.68</b>	<b>34.12</b>	<b>27.97</b>	<b>27.37</b>	<b>21.96</b>

inference and not explicitly optimized for grounding tasks. To tackle the issues, we propose to distill the coarse-to-fine semantic matching knowledge into a supervised 3DVG pipeline. Our approach offers multiple benefits, including reduced inference costs and the ability to capitalize on more powerful 3DVG architectures and established learning objectives tailored for 3DVG tasks. By incorporating knowledge distillation, our framework can be integrated with any advanced supervised 3DVG pipeline, enhancing the flexibility and practicality of our method.

For candidates, we calculate the reward according to their rank of  $\mathcal{L}_{recon}^k$ . The reward is reduced from one to zero, under the assumption that lower reconstruction loss gets better reward. The distilled pseudo labels  $\mathbf{d} = \{d_1, \dots, d_{M_p}\}$  can be generated by filling the rewards of candidates to their original indices and padding the non-candidate indices with zeros, following by a SoftMax operation. After all, we distill the knowledge by aligning the predict scores  $\mathbf{s} = \{s_1, \dots, s_{M_p}\}$  to the pseudo labels, where the predict scores are obtained from the powerful matching architecture. The distillation loss is:

$$\mathcal{L}_{distill} = - \sum_{i=1}^{M_p} d_i \log \left( \frac{\exp(s_i)}{\sum_{j=1}^{M_p} \exp(s_j)} \right). \quad (6)$$

### 3.6. Training and Inference

**Multi-Task Loss** We train the model end-to-end via a multi-task loss function, formulated by

$$\mathcal{L}_{overall} = \mathcal{L}_{distill} + \lambda_1 \mathcal{L}_{cls} + \lambda_2 \mathcal{L}_{match} + \lambda_3 \mathcal{L}_{recon} \quad (7)$$

where  $\lambda_1$ ,  $\lambda_2$  and  $\lambda_3$  are hyper-parameters to balance four parts of the loss function.

**Inference.** Thanks to the knowledge distillation, all we need in the inference phase is the two-stage 3DVG pipeline.

We get the predict score  $\mathbf{s} \in \mathbb{R}^{M_p}$  from the matching architecture, and the index of the predicted best-match proposal is  $\text{argmax}(\mathbf{s})$ . Then, we obtain the corresponding 3D bounding box of this object proposal.

## 4. Experiments

### 4.1. Datasets

**ScanRefer.** The ScanRefer [4] dataset contain 51,583 descriptions of 11,046 objects from 800 ScanNet [6] scenes. On average, each scene has 64.48 sentences and 13.81 objects. The data can be divided into “Unique” and “Multiple”, depending on whether there are multiple objects of the same category as the target in the scene.

**Nr3D/Sr3D.** The Nr3D/Sr3D dataset [1] is also based on the 3D scene dataset ScanNet [6]. Nr3D contains 41,503 human utterances collected by ReferItGame, and Sr3D contains 83,572 sentences automatically generated based on a “target-spatial relationship-anchor object” template. Similar to the definition of “Unique” and “Multiple” in ScanRefer, Nr3D/Sr3D can be split into “easy” and “hard” subsets. The “view-dep.” and “view-indep.” subsets depend on whether the description is related to the speaker’s view.<sup>1</sup>

### 4.2. Evaluation Metric.

To evaluate the performance of our method and baselines on these three datasets, we adopt the “ $R@n, IoU@m$ ” metric. Specifically, this metric represents the percentage of at least one of the top- $n$  predicted proposals having an IoU

<sup>1</sup>In the Nr3D/Sr3D datasets, the supervised task involves selecting the correct matching 3D box from a set of given boxes, with the instance matching accuracy serving as the evaluation metric. However, in the weakly-supervised setting, we predict the boxes from scratch and assess the IoU metrics, which cannot be directly compared to the results of supervised methods.

Table 2. Performance comparison on Nr3D and Sr3D dataset. ‘‘SUN’’ and ‘‘SCAN’’ denotes that the 3D object detector is pretrained on SUN RGB-D [32] or ScanNet [6], respectively. For the ‘‘ $R@n, IoU@m$ ’’ metric,  $n = 3$  and  $m \in \{0.25, 0.5\}$ .

