

# SAT: 2D Semantics Assisted Training for 3D Visual Grounding

Zhengyuan Yang<sup>1</sup> Songyang Zhang<sup>1</sup> Liwei Wang<sup>2</sup> Jiebo Luo<sup>1</sup>

<sup>1</sup>University of Rochester

<sup>2</sup>The Chinese University of Hong Kong

{zyang39, szhang83, jluo}@cs.rochester.edu lwwang@cse.cuhk.edu.hk

## Abstract

3D visual grounding aims at grounding a natural language description about a 3D scene, usually represented in the form of 3D point clouds, to the targeted object region. Point clouds are sparse, noisy, and contain limited semantic information compared with 2D images. These inherent limitations make the 3D visual grounding problem more challenging. In this study, we propose 2D Semantics Assisted Training (SAT) that utilizes 2D image semantics in the training stage to ease point-cloud-language joint representation learning and assist 3D visual grounding. The main idea is to learn auxiliary alignments between rich, clean 2D object representations and the corresponding objects or mentioned entities in 3D scenes. SAT takes 2D object semantics, i.e., object label, image feature, and 2D geometric feature, as the extra input in training but does not require such inputs during inference. By effectively utilizing 2D semantics in training, our approach boosts the accuracy on the Nr3D dataset from 37.7% to 49.2%, which significantly surpasses the non-SAT baseline with the identical network architecture and inference input. Our approach outperforms the state of the art by large margins on multiple 3D visual grounding datasets, i.e., +10.4% absolute accuracy on Nr3D, +9.9% on Sr3D, and +5.6% on ScanRef.

## 1. Introduction

Visual grounding provides machines the ability to ground a language description to the targeted visual region. The task has received wide attention in both datasets [54, 31, 19] and methods [16, 46, 53, 50]. However, most previous visual grounding studies remain on images [54, 31, 19] and videos [57, 38, 51], which contain 2D projections of inherently 3D visual scenes. The recently proposed 3D visual grounding task [1, 4] aims to ground a natural language description about a 3D scene to the region referred to by a language query (in the form of a 3D bounding box). The 3D visual grounding task has various applications, including autonomous agents [40, 47], human-machine interac-

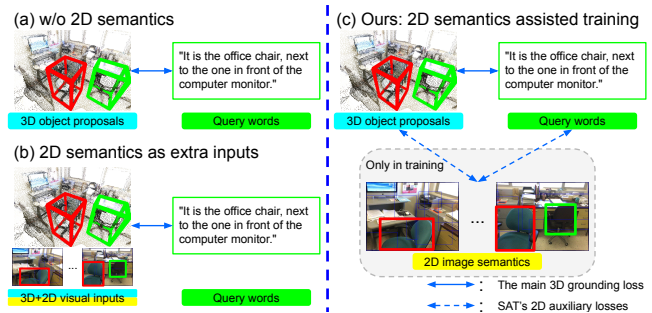


Figure 1. 3D visual grounding aims to ground a language query to a targeted 3D object region, as shown by the green 3D bounding box. (a) Previous 3D visual grounding studies are trained with a sole 3D grounding loss that maximizes the similarity between positive object-query pairs. However, the sole objective is less effective as point clouds are sparse and noisy. (b) 2D semantics contain rich and clean object representations and can be used as extra visual inputs to assist 3D grounding. However, requiring extra 2D inputs in inference limits potential application scenarios. (c) Our proposed 2D Semantics Assisted Training (SAT) uses 2D semantics only in training and does not require extra inputs in inference. The green and red boxes are the targeted and distracting objects.

tion in augmented/mixed reality [20, 22], intelligent vehicles [29, 12], and so on.

Visual grounding tries to learn a good joint representation between visual and text modalities, i.e., the 3D point cloud and language query in 3D visual grounding. As shown in Figure 1 (a), previous 3D grounding studies [1, 4, 17, 55] directly learn the joint representation with a sole 3D grounding objective of maximizing the positive object-query pairs' similarity scores. Specifically, the model first generates a set of 3D object proposals and then fuses each proposal with the language query to predict a similarity score. The framework is trained with a sole objective that maximizes the paired object-query scores and minimizes the unpaired ones' scores. However, direct joint representation learning is challenging and less effective since 3D point clouds are inherently sparse, noisy, and contain limited semantic information. Given that the 2D object representation provides rich and clean semantics, we explore using 2D image semantics to help 3D visual grounding.

How to assist 3D tasks with 2D image semantics remains an open problem. Previous studies on 3D object detection and segmentation have proposed a series of methods that take 2D semantics as extra visual inputs to assist 3D tasks. Representative approaches include aligning the 2D object detection results with 3D bounding boxes [34, 48, 25] and concatenating the image visual feature with 3D points [24, 14, 44, 41, 32]. However, these methods require extra 2D inputs in both training and inference. The necessity of extra input 2D data during inference limits potential application scenarios since 2D inputs might not exist in inference or require extra pre-processing, such as 2D-3D matching and 2D detection. Instead of as extra visual inputs in both training and inference (as shown in Figure 1 (b)), we explore using 2D semantics only in training to assist 3D visual grounding.

In this study, we propose *2D Semantics Assisted Training* (SAT), which utilizes 2D image semantics (in the form of object label, image feature, and 2D geometric feature) to ease joint representation learning between the 3D scene and language query. As shown in Figure 1 (c), in addition to the main 3D visual grounding loss [1, 4] that maximizes the score between the paired 3D object and language query, SAT introduces auxiliary loss functions that align objects in 2D images with the corresponding ones in 3D point clouds or language queries. The learned auxiliary alignments effectively distill the rich and clean 2D object representation to assist 3D visual grounding. Specifically, in SAT, we study the training loss design for auxiliary alignments and the encoding method for 2D semantics features. For the former, we propose an object correspondence loss based on the triplet loss [18, 10, 45, 26] for 3D and 2D object alignment. For the latter, we propose a transformer attention mask that generates good 2D semantics features and prevents leaking 2D inputs to the output module.

We experiment with the SAT approach on a transformer-based model [42] we propose and name as 3D grounding transformer. We benchmark SAT on the Nr3D [1], Sr3D [1], and ScanRef [4] datasets. The extra 2D semantics, together with SAT’s specially designed way of using them, effectively help the model learn a better 3D object point cloud representation and ease joint representation learning. With the same network architecture and inference inputs, SAT improves the grounding accuracy on Nr3D from the non-SAT baseline’s 37.7% to 49.2%.

In summary, our main contributions are:

- We propose *2D Semantics Assisted Training* (SAT) that assists 3D visual grounding with 2D semantics. To the best of our knowledge, SAT is the first method that helps 3D tasks with 2D semantics in training but does not require 2D inputs during inference.
- With the proposed object correspondence loss and the 2D semantics encoding method, SAT effectively utilizes 2D semantics to learn a better 3D object repre-

sentation, which leads to significant accuracy improvements on the Nr3D [1] (+10.4% in absolute accuracy), Sr3D [1] (+9.9%), and ScanRef [4] (+5.6%) datasets.

