Supplementary Material
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

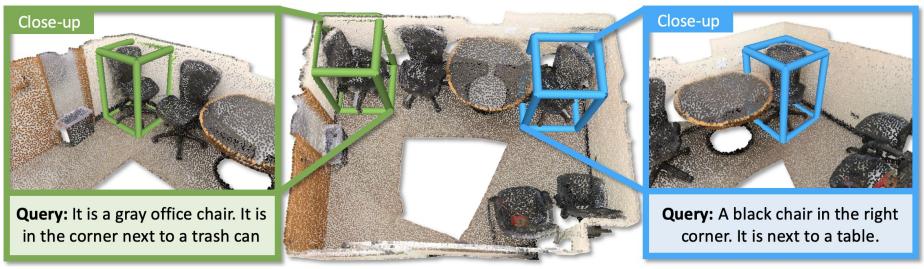


Fig. 1: ScanRefer localizes objects in a scene given a language description as input. In many cases, including this example, there are multiple objects from the same category in a single scene which makes the problem challenging and interesting at the same time.

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. 2. 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 [1]. In total, we annotate 11,046 objects from 265 categories from ScanNet [1]. 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. 3 shows the distribution of finer-grained objects in the category "Others". For each of the 18 coarse categories, Fig. 4 shows the average and maximum

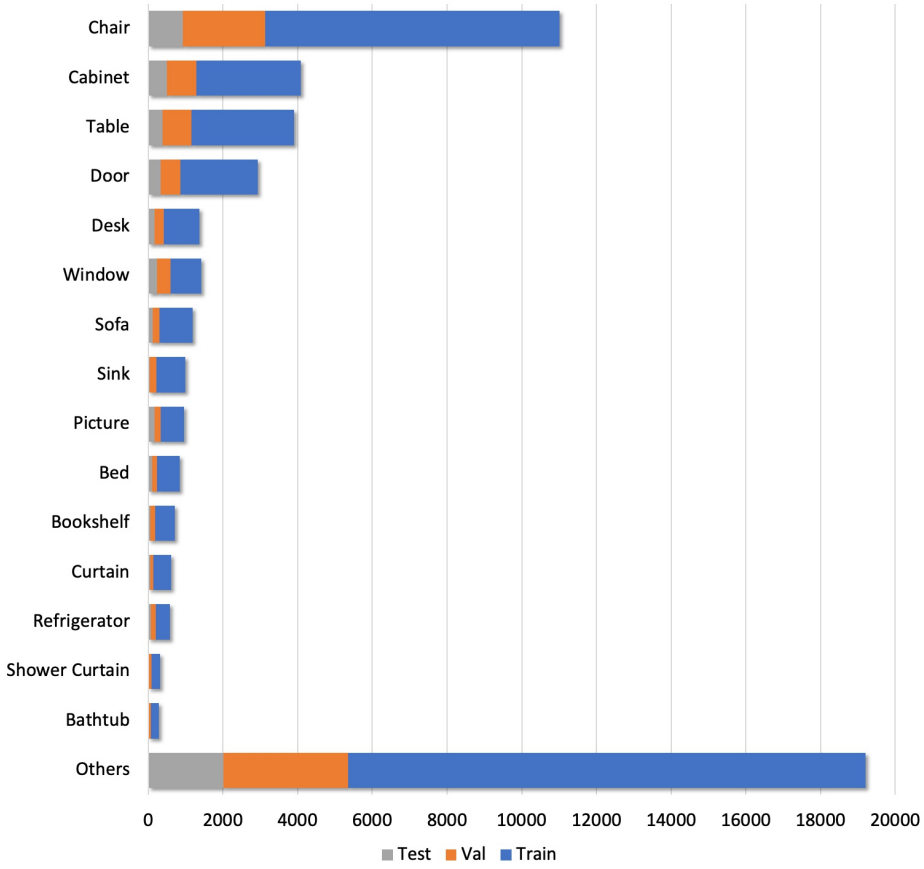


Fig. 2: Distribution of categories of objects in the ScanRefer dataset with annotated language descriptions.

	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 1: ScanRefer dataset statistics on Train and Val splits.

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.

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 2: 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.

We also provide detailed statistics in our training and validation splits in Tab. 1. To further address the difficulty of our task, we present additional details about the "unique" and "multiple" subsets in Tab. 2. 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 (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. 1). The "multiple" subset refers to cases where there are multiple objects of the 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. 1). As shown in Tab. 2, 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. 5a. 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. 5b. 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.