	Method	Easy		Hard		View-dep.		View-indep.		Overall	
		$m=0.25$	$m=0.5$	$m=0.25$	$m=0.5$	$m=0.25$	$m=0.5$	$m=0.25$	$m=0.5$	$m=0.25$	$m=0.5$
Nr3D											
SUN	Upper Bound	40.24	24.62	40.62	23.80	40.66	24.88	40.32	23.82	40.44	24.20
	Random	6.70	2.40	6.34	2.75	6.59	2.91	6.47	2.41	6.51	2.59
	MIL-Margin [10]	9.93	5.63	7.79	4.03	8.71	4.77	8.88	4.81	8.82	4.80
	MIL-NCE [12]	9.93	5.42	7.77	4.79	8.45	4.67	9.00	5.32	8.81	5.09
	Ours	10.93	6.36	9.83	6.18	10.77	6.53	10.13	6.13	10.36	6.27
SCAN	Upper Bound	62.43	44.75	58.98	44.18	59.15	42.91	61.44	45.29	60.64	44.45
	Random	8.81	5.66	7.57	4.97	7.28	4.80	8.65	5.61	8.17	5.30
	MIL-Margin [10]	14.25	10.64	9.79	7.68	10.64	8.35	12.63	9.50	11.93	9.10
	MIL-NCE [12]	17.29	13.53	9.61	7.59	11.96	9.44	14.01	10.98	13.29	10.44
	Ours	27.29	21.10	17.98	14.42	21.60	16.80	22.91	18.07	22.45	17.62
Sr3D											
SUN	Upper Bound	39.22	23.69	39.58	21.83	25.93	13.30	39.92	23.24	39.33	22.82
	Random	6.53	2.28	4.61	2.17	1.86	0.80	6.05	2.32	5.96	2.25
	MIL-Margin [10]	8.52	4.84	5.66	3.98	3.19	2.66	7.86	4.67	7.67	4.59
	MIL-NCE [12]	8.66	4.92	4.10	2.78	2.46	0.93	7.56	4.42	7.30	4.28
	Ours	10.31	6.60	8.57	6.23	4.19	1.86	10.09	6.69	9.79	6.49
SCAN	Upper Bound	65.42	46.75	58.46	42.69	53.59	34.84	63.77	46.01	63.34	45.54
	Random	8.50	5.38	6.85	4.55	5.59	3.72	8.12	5.20	8.01	5.13
	MIL-Margin [10]	12.55	9.82	9.59	7.50	9.57	7.98	11.76	9.18	11.67	9.13
	MIL-NCE [12]	17.45	12.51	9.61	7.14	12.37	7.97	15.22	11.03	15.11	10.90
	Ours	29.40	24.87	21.00	17.47	20.21	17.15	27.19	22.90	26.89	22.66

greater than  $m$  when compared to the ground-truth target bounding box. In our setting,  $n \in 1, 3$  and  $m \in 0.25, 0.5$ .

### 4.3. Implementation Details.

In our practice, we use the pretrained GroupFree model [22] as our 3D object detector and distill the learned semantic matching knowledge to the matching architecture proposed in 3DJCG [3]. The input point number  $N_p$ , the proposal number  $M_p$ , and the candidate number  $K$  are set to 50000, 256 and 8, respectively. More details can be found in the supplementary material.

### 4.4. Compared Methods

**Random.** We randomly select a candidate from all the proposals as the predicted result.

**MIL-Margin.** The MIL-Margin method [10] proposes a max margin loss to enforce the score between a sentence and a paired scene to be higher than non-paired scenes, and vice versa.

**MIL-NCE.** The MIL-NCE method [12] maximizes the InfoNCE lower bound on mutual information between the sentence and proposals from the paired scene, compared to non-corresponding pairs of scenes and sentences.

**Upper Bound.** The quality of the bounding boxes generated by the 3D object detector determines the upper bound performance of our model. We consider the maximum IoU

between all the  $M_p$  object proposals and the ground-truth bounding box as the upper bound.