## 2. Related Work

**3D visual grounding.** 3D visual grounding aims to ground the language referred object in a 3D scene (in the form of RGB-XYZ point clouds) to a 3D bounding box. Two recent works Referit3D [1] and ScanRef [4], independently proposed datasets and baseline methods for the 3D visual grounding task. Both works [1, 4] augment the 3D scans in the ScanNet [7] dataset with the manually annotated language queries to construct the 3D visual grounding datasets. Previous 3D grounding studies [1, 4, 17, 55, 11, 36, 58] follow a two-stage framework. In the first stage, multiple 3D object proposals are generated either with ground truth objects [1] or a 3D object detector [4, 33]. In the second stage, 3D object proposal features are fused with the language query to predict each proposal’s matching scores. A softmax grounding loss is applied to maximize the score between the paired object proposal and language query.

We find that the sole objective of similarity score maximization is less effective because the point clouds for object proposals are sparse and noisy. In this study, we explore using 2D image semantics to assist 3D visual grounding.

**2D semantics in 3D tasks.** Studies on 3D object detection and segmentation have explored using 2D image semantics to assist 3D tasks. There exist two representative approaches, *i.e.*, 1) projecting image object detection results into 3D space to assist 3D box prediction [34, 48, 25] and 2) concatenating the image feature with each point in the 3D scene as the extra information for the 3D tasks [24, 14, 44, 41, 32]. ImVoteNet [32] fuses the image object detection results with 3D points.

Previous studies use 2D image semantics as the extra inputs to 3D tasks and thus require the extra 2D information in both training and inference. Despite the performance improvement, the extra 2D inputs potentially limit the application scenarios since the 2D information either does not exist in inference or requires tedious pre-processing, such as 2D-3D matching and 2D object detection. In this study, we explore using 2D semantics only in training to assist 3D visual grounding.

**Image visual grounding.** 3D visual grounding is related to the image visual grounding task [19, 30, 54, 28]. There are mainly two approaches in image visual grounding, namely the one- and two-stage frameworks. The one-stage methods [50, 37, 49, 8] fuse the language query with each pixel/patch in image and predict grounding boxes densely at all spatial locations. The two-stage methods [54, 45, 53] first generate object proposals based on the visual objectiveness. The methods then compare each proposal with the language query to select the grounding prediction.

We follow previous 3D grounding studies [1, 4, 17] and experiment with our proposed SAT on a two-stage framework introduced in Section 3. We focus on using 2D semantics to assist 3D grounding in this study and leave the exploration of alternative frameworks to future studies.

### 3. 3D Grounding Transformer

Before introducing our proposed 2D semantics assisted training (SAT), we first overview the problem modeling of the 3D visual grounding task, and a transformer-based network architecture that we experiment on, named the 3D grounding transformer.

#### 3.1. 3D visual grounding inputs

The input to the 3D visual grounding task is a 3D scene  $S \in \mathbb{R}^{N \times 6}$  in the form of RGB-XYZ point clouds with  $N$  points and a natural language query with  $K$  words. In the training stage, SAT takes the extra input of 2D semantics extracted from the original ScanNet videos [7] to ease joint representation learning. We detail how SAT represents and utilizes 2D semantics in following sections.

#### 3.2. Embedding for all modalities

**3D scene embedding.** Following previous studies [1, 4, 17], we assume the access to  $M$  3D object proposals (in the form of point cloud object segments) in scene  $S$ . The proposals are either generated with ground truth objects as in Referit3D [1] or by a detection network [33] as in ScanRef [4]. After getting the proposals, we normalize each object’s center and size [43], and encode the point cloud segment of each proposal into a feature vector  $x_m^{pc}$  with PointNet++ [35, 1, 4, 43]. We obtain the  $d$ -dimensional 3D proposal embedding  $\{O_1, \dots, O_M\}$  with two learned linear transforms, where

$$O_m = \text{LN}(W_1 x_m^{pc}) + \text{LN}(W_2 x_m^{\text{offset}}).$$

$x_m^{pc}$  is the PointNet++’s output feature.  $x_m^{\text{offset}}$  is a 4D vector with the normalization offset, *i.e.*, the center offsets  $(x, y, z)$  and the original size  $r$  for proposal  $m$ .  $W_1, W_2$  are learned projection matrices.  $\text{LN}(\cdot)$  is layer normalization [3].

**2D semantics embedding.** For each 3D proposal  $m$ , we project its point clouds onto  $L$  sampled frames in the original ScanNet videos [7] and get the corresponded 2D image semantics (image region, 2D bounding box, object class). In sampled frame  $l \in \{1, \dots, L\}$ , we represent the 2D semantics for proposal  $m$  by its visual feature  $x_{m,l}^{ROI}$  (Region of interest feature from a visual genome [23] pre-trained Faster-RCNN detector [39]), semantic feature  $x_{m,l}^{cls}$  (one-hot class vector), and geometric feature  $x_{m,l}^{geo}$  (2D bounding box coordinates and frame’s camera pose). We obtain the  $d$ -dimensional 2D semantics  $I_{m,l}$  with linear transforms:

$$I_{m,l} = \text{LN}(W_3 x_{m,l}^{ROI} + W_4 x_{m,l}^{cls}) + \text{LN}(W_5 x_{m,l}^{geo}), \quad (1)$$

where  $W_3, W_4, W_5$  are learned projection matrices and  $\text{LN}(\cdot)$  is layer normalization. We note that a 3D proposal  $O_m$  corresponds to multiple 2D semantic feature vector  $I_{m,l}$  obtained from different frames  $l$ . We randomly choose one of  $I_{m,l}, l \in \{1, \dots, L\}$  as the corresponding 2D semantics in each epoch of training in SAT. We refer to the sampled  $d$ -dimensional 2D semantic vector as  $I_m$ , which corresponds to the 3D proposal  $O_m$ .

**Text embedding.** Given a query with  $K$  words, we embed the text input with a pre-trained BERT model [9] into a set of  $d$ -dimensional word feature vectors  $\{Q_1, \dots, Q_K\}$ . We fine-tune the BERT text encoder during training.

#### 3.3. Fusion and grounding module

After respectively embedding each modality into multiple  $d$ -dimension feature vectors, we apply a stack of transformer layers [42] to fuse the input modalities (query words, 3D objects proposals, and if training 2D semantics). We denote the transformer’s output features at the language, 3D proposal, and 2D semantics positions as  $F^Q, F^O$ , and  $F^I$ .

An output grounding module that consists of two fully connected layers projects fused features  $\{F_1^O, \dots, F_M^O\}$  into a set of  $M$  grounding scores  $\{S_1^O, \dots, S_M^O\}$ , respectively. The object proposal  $m$  with the highest grounding score is selected as the final grounding prediction.

## 4. 2D Semantics Assisted Training (SAT)

SAT learns auxiliary alignments between the 2D object semantics and the objects in 3D scenes/language queries to assist 3D visual grounding. Figure 2 overviews SAT in training and inference with the 3D grounding transformer.

