ScanRefer: 3D Object Localization in RGB-D Scans using Natural Language

Dave Zhenyu Chen[1] Angel X. Chang[2] Matthias Nießner[1]

[1]Technical University of Munich [2]Simon Fraser University

Fig. 1: We introduce the task of object localization in 3D scenes using natural language. Given as input a 3D scene and a natural language expression, we predict the bounding box for the target 3D object (right). The counterpart 2D task (left) does not capture the physical extent of the 3D objects.

Abstract. We introduce the task of 3D object localization in RGB-D scans using natural language descriptions. As input, we assume a point cloud of a scanned 3D scene along with a free-form description of a specified target object. To address this task, we propose **ScanRefer**, learning a fused descriptor from 3D object proposals and encoded sentence embeddings. This fused descriptor correlates language expressions with geometric features, enabling regression of the 3D bounding box of a target object. We also introduce the ScanRefer dataset, containing 51,583 descriptions of 11,046 objects from 800 ScanNet [9] scenes. ScanRefer is the first large-scale effort to perform object localization via natural language expression directly in 3D [1].

1 Introduction

In recent years, there has been tremendous progress in both semantic understanding and localization of objects in 2D images from natural language (also known as visual grounding). Datasets such as ReferIt [28], RefCOCO [71], and

[1] Project page: `https://daveredrum.github.io/ScanRefer/`

Flickr30K Entities [47] have enabled the development of various methods for visual grounding in 2D [23, 22, 39]. However, these methods and datasets are restricted to 2D images, where object localization fails to capture the true 3D extent of an object (see Fig. 1, left). This is a limitation for applications ranging from assistive robots to AR/VR agents where understanding the global 3D context and the physical size is important, e.g., finding objects in large spaces, interacting with them, and understanding their spatial relationships. Early work by Kong et al. [31] looked at coreference in 3D, but was limited to single-view RGB-D images.

In this work, we address these shortcomings by proposing the task of object localization using natural language directly in 3D space. Specifically, we develop a neural network architecture that localizes objects in 3D point clouds given natural language descriptions referring to the underlying objects; i.e., for a given text description in a 3D scene, we predict a corresponding 3D bounding box matching the best-described object. To facilitate the task, we collect the ScanRefer dataset, which provides natural language descriptions for RGB-D scans in ScanNet [9]. In total, we acquire $51,583$ descriptions of $11,046$ objects. To the best of our knowledge, our ScanRefer dataset is the first large-scale effort that combines 3D scene semantics and free-form descriptions. In summary, our contributions are as follows:

- We introduce the task of localizing objects in 3D environments using natural language descriptions.
- We provide the ScanRefer dataset containing $51,583$ human-written free-form descriptions of $11,046$ objects in 3D scans.
- We propose a neural network architecture for localization based on language descriptions that directly fuses features from 2D images and language expressions with 3D point cloud features.
- We show that our end-to-end method outperforms the best 2D visual grounding method that simply backprojects its 2D predictions to 3D by a significant margin (9.04 Acc@0.5IoU vs. 22.39 Acc@0.5IoU).

2 Related Work

Grounding Referring Expressions in Images. There has been much work connecting images to natural language descriptions across tasks such as image captioning [27, 26, 59, 64], text-to-image retrieval [61, 25], and visual grounding [23, 39, 70]. The task of visual grounding (with variants also known as referring expression comprehension or phrase localization) is to localize a region described by a given referring expression, the query. Localization can be specified by a 2D bounding box [28, 47, 39] or a segmentation mask [22], with the input description being short phrases [28, 47] or more complex descriptions [39]. Recently, Acharya et al [1] proposed visual query detection where the input is a question. The focus of our work is to lift this task to 3D, focusing on complex descriptions that can localize an unique object in a scene.

dataset	#objects	#expressions	AvgLeng	data format	3D context
ReferIt [28]	96,654	130,364	3.51	image	-
RefCOCO [71]	50,000	142,209	3.50	image	-
Google RefExp [39]	49,820	95,010	8.40	image	-
SUN-Spot [41]	3,245	7,990	14.04	image	depth
REVERIE [52]	4,140	21,702	18.00	image	panoramic image
ScanRefer (ours)	**11,046**	**51,583**	**20.27**	**3D scan**	**depth, size, location, etc.**

Table 1: Comparison of referring expression datasets in terms of the number of objects (#objects), number of expressions (#expressions), average lengths of the expressions, data format and the 3D context.

Existing methods focus on predicting 2D bounding boxes [23, 55, 61, 60, 46, 71, 70, 12, 37] and some predict segmentation masks [22, 35, 33, 40, 69, 6]. A two-stage pipeline is common, where first an object detector, either unsupervised [74] or pretrained [54], is used to propose regions of interest, and then the regions are ranked by similarity to the query, with the highest scoring region provided as the final output. Other methods address the referring expression task with a single stage end-to-end network [22, 43, 68]. There are also approaches that incorporate syntax [36, 17], use graph attention networks [62, 66, 67], speaker-listener models [39, 72], weakly supervised methods [63, 73, 11] or tackle zero-shot settings for unseen nouns [56].

However, all these methods operate on 2D image datasets [47, 28, 71]. A recent dataset [41] integrates RGB-D images but lacks the complete 3D context beyond a single image. Qi et al. [52] study referring expressions in an embodied setting, where semantic annotations are projected from 3D to 2D bounding boxes on images observed by an agent. Our contribution is to lift NLP tasks to 3D by introducing the first large-scale effort that couples free-form descriptions to objects in 3D scans. Tab. 1 summarizes the difference between our ScanRefer dataset and existing 2D datasets.

Object Detection in 3D. Recent work on 3D object detection on volumetric grids [20, 19, 32, 42, 13] has been applied to several 3D RGB-D datasets [58, 9, 4]. As an alternative to regular grids, point-based methods, such as PointNet [50] or PointNet++ [51], have been used as backbones for 3D detection and/or object instance segmentation [65, 14]. Recently, Qi et al. [49] introduced VoteNet, a 3D object detection method for point clouds based on Hough Voting [21]. Our approach extracts geometric features in a similar fashion, but backprojects 2D feature information since the color signal is useful for describing 3D objects with natural language.

3D Vision and Language. Vision and language research is gaining popularity in image domains (e.g., image captioning [26, 59, 64, 38], image-text matching [15, 30, 34, 24, 16], and text-to-image generation [53, 16, 57]), but there is little work on vision and language in 3D. Chen et al. [7] learn a joint embedding of 3D shapes from ShapeNet [5] and corresponding natural language descriptions. Achlioptas et al. [3] disambiguate between different objects using language. Recent work has started to investigate grounding of language to 3D by identifying

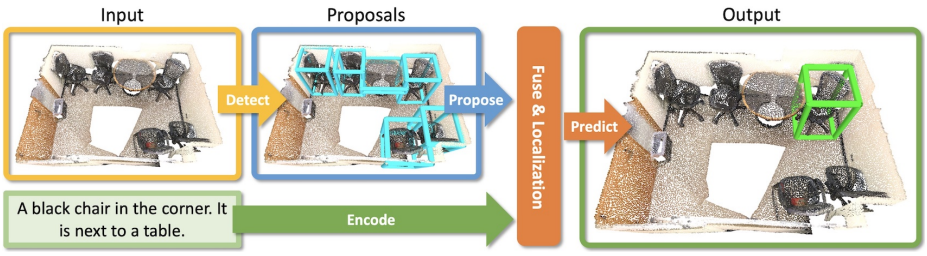


Fig. 2: Our task: ScanRefer takes as input a 3D scene point cloud and a description of an object in the scene, and predicts the object bounding box.

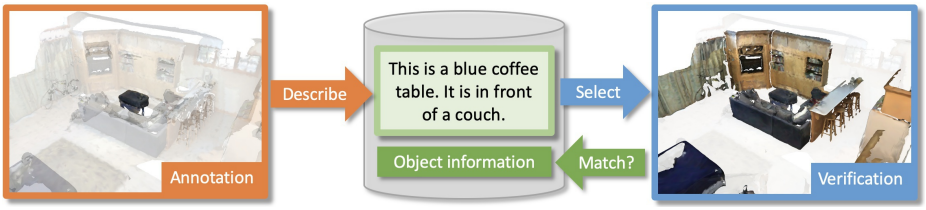


Fig. 3: Our data collection pipeline. The annotator writes a description for the focused object in the scene. Then, a verifier selects the objects that match the description. The selected object is compared with the target object to check that it can be uniquely identified by the description.

3D bounding boxes of target objects for simple arrangements of primitive shapes of different colors [48]. Instead of focusing on isolated objects, we consider large 3D RGB-D reconstructions that are typical in semantic 3D scene understanding. A closely related work by Kong et al. [31] studied the problem of coreference in text description of single-view RGB-D images of scenes, where they aimed to connect noun phrases in a scene description to 3D bounding boxes of objects. Concurrent with this work, Achlioptas et al. [2] introduces a new dataset and task that focuses on disambiguating objects from the same category with known localizations.

3 Task

We introduce the task of object localization in 3D scenes using natural language (Fig. 2). The input is a 3D scene and free-form text describing an object in the scene. The scene is represented as a point cloud with additional features such as colors and normals for each point. The goal is to predict the 3D bounding box of the object that matches the input description.

4 Dataset

The ScanRefer dataset is based on ScanNet [9] which is composed of 1,613 RGB-D scans taken in 806 unique indoor environments. We provide 5 descriptions for

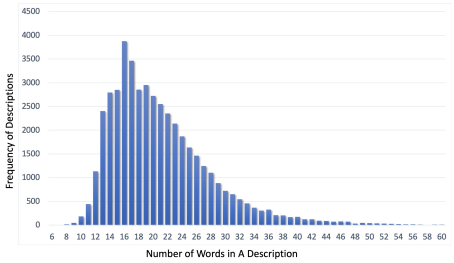


Fig. 4: Description lengths

Number of descriptions	51,583
Number of scenes	800
Number of objects	11,046
Number of objects per scene	13.81
Number of descriptions per scene	64.48
Number of descriptions per object	4.67
Size of vocabulary	4,197
Average length of descriptions	20.27

Table 2: ScanRefer dataset statistics.

(a) (b) (c) (d) (e)

Fig. 5: Word clouds of terms for (a) object names (b) colors (c) shapes (d) sizes, and (e) spatial relations for the ScanRefer dataset. Bigger fonts indicate more frequent terms in the descriptions.

each object in each scene, focusing on complete coverage of all objects that are present in the reconstruction. Here, we summarize the annotation process and statistics of our dataset (see supplement for more details).

4.1 Data Collection

We deploy a web-based annotation interface on Amazon Mechanical Turk (AMT) to collect object descriptions in the ScanNet scenes. The annotation pipeline consists of two stages: i) description collection, and ii) verification (Fig. 3). From each scene, we select objects to annotate by restricting to indoor furniture categories and excluding structural objects such as "Floor" and "Wall". We manually check the selected objects are recognizable and filter out objects with reconstructions that are too incomplete or hard to identify.

Annotation The 3D web-based UI shows each object in context. The workers see all objects other than the target object faded out and a set of captured image frames to compensate for incomplete details in the reconstructions. The initial viewpoint is random but includes the target object. Camera controls allow for adjusting the camera view to better examine the target object. We ask the annotator to describe the appearance of the target and its spatial location relative to other objects. To ensure the descriptions are informative, we require the annotator to provide at least two full sentences. We batch and randomize the tasks so that each object is described by five different workers.