### 4.5. Quantitative Comparison

The performance results of our methods and baselines on ScanRefer and Nr3D/Sr3D are reported in Table 1 and Table 2, respectively, with the best results highlighted in **bold**. The comparison to supervised methods is presented in Table 3. Although the 3D object detector pre-trained on ScanNet implicitly utilizes ground truth boxes on ScanNet, the object-sentence annotations are still unseen, and pre-training on ScanNet is only used to obtain more accurate proposals. To fully avoid annotations in ScanNet, we also evaluate results using a detector pre-trained on SUN RGB-D [32]. Despite the degradation caused by out-of-domain data, our method still shows significant improvement over baselines. By analyzing the evaluation results, we can observe the following facts:

- Our method achieves significant improvements over the Random method on all datasets, indicating the effectiveness of the coarse-to-fine semantic matching model in analyzing the similarity between objects and sentences when true object-sentence pairs are unavailable.
- The results show that our method outperforms widely

Table 3. Comparison to supervised methods on ScanRefer.

Method	Backbone	R@1	
		m=0.25	m=0.5
ScanRefer [4]	VoteNet	41.19	27.40
SAT [37]	VoteNet	44.54	30.14
3DVG-Transformer [40]	VoteNet	47.57	34.67
3DJCG [3]	VoteNet	49.56	37.33
Ours	VoteNet	25.87	16.63
Ours	GroupFree	<b>27.37</b>	<b>21.96</b>

Table 4. Ablation studies on the coarse-to-fine semantic matching model. The experiments of all the ablation study are conducted on ScanRefer dataset. “R@3, SUN” refers to  $n = 3$  and the object detector is pretrained on SUN RGB-D.

$\mathcal{L}_{cls}$	$\mathcal{L}_{match}$	$\mathcal{L}_{recon}$	R@3, SUN		R@3, SCAN	
			m=0.25	m=0.5	m=0.25	m=0.5
			9.03	3.96	12.28	8.53
	✓		10.53	6.29	17.41	14.28
✓			13.10	8.17	32.71	25.90
✓	✓		13.76	8.48	33.03	26.58
✓	✓	✓	<b>14.78</b>	<b>9.55</b>	<b>34.12</b>	<b>27.97</b>

used MIL-based weakly supervised methods by a large margin, and even approaches the upper bound in the “Unique” subset of ScanRefer. This suggests that our proposed model can deeply exploit the alignment relationship between 3D scenes and sentences and identify the most semantically relevant object proposals.

- Our coarse-to-fine semantic matching model significantly improves performance in the challenging “Multiple” subset of ScanRefer and “Hard” subset of Nr3D/Sr3D, where there are multiple interfering objects with the same category as the target object. This problem requires a comprehensive understanding of the sentence to distinguish the described object, which our model handles efficiently with the keywords semantic reconstruction module.
- The performance improvement with the SUN RGB-D pre-trained backbone is relatively small on Nr3D and Sr3D datasets, possibly because the target objects are inherently more challenging to detect, and the pre-trained detector performs poorly due to out-of-distribution data. The low grounding upper bound and inaccurate proposals make the training phase unstable. Nevertheless, our method outperforms all baselines, and when the detector is more reliable, our semantic matching model shows much more significant advantages on Nr3D/Sr3D.

#### 4.6. Ablation Study

To further assess the effectiveness of each component, we conduct ablation studies on the ScanRefer dataset.

Table 5. Ablation study on the candidate number  $K$ .

$K$	R@3, SUN		R@3, SCAN	
	m=0.25	m=0.5	m=0.25	m=0.5
4	14.04	9.05	33.68	27.50
<b>8</b>	<b>14.78</b>	<b>9.55</b>	<b>34.12</b>	<b>27.97</b>
16	14.37	9.39	33.23	26.99
32	14.26	9.21	31.29	25.90

Table 6. Ablation study on the knowledge distillation. The matching time is evaluated for one batch.