We study two technical problems in SAT. First, in Section 4.1, we propose the auxiliary training objectives that align 2D semantics with the 3D scene and language query. Second, in Section 4.2, we introduce the 2D semantics encoding method that generates the fused feature  $F^I$  from 2D inputs  $I$ . We use  $F^I$  in computing the auxiliary losses.

### 4.1. Training objectives

In addition to the main training objective between the 3D scene and language query, SAT introduces auxiliary training objectives to align 2D semantics with the 3D scene and language query. We apply the “visual grounding loss” between the query and 3D/2D visual inputs. We propose an “object correspondence loss” between the 3D and 2D objects.

**3D visual grounding loss.** We first introduce the main visual grounding loss  $\mathcal{L}_{VG}^O$  between the 3D scene and language query [27, 50, 1, 4]. Visual grounding loss  $\mathcal{L}_{VG}$  is a softmax loss over grounding scores  $S_m^O$  for proposals  $m \in \{1, \dots, M\}$ . The proposal with the highest Intersection over Union (IoU) with the ground truth region is labeled 1 and all remaining ones have label 0 (the highest IoU

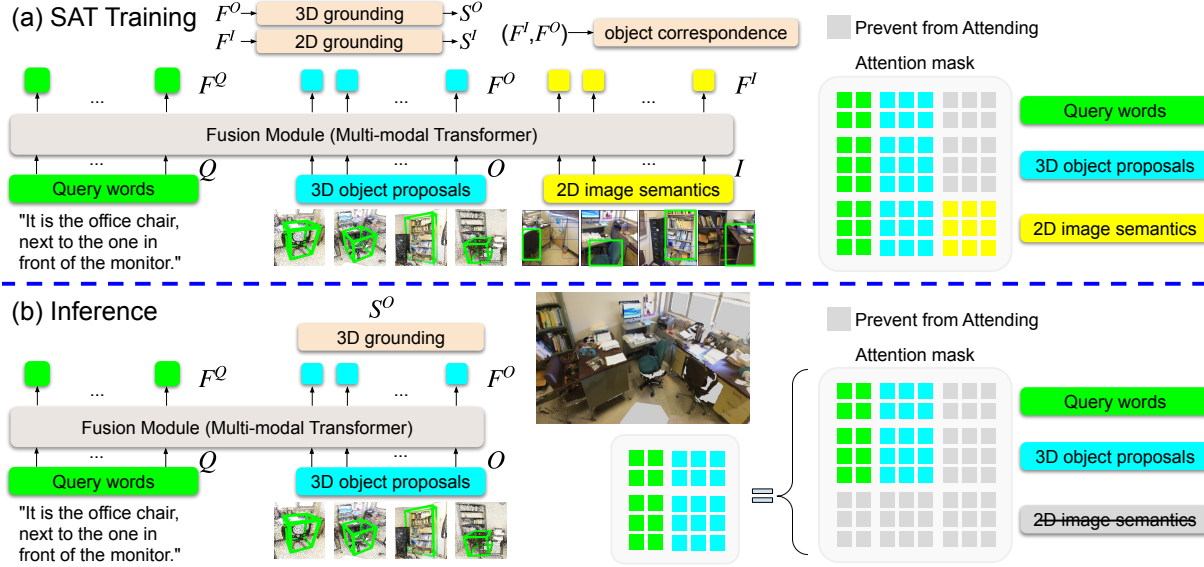


Figure 2. The proposed 2D semantics assisted training (SAT) for 3D visual grounding. (a) In training, SAT takes 2D semantics as extra input and helps 3D visual grounding with the auxiliary objectives of 2D visual grounding  $\mathcal{L}_{VG}^I$  and object correspondence prediction  $\mathcal{L}_{cor}$ . (b) In inference, SAT does not require 2D inputs and is easy to use. SAT’s attention mask prevents query words and 3D proposals from attending on 2D image semantics  $I$  in training (top five rows of the mask), avoiding performance drop in inference when  $I$  is not available.

equals 1.0 when experimented with ground truth object proposals).  $\mathcal{L}_{VG}^O$  encourages the model to generate high scores for positive proposals. In inference, the proposal with the highest score  $S^O$  is selected as the final prediction.

**2D visual grounding loss.** We apply the 2D grounding loss  $\mathcal{L}_{VG}^I$  with the same form as  $\mathcal{L}_{VG}^O$  between the 2D semantics and language query. A separate grounding head with two fully connected layers projects the fused features  $\{F_1^I, \dots, F_M^I\}$  into the 2D grounding scores  $\{S_1^I, \dots, S_M^I\}$ .  $\mathcal{L}_{VG}^I$  is the softmax loss computed over  $S^I$ .

**Object correspondence loss.** The proposed object correspondence loss learns the correspondence between the objects in 3D scenes and the ones in 2D images. We design the object correspondence loss as a triplet loss [18, 10, 45, 26]:

$$\mathcal{L}_{cor} = \sum_{m=1}^M \left\{ [\alpha - s(F_m^O, F_m^I) + s(F_m^O, F_i^I)]_+ + [\alpha - s(F_m^O, F_m^I) + s(F_j^O, F_m^I)]_+ \right\},$$

where  $s(\cdot)$  is the similarity function. We use the inner product over the L2 normalized feature  $F^O$  and  $F^I$  as  $s(\cdot)$  in our experiments.  $\alpha$  is the margin with a default value of 0.1.  $i, j$  are the index for the hard negatives where  $i = \operatorname{argmax}_{i \neq m} s(F_m^O, F_i^I)$  and  $j = \operatorname{argmax}_{j \neq m} s(F_j^O, F_m^I)$ . We compute the object correspondence among the 3D and 2D object proposals  $m$  within each sample (3D scene). We do not construct negatives across different 3D scenes.

We optimize the model with the following loss function:

$$\mathcal{L} = \mathcal{L}_{VG}^O + \mathcal{L}_{VG}^I + \mathcal{L}_{cor} * w_{cor} + (\mathcal{L}_{cls}^O + \mathcal{L}_{cls}^Q) * w_{cls}, \quad (2)$$

where  $w_{cor}$  is the weight for the object correspondence loss with a default value of 10. In addition to 3D/2D grounding loss  $\mathcal{L}_{VG}$  and object correspondence loss  $\mathcal{L}_{cor}$ , we add query and object classification losses  $\mathcal{L}_{cls}^Q$  and  $\mathcal{L}_{cls}^O$  as in Referit3D [1]. The query feature  $Q_0$  and proposal feature  $O$  are projected with fully connected layers to predict the object classes for the language query and 3D proposals. We follow the classification loss weight  $w_{cls}$  of 0.5 [1]. Ablation studies on the losses are in the supplementary material.