Verification We recruit trained workers (students) to verify that the descriptions are discriminative and correct. Verifiers are shown the 3D scene and a description, and are asked to select the objects (potentially multiple) in the

1. There is a brown wooden chair placed right against the wall.
2. This is a triangular shape table. The table is near the armchair.
3. The little nightstand. The nightstand is on the right of the bed.
4. This is a short trash can. It is in front of a taller trash can.
5. The couch is the biggest one below the picture. The couch has three seats and is brown.
6. This is a gray desk chair. This chair is the last one on the side closest to the open door.
7. The kitchen counter is covering the lower cabinets. The kitchen counter is under the upper cabinets that are mounted above.
8. This is a round bar stool. It is third from the wall.

Table 3: Examples from our dataset illustrating different types of phrases such as attributes (1-8) and parts (5), comparatives (4), superlatives (5), intra-class spatial relations (6), inter-class spatial relations (7) and ordinal numbers (8).

scene that match the description. Descriptions that result in the wrong object or multiple objects are filtered out. Verifiers also correct spelling and wording issues in the description when necessary. We filter out 2,823 invalid descriptions that do not match the target objects and fix writing issues for 2,129 descriptions.

4.2 Dataset Statistics

We collected 51,583 descriptions for 800 ScanNet scenes[2]. On average, there are 13.81 objects, 64.48 descriptions per scene, and 4.67 descriptions per object after filtering (see Tab. 2 for basic statistics, Tab. 3 for sample descriptions, and Fig. 4 for the distribution of the description lengths). The descriptions are complex and diverse, covering over 250 types of common indoor objects, and exhibiting interesting linguistic phenomena. Due to the complexity of the descriptions, one of the key challenges of our task is to determine what parts of the description describe the target object, and what parts describe neighboring objects. Among those descriptions, 41,034 mention object attributes such as color, shape, size, etc. We find that many people use spatial language (98.7%), color (74.7%), and shape terms (64.9%). In contrast, only 14.2% of the descriptions convey size information. Fig 5 shows commonly used object names and attributes. Tab. 3 shows interesting expressions, including comparatives ("taller") and superlatives ("the biggest one"), as well as phrases involving ordinals such as "third from the wall". Overall, there are 672 and 2,734 descriptions with comparative and superlative phrases. We provide more detailed statistics in the supplement.

5 Method

Our architecture consists of two main modules: 1) detection & encoding; 2) fusion & localization (Fig. 6). The detection & encoding module encodes the input point cloud and description, and outputs the object proposals and the language embedding, which are fed into the fusion module to mask out invalid

[2] 6 scenes are excluded since they do not contain any objects to describe

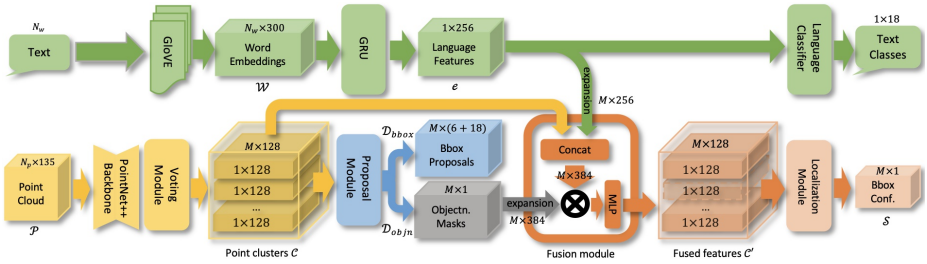


Fig. 6: ScanRefer architecture: The PointNet++ [51] backbone takes as input a point cloud and aggregates it to high-level point feature maps, which are then clustered and fused as object proposals by a voting module similar to Qi et al. [49]. Object proposals are masked by the objectness predictions, and then fused with the sentence embedding of the input descriptions, which is obtained by a GloVE [45] + GRU [8] embedding. In addition, an extra language-to-object classifier serves as a proxy loss. We apply a softmax function in the localization module to output the confidence scores for the object proposals.

object proposals and produce the fused features. Finally, the object proposal with the highest confidence predicted by the localization module is chosen as the final output.

5.1 Data Representations

Point clouds We randomly sample N_P vertices of one scan from ScanNet as the input point cloud $\mathcal{P} = \{(p_i, f_i)\}$, where $p_i \in \mathcal{R}^3$ represents the point coordinates in 3D space and f_i stands for additional point features such as colors and normals. Note that the point coordinates p_i provides only geometrical information and does not contain other visual information such as color and texture. Since descriptions of objects do refer to attributes such as color and texture, we incorporate visual appearance by adapting the feature projection scheme in Dai et al. [10] to project multi-view image features $v_i \in \mathcal{R}^{128}$ to the point cloud. The image features are extracted using a pre-trained ENet [44]. Following Qi et al. [49], we also append the height of the point from the ground and normals to the new point features $f_i' \in \mathcal{R}^{135}$. The final point cloud data is prepared offline as $\mathcal{P}' = \{(p_i, f_i')\} \in \mathcal{R}^{N_P \times 135}$. We set N_P to $40,000$ in our experiments.
Descriptions We tokenize the input description with SpaCy [18] and the N_W tokens to 300-dimensional word embedding vectors $\mathcal{W} = \{w_j\} \in \mathcal{R}^{N_W \times 300}$ using pretrained GloVE word embeddings [45].

5.2 Network Architecture

Our method takes as input the preprocessed point cloud $\mathcal{P}'$ and the word embedding sequence $\mathcal{W}$ representing the input description and outputs the 3D

bounding box for the proposal which is most likely referred to by the input description. Conceptually, our localization pipeline consists of the following four stages: detection, encoding, fusion and localization.

Detection As the first step in our network, we detect all probable objects in the given point cloud. To construct our detection module, we adapt the PointNet++ [51] backbone and the voting module in Qi et al. [49] to process the point cloud input and aggregate all object candidates to individual clusters. The output from the voting module is a set of point clusters $\mathcal{C} \in \mathcal{R}^{M\times 128}$ representing all object proposals with enriched point features, where M is the upper bound of the number of proposals. Next, the proposal module takes in the point clusters and processes those clusters to predict the objectness mask $\mathcal{D}_{\text{objn}} \in \mathcal{R}^{M\times 1}$ and the axis-aligned bounding boxes $\mathcal{D}_{\text{bbox}} \in \mathcal{R}^{M\times(6+18)}$ for all M proposals, where each $\mathcal{D}^i_{\text{bbox}} = (c_x, c_y, c_z, r_x, r_y, r_z, l)$ consists of the box center c, the box lengths r and a vector $l \in \mathcal{R}^{18}$ representing the semantic predictions.

Encoding The sequences of word embedding vectors of the input description are fed into a GRU cell [8] to aggregate the textual information. We take the final hidden state $e \in \mathcal{R}^{256}$ of the GRU cell as the final language embedding.

Fusion The outputs from the previous detection and encoding modules are fed into the fusion module (orange block in Fig. 6, see supplemental for details) to integrate the point features together with the language embeddings. Specifically, each feature vector $c_i \in \mathcal{R}^{128}$ in the point cluster $\mathcal{C}$ is concatenated with the language embedding $e \in \mathcal{R}^{256}$ as the extended feature vector, which is then masked by the predicted objectness mask $\mathcal{D}^i_{\text{objn}} \in \{0, 1\}$ and fused by a multi-layer perceptron as the the final fused cluster features $C' = \{c'_i\} \in \mathcal{R}^{M\times 128}$.

Localization The localization module aims to predict which of the proposed bounding boxes corresponds to the description. Point clusters with fused cluster features $\mathcal{C}' = \{c'_i\}$ are processed by a single layer perceptron to produce the raw scores of how likely each box is the target box. We use a softmax function to squash all the raw scores into the interval of $[0, 1]$ as the localization confidences $S = \{s_i\} \in \mathcal{R}^{M\times 1}$ for the proposed M bounding boxes.

5.3 Loss Function

Localization loss For the predicted localization confidence $s_i \in [0, 1]$ for object proposal $\mathcal{D}^i_{\text{bbox}}$, the target label is represented as $t_i \in \{0, 1\}$. Following the strategy of Yang et al. [68], we set the label t_j for the j^{th} box that has the highest IoU score with the ground truth box as 1 and others as 0. We then use a cross-entropy loss as the localization loss $\mathcal{L}_{\text{loc}} = -\sum_{i=1}^{M} t_i \log(s_i)$.

Object detection loss We use the same detection loss $\mathcal{L}_{det}$ as introduced in Qi et al. [49] for object proposals $\mathcal{D}^i_{\text{bbox}}$ and $\mathcal{D}^i_{\text{objn}}$: $\mathcal{L}_{\text{det}} = \mathcal{L}_{\text{vote-reg}} + 0.5\mathcal{L}_{\text{objn-cls}} + \mathcal{L}_{\text{box}} + 0.1\mathcal{L}_{\text{sem-cls}}$, where $\mathcal{L}_{\text{vote-reg}}$, $\mathcal{L}_{\text{objn-cls}}$, $\mathcal{L}_{\text{box}}$ and $\mathcal{L}_{\text{sem-cls}}$ represent the vote regression loss (defined in Qi et al. [49]), the objectness binary classification loss, box regression loss and the semantic classification loss for the 18 ScanNet benchmark classes, respectively. We ignore the bounding box orientations in our task and simplify $\mathcal{L}_{\text{box}}$ as $\mathcal{L}_{\text{box}} = \mathcal{L}_{\text{center-reg}} + 0.1\mathcal{L}_{\text{size-cls}} + \mathcal{L}_{\text{size-reg}}$, where

$\mathcal{L}_{\text{center-reg}}$, $\mathcal{L}_{\text{size-cls}}$ and $\mathcal{L}_{\text{size-reg}}$ are used for regressing the box center, classifying the box size and regressing the box size, respectively. We refer readers to Qi et al. [49] for more details.

Language to object classification loss To further supervise the training, we include an object classification loss based on the input description. We consider the 18 ScanNet benchmark classes (excluding the label "Floor" and "Wall"). The language to object classification loss $\mathcal{L}_{\text{cls}}$ is a multi-class cross-entropy loss.

Final loss The final loss is a linear combination of the localization loss, object detection loss and the language to object classification loss: $\mathcal{L} = \alpha\mathcal{L}_{\text{loc}} + \beta\mathcal{L}_{\text{det}} + \gamma\mathcal{L}_{\text{cls}}$, where α, β and γ are the weights for the individual loss terms. After fine-tuning on the validation split, we set those weights to 0.1, 10, and 1 in our experiments to ensure the loss terms are roughly of the same magnitude.

5.4 Training and Inference

Training During training, the detection and encoding modules propose object candidates as point clusters, which are then fed into the fusion and localization modules to fuse the features from the previous module and predict the final bounding boxes. We train the detection backbone end-to-end with the detection loss. In the localization module, we use a softmax function to compress the raw scores to $[0, 1]$. The higher the predicted confidence is, the more likely the proposal will be chosen as output. To filter out invalid object proposals, we use the predicted objectness mask to ensure that only positive proposals are taken into account. We set the maximum number of proposals M to 256 in practice.