Distill Target	R@3, SUN		R@3, SCAN		Matching Phase	
	m=0.25	m=0.5	m=0.25	m=0.5	Time	Params
w/o distill.	13.88	9.09	31.71	26.38	31.4 ms	5.85 M
SAT [37]	14.00	9.15	33.70	27.85	6.78 ms	1.85 M
3DVG-Trans [40]	14.01	9.10	<b>34.61</b>	<b>28.22</b>	8.38 ms	1.91 M
3DJCG [3]	<b>14.78</b>	<b>9.55</b>	34.12	27.97	8.22 ms	1.93 M

##### 4.6.1 Effectiveness of Semantic Matching Model

**Coarse-to-Fine Matching Scores.** We aim to examine the effect of each module in the coarse-to-fine semantic matching model.  $\mathcal{L}_{cls}$  and  $\mathcal{L}_{match}$  denote whether to use class similarity and feature similarity for coarsely selecting the top- $K$  candidates, respectively.  $\mathcal{L}_{recon}$  represents using the reconstruct module to finely generate distilled pseudo labels for the selected  $K$  candidates. If  $\mathcal{L}_{recon}$  is not used, all the selected  $K$  candidates’s rewards are set directly to 1. Table 4 shows that using  $\mathcal{L}_{cls}$  or  $\mathcal{L}_{match}$  alone can effectively aid the model in learning object-sentence pairs from the caption-level annotations, while joint usage of  $\mathcal{L}_{cls}$  and  $\mathcal{L}_{match}$  leads to better performance. Furthermore, the last two rows suggest that the fine-grained semantic matching module can effectively and comprehensively analyze the semantic similarity between the selected  $K$  candidates and the sentence query, and further enhance the performance.

**Number of Coarse-grain Candidate.** We analyze the performance for varying numbers of coarse-grained candidates,  $K \in \{4, 8, 16, 32\}$ . As shown in Table 5, we observe that selecting 8 candidates yields the best results for fine-grained semantic matching. We tentatively infer the reason is that too small  $K$  leaves out the possible proposal that covers the target object, while too large  $K$  leads to selecting many proposals that are not relevant to the description due to the numerous objects in a 3D scene.

##### 4.6.2 Effectiveness of Knowledge Distillation

To investigate the effect of the semantic distillation in terms of performance and efficiency, we construct the baseline without knowledge distillation that removes the distillation loss  $\mathcal{L}_{distill}$  during training and directly uses the coarse-to-fine semantic matching model for inference. As shown in Table 6, distilling the semantic matching knowledge into the matching module of a two-stage 3DVG model brings