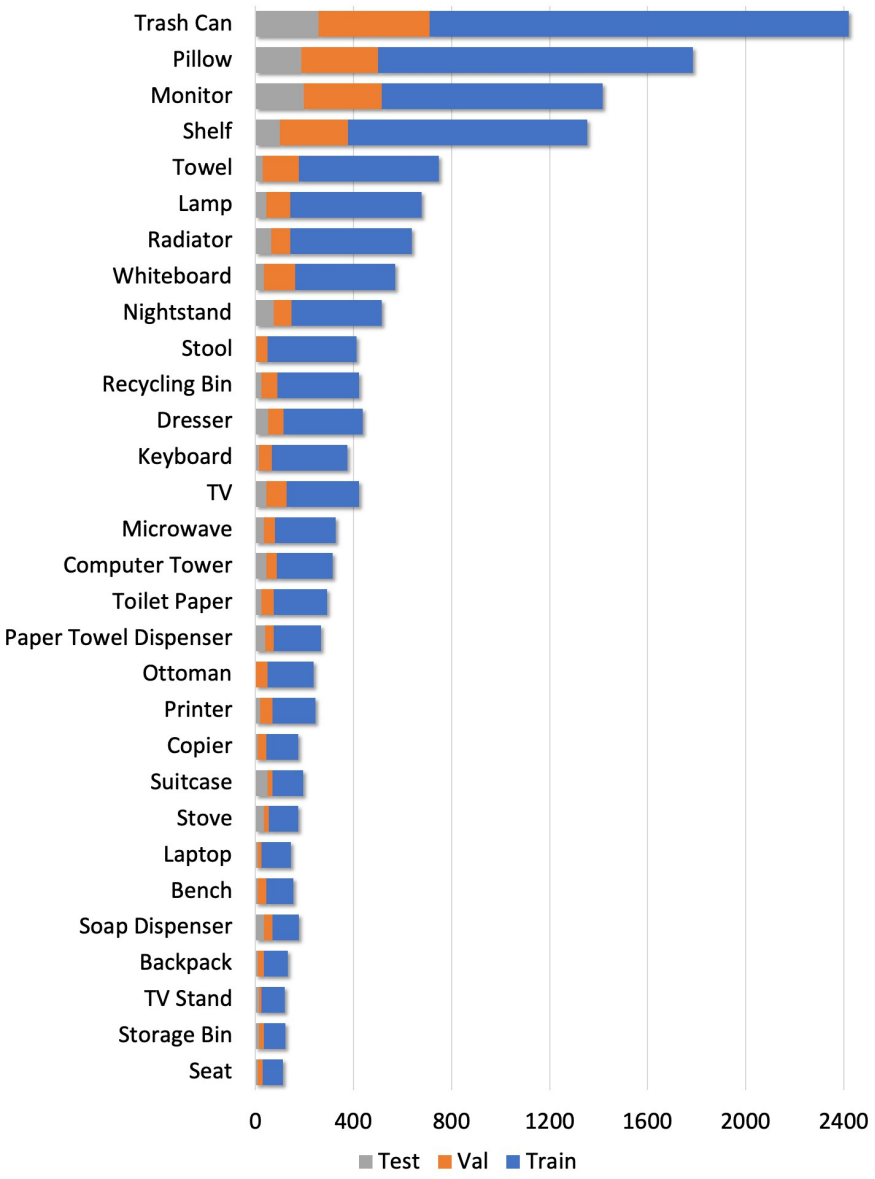


Fig. 3: Distribution of the top 30 categories in the "Others" category of the Train/Val/Test splits of the ScanRefer dataset (sorted in descending order according to the number of objects in the Train split).

– The location of that object in the scene, e.g., "the chair is in the center of this room".

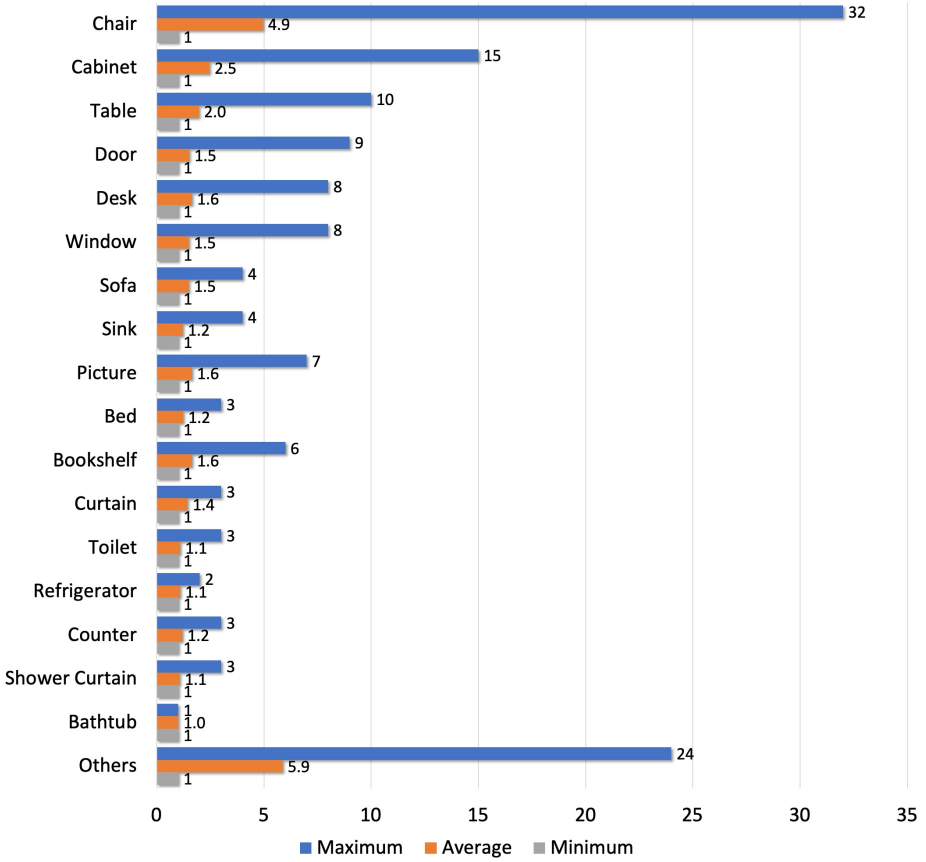