Inference Since there can be overlapping detections, we apply a non-maximum suppression module to suppress those overlapping proposals in the inference step. The remaining object proposals are fed into the localization module to predict the final score for each proposal. The number of object proposals is less than the upper bound M in the training step.

Implementation Details We implement our architecture using PyTorch and train the model end-to-end using ADAM [29] with a learning rate of 1e−3. We train the model for roughly $130,000$ iterations until convergence. To avoid overfitting, we set the weight decay factor to 1e−5 and apply data augmentations to our training data. For point clouds, we apply rotation about all three axes by a random angle in $[-5°, 5°]$ and randomly translate the point cloud within 0.5 meters in all directions. We rotate around all axes (not just up), since the ground alignment in ScanNet is imperfect.

6 Experiments

Train/Val/Test Split. Following the official ScanNet [9] split, we split our data into train/val/test sets with 36,665, 9,508 and 5,410 samples respectively, ensuring disjoint scenes for each split. Results and analysis are conducted on the val split (except for results in Tab. 4 bottom). The test set is hidden and will be reserved for the ScanRefer benchmark.

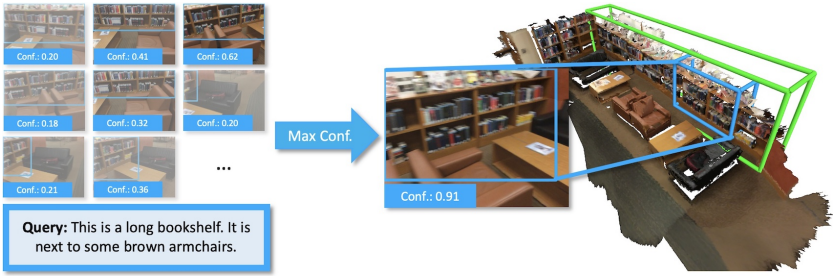


Fig. 7: Object localization in an image using a 2D grounding method and back-projecting the result to the 3D scene (blue box) vs. directly localizing in the 3D scene (green box). Grounding in 2D images suffers from the limited view of a single frame, which results in inaccurate 3D bounding boxes.

Metric. To evaluate the performance of our method, we measure the thresholded accuracy where the positive predictions have higher intersection over union (IoU) with the ground truths than the thresholds. Similar to work with 2D images, we use Acc@kIoU as our metric, where the threshold value k for IoU is set to 0.25 and 0.5 in our experiments.

Baselines. We design several baselines by 1) evaluating our language localization module on ground truth bounding boxes, 2) adapting 3D object detectors, and 3) adapting 2D referring methods to 3D using back-projection.

OracleCatRand & OracleRefer: To examine the difficulty of our task, we use an oracle with ground truth bounding boxes of objects, and predict the box by simply selecting a random box that matches the object category (OracleCatRand) or our trained fusion and localization modules (OracleRefer).

VoteNetRand & VoteNetBest: From the predicted object proposals of the VoteNet backbone [49], we select one of the bounding box proposals, either by selecting a box randomly with the correct semantic class label (VoteNetRand) or the best matching box given the ground truth (VoteNetBest). VoteNetBest provides an upper bound on how well the object detection component works for our task, while VoteNetRand provides a measure of whether additional information beyond the semantic label is required.

SCRC & One-stage: 2D image baselines for referring expression comprehension by extending SCRC [23] and One-stage [68] to 3D using back-projection. Since 2D referring expression methods operate on a single image frame, we construct a 2D training set by using the recorded camera pose associated with each annotation to retrieve the frame from the scan video with the closest camera pose. At inference time, we sample frames from the scans (using every 20th frame) and predict the target 2D bounding boxes in each frame. We then select the 2D bounding box with the highest confidence score from the bounding box candidates and project it to 3D using the depth map for that frame (see Fig. 7).

Ours: We compare our full end-to-end model against using a pretrained VoteNet backbone with a trained GRU [8] for selecting a matching bounding box.

	unique		multiple		overall	
	Acc@0.25	Acc@0.5	Acc@0.25	Acc@0.5	Acc@0.25	Acc@0.5
OracleCatRand (GT boxes + RandCat)	100.00	100.00	18.09	17.84	29.99	29.76
OracleRefer (GT boxes + GRU)	74.09	73.55	32.57	32.00	40.63	40.06
VoteNetRand (VoteNet[49] + RandCat)	34.34	19.35	5.73	2.81	10.00	5.28
VoteNetBest (VoteNet[49] + Best)	88.85	85.50	46.63	46.42	55.10	54.33
SCRC [23] + backproj	24.03	9.22	17.77	5.97	18.70	6.45
One-stage [68] + backproj	29.32	22.82	18.72	6.49	20.38	9.04
Ours (VoteNet[48] + GRU)	**77.33**	51.73	30.43	19.46	39.52	25.72
Ours (end-to-end)	76.33	**53.51**	**32.73**	**21.11**	**41.19**	**27.40**
Test results (ScanRefer benchmark)						
OracleRefer (GT boxes + GRU)	72.37	71.84	31.81	31.26	39.69	39.13
VoteNetBest (VoteNet[49] + Best)	86.78	83.85	45.54	45.33	53.82	53.07
Ours (VoteNet[48] + GRU)	**72.55**	**47.24**	32.90	19.16	41.79	25.45
Ours (end-to-end)	71.06	46.66	**35.17**	**20.92**	**43.22**	**26.69**

Table 4: Comparison of localization results obtained by our ScanRefer and baseline models. We measure percentage of predictions whose IoU with the ground truth boxes are greater than 0.25 and 0.5. We also report scores on "unique" and "multiple" subsets; unique means that there is only a single object of its class in the scene. We outperform all baselines by a significant margin.

6.1 Task Difficulty

To understand how informative the input description is beyond capturing the object category, we analyze the performance of the methods on "unique" and "multiple" subsets with 1,875 and 7,663 samples from val split, respectively. The "unique" subset contains samples where only one unique object from a certain category matches the description, while the "multiple" subset contains ambiguous cases where there are multiple objects of the same category. For instance, if there is only one refrigerator in a scene, it is sufficient to identify that the sentence refers to a refrigerator. In contrast, if there are multiple objects of the same category in a scene (e.g., chair), the full description must be taken into account. From the OracleCatRand baseline, we see that information from the description, other than the object category, is necessary to disambiguate between multiple objects (see Tab. 4 Acc@0.5IoU multiple). From the OracleRefer baseline, we see that using our fused language module, we are able to improve beyond over selecting a random object of the same category (multiple Acc@0.5IoU increases from 17.84% to 32.00%), but we often fail to identify the correct object category (unique Acc@0.5IoU drops from 100.0% to 73.55%).

6.2 Quantitative Analysis

We evaluate the performance of our model against baselines on the val and the hidden test split of ScanRefer which serves as the ScanRefer benchmark (see Tab. 4). Note that for all results using Ours and VoteNet for object proposal,

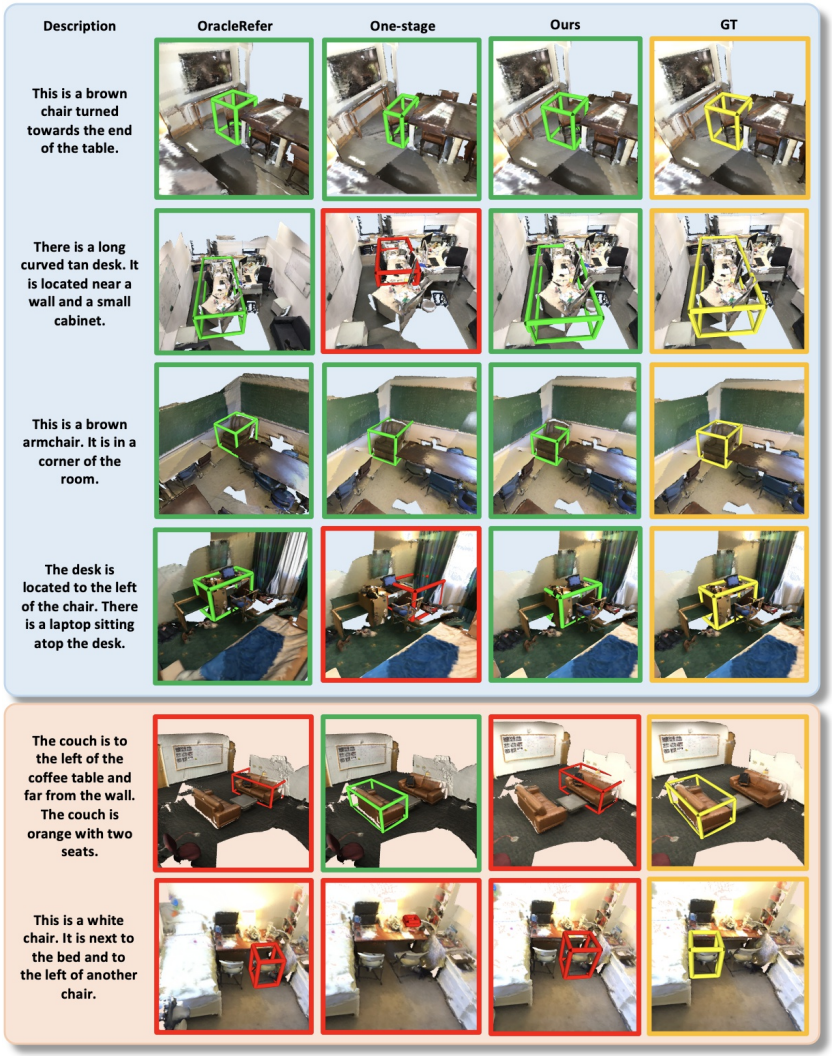


Fig. 8: Qualitative results from baseline methods and ScanRefer. Predicted boxes are marked green if they have an IoU score higher than 0.5, otherwise they are marked red. We show examples where our method produced good predictions (blue block) as well as failure cases (orange block). Image best viewed in color.

we take the average of 5 differently seeded subsamplings (of seed points and vote points) during inference (see supplemental for more details on experimental variance). Training the detection backbone jointly with the localization module (end-to-end) leads to a better performance when compared to the model trained separately (VoteNet[49] + GRU). However, as the accuracy gap between VoteNetBest and ours (end-to-end) indicates, there is still room for improving

	unique		multiple		overall	
	Acc@0.25	Acc@0.5	Acc@0.25	Acc@0.5	Acc@0.25	Acc@0.5
Ours (xyz)	63.98	43.57	29.28	18.99	36.01	23.76
Ours (xyz+rgb)	63.24	41.78	30.06	19.23	36.50	23.61
Ours (xyz+rgb+normals)	64.63	43.65	31.89	20.77	38.24	25.21
Ours (xyz+multiview)	77.20	52.69	32.08	19.86	40.84	26.23
Ours (xyz+multiview+normals)	**78.22**	52.38	33.61	20.77	**42.27**	26.90
Ours (xyz+lobjcls)	64.31	44.04	30.77	19.44	37.28	24.22
Ours (xyz+rgb+lobjcls)	65.00	43.31	30.63	19.75	37.30	24.32
Ours (xyz+rgb+normals+lobjcls)	67.64	46.19	32.06	**21.26**	38.97	26.10
Ours (xyz+multiview+lobjcls)	76.00	50.40	**34.05**	20.73	42.19	26.50
Ours (xyz+multiview+normals+lobjcls)	76.33	**53.51**	32.73	21.11	41.19	**27.40**

Table 5: Ablation study with different features. We measure the percentages of predictions whose IoU with the ground truth boxes are greater than 0.25 and 0.5. Unique means that there is only a single object of its class in the scene.

the match between language inputs and the visual signals. For the val split, we also include additional experiments on the 2D baselines and a comparison with VoteNetRand. With just category information, VoteNetRand is able to perform relatively well on the "unique" subset, but has trouble identifying the correct object in the "multiple" case. However, the gap between the VoteNetRand and OracleCatRand for the "unique" case shows that 3D object detection still need to be improved. Our method is able to improve over the bounding box predictions from VoteNetRand, and leverages additional information in the description to differentiate between ambiguous objects. It adapts better to the 3D context compared to the 2D methods (SCRC and One-stage) which is limited by the view of a single frame (see Fig. 7 and Fig. 8).