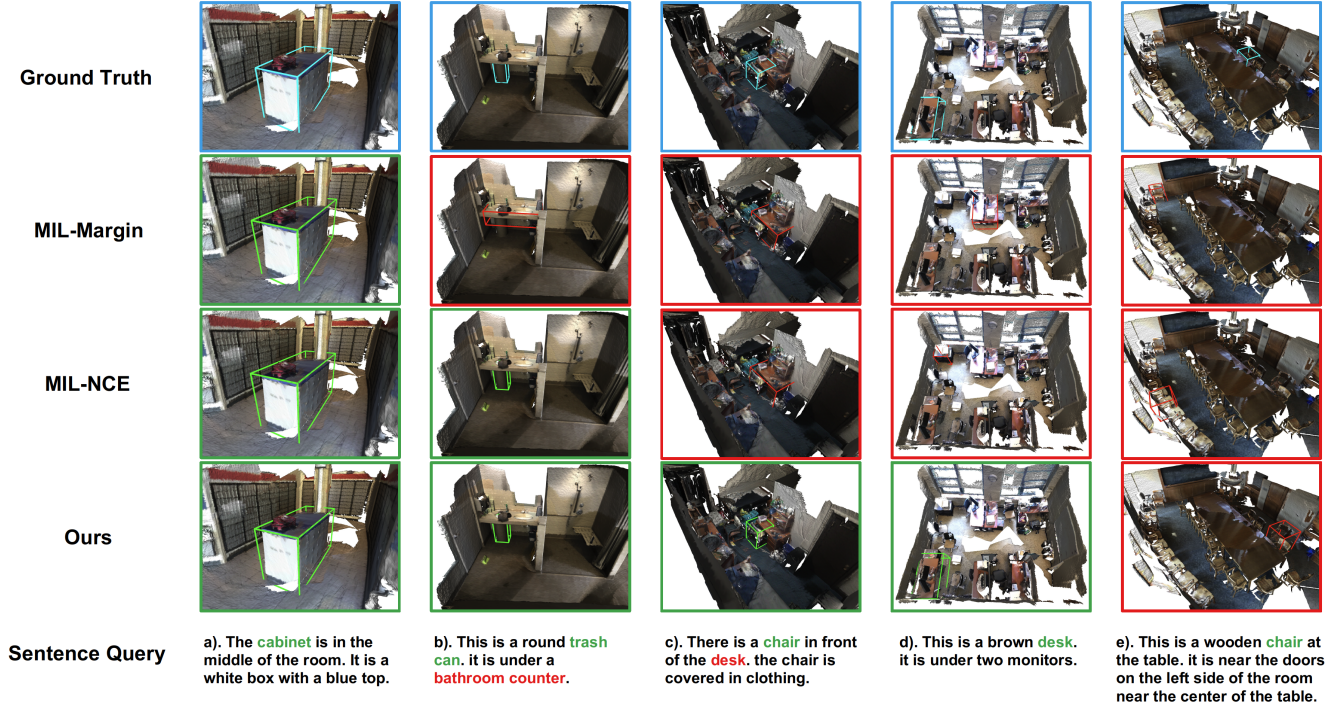


Figure 3. Qualitative Comparison between MIL-based methods and Ours.

a significant performance improvement. The well-studied structure of the existing 3DVG model enhances the generalization ability of our method. As for the efficiency, we observe that the distilled matching module is  $3\times$  smaller and  $4\times$  faster than the coarse-to-fine semantic matching model, demonstrating that the distilling operation reduces inference costs significantly. Meanwhile, we try to distill the knowledge into the matching modules of different supervised methods (SAT [37], 3DVG-Transformer [40], and 3DJCG [3]), the results show that the distillation fits well to different architectures.

#### 4.7. Qualitative Comparison

As depicted in Figure 3, we visualize the predicted bounding boxes in the corresponding 3D scene, where the green box denotes that the method predicts the correct object ( $\text{IoU} \geq 0.5$  with the true box), and the red box indicates a wrong prediction.

In case (a), the target object *cabinet* is the only cabinet in the simple scene. So, both MIL-based methods and our method can predict the box well. In case (b) / (c), the target object is a *trash can* / *chair*. MIL-based methods may be misled by the presence of another object (*bathroom counter* / *desk*) in the sentence. While our method can filter out the objects that do not belong to the target category, benefiting from the coarse-grained candidate selection module. In case (d), there are six *desks* in the scene. The MIL-based methods fail to localize the correct object, even though they

figure out the target object (category) is *desk*. With the fine-grained semantic matching module, our methods can better differentiate among these six *desks* and choose the one best-matched to the sentence (“brown” and “under two monitors”). In case (e), the scene contains 32 different *chairs*. Unfortunately, both our method and MIL-based methods fail in this case. However, we consider our method’s predicted result acceptable. Firstly, the sentence query’s expressions, such as “near the doors” and “near the center”, are ambiguous and cannot give a precise location of the target object. Secondly, our method’s predicted *chair* is also consistent with the sentence description and is close to the true *chair*.

#### 5. Conclusion

In this paper, we raise the weakly-supervised 3D visual grounding setting, using only coarse scene-sentence correspondences to learn the object-sentence links. The weak supervision gets rid of time-consuming and expensive manual annotations of accurate bounding boxes, which makes this problem more realistic but more challenging. To tackle this, we propose a novel semantic matching method to analyze the object-sentence semantic similarity in a coarse-to-fine manner. Moreover, we distill the semantic matching knowledge into the existing 3D visual grounding architecture, effectively reducing the inference cost and further improving performance. The sufficient experiments on large-scale datasets verify the effectiveness of our method.