## 4.2. 2D semantics encoding

SAT uses the fused 2D semantic feature  $F^I$  to compute 2D visual grounding loss  $\mathcal{L}_{VG}^I$  and object correspondence loss  $\mathcal{L}_{cor}$ . In this subsection, we introduce how to encode  $F^I$  from 2D semantics  $I$ . We show that a simple yet effective approach to encode  $F^I$  is by introducing proper attention masks in the multi-modal transformer. Specifically, we adopt the same stack of transformer layers to jointly encode the three input modalities  $Q$ ,  $O$ , and  $I$ . We design the attention mask in Figure 2 (a) such that  $F^Q$  and  $F^O$  do not directly attend to 2D inputs  $I$  (the top five rows of the mask). In this way, the proposed mask prevents the model from directly using 2D inputs  $I$  for grounding prediction and thus avoids the performance drop in inference when  $I$  is not available. Meanwhile, the proposed attention mask allows the model to reference both 2D semantics  $I$  and other input features  $Q$  and  $O$  when generating  $F^I$  (the bottom three rows of the mask).

We find that both properties of the proposed mask, *i.e.*, masking 2D inputs  $I$  from  $F^Q$  and  $F^O$ , and referencing  $Q$



and  $O$  when generating  $F^I$ , are critical to SAT’s success. We discuss alternative methods as follow. **1)** Methods that do not mask  $I$  from  $F^Q$  and  $F^O$  will leak 2D inputs to  $F^O$  and partially rely on  $I$  to generate the grounding prediction in training. Therefore, the grounding accuracy drops catastrophically in inference when no 2D inputs  $I$  are available. **2)** Encoding  $F^Q/F^O$  and  $F^I$  independently with  $Q/O$  and  $I$  avoid the 2D input leakage. However, without referencing the scene context  $Q$  and  $O$ , the 2D feature  $F^I$  fails to generate relevant object representations that effectively help 3D visual grounding. We show related ablations in Section 5.4.

## 5. Experiments

### 5.1. Datasets

**Nr3D.** The Natural Reference in 3D (Nr3D) dataset [1] augments the indoor 3D scene dataset ScanNet [7] with 41, 503 natural language queries annotated by Amazon Mechanical Turk (AMT) workers. There exist 707 unique indoor scenes with targets belong to one of the 76 object classes. There are multiple but no more than six distractors (objects in the same class as the target) in the scene for each target. The dataset splits follow the official ScanNet [7] splits.

**Sr3D/Sr3D+.** The Spatial Reference in 3D (Sr3D) dataset [1] contains 83, 572 queries automatically generated based on a “target”-“spatial relationship”-“anchor object” template. The Sr3D+ dataset further enlarges Sr3D with the samples that do not have multiple distractors in the scene and ends up with 114, 532 queries.

**ScanRef.** The ScanRef dataset [4] augments the 800 3D indoor scenes in the ScanNet [7] dataset with 51, 583 language queries. ScanRef follows the official ScanNet [7] splits and contains 36, 665, 9, 508, and 5, 410 samples in train/val/test sets, respectively.

### 5.2. Experiment settings

**Evaluation metric.** We follow the experiment settings in Referit3D [1] and ScanRef [4] for experiments with ground truth and detector-generated proposals, respectively. Specifically, Referit3D [1] assumes the access to ground truth objects as the 3D proposals and converts the grounding task into a classification problem. The models are evaluated by the accuracy, *i.e.*, whether the model correctly selects the referred object among  $M$  proposals. We choose this “using ground truth proposal” setting as the default setting and present the results on all experimented datasets (Nr3d, Sr3d, and ScanRef).

Alternatively, ScanRef [4] adopts a 3D object detector [33] to generate object proposals. On the ScanRef dataset, we also evaluate models using  $\text{Acc}@k\text{IoU}$ , *i.e.*, the fraction of language queries whose predicted box overlaps the ground truth with  $\text{IoU} > k\text{IoU}$ . We experiment with the IoU threshold  $k\text{IoU}$  of 0.25 and 0.5. For clarity, we present

the experiments with ground truth proposals in the main paper and postpone the experiments of “SAT with detector-generated proposals” to the supplementary material.

**Implementation details.** We set the dimension  $d$  in all transformer layers as 768. We experiment with a text transformer with 3 layers and a fusion transformer with 4 layers [15, 52]. The text transformer is initialized from the first three layers of BERT<sub>BASE</sub> [9], and the fusion transformer is trained from scratch. We sample 1024 points for each 3D proposal from its point cloud segment and encode the proposal with PointNet++ [35]. We follow the max sentence length and proposal numbers in Referit3D [1] and ScanRef [4] when experimented on Nr3D/Sr3D and ScanRef, respectively. The model is trained with the Adam [21] optimizer with a batch size of 16. We set an initial learning rate of  $10^{-4}$  and reduce the learning rate by a multiplicative factor of 0.65 every 10 epochs for a total of 100 epochs.

**Compared methods.** We compare SAT with the state-of-the-art methods [1, 4, 17, 55] and the non-SAT baseline. “Non-SAT” adopts the same “3D grounding transformer” architecture used in “SAT.” The only difference is that “non-SAT” does not include 2D semantics in training and thus does not use the auxiliary losses  $\mathcal{L}_{VG}^I$  and  $\mathcal{L}_{cor}$ . With the same network architecture and experiment settings, “non-SAT” is a directly comparable baseline to “SAT.” The performance difference shows how much SAT could help the 3D visual grounding task.

### 5.3. 3D visual grounding results

**Nr3D.** Table 1 reports the grounding accuracy on the Nr3D [1] dataset. Both “non-SAT” and “SAT” use the 3D grounding transformer introduced in Section 3. For SAT’s reported accuracy, we encode 2D semantics  $I$  from the visual feature  $x^{ROI}$ , object semantic feature  $x^{cls}$ , and geometric feature  $x^{geo}$  following Eq. 1. We postpone the ablation studies on the types of 2D semantics to Section 5.4. Different columns show the results with different training data, *i.e.*, using Nr3D’s training set only, or jointly training with Sr3D/Sr3D+’s training set. We take “SAT-Nr3D” as the default setting and refer to it as “SAT.” We refer to the experiments with extra data as “SAT w/ Sr3D/Sr3D+.”

The top five rows of Table 1 show that our baseline “non-SAT” already achieves comparable performance to the state of the art (non-SAT: 37.7%, InstanceRefer [55]: 38.8%). By effectively utilizing 2D semantics in training, our proposed SAT improves the non-SAT baseline accuracy from 37.7% to 49.2%, with the identical model architecture and inference inputs. SAT also outperforms the state-of-the-art accuracy [55] of 38.8% a large margin of +10.4%. Jointly using the Sr3D/Sr3D+ training data further improves the grounding accuracy. As shown in the last row, “SAT w/ Sr3D+” improves “SAT-Nr3D” from 49.2% to 56.5%.