6.3 Qualitative Analysis

Fig. 8 shows results produced by OracleRefer, One-stage, and our method. The successful localization cases in the green boxes show our architecture can handle the semantic correlation between the scene contexts and the textual descriptions. In contrast, even provided with a pool of ground truth proposals, OracleRefer sometimes still fails to predict correct bounding boxes, while One-stage is limited by the single view and hence cannot produce accurate bounding boxes in 3D space. The failure case of OracleRefer suggests that our fusion & localization module can still be improved. Some failure cases of our method are displayed in the orange block in Fig. 8, indicating that our architecture cannot handle all spatial relations to distinguish between ambiguous objects.

6.4 Ablation Studies

We conduct an ablation study on our model to examine what components and point cloud features contribute to the performance (see Tab. 5).

Does a language-based object classifier help? To show the effectiveness of the extra supervision on input descriptions, we conduct an experiment with the language to object classifier (+lobjcls) and without. Architectures with a language to object classifier outperform ones without it. This indicates that it is helpful to predict the category of the target object based on the input description.
Do colors help? We compare our method trained with the geometry and multi-view image features (xyz+multiview+lobjcls) with a model trained with only geometry (xyz+lobjcls) and one trained with RGB values from the reconstructed meshes (xyz+rgb+lobjcls). ScanRefer trained with geometry and pre-processed multi-view image features outperforms the other two models. The performance of models with color information are higher than those that use only geometry.
Do other features help? We include normals from the ScanNet meshes to the input point cloud features and compare performance against networks trained without them. The additional 3D information improves performance. Our architecture trained with geometry, multi-view features, and normals (xyz+multiview+normals+lobjcls) achieves the best performance among all ablations.

7 Conclusion

In this work, we introduce the task of localizing a target object in a 3D point cloud using natural language descriptions. We collect the ScanReferdataset which contains 51,583 unique descriptions for 11,046 objects from 800 ScanNet [9] scenes. We propose an end-to-end method for localizing an object with a free-formed description as reference, which first proposes point clusters of interest and then matches them to the embeddings of the input sentence. Our architecture is capable of learning the semantic similarities of the given contexts and regressing the bounding boxes for the target objects. Overall, we hope that our new dataset and method will enable future research in the 3D visual language field.

Acknowledgements

We would like to thank the expert annotators Josefina Manieu Seguel and Rinu Shaji Mariam, all anonymous workers on Amazon Mechanical Turk and the student volunteers (Akshit Sharma, Yue Ruan, Ali Gholami, Yasaman Etesam, Leon Kochiev, Sonia Raychaudhuri) at Simon Fraser University for their efforts in building the ScanRefer dataset, and Akshit Sharma for helping with statistics and figures. This work is funded by Google (AugmentedPerception), the ERC Starting Grant Scan2CAD (804724), and a Google Faculty Award. We would also like to thank the support of the TUM-IAS Rudolf Mößbauer and Hans Fischer Fellowships (Focus Group Visual Computing), as well as the the German Research Foundation (DFG) under the Grant *Making Machine Learning on Static and Dynamic 3D Data Practical*. Angel X. Chang is supported by the Canada CIFAR AI Chair program. Finally, we thank Angela Dai for the video voice-over.

Bibliography

[1] Acharya, M., Jariwala, K., Kanan, C.: VQD: Visual query detection in natural scenes. In: Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL) (2019)

[2] Achlioptas, P., Abdelreheem, A., Xia, F., Elhoseiny, M., Guibas, L.: ReferIt3D: Neural listeners for fine-grained 3D object identification in real-world scenes. In: Proceedings of the European Conference on Computer Vision (ECCV) (2020)

[3] Achlioptas, P., Fan, J., Hawkins, R.X., Goodman, N.D., Guibas, L.J.: ShapeGlot: Learning language for shape differentiation. In: Proc. International Conference on Computer Vision (ICCV) (2019)

[4] Chang, A., Dai, A., Funkhouser, T., Halber, M., Niessner, M., Savva, M., Song, S., Zeng, A., Zhang, Y.: Matterport3D: Learning from RGB-D data in indoor environments. In: Proceedings of the International Conference on 3D Vision (3DV) (2017)

[5] Chang, A.X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., Xiao, J., Yi, L., Yu, F.: ShapeNet: An information-rich 3D model repository. arXiv preprint arXiv:1512.03012 (2015)

[6] Chen, D.J., Jia, S., Lo, Y.C., Chen, H.T., Liu, T.L.: See-through-text grouping for referring image segmentation. In: Proceedings of the IEEE International Conference on Computer Vision (2019)

[7] Chen, K., Choy, C.B., Savva, M., Chang, A.X., Funkhouser, T., Savarese, S.: Text2Shape: Generating shapes from natural language by learning joint embeddings. In: Proc. Asian Conference on Computer Vision (ACCV) (2018)

[8] Chung, J., Gulcehre, C., Cho, K., Bengio, Y.: Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555 (2014)

[9] Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M.: ScanNet: Richly-annotated 3D reconstructions of indoor scenes. In: Proc. Computer Vision and Pattern Recognition (CVPR) (2017)

[10] Dai, A., Nießner, M.: 3DMV: Joint 3D-multi-view prediction for 3D semantic scene segmentation. In: Proceedings of the European Conference on Computer Vision (2018)

[11] Datta, S., Sikka, K., Roy, A., Ahuja, K., Parikh, D., Divakaran, A.: Align2Ground: Weakly supervised phrase grounding guided by image-caption alignment. In: Proceedings of the IEEE International Conference on Computer Vision (2019)

[12] Dogan, P., Sigal, L., Gross, M.: Neural sequential phrase grounding (SeqGROUND). In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2019)

[13] Elich, C., Engelmann, F., Schult, J., Kontogianni, T., Leibe, B.: 3D-BEVIS: Birds-eye-view instance segmentation. arXiv preprint arXiv:1904.02199 (2019)
[14] Engelmann, F., Kontogianni, T., Leibe, B.: Dilated point convolutions: On the receptive field of point convolutions. arXiv preprint arXiv:1907.12046 (2019)
[15] Feng, F., Wang, X., Li, R.: Cross-modal retrieval with correspondence autoencoder. In: Proceedings of the 22nd ACM international conference on Multimedia. pp. 7–16. ACM (2014)
[16] Gu, J., Cai, J., Joty, S.R., Niu, L., Wang, G.: Look, imagine and match: Improving textual-visual cross-modal retrieval with generative models. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7181–7189 (2018)
[17] Hong, R., Liu, D., Mo, X., He, X., Zhang, H.: Learning to compose and reason with language tree structures for visual grounding. IEEE transactions on pattern analysis and machine intelligence (2019)
[18] Honnibal, M., Montani, I.: spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing (2017), to appear
[19] Hou, J., Dai, A., Nießner, M.: 3D-SIC: 3D semantic instance completion for RGB-D scans. arXiv preprint arXiv:1904.12012 (2019)
[20] Hou, J., Dai, A., Nießner, M.: 3D-SIS: 3D semantic instance segmentation of RGB-D scans. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2019)
[21] Hough, P.V.: Machine analysis of bubble chamber pictures. In: Conf. Proc. vol. 590914, pp. 554–558 (1959)
[22] Hu, R., Rohrbach, M., Darrell, T.: Segmentation from natural language expressions. In: European Conference on Computer Vision. pp. 108–124. Springer (2016)
[23] Hu, R., Xu, H., Rohrbach, M., Feng, J., Saenko, K., Darrell, T.: Natural language object retrieval. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4555–4564 (2016)
[24] Huang, Y., Wang, W., Wang, L.: Instance-aware image and sentence matching with selective multimodal LSTM. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2310–2318 (2017)
[25] Huang, Y., Wu, Q., Song, C., Wang, L.: Learning semantic concepts and order for image and sentence matching. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6163–6171 (2018)
[26] Karpathy, A., Fei-Fei, L.: Deep visual-semantic alignments for generating image descriptions. In: Proceedings of the IEEE conference on computer vision and pattern recognition (2015)
[27] Karpathy, A., Joulin, A., Fei-Fei, L.: Deep fragment embeddings for bidirectional image sentence mapping. In: Advances in neural information processing systems (2014)
[28] Kazemzadeh, S., Ordonez, V., Matten, M., Berg, T.: ReferItGame: Referring to objects in photographs of natural scenes. In: Proceedings of the 2014

conference on empirical methods in natural language processing (EMNLP). pp. 787–798 (2014)
[29] Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
[30] Kiros, R., Salakhutdinov, R., Zemel, R.S.: Unifying visual-semantic embeddings with multimodal neural language models. arXiv preprint arXiv:1411.2539 (2014)
[31] Kong, C., Lin, D., Bansal, M., Urtasun, R., Fidler, S.: What are you talking about? text-to-image coreference. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3558–3565 (2014)
[32] Lahoud, J., Ghanem, B., Pollefeys, M., Oswald, M.R.: 3D instance segmentation via multi-task metric learning. arXiv preprint arXiv:1906.08650 (2019)
[33] Li, R., Li, K., Kuo, Y.C., Shu, M., Qi, X., Shen, X., Jia, J.: Referring image segmentation via recurrent refinement networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2018)
[34] Li, S., Xiao, T., Li, H., Yang, W., Wang, X.: Identity-aware textual-visual matching with latent co-attention. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1890–1899 (2017)
[35] Liu, C., Lin, Z., Shen, X., Yang, J., Lu, X., Yuille, A.: Recurrent multimodal interaction for referring image segmentation. In: Proceedings of the IEEE International Conference on Computer Vision (2017)
[36] Liu, D., Zhang, H., Wu, F., Zha, Z.J.: Learning to assemble neural module tree networks for visual grounding. In: Proceedings of the IEEE International Conference on Computer Vision (2019)
[37] Liu, X., Wang, Z., Shao, J., Wang, X., Li, H.: Improving referring expression grounding with cross-modal attention-guided erasing. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2019)
[38] Lu, J., Xiong, C., Parikh, D., Socher, R.: Knowing when to look: Adaptive attention via a visual sentinel for image captioning. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 375–383 (2017)
[39] Mao, J., Huang, J., Toshev, A., Camburu, O., Yuille, A.L., Murphy, K.: Generation and comprehension of unambiguous object descriptions. In: Proceedings of the IEEE conference on computer vision and pattern recognition (2016)
[40] Margffoy-Tuay, E., Pérez, J.C., Botero, E., Arbeláez, P.: Dynamic multimodal instance segmentation guided by natural language queries. In: Proceedings of the European Conference on Computer Vision (ECCV) (2018)
[41] Mauceri, C., Palmer, M., Heckman, C.: SUN-Spot: An RGB-D dataset with spatial referring expressions. In: Proceedings of the IEEE International Conference on Computer Vision Workshops. pp. 0–0 (2019)
[42] Narita, G., Seno, T., Ishikawa, T., Kaji, Y.: PanopticFusion: Online volumetric semantic mapping at the level of stuff and things. arXiv preprint arXiv:1903.01177 (2019)