## References

- [1] Panos Achlioptas, Ahmed Abdelreheem, Fei Xia, Mohamed Elhoseiny, and Leonidas Guibas. Referit3d: Neural listeners for fine-grained 3d object identification in real-world scenes. In *European Conference on Computer Vision*, pages 422–440. Springer, 2020. 2, 5
- [2] Steven Bird, Ewan Klein, and Edward Loper. *Natural language processing with Python: analyzing text with the natural language toolkit*. ” O’Reilly Media, Inc.”, 2009. 2
- [3] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16464–16473, 2022. 1, 2, 6, 7, 8, 11
- [4] Dave Zhenyu Chen, Angel X Chang, and Matthias Nießner. Scanrefer: 3d object localization in rgb-d scans using natural language. In *European Conference on Computer Vision*, pages 202–221. Springer, 2020. 1, 2, 5, 7
- [5] Zhenyu Chen, Ali Gholami, Matthias Nießner, and Angel X Chang. Scan2cap: Context-aware dense captioning in rgb-d scans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3193–3203, 2021. 2
- [6] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017. 1, 2, 5, 6
- [7] Samyak Datta, Karan Sikka, Anirban Roy, Karuna Ahuja, Devi Parikh, and Ajay Divakaran. Align2ground: Weakly supervised phrase grounding guided by image-caption alignment. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2601–2610, 2019. 2
- [8] Jiajun Deng, Zhengyuan Yang, Tianlang Chen, Wengang Zhou, and Houqiang Li. Transvg: End-to-end visual grounding with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1769–1779, 2021. 2
- [9] John David N Dionisio, William G Burns III, and Richard Gilbert. 3d virtual worlds and the metaverse: Current status and future possibilities. *ACM Computing Surveys (CSUR)*, 45(3):1–38, 2013. 2
- [10] Fartash Faghri, David J Fleet, Jamie Ryan Kiros, and Sanja Fidler. Vse++: Improving visual-semantic embeddings with hard negatives. *arXiv preprint arXiv:1707.05612*, 2017. 2, 5, 6
- [11] Qi Feng, Vitaly Ablavsky, and Stan Sclaroff. Cityflow-nl: Tracking and retrieval of vehicles at city scale by natural language descriptions. *arXiv preprint arXiv:2101.04741*, 2021. 2
- [12] Tanmay Gupta, Arash Vahdat, Gal Chechik, Xiaodong Yang, Jan Kautz, and Derek Hoiem. Contrastive learning for weakly supervised phrase grounding. In *European Conference on Computer Vision*, pages 752–768. Springer, 2020. 2, 5, 6
- [13] 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. 2
- [14] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020. 2
- [15] Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2(7), 2015. 2
- [16] 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*, volume 35, pages 1610–1618, 2021. 2
- [17] Shijia Huang, Yilun Chen, Jiaya Jia, and Liwei Wang. Multi-view transformer for 3d visual grounding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15524–15533, 2022. 1, 2
- [18] Maximilian Ilse, Jakub Tomczak, and Max Welling. Attention-based deep multiple instance learning. In *International conference on machine learning*, pages 2127–2136. PMLR, 2018. 2
- [19] Aishwarya Kamath, Mannat Singh, Yann LeCun, Gabriel Synnaeve, Ishan Misra, and Nicolas Carion. Mdetrm: Modulated detection for end-to-end multi-modal understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1780–1790, 2021. 2
- [20] Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. Referitgame: Referring to objects in photographs of natural scenes. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 787–798, 2014. 2
- [21] Xuejing Liu, Liang Li, Shuhui Wang, Zheng-Jun Zha, Dechao Meng, and Qingming Huang. Adaptive reconstruction network for weakly supervised referring expression grounding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2611–2620, 2019. 2
- [22] Ze Liu, Zheng Zhang, Yue Cao, Han Hu, and Xin Tong. Group-free 3d object detection via transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2949–2958, 2021. 2, 6, 11
- [23] Ilya Loshchilov and Frank Hutter. Fixing weight decay regularization in adam. 2018. 11
- [24] Oded Maron and Tomás Lozano-Pérez. A framework for multiple-instance learning. *Advances in neural information processing systems*, 10, 1997. 2
- [25] Vivek Mittal. Attngrounder: Talking to cars with attention. In *European Conference on Computer Vision*, pages 62–73. Springer, 2020. 2
- [26] Stylianos Mystakidis. Metaverse. *Encyclopedia*, 2(1):486–497, 2022. 2
- [27] Alexander Neubeck and Luc Van Gool. Efficient non-maximum suppression. In *18th International Conference on Pattern Recognition (ICPR’06)*, volume 3, pages 850–855. IEEE, 2006. 11