Analyses reveal that SAT learns a better 3D object repre-

Table 1. The 3D grounding accuracy on Nr3D [1] with different training data (Nr3D training set only or with extra data from Sr3D/Sr3D+).

| Method             | Nr3D             | w/ Sr3D          | w/ Sr3D+         |
|--------------------|------------------|------------------|------------------|
| V + L [1]          | 26.6±0.5%        | -                | -                |
| Ref3DNet [1]       | 35.6±0.7%        | 37.2±0.3%        | 37.6±0.4%        |
| TGNN [17]          | 37.3±0.3%        | -                | -                |
| InstanceRefer [55] | 38.8±0.4%        | -                | -                |
| non-SAT            | 37.7±0.3%        | 43.9±0.3%        | 45.9±0.2%        |
| SAT (Ours)         | <b>49.2±0.3%</b> | <b>53.9±0.2%</b> | <b>56.5±0.1%</b> |

Table 3. The accuracy on ScanRef [4] with different training data (ScanRef training set only or with extra data from Nr3D/Sr3D+).

| Method       | ScanRef          | w/ Nr3D          | w/ Sr3D+         |
|--------------|------------------|------------------|------------------|
| Ref3DNet [1] | 46.9±0.2%        | 47.5±0.4%        | 47.0±0.3%        |
| non-SAT      | 48.2±0.2%        | 50.2±0.1%        | 51.7±0.1%        |
| SAT (Ours)   | <b>53.8±0.1%</b> | <b>57.0±0.3%</b> | <b>56.5±0.2%</b> |

sensation with the assist of 2D semantics, which leads to the 11.5% improvement over the non-SAT baseline. The improvement brought by Sr3D/Sr3D+ mainly comes from better modeling the spatial relationships in queries. We present these analyses in Section 6.

**Sr3D.** Table 2 shows the grounding accuracy on Sr3D [1]. We draw similar conclusions from Table 2 as in Table 1 that 1) SAT significantly improves the grounding accuracy from 47.4% to 57.9%, 2) SAT outperforms the previous state of the art [1, 17, 55] by large margins, and 3) extra training data (Nr3D) further boosts the accuracy from 57.9% to 60.7%.

**ScanRef.** Table 3 reports the grounding accuracy on the ScanRef dataset [4] with ground truth object proposals. We observe a significant improvement of “SAT” over the non-SAT baseline (SAT: 53.8%, non-SAT: 48.2%). Extra training data from Nr3D and Sr3D+ further improves the accuracy from 53.8% to 57.0% and 56.5%, respectively.

In addition to the ground-truth object proposals [1], we experiment with the proposal setting in ScanRef [4] that generates proposals with a 3D detector [33]. To apply SAT, we first compute the ground truth 2D semantics offline. In training, we match each predicted 3D proposal with a 2D semantics object that has the largest IoU with the 3D proposal. We then evaluate the models with the Acc@0.25 and Acc@0.50 metrics. SAT achieves the Acc@0.25 and Acc@0.50 of 44.54% and 30.14%, outperforming the non-SAT baseline of 38.92% and 26.40% by a large margin. We introduce the details of “SAT with detector-generated proposals” in the supplementary material.

#### 5.4. Ablation studies

**Multi-modal transformer masks.** SAT’s attention mask in the multi-modal transformer has two properties, *i.e.*, 1) masking 2D semantics  $I$  from  $F^Q$  and  $F^O$ , and 2) referencing context  $Q$  and  $O$  when generating  $F^I$ . We verify the importance of both properties with the ablation studies in Table 4. In training, we replace our proposed transformer’s

Table 2. The 3D grounding accuracy on Sr3D [1] with different training data (Sr3D training set only or with extra data from Nr3D).

| Method             | Sr3D             | w/ Nr3D          |
|--------------------|------------------|------------------|
| V + L [1]          | 33.0±0.4%        | -                |
| Ref3DNet [1]       | 40.8±0.2%        | 41.5±0.2%        |
| TGNN [17]          | 45.0±0.2%        | -                |
| InstanceRefer [55] | 48.0±0.3%        | -                |
| non-SAT            | 47.4±0.2%        | 50.1±0.1%        |
| SAT (Ours)         | <b>57.9±0.1%</b> | <b>60.7±0.2%</b> |

Table 4. Ablation studies on 2D semantics embedding with different transformer attention masks. The gray mask color indicates prevent from attending. SAT’s attention mask is in Figure 2 (a).

| Method     | Accuracy         |
|------------|------------------|
| non-SAT    | 37.7±0.3%        |
| SAT-mask A | 33.9±0.2%        |
| SAT-mask B | 43.9±0.2%        |
| SAT (Ours) | <b>49.2±0.3%</b> |




Table 5. Ablation studies on different types of 2D semantics inputs as in Eq 1. We highlight the default “SAT” setting by underline.

|     | $+x^{geo}$ | $+x^{cls}$ | $+x^{ROI}$ | Acc.             |
|-----|------------|------------|------------|------------------|
| (a) | -          | -          | -          | 37.7±0.3%        |
| (b) | ✓          | -          | -          | 39.4±0.3%        |
| (c) | ✓          | ✓          | -          | 48.1±0.2%        |
| (d) | ✓          | -          | ✓          | 46.5±0.1%        |
| (e) | -          | ✓          | ✓          | 43.2±0.2%        |
| (f) | ✓          | ✓          | ✓          | <u>49.2±0.3%</u> |

attention mask in Figure 2 (a) with the alternative masks A/B in Table 4. In inference, the model removes the extra 2D semantics input and follows the standard inference setting as in Figure 2 (b).

Mask A does not mask 2D semantics  $I$  from  $F^Q$  and  $F^O$ . We observe that the model directly relies on the extra 2D inputs  $I$  for grounding prediction. Consequently, the grounding accuracy drops catastrophically to 33.9% when no 2D inputs are available in inference. Mask B encodes  $F^I$  with 2D semantics  $I$  alone. Without referencing the scene context in  $Q$  and  $O$ , the 2D feature  $F^I$  fails to provide a relevant object representation and is thus less effective in helping 3D visual grounding. Although outperforming the non-SAT baseline accuracy of 37.7%, “SAT-mask B” performs much worse than the SAT with our proposed attention mask (SAT-mask B: 43.9%, SAT: 49.2%).

**Types of 2D context inputs.** Table 5 shows the ablation studies on the types of 2D semantics. The combination of  $x^{ROI}$ ,  $x^{cls}$ , and  $x^{geo}$  are projected into a  $d$ -dimension 2D semantics feature  $I$  following Eq. 1.

Compared with the non-SAT baseline accuracy of 37.7% in row (a), SAT with any 2D semantics significantly boosts the grounding accuracy (rows (b-f)). Jointly using visual feature  $x^{ROI}$ , semantic feature  $x^{cls}$ , and geometric feature  $x^{geo}$  achieves the best accuracy of 49.2%, as in row (f).

## 6. How does SAT help?

In this section, we analyze how does SAT help 3D visual grounding. We draw three major conclusions: **1)** SAT learns a better 3D object representation  $F^O$  with the assist of 2D semantics  $I$ , leading to consistent performance improvements over samples with different target classes, number of distractors, query lengths/types, *etc.* (Section 6.1). **2)** Training with the extra data in Sr3D/Sr3D+ mainly benefits the queries with spatial relationship referring (Section 6.2). **3)** The performance gap between SAT and the methods that require extra 2D inputs in inference is small, indicating the effectiveness of SAT in utilizing 2D semantics (Section 6.3). Finally, we present qualitative examples in Section 6.4.