[43] Nguyen, A., Do, T.T., Reid, I., Caldwell, D.G., Tsagarakis, N.G.: Object captioning and retrieval with natural language. arXiv preprint arXiv:1803.06152 (2018)
[44] Paszke, A., Chaurasia, A., Kim, S., Culurciello, E.: ENet: A deep neural network architecture for real-time semantic segmentation. arXiv preprint arXiv:1606.02147 (2016)
[45] Pennington, J., Socher, R., Manning, C.: Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (2014)
[46] Plummer, B.A., Kordas, P., Hadi Kiapour, M., Zheng, S., Piramuthu, R., Lazebnik, S.: Conditional image-text embedding networks. In: Proceedings of the European Conference on Computer Vision (2018)
[47] Plummer, B.A., Wang, L., Cervantes, C.M., Caicedo, J.C., Hockenmaier, J., Lazebnik, S.: Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In: Proceedings of the IEEE international conference on computer vision (2015)
[48] Prabhudesai, M., Tung, H.Y.F., Javed, S.A., Sieb, M., Harley, A.W., Fragkiadaki, K.: Embodied language grounding with implicit 3D visual feature representations. arXiv preprint arXiv:1910.01210 (2019)
[49] Qi, C.R., Litany, O., He, K., Guibas, L.J.: Deep hough voting for 3D object detection in point clouds. In: Proceedings of the IEEE International Conference on Computer Vision (2019)
[50] Qi, C.R., Su, H., Mo, K., Guibas, L.J.: PointNet: Deep learning on point sets for 3D classification and segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 652–660 (2017)
[51] Qi, C.R., Yi, L., Su, H., Guibas, L.J.: PointNet++: Deep hierarchical feature learning on point sets in a metric space. In: Advances in neural information processing systems (2017)
[52] Qi, Y., Wu, Q., Anderson, P., Liu, M., Shen, C., Hengel, A.v.d.: REVERIE: Remote embodied visual referring expression in real indoor environments. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2020)
[53] Reed, S., Akata, Z., Yan, X., Logeswaran, L., Schiele, B., Lee, H.: Generative adversarial text to image synthesis. arXiv preprint arXiv:1605.05396 (2016)
[54] Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: Towards real-time object detection with region proposal networks. In: Advances in neural information processing systems (2015)
[55] Rohrbach, A., Rohrbach, M., Hu, R., Darrell, T., Schiele, B.: Grounding of textual phrases in images by reconstruction. In: European Conference on Computer Vision. pp. 817–834 (2016)
[56] Sadhu, A., Chen, K., Nevatia, R.: Zero-shot grounding of objects from natural language queries. In: Proceedings of the IEEE International Conference on Computer Vision (2019)
[57] Sharma, S., Suhubdy, D., Michalski, V., Kahou, S.E., Bengio, Y.: ChatPainter: Improving text to image generation using dialogue. arXiv preprint arXiv:1802.08216 (2018)

[58] Song, S., Lichtenberg, S.P., Xiao, J.: SUN RGB-D: A RGB-D scene understanding benchmark suite. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 567–576 (2015)
[59] Vinyals, O., Toshev, A., Bengio, S., Erhan, D.: Show and tell: A neural image caption generator. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3156–3164 (2015)
[60] Wang, L., Li, Y., Huang, J., Lazebnik, S.: Learning two-branch neural networks for image-text matching tasks. IEEE Transactions on Pattern Analysis and Machine Intelligence (2018)
[61] Wang, L., Li, Y., Lazebnik, S.: Learning deep structure-preserving image-text embeddings. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5005–5013 (2016)
[62] Wang, P., Wu, Q., Cao, J., Shen, C., Gao, L., Hengel, A.v.d.: Neighbourhood watch: Referring expression comprehension via language-guided graph attention networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2019)
[63] Xiao, F., Sigal, L., Jae Lee, Y.: Weakly-supervised visual grounding of phrases with linguistic structures. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2017)
[64] Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., Zemel, R., Bengio, Y.: Show, attend and tell: Neural image caption generation with visual attention. In: International conference on machine learning. pp. 2048–2057 (2015)
[65] Yang, B., Wang, J., Clark, R., Hu, Q., Wang, S., Markham, A., Trigoni, N.: Learning object bounding boxes for 3D instance segmentation on point clouds. arXiv preprint arXiv:1906.01140 (2019)
[66] Yang, S., Li, G., Yu, Y.: Cross-modal relationship inference for grounding referring expressions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2019)
[67] Yang, S., Li, G., Yu, Y.: Dynamic graph attention for referring expression comprehension. In: Proceedings of the IEEE International Conference on Computer Vision (2019)
[68] Yang, Z., Gong, B., Wang, L., Huang, W., Yu, D., Luo, J.: A fast and accurate one-stage approach to visual grounding. In: Proceedings of the IEEE International Conference on Computer Vision (2019)
[69] Ye, L., Rochan, M., Liu, Z., Wang, Y.: Cross-modal self-attention network for referring image segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2019)
[70] Yu, L., Lin, Z., Shen, X., Yang, J., Lu, X., Bansal, M., Berg, T.L.: MAttNet: Modular attention network for referring expression comprehension. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2018)
[71] Yu, L., Poirson, P., Yang, S., Berg, A.C., Berg, T.L.: Modeling context in referring expressions. In: European Conference on Computer Vision. pp. 69–85. Springer (2016)

[72] Yu, L., Tan, H., Bansal, M., Berg, T.L.: A joint speaker-listener-reinforcer model for referring expressions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2017)
[73] Zhao, F., Li, J., Zhao, J., Feng, J.: Weakly supervised phrase localization with multi-scale anchored transformer network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2018)
[74] Zitnick, C.L., Dollár, P.: Edge boxes: Locating object proposals from edges. In: European conference on computer vision. pp. 391–405. Springer (2014)

Supplementary Material

Fig. 9: ScanRefer localizes objects in a scene given a language description as input. In many cases, including this example, there are multiple objects from the same category in a single scene which makes the problem challenging and interesting at the same time.

In this supplementary material, we provide addition details on the data collection and statistic of the ScanRefer dataset (Section A); we also provide implementation details of our localization network (Section B), as well as additional quantitative (Section C) and qualitative comparisons (Section D).

A Dataset

A.1 Statistics

We present the distribution of categories of the ScanRefer dataset in Fig. 10. ScanRefer provides a large coverage of furniture (e.g., chair, table, cabinet, bed, etc.) in indoor environments with various sizes, colors, materials, and locations. We use the same category names as in the original ScanNet dataset [9]. In total, we annotate 11,046 objects from 265 categories from ScanNet [9]. Following the ScanNet voxel labeling task, we aggregate these finer-grained categories into 17 coarse categories and group the remaining object types into "Others" for a total of 18 object categories that we use to train the language-based object classifier.

Fig. 11 shows the distribution of finer-grained objects in the category "Others". For each of the 18 coarse categories, Fig. 12 shows the average and maximum number of objects for that category in a scene in which an object of that category appears. For instance, for scenes that contains a bed, the average number of beds is 1.22 and the maximum is 3.

We also provide detailed statistics in our training and validation splits in Tab. 6. To further address the difficulty of our task, we present additional details about the "unique" and "multiple" subsets in Tab. 7. The "unique" subset consists of cases where there is just one unique object of that category (from the 18 ScanNet classes), in the scene. In these cases, the object can be localized

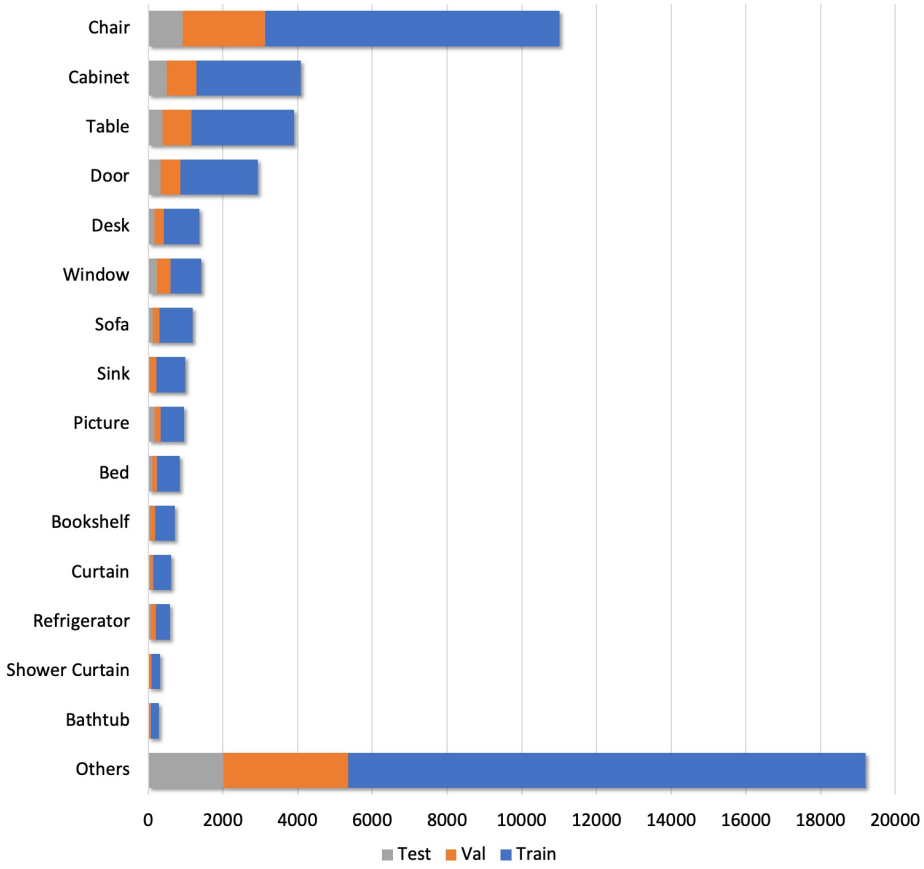


Fig. 10: Distribution of categories of objects in the ScanRefer dataset with annotated language descriptions.

	Train	Val	Test	Total
Number of descriptions	36,665	9,508	5,410	51,583
Number of scenes	562	141	97	800
Number of objects	7,875	2,068	1,103	11,046
Number of objects per scene	14.01	14.67	11.37	14.14
Number of descriptions per scene	65.24	67.43	55.77	65.68
Number of descriptions per object	4.66	4.60	4.90	4.64

Table 6: ScanRefer dataset statistics on Train and Val splits.