- [28] 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 *Proceedings of the IEEE international conference on computer vision*, pages 2641–2649, 2015. 2
- [29] Charles R Qi, Or Litany, Kaiming He, and Leonidas J Guibas. Deep hough voting for 3d object detection in point clouds. In *proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9277–9286, 2019. 2
- [30] 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. 11
- [31] Anna Rohrbach, Marcus Rohrbach, Ronghang Hu, Trevor Darrell, and Bernt Schiele. Grounding of textual phrases in images by reconstruction. In *European Conference on Computer Vision*, pages 817–834. Springer, 2016. 2
- [32] Shuran Song, Samuel P Lichtenberg, and Jianxiong Xiao. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 567–576, 2015. 5, 6
- [33] Liwei Wang, Jing Huang, Yin Li, Kun Xu, Zhengyuan Yang, and Dong Yu. Improving weakly supervised visual grounding by contrastive knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14090–14100, 2021. 2
- [34] Xin Wang, Qiuyuan Huang, Asli Celikyilmaz, Jianfeng Gao, Dinghan Shen, Yuan-Fang Wang, William Yang Wang, and Lei Zhang. Reinforced cross-modal matching and self-supervised imitation learning for vision-language navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6629–6638, 2019. 2
- [35] Fei Xia, Amir R Zamir, Zhiyang He, Alexander Sax, Jitendra Malik, and Silvio Savarese. Gibson env: Real-world perception for embodied agents. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9068–9079, 2018. 2
- [36] Zhengyuan Yang, Tianlang Chen, Liwei Wang, and Jiebo Luo. Improving one-stage visual grounding by recursive subquery construction. In *European Conference on Computer Vision*, pages 387–404. Springer, 2020. 2
- [37] Zhengyuan Yang, Songyang Zhang, Liwei Wang, and Jiebo Luo. Sat: 2d semantics assisted training for 3d visual grounding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1856–1866, 2021. 1, 2, 7, 8
- [38] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1307–1315, 2018. 2
- [39] Zhihao Yuan, Xu Yan, Yinghong Liao, Ruimao Zhang, Sheng Wang, Zhen Li, and Shuguang Cui. Instancerefer: Cooperative holistic understanding for visual grounding on point clouds through instance multi-level contextual referring. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1791–1800, 2021. 1, 2
- [40] Lichen Zhao, Daigang Cai, Lu Sheng, and Dong Xu. 3dvg-transformer: Relation modeling for visual grounding on point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2928–2937, 2021. 1, 2, 7, 8, 11

## A. Qualitative Study

### A.1. Coarse-to-fine Visualization

Figure 4 shows the visualization results of the coarse-to-fine semantic matching pipeline. Given the input point cloud, we first extract  $M_p = 256$  proposals by the 3D object detector. (When doing visualization, we employ the Non-Maximum Suppression[27] algorithm on these proposals to filter out some duplicate or overlapping boxes.) Then we conduct the coarse-grained candidate selection among these proposals to get  $K = 8$  candidates. These candidates do the fine-grained semantic matching with the sentence query one by one to finally produce the best-match result.

In all three cases, the coarse-grained module effectively selects 8 candidates related to the target object (mostly belonging to the target category). For example, in case (b), the sentence asks for a black chair near a table and facing the wall. Firstly, the module filters out objects (*tables*, *doors*) that do not belong to the target category (*chair*). Secondly, with the assistance of the feature similarity matrix, the module’s selection is also consistent with the sentence description (“by the table” and “facing the wall”) to some degree.

After coarse-grained selection, the reconstruction-based semantic matching process aims to differentiate these candidates in a fine-grained way. We get the correct answer in both cases (a) and (b), but we fail in case (c). Actually, in case (c), the candidates we get do not contain the target object. Considering that there are 32 different chairs in this scene, it’s very likely that the 8 candidates miss the target object since we are not expecting the coarse-grained part to have a deep understanding of the objects in the same category (*chair*). Even in such a condition, the fine-grained part still gets the best-match *chair* (“near the doors”, “on the left side”, and “near the center”) among the 8 candidates, which demonstrates our model’s strong ability in semantic matching.