### 6.1. Linear probing

How could SAT achieve the large improvement over the non-SAT baseline and the state of the art? We conjecture that SAT learns a better 3D representation  $F^O$  for noisy object point clouds with the assist of 2D semantics. Consequently, we observe consistent 3D grounding accuracy improvements on samples with different target object classes, number of distractors, query lengths/types, *etc.*, as shown in the performance breakdown in the supplementary material.

We use linear probing [56, 13, 6] to evaluate the quality of the learned 3D object representations  $F^O$  in different models. Specifically, we keep the pre-trained grounding network fixed and train a linear classifier that maps each proposal feature  $F_m^O$  into one of Nr3D’s 607 object classes. Because no classification annotation is seen during the grounding network training, we evaluate learned representations  $F^O$  by the object classification accuracy. Similar to the use of linear probing in representation learning [56, 13, 6], we consider a higher linear probing accuracy the indicator of a better 3D object representation  $F^O$ .

Table 6 shows the linear probing accuracy on Nr3D. SAT improves the linear probing accuracy from 35.7% to 60.1%, compared with the non-SAT baseline. The significant improvement supports the conjecture that SAT learns a better 3D object representation with 2D semantics in training. We observe similar improvements in the full fine-tuning setting, where all layers are updated for object classification. It is worth noting that SAT’s effectiveness in generating better 3D object representations may hold the promise of benefiting not only 3D vision-language tasks such as grounding [1, 4] and captioning [5], but also 3D semantic understanding tasks such as 3D scene graph prediction [2, 43].

### 6.2. Spatial relationship referring

Our second observation is that the extra data in Sr3D/Sr3D+ helps the queries with spatial relationship referring. On Nr3D’s subset with spatial queries (76.7% of the samples), the extra Sr3D+ training data leads to an 8.4% improvement on “SAT-Nr3D” from 48.4% to 56.8%. In

Table 6. Linear probing accuracy on Nr3D.

| Method       | Linear probing | Full fine-tuning |
|--------------|----------------|------------------|
| non-SAT      | 35.7%          | 63.4%            |
| SAT          | 60.1%          | 65.4%            |
| SAT w/ Sr3D+ | 61.7%          | 67.6%            |

contrast, the improvement is only 3.9% on the remaining samples (from 50.9% to 54.8%). Furthermore, we observe larger improvements on subsets that contain the frequently appeared spatial words in Sr3D/Sr3D+, *e.g.*, “closest” of +11.5% and “farthest” of +13.5%.

### 6.3. 2D semantics as extra inputs

In this subsection, we compare SAT with the methods that require extra 2D inputs in both training and inference. We design two methods that directly use 2D semantics as extra inputs, namely the “2D input aligned” and “2D input unaligned.” Both methods use the same network architecture as SAT and take extra 2D inputs in both training and inference. For “2D input aligned,” we concatenate 2D semantics  $I_m$  with 3D proposal feature  $O_m$  and use the extended proposal feature as  $O_m$  in both training and inference. The input sequence length for “2D input aligned” is  $M + K$ . We train “2D input aligned” with the main grounding loss  $\mathcal{L}_{VG}^O$  and the classification loss  $\mathcal{L}_{cls}$  in Eq. 2. For “2D input unaligned,” we input 2D semantics  $I_m$  as extra input tokens to the multi-modal transformer in both training and inference. The input sequence length for “2D input unaligned” is  $2M + K$ . We train “2D input unaligned” with the same loss  $\mathcal{L}$  in Eq. 2 as SAT.

Table 7 shows the experiment results on Nr3D with no 2D semantics (upper portion), with 2D inputs only in training (middle portion), and in both training and inference (bottom portion). The “hard” subset contains more than 2 distractors and remaining samples belong to “easy.” We observe a marginal accuracy gap of 1% between “SAT” and “2D input aligned/unaligned” (“overall” in row (e): 49.2%, rows (h,i): 50.3%). The comparable performance indicates SAT’s effectiveness in utilizing 2D semantics to help 3D visual grounding. Meanwhile, SAT does not require extra 2D inputs in inference as “2D input aligned/unaligned,” and thus is easier to use.

### 6.4. Qualitative insights

The left four examples of Figure 3 show representative failure cases of “non-SAT” that can be corrected by “SAT.” We group common cases into three scenarios. **1) Object:** SAT improves non-SAT by better recognizing the object classes. Non-SAT occasionally fails to ground the head noun and generates the object prediction in a different class, *e.g.*, “bed” instead of the referred “desk” in Figure 3 (a). **2) Relationship:** We observe that SAT is better in modeling relationships in language queries, despite no specific modules are proposed in SAT for relationship understanding. For example, in Figures 3 (b,c), SAT correctly understands the relationship “attach to” and “over.” We con-



Table 7. The benefit of using 2D semantics in 3D visual grounding. The upper/middle/bottom portion of the table shows the results that do not use 2D semantics/use only in training/use in both training and inference (as extra inputs). The results with extra training data (Sr3D/Sr3D+) are shown in gray. Our SAT (#(e)) shows comparable performance to oracles that require 2D inputs in inference (#(h,i)).

|     |                     | Extra data | 2D semantics |      | Overall          | Easy      | Hard      | View-dep. | View-indep. |
|-----|---------------------|------------|--------------|------|------------------|-----------|-----------|-----------|-------------|
|     |                     |            | Train        | Test |                  |           |           |           |             |
| (a) | Ref3DNet [1]        | -          | ✗            | ✗    | 35.6±0.7%        | 43.6±0.8% | 27.9±0.7% | 32.5±0.7% | 37.1±0.8%   |
| (b) | TGNN [17]           | -          | ✗            | ✗    | 37.3±0.3%        | 44.2±0.4% | 30.6±0.2% | 35.8±0.2% | 38.0±0.3%   |
| (c) | InstanceRefer [55]  | -          | ✗            | ✗    | 38.8±0.4%        | 46.0±0.5% | 31.8±0.4% | 34.5±0.6% | 41.9±0.4%   |
| (d) | non-SAT             | -          | ✗            | ✗    | 37.7±0.3%        | 44.5±0.5% | 31.2±0.2% | 34.1±0.3% | 39.5±0.4%   |
| (e) | SAT (Ours)          | -          | ✓            | ✗    | <b>49.2±0.3%</b> | 56.3±0.5% | 42.4±0.4% | 46.9±0.3% | 50.4±0.3%   |
| (f) | SAT w/ Sr3D (Ours)  | Sr3D       | ✓            | ✗    | 53.9±0.2%        | 61.5±0.1% | 46.7±0.3% | 52.7±0.7% | 54.5±0.3%   |
| (g) | SAT w/ Sr3D+ (Ours) | Sr3D+      | ✓            | ✗    | 56.5±0.1%        | 64.9±0.2% | 48.4±0.1% | 54.4±0.3% | 57.6±0.1%   |
| (h) | 2D input aligned    | -          | ✓            | ✓    | 50.0±0.1%        | 62.0±0.2% | 38.5±0.3% | 44.7±0.3% | 52.6±0.3%   |
| (i) | 2D input unaligned  | -          | ✓            | ✓    | 50.3±0.4%        | 58.5±0.7% | 42.4±0.5% | 48.1±0.4% | 51.3±0.5%   |
| (j) | 2D input aligned    | Sr3D+      | ✓            | ✓    | 59.7±0.1%        | 71.0±0.3% | 48.8±0.5% | 52.9±0.3% | 63.1±0.2%   |
| (k) | 2D input unaligned  | Sr3D+      | ✓            | ✓    | 61.0±0.3%        | 69.0±0.6% | 53.2±0.3% | 58.4±0.3% | 62.2±0.5%   |