(assuming perfect object detection) just by identifying the semantic class of the target object from the description (e.g., localizing the table in the scene Fig. 9). The "multiple" subset refers to cases where there are multiple objects of the

Number of objects per scene	Unique	Multiple	Overall
total	3.00	11.81	14.14
same category as the target object	1.00	4.96	2.98

Table 7: Average number of objects (per scene) for the "Unique" and "Multiple" subsets of the ScanRefer dataset. Assuming ground truth bounding boxes, there are on average 14 different objects for to disambiguate between. For the "Multiple" subset, there are on average 5 objects to disambiguate between even if we could match the semantic class perfectly.

same category as the target object in the scene, thus requiring disambiguation between multiple objects of the same time (e.g., localizing a specific chair in the scene in Fig. 9). As shown in Tab. 7, since there are on average more objects of the same category as the target object in the "multiple" subset than in the "unique", it is more challenging to correctly localize the target object in the "multiple" subset.

A.2 Collection Details

In this section, we provide more details of the data annotation and verification processes of ScanRefer. The data collection took place over one month and involved 1,929 AMT workers. Together, the description collection and verification took around 4,984 man hours in total.

Annotation We deploy our web-based annotation application on Amazon Mechanical Turk (AMT) to collect object descriptions in the reconstructed RGB-D scans, as shown in Fig. 13a. To ensure that the initial descriptions are written in proper English, we restrict the workers to be from the United States, the United Kingdom, Canada, and Australia. The workers are asked to finish a batch of 5 description tasks within a time limit of 2 hours once the assignment is accepted on AMT. To ensure the descriptions are diverse and linguistically rich, we require that each description consists of at least two sentences. Before the annotation task begins, the AMT workers are also presented with the instructions shown in Fig. 13b. We request that the workers provide the following information in the descriptions:

- The appearance of the object such as shape, color, material and so on.
- The location of that object in the scene, e.g., "the chair is in the center of this room".
- The relative position to other objects in the scene, for instance, "this chair is the second one from the left".

Verification After collecting the descriptions from AMT, we do a quick inspection of the descriptions and manually filter and reject obvious bad descriptions

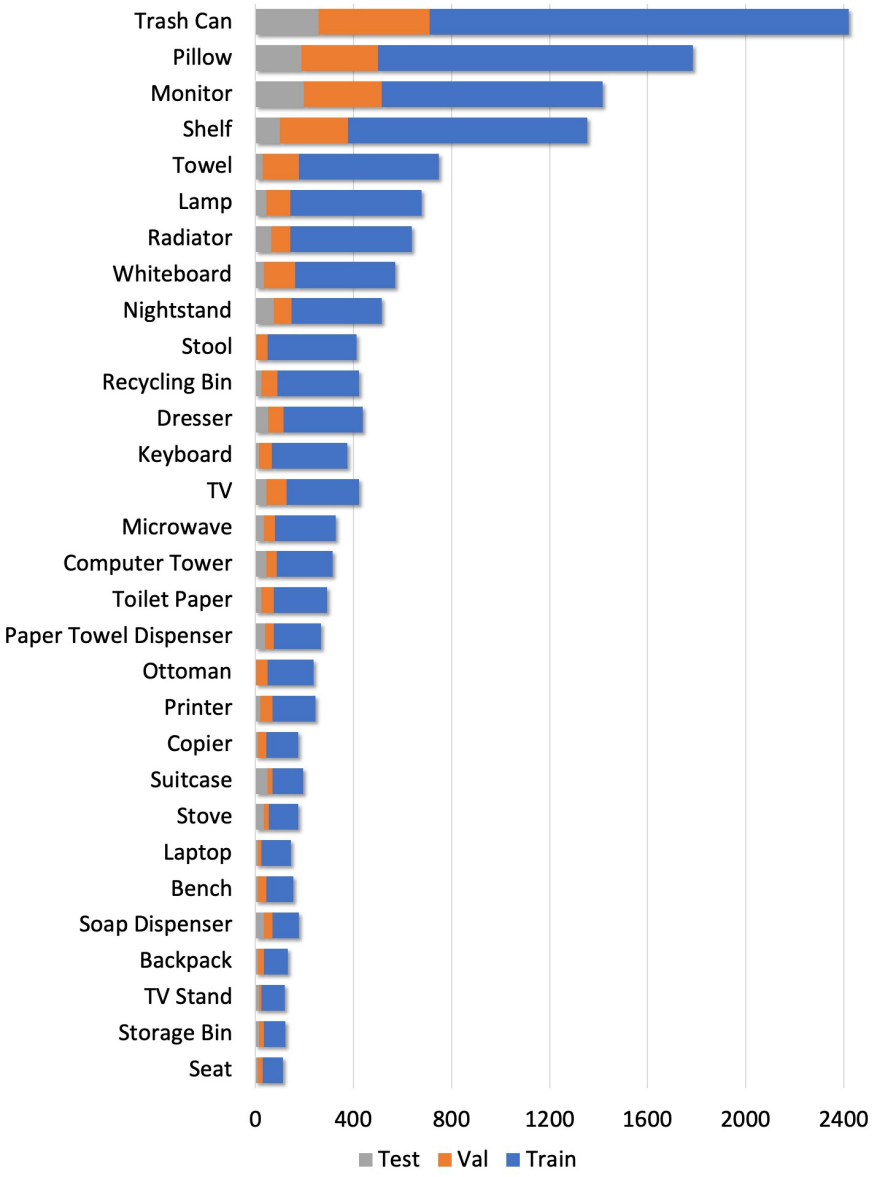


Fig. 11: Distribution of the top 30 categories in the "Others" category of the Train/Val/Test splits of the ScanRefer dataset (sorted in descending order according to the number of objects in the Train split).

before we start the verification process. We then verify the collected object descriptions by recruiting trained students to perform the verification task on our WebGL-based application, as shown in Fig. 14a. To ensure that the descriptions

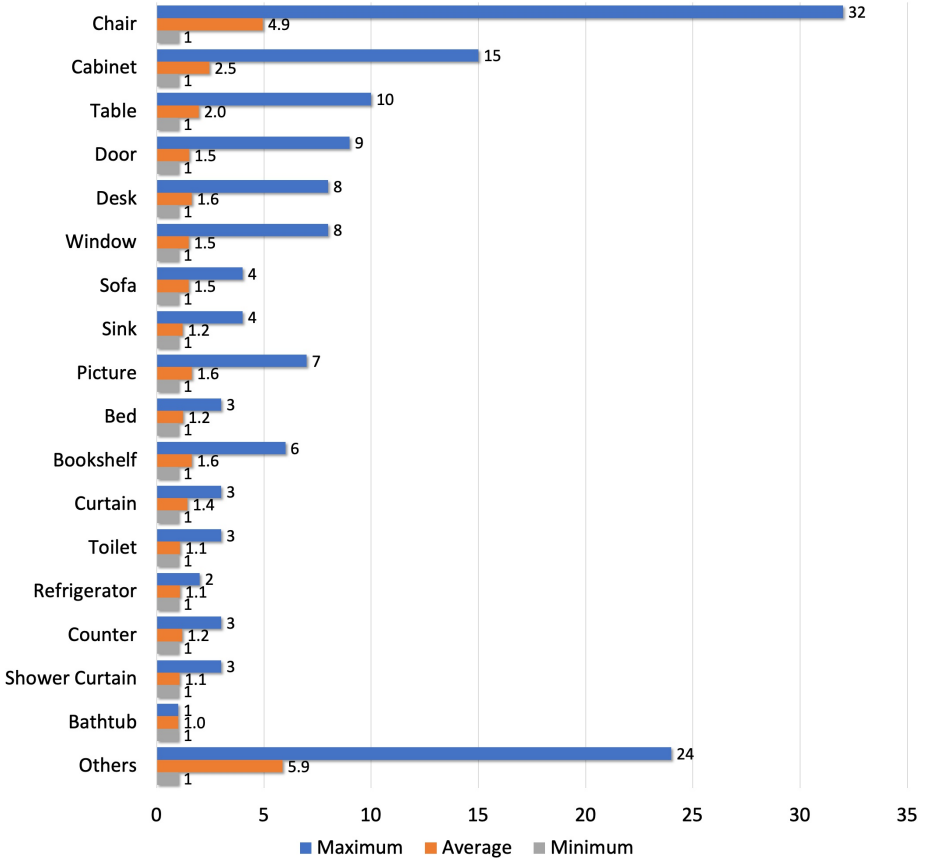


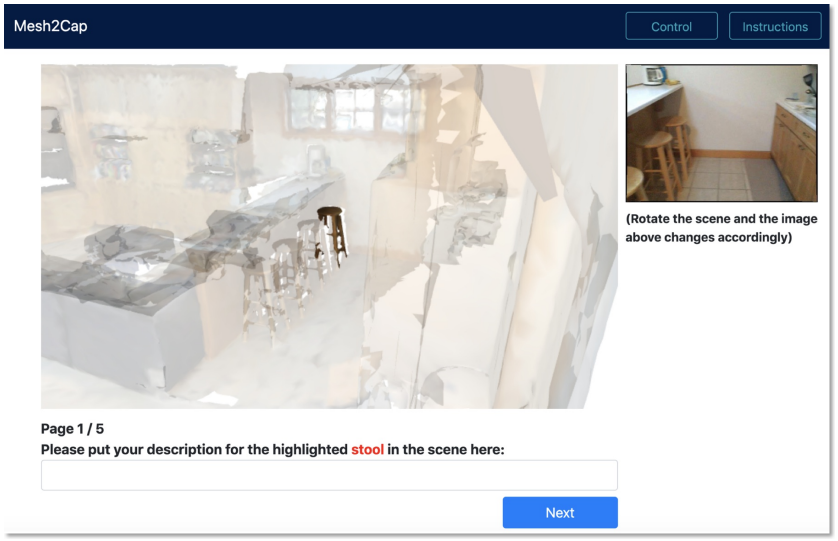
Fig. 12: Average and maximum numbers of objects in each category per scene in the ScanRefer dataset. For each category, we only consider scenes that contains the corresponding objects.

provided are discriminative (e.g., can pick out which one of the chairs is being described), the verifiers are asked to select the objects in the scene that match the descriptions the best. The verifiers are also asked to fix any spelling and wording issues, e.g., "hair" instead of "chair", and submit the corrected descriptions to our database. To guide the trained verifiers, we provide the verification instructions as shown in Fig. 14b.