## B. Implementation Details

### B.1. Model Setting

In this section, we give a detailed description of our model setting. We consider the whole two-stage 3DVG pipeline as off-the-shelf modules. In our practice, we use the pre-trained GroupFree[22] model as the 3D object detector, which contains a PointNet++[30] backbone, 6 layers of transformer decoder and the proposal predict head. The proposal predict head outputs the bounding boxes and the class predictions of  $M_p = 256$  object proposals. Then we employ the same attribute encoder, textual encoder and predict module as the supervised 3DJCG[3] model. The attribute encoder aggregates the attribute features (27-dimensional box center and corner coordinates and the 128-dimensional multi-view RGB features) and the

initial object features (produced by the object detector) with 2-layer multi-head self-attention modules. The input sentence query is firstly encoded by a GloVe module, and then input to a GRU cell, which produces the text feature. The predict module is simply a 1-layer multi-head cross-attention module between the text feature (Key & Value) and the object features (Query). For the reconstruct module, We mask the input sentence with a random ratio  $p = 0.3$ . We use NLTK to parse the sentences, and the verbs and nouns are treated as important words. The core of the reconstruct module is a 3-layer transformer decoder. The input point number  $N_p$ , the proposal number  $M_p$ , and the candidate number  $K$  are set to 50000, 256 and 8, respectively. The dimension of all hidden layers is 288. When calculating the candidates’ rewards, we find that instead of reducing from one to zero linearly in steps of  $1/(K - 1)$ , applying a square operation over them is better for increasing the discrimination between the optimal and suboptimal candidates.

### B.2. Training and Inference

We follow the weakly supervised setting where none of the object-sentence and bounding box annotations are used during training. We follow [40] to use 8 sentence queries for each scene to accelerate the training process. It takes 20 epochs to train our framework with a batch size of 12 (*i.e.* there are 96 sentence queries from 12 point clouds in each batch). For the stability of the reconstruction module, we start its training at the second epoch and ignore the highest  $K/2$  reconstruction losses after the third epoch. The learning rate is set to  $1e-3$  with cosine annealing strategy. We employ AdamW optimizer[23] with the weight decay of  $5e-4$ . The hyper-parameters  $\lambda_1$ ,  $\lambda_2$  and  $\lambda_3$  are set to 2, 2 and 1, respectively.

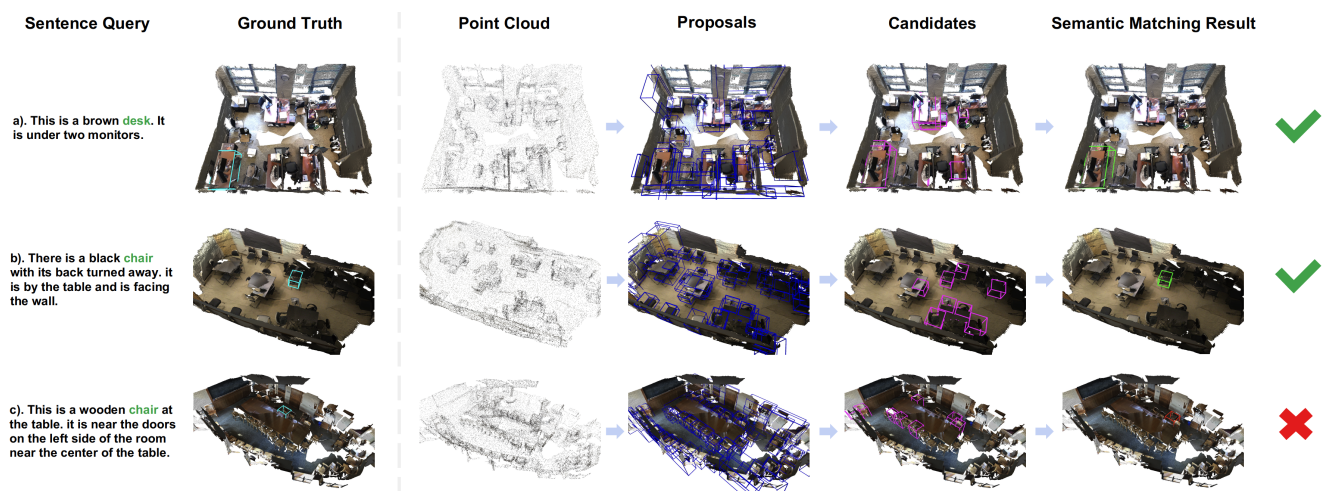


Figure 4. Visualization of the coarse-to-fine semantic matching process. With knowledge distillation, we can directly get matching result from proposals during inference (skipping the time-consuming coarse-to-fine matching).