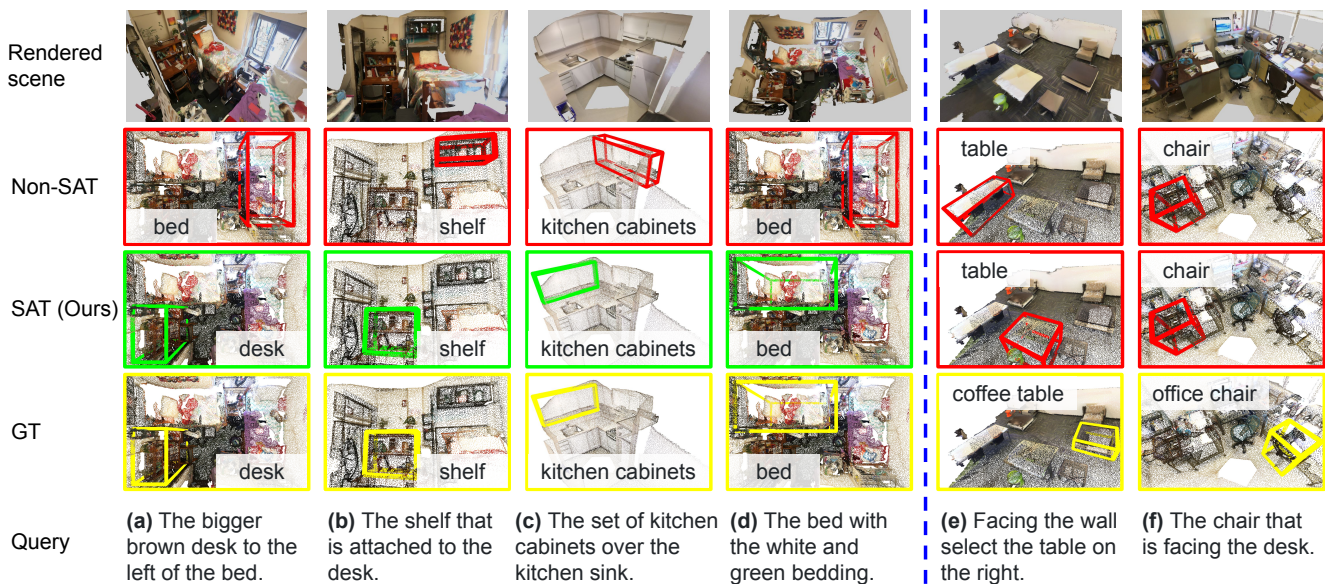


Figure 3. The failure cases of non-SAT that can be corrected by SAT (the left four examples), and SAT’s representative failure cases (the right two examples). The green/red/yellow colors indicate the correct/incorrect/ground truth boxes. The object class for each box is shown in text next to the 3D box. We provide rendered scenes in first row to better visualize the scene layout. Best viewed zoomed in and in color.

ture that SAT learns a better object representation for both foreground and background objects, which benefits the relationship modeling. **3) Color and shape:** SAT also performs better in understanding color and shape-related language queries, *e.g.*, “white and green” in Figure 3 (d).

The right two examples of Figure 3 show SAT’s representative failure cases. Figure 3 (e) shows a failure case that requires understanding “facing the wall.” Although SAT improves both view-dependent and independent samples (*c.f.* Table 7 “View-dep.” column), view understanding remains an unsolved problem. Figure 3 (f) shows a failure case caused by ambiguous queries. The model predicts the “chair facing the desk” instead of the referred “office chair facing the desk” in the ground truth. We observe that the model and human annotators occasionally confuse objects in similar categories, such as chair/office-chair (Fig-

ure 3 (f)), table/coffee-table (Figure 3 (e)), *etc.*

## 7. Conclusion

We have presented 2D semantics assisted training (SAT) for 3D visual grounding. SAT uses 2D semantics in training to assist 3D visual grounding and eases joint representation learning between the 3D scene and language query. With identical network and inference inputs, SAT beats the non-SAT baseline by 11.5% in absolute accuracy. SAT leads to the new state of the art on multiple datasets and outperforms previous works by large margins. Analyses show that SAT effectively uses 2D semantics to learn a better 3D point cloud object representation that helps 3D visual grounding.

## Acknowledgment

This work is supported in part by NSF awards IIS-1704337 and IIS-1813709, as well as our corporate sponsors.



## 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 *ECCV*, pages 422–440. Springer, 2020. 1, 2, 3, 4, 5, 6, 7, 8
- [2] Iro Armeni, Zhi-Yang He, JunYoung Gwak, Amir R Zamir, Martin Fischer, Jitendra Malik, and Silvio Savarese. 3d scene graph: A structure for unified semantics, 3d space, and camera. In *CVPR*, pages 5664–5673, 2019. 7
- [3] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 3
- [4] Dave Zhenyu Chen, Angel X Chang, and Matthias Nießner. Scanrefer: 3d object localization in rgb-d scans using natural language. In *ECCV*, 2020. 1, 2, 3, 5, 6, 7
- [5] Dave Zhenyu Chen, Ali Gholami, Matthias Nießner, and Angel X Chang. Scan2cap: Context-aware dense captioning in rgb-d scans. In *CVPR*, 2021. 7
- [6] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pre-training from pixels. In *International Conference on Machine Learning*, pages 1691–1703. PMLR, 2020. 7
- [7] 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. 2, 3, 5
- [8] Jiajun Deng, Zhengyuan Yang, Tianlang Chen, Wengang Zhou, and Houqiang Li. Transvg: End-to-end visual grounding with transformers. *arXiv preprint arXiv:2104.08541*, 2021. 2
- [9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 3, 5
- [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, 4
- [11] 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. *arXiv preprint arXiv:2103.16381*, 2021. 2
- [12] 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. 1
- [13] Priya Goyal, Dhruv Mahajan, Abhinav Gupta, and Ishan Misra. Scaling and benchmarking self-supervised visual representation learning. In *ICCV*, pages 6391–6400, 2019. 7
- [14] Ji Hou, Angela Dai, and Matthias Nießner. 3d-sis: 3d semantic instance segmentation of rgb-d scans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4421–4430, 2019. 2
- [15] Ronghang Hu, Amanpreet Singh, Trevor Darrell, and Marcus Rohrbach. Iterative answer prediction with pointer-augmented multimodal transformers for textvqa. In *CVPR*, pages 9992–10002, 2020. 5
- [16] Ronghang Hu, Huazhe Xu, Marcus Rohrbach, Jiashi Feng, Kate Saenko, and Trevor Darrell. Natural language object retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4555–4564, 2016. 1
- [17] Pin-Hao Huang, Han-Hung Lee, Hwann-Tzong Chen, and Tyng-Luh Liu. Text-guided graph neural networks for referring 3d instance segmentation. In *AAAI*, 2021. 1, 2, 3, 5, 6, 8
- [18] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3128–3137, 2015. 2, 4
- [19] 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. 1, 2
- [20] Kangsoo Kim, Mark Billingham, Gerd Bruder, Henry Been-Lirn Duh, and Gregory F Welch. Revisiting trends in augmented reality research: A review of the 2nd decade of ismar (2008–2017). *IEEE transactions on visualization and computer graphics*, 24(11):2947–2962, 2018. 1
- [21] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5
- [22] Bernard C Kress and William J Cummings. 11-1: Invited paper: Towards the ultimate mixed reality experience: Hologens display architecture choices. In *SID symposium digest of technical papers*, volume 48, pages 127–131. Wiley Online Library, 2017. 1
- [23] 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(1):32–73, 2017. 3
- [24] Jason Ku, Melissa Mozifian, Jungwook Lee, Ali Harakeh, and Steven L Waslander. Joint 3d proposal generation and object detection from view aggregation. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1–8. IEEE, 2018. 2
- [25] Jean Lahoud and Bernard Ghanem. 2d-driven 3d object detection in rgb-d images. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4622–4630, 2017. 2
- [26] Kunpeng Li, Yulun Zhang, Kai Li, Yuanyuan Li, and Yun Fu. Visual semantic reasoning for image-text matching. In *ICCV*, pages 4654–4662, 2019. 2, 4
- [27] Ruotian Luo and Gregory Shakhnarovich. Comprehension-guided referring expressions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7102–7111, 2017. 3
- [28] Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L Yuille, and Kevin Murphy. Generation and comprehension of unambiguous object descriptions. In

- Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 11–20, 2016. 2
- [29] Vivek Mittal. Attngrinder: Talking to cars with attention. In *European Conference on Computer Vision*, pages 62–73. Springer, 2020. 1
- [30] 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
- [31] 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. *International journal of computer vision*, 123(1):74–93, 2017. 1
- [32] Charles R Qi, Xinlei Chen, Or Litany, and Leonidas J Guibas. Invotenet: Boosting 3d object detection in point clouds with image votes. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4404–4413, 2020. 2
- [33] Charles R Qi, Or Litany, Kaiming He, and Leonidas J Guibas. Deep hough voting for 3d object detection in point clouds. In *ICCV*, pages 9277–9286, 2019. 2, 3, 5, 6
- [34] Charles R Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J Guibas. Frustum pointnets for 3d object detection from rgb-d data. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 918–927, 2018. 2
- [35] Charles R Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *arXiv preprint arXiv:1706.02413*, 2017. 3, 5
- [36] Junha Roh, Karthik Desingh, Ali Farhadi, and Dieter Fox. Languagerefer: Spatial-language model for 3d visual grounding. *arXiv preprint arXiv:2107.03438*, 2021. 2
- [37] Arka Sadhu, Kan Chen, and Ram Nevatia. Zero-shot grounding of objects from natural language queries. In *ICCV*, pages 4694–4703, 2019. 2
- [38] Arka Sadhu, Kan Chen, and Ram Nevatia. Video object grounding using semantic roles in language description. In *CVPR*, pages 10417–10427, 2020. 1
- [39] Amaia Salvador, Xavier Giró-i Nieto, Ferran Marqués, and Shin’ichi Satoh. Faster r-cnn features for instance search. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 9–16, 2016. 3
- [40] Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, et al. Habitat: A platform for embodied ai research. In *ICCV*, pages 9339–9347, 2019. 1
- [41] Shuran Song and Jianxiong Xiao. Deep sliding shapes for amodal 3d object detection in rgb-d images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 808–816, 2016. 2
- [42] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017. 2, 3
- [43] Johanna Wald, Helisa Dharmo, Nassir Navab, and Federico Tombari. Learning 3d semantic scene graphs from 3d indoor reconstructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3961–3970, 2020. 3, 7
- [44] Chen Wang, Danfei Xu, Yuke Zhu, Roberto Martín-Martín, Cewu Lu, Li Fei-Fei, and Silvio Savarese. Densefusion: 6d object pose estimation by iterative dense fusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3343–3352, 2019. 2
- [45] Liwei Wang, Yin Li, Jing Huang, and Svetlana Lazebnik. Learning two-branch neural networks for image-text matching tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(2):394–407, 2018. 2, 4
- [46] Liwei Wang, Yin Li, and Svetlana Lazebnik. Learning deep structure-preserving image-text embeddings. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5005–5013, 2016. 1
- [47] 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. 1
- [48] Danfei Xu, Dragomir Anguelov, and Ashesh Jain. Pointfusion: Deep sensor fusion for 3d bounding box estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 244–253, 2018. 2
- [49] Zhengyuan Yang, Tianlang Chen, Liwei Wang, and Jiebo Luo. Improving one-stage visual grounding by recursive subquery construction. In *ECCV*, 2020. 2
- [50] Zhengyuan Yang, Boqing Gong, Liwei Wang, Wenbing Huang, Dong Yu, and Jiebo Luo. A fast and accurate one-stage approach to visual grounding. In *ICCV*, pages 4683–4693, 2019. 1, 2, 3
- [51] Zhengyuan Yang, Tushar Kumar, Tianlang Chen, Jingsong Su, and Jiebo Luo. Grounding-tracking-integration. *IEEE Transactions on Circuits and Systems for Video Technology*, 2020. 1
- [52] Zhengyuan Yang, Yijuan Lu, Jianfeng Wang, Xi Yin, Dinei Florencio, Lijuan Wang, Cha Zhang, Lei Zhang, and Jiebo Luo. Tap: Text-aware pre-training for text-vqa and text-caption. In *CVPR*, 2021. 5
- [53] 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. 1, 2
- [54] Licheng Yu, Patrick Poirson, Shan Yang, Alexander C Berg, and Tamara L Berg. Modeling context in referring expressions. In *European Conference on Computer Vision*, pages 69–85. Springer, 2016. 1, 2
- [55] 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. *arXiv preprint arXiv:2103.01128*, 2021. 1, 2, 5, 6, 8

- [56] Richard Zhang, Phillip Isola, and Alexei A Efros. Split-brain autoencoders: Unsupervised learning by cross-channel prediction. In *CVPR*, pages 1058–1067, 2017. [7](#)
- [57] Zhu Zhang, Zhou Zhao, Yang Zhao, Qi Wang, Huasheng Liu, and Lianli Gao. Where does it exist: Spatio-temporal video grounding for multi-form sentences. In *CVPR*, pages 10668–10677, 2020. [1](#)
- [58] Yusheng Zhao, Junyu Luo, Tianrui Hui, Shaofei Huang, Aixi Zhang, Si Liu, et al. Transrefer3d: Entity-and-relation aware transformer for fine-grained 3d visual grounding. *arXiv preprint arXiv:2108.02388*, 2021. [2](#)