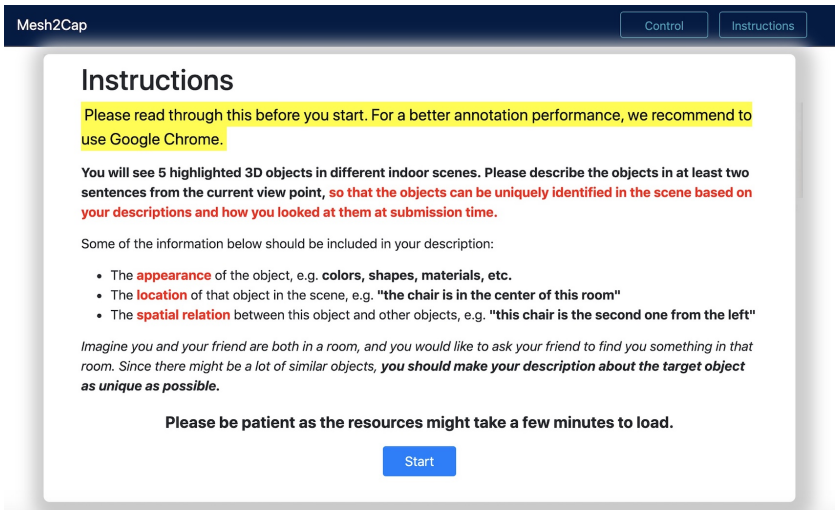
B Additional Implementation Details

B.1 Fusion Module

Fig. 15 shows the feature fusion process in our localization pipeline. Concretely, the fusion module first concatenates the point clusters $C = c_i \in \mathcal{R}^{M \times 128}$ and



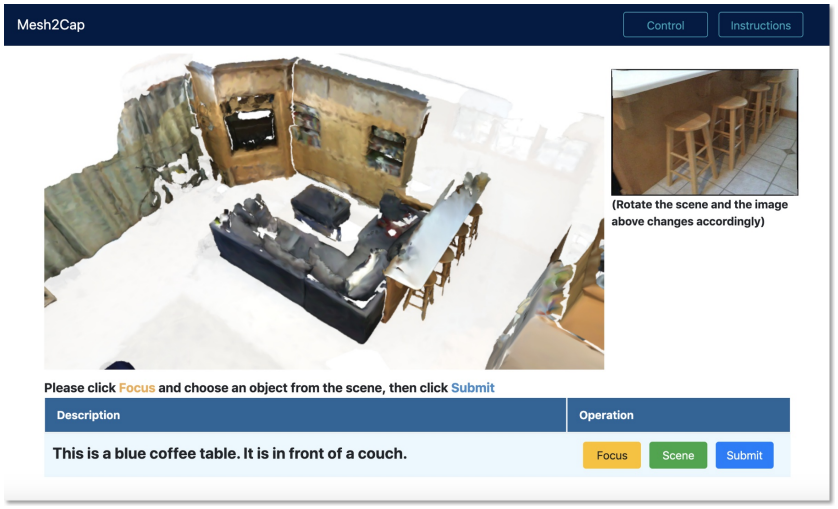
(a) Annotation interface for Amazon Mechanical Turk workers used to create the ScanRefer dataset.



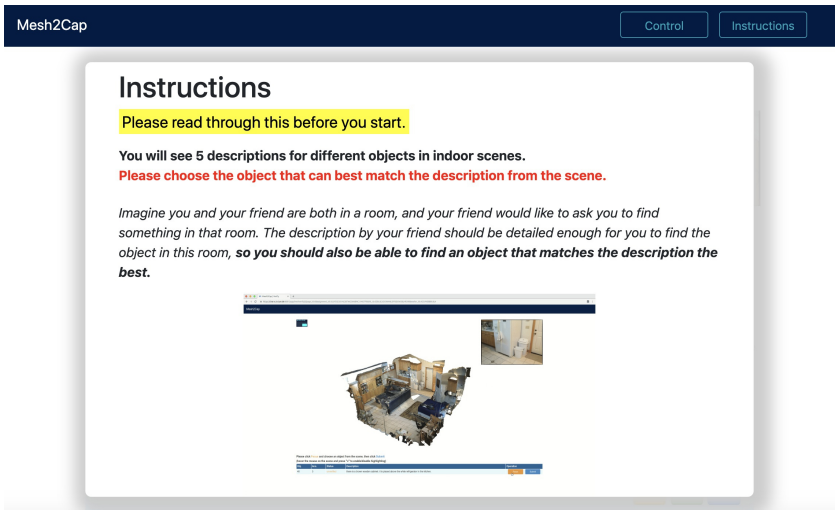
(b) Annotation instructions shown to the Amazon Mechanical Turk workers.

Fig. 13: (a) Our web-based annotation interface: annotators are requested to describe a batch of 5 target objects. The viewpoint can be adjusted by the user while the image on the right is chosen based on the camera view. (b) Screenshot of the instructions for the Amazon Mechanical Turk workers before providing descriptions for objects.

expanded language embedding $E = e' \in \mathcal{R}^{M \times 256}$, then multiply the expanded



(a) Verification interface used by trained student verifiers in order to verify each annotation done earlier by the annotation Amazon Mechanical Turk workers.



(b) Verification instructions shown to the trained student verifiers.

Fig. 14: (a) Our web-based verification interface: verifiers are asked to select objects that match the provided descriptions from the collection step. The ambiguous descriptions, which can be used to match multiple objects in the scene, are excluded from the final dataset. (b) Screenshot of the instructions that the trained verifiers have to go through before starting the verification.

objectness mask $D'_{objn} \in \mathcal{R}^{M \times 384}$ to filter out invalid object proposals. A multi-

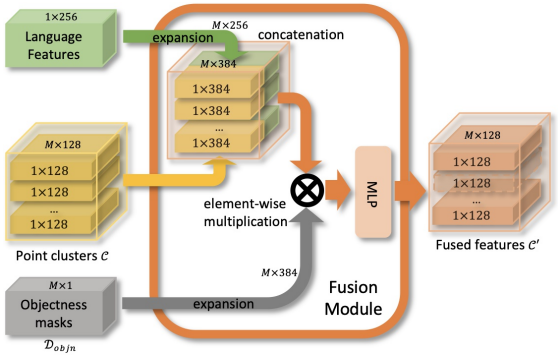


Fig. 15: The fusion module takes as input the aggregated point clusters, the language embeddings, and the predicted objectness masks. It first concatenates the point clusters with the expanded language features as the raw fused features, of which the invalid ones will be masked out by the predicted objectness masks. Finally, a multi-layer perceptron takes in the raw fused features and outputs the final fused multimodal point features.

	cab.	bed	chair	sofa	tabl.	door	wind.	bkshf.	pic.	cntr.	desk	curt.	fridg.	showr.	toil.	sink	bath.	others	mAP
[a]	4.77	85.51	64.42	72.74	30.39	11.17	6.62	17.32	0.35	2.16	35.79	7.80	16.69	16.96	76.74	16.77	69.57	5.68	30.08
[b]	9.93	**88.43**	67.12	69.44	**39.76**	12.20	5.11	20.27	0.02	9.27	41.52	16.10	**30.79**	5.77	77.32	14.93	61.02	7.82	32.05
[c]	7.01	88.01	67.13	73.69	32.87	12.36	9.01	17.61	0.31	9.27	44.78	16.25	20.29	3.55	76.50	12.33	72.24	8.08	31.74
[d]	11.16	87.20	**70.58**	75.17	36.76	11.47	6.72	13.40	1.09	7.08	48.38	11.64	19.96	4.29	85.29	**18.20**	72.83	**10.74**	32.89
[e]	7.22	87.72	67.24	72.42	33.66	11.55	8.80	20.16	0.14	**9.82**	46.07	15.91	22.48	2.67	77.82	13.17	68.14	8.01	31.83
[f]	**12.74**	83.91	69.94	72.17	36.11	**13.38**	8.42	17.52	**1.99**	6.58	**46.65**	**17.65**	24.04	**31.30**	75.99	10.31	61.92	9.78	**33.36**
[g]	10.53	84.00	63.48	**75.27**	30.62	7.78	8.45	18.08	1.18	5.47	39.27	10.14	18.83	8.93	69.99	9.36	**75.59**	7.97	30.27
[h]	11.11	85.63	67.81	71.04	34.96	9.54	6.22	16.37	1.67	6.28	36.07	12.93	17.40	7.46	68.74	11.77	65.69	7.71	29.91
[i]	10.72	86.71	69.86	72.77	32.60	16.33	8.16	19.64	1.14	7.08	42.21	14.31	22.99	6.92	**86.09**	8.06	65.51	8.79	32.22
[j]	9.76	87.93	65.93	72.59	31.60	9.48	9.05	**23.86**	0.37	6.69	42.22	13.86	21.42	16.35	80.41	12.30	57.80	7.40	31.61
[k]	8.92	88.20	70.37	73.93	32.89	10.54	**9.21**	14.05	0.48	6.91	44.74	6.54	17.76	27.64	81.18	12.86	62.40	9.06	32.09

Table 8: Object detection results measured using mean average precision (mAP) at IOU of 0.5 for the 18 difference classes for [a] VoteNet [49], [b] Ours (xyz), [c] Ours (xyz+rgb), [d] Ours (xyz+rgb+normals), [e] Ours (xyz+multiview), [f] Ours (xyz+multiview+normals), [g] Ours (xyz+lobjcls), [h] Ours (xyz+rgb+lobjcls), [i] Ours (xyz+rgb+normals+lobjcls), [j] Ours (xyz+multiview+lobjcls), [k] Ours (xyz+multiview+normals+lobjcls). Training with point normals (compare rows [d,f] to rows [c,e]) and multiview features (compare rows [e,f] to rows [c,d]) clearly leads to better performance. As expected, models with the language-based object classifier (rows [g-k]) does not results in better object detection compared to models without such a module (rows [b-f]).

layer perceptron maps the filtered feature maps into the final fused features $\mathcal{C}' \in \mathcal{R}^{M \times 128}$ as the output of the fusion module.

C Additional quantitative analysis

C.1 Object Detection Results

In order to evaluate the 3D object detection, we conduct ablations of our architecture with different point cloud features as well as ablating the inclusion of the language-based object classifier (see Tab. 8). We also compare against the the object detection results of VoteNet [49]. We use the mean average precision (mAP) thresholded by IoU value 0.5 as our evaluation metric and examine the object detection results for different object categories. We exclude structural objects such as "Floor" and "Wall". We group all categories which are not in the ScanNet benchmark categories [9] including "Otherfurnitures", "Otherstructure", and "Otherprop" into the "Others" category in our evaluation. Note that the "Others" category in our evaluation includes additional types of objects, such as "Pillow" and "Keyboard", with respect to those in the "Otherfurniture" category of the ScanNet benchmark.

While our 3D object detector is robust in identifying and separating out instances of large objects that are typically placed away from walls (e.g., bed, chair, sofa, toilet, bathtub), it is not as reliable at identifying instances of flat objects (e.g., picture, window, door) and objects with unclear instance boundaries (e.g., cabinet, shelving) and smaller objects (e.g., sink, others). Overall, our best 3D object detector only achieves a mAP of 33%, suggesting that improving 3D object detection, especially better instance detection for the "other" category, is a key challenge in our task of localizing objects in 3D using natural language.

As shown in Tab. 8, including point normals as extra point features (rows [d,f]) in training increases the detection results when compared to the models trained without the normals (rows [c,e]). Also, training with extracted high-level color features from the multi-view images (rows [e,f]) also produces better detection results compared with the results from models trained with just the raw RGB values (rows [c,d]). Note that networks equipped with the language-based object classifier (rows [g-k]) fail to produce better detection results compared to the ones without the extra language classifier module (rows [b-f]). This behavior is expected as the description provides additional information which helps to differentiate between objects of the same category; but it has no information for helping with object detection.

C.2 Training and Evaluation Variance

Since there is a random sampling of 40,000 points from the original point cloud in the VoteNet [49] detection backbone, we conduct experiments to measure the training and evaluation variance across multiple runs. As shown in Tab. 9 and Tab. 10, due to random sampling, there is a stddev of 0.30 across training runs and a stddev of 0.37 across evaluation runs. For more reliable results, we average the results of 5 evaluation runs with different random seeds when using VoteNet.

random seed	unique Acc@0.5	multiple Acc@0.5	overall Acc@0.5
2	46.83	20.57	25.66
4	47.96	19.45	24.98
8	45.96	20.05	25.07
standard deviation	0.82	0.46	0.30
mean	46.92	20.02	25.23

Table 9: Variance between training runs. We train our model (xyz+rgb+lobjcls) with three different random seeds (2, 4, 8) and evaluate the trained model using a fixed random seed 42. We have a training stddev of 0.30.

random seed	unique Acc@0.5	multiple Acc@0.5	overall Acc@0.5
42	48.89	22.24	27.40
2	49.28	22.05	27.34
4	48.68	21.56	26.82
8	48.29	21.99	27.09
16	50.35	21.42	27.03
32	49.55	21.75	27.14
64	49.61	22.25	27.56
128	49.28	21.57	26.95
256	49.88	21.98	27.39
512	47.29	21.99	28.12
standard deviation	0.87	0.29	0.37
mean	49.11	21.88	27.28

Table 10: Variance between evaluation runs due to the random sampling of points in the VoteNet [49]. We train our model (xyz+multiview+normal+lobjcls) with the a fixed random seed of 42 and evaluate the trained model using 10 different random seeds as shown in the first column. We have a evaluation stddev of 0.37.

C.3 Additional Ablation Study

In Tab. 11, we examine what happens when we feed different language inputs into our pipeline.

Does our method really learn from the full descriptions? To evaluate the impact of information from the full descriptions versus just the identification of the type of object to locate, we compare using the full description as input versus using the semantic label or the object name as the input. For example, for a target object "trash can" with the description *This is a short trash can. It is in front of a taller trash can.*, we input "trash can" as the object name and "others" as the semantic label (see Sec. A.1 for list of semantic classes). The results in Tab. 11 show that using the full descriptions improves the localization performance compared to using just the semantic labels as input. Comparing the

	unique		multiple		overall	
	Acc@0.25	Acc@0.5	Acc@0.25	Acc@0.5	Acc@0.25	Acc@0.5
Ours (semantic labels)	61.60	39.04	28.26	18.98	34.72	21.88
Ours (object names)	70.53	44.69	32.34	20.33	39.75	25.05
Ours (first sentences)	73.52	46.60	33.71	21.20	41.44	26.12
Ours (whole descriptions)	**76.33**	**53.51**	**32.73**	**21.11**	**41.19**	**27.40**

Table 11: Ablation study with different input lengths. We measure the percentages of predictions whose IoU with the ground truth boxes are greater than 0.25 and 0.5. Unique means that there is only a single object of its class in the scene. Obviously, the richer information the descriptions contain, the better our localization pipeline performs.

performance of using semantic labels and object names, we see that inputting the semantic labels helps with the performance in the "unique" scenarios where there is only one object from a certain category, but suffers in the "multiple" scenarios where more information is needed to distinguish between objects that are grouped into the same broad category (e.g., "trash can" and "laptop" would both be categorized as "other", and "armchair" would provide more information than just the coarse semantic label "chair").

Are the first sentences enough for the task? Since we deliberately collect at least two sentences as descriptions for the objects to ensure the richness of information, we also conduct experiments to show that the full description (with potentially multiple sentences) result in better performance than using only the first sentences. As Tab. 11 shows, the model trained on longer descriptions performs better than the one trained just on the first sentences.

D Additional Qualitative Analysis

We present additional examples of localization results by our method and the baselines for further qualitative analysis.

Qualitative results comparing VoteNet [49]+GRU and VoteNetBest with out method We show more qualitative results in Fig. 16 to display the difference in performance between these three methods. As shown in the first column in Fig. 16, using a pretrained VoteNet [49] detection backbone provides reasonable bounding box around objects, but still performs slightly worse than our method where we train the detection backbone and localization module in an end-to-end fashion (see the third column "ours").

More qualitative examples comparing OracleRefer and One-stage (with 2D to 3D backprojection) with our method To illustrate the difference in performance between the methods, we provide more qualitative results. We split the localization results into "unique" (Fig. 17) and "multiple" (Fig. 18 & Fig. 19) subsets. As shown in Fig. 17, for the "unique" subset, our method is able to identify and localize the object. In contrast, the 2D method (One-Stage), is able to

Fig. 16: Additional qualitative analysis comparing our method with VoteNet [49]+GRU and VoteNetBest.

identify the rough location of the object, but the backprojected 3D bounding box does not match the ground truth very well. For the "multiple" subset, there are challenging cases where our method fails to localize the target object. Fig. 18 and 19 show that our method is able to localize objects correctly (Fig. 18 rows 1,5, Fig. 19 rows 1-3,5-6) even when there are other objects of the same category in the scene. Our method is sometimes limited by the accuracy of the object detector, which tends to produce inaccurate bounding boxes for small objects such as pictures (Fig. 18 row 2). This indicates that the object detection can still be improved. Our method also has trouble disambiguating between objects based on spatial relations (Fig. 18 rows 3-4,6). For instance, for comparative

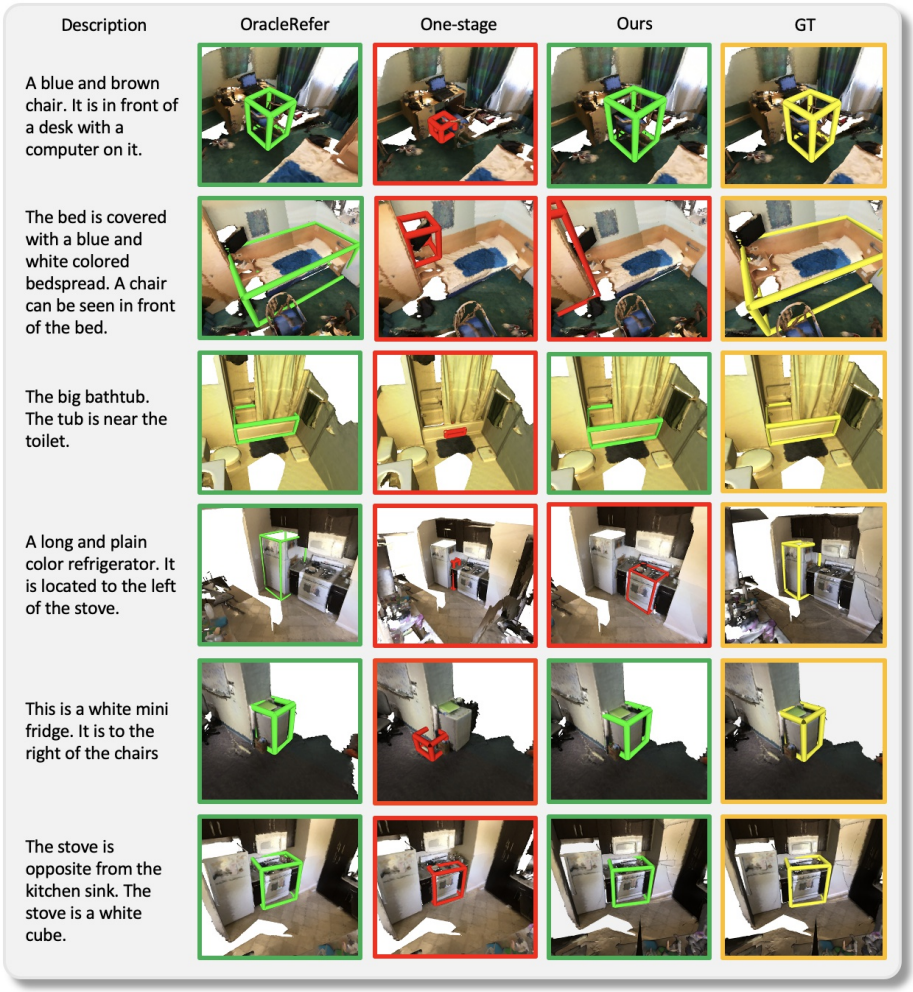


Fig. 17: Additional qualitative analysis in the "unique" scenarios where there is only one object from a certain category. Our method is capable of localizing the target object in a 3D indoor scene with the help of the free-form description.

phrases (e.g., "leftmost" or "rightmost") or counting (e.g., "the second one from the left"), the model fails to pick out the correct object (Fig. 18 rows 4).

Fig. 18: Additional qualitative analysis for the "multiple" subset where there are multiple objects with the same category as the target objects. While our methods can correctly localize the target object in some cases (rows 1,5), it often fails due to the limited accuracy of the object detector (row 2) or difficulty disambiguating between multiple instances (rows 3,4,6).

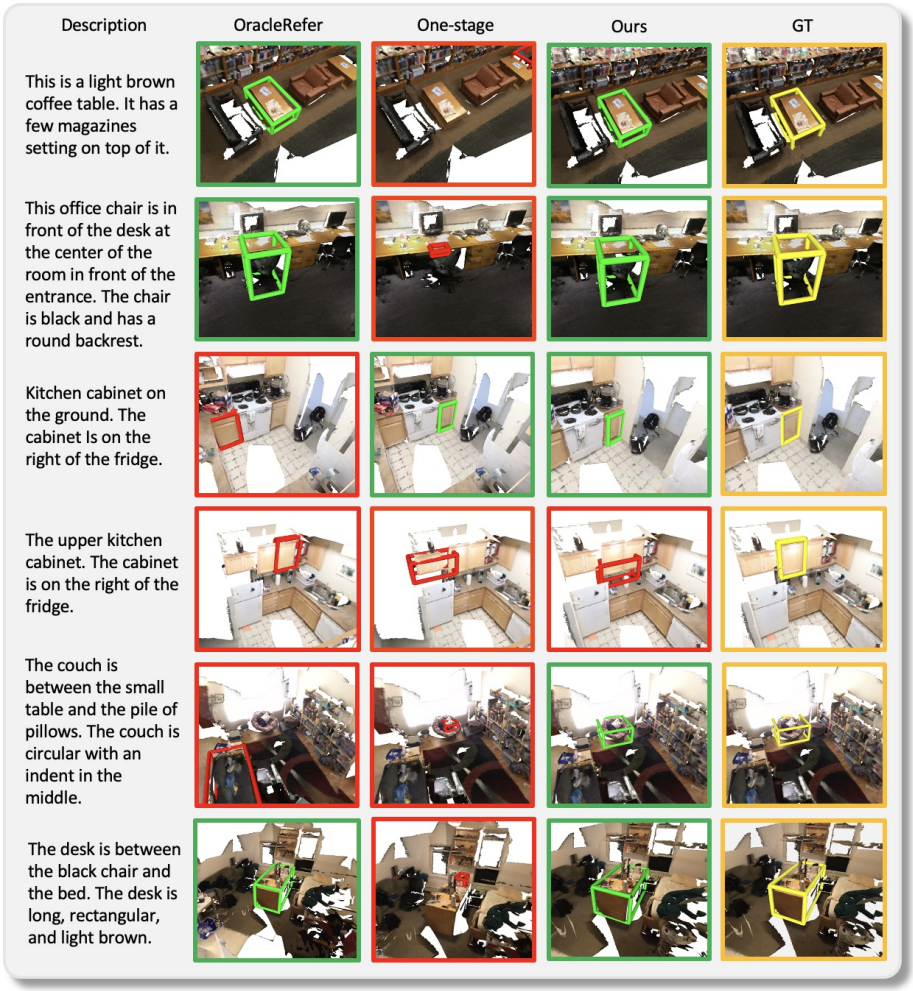


Fig. 19: Additional qualitative analysis for the "multiple" subset where there are multiple objects with the same category as the target objects. While our methods can correctly localize the target object in some cases (rows 1-3,5-6), it can fail due to the limited accuracy of the object detector and difficulty handling spatial relations (rows 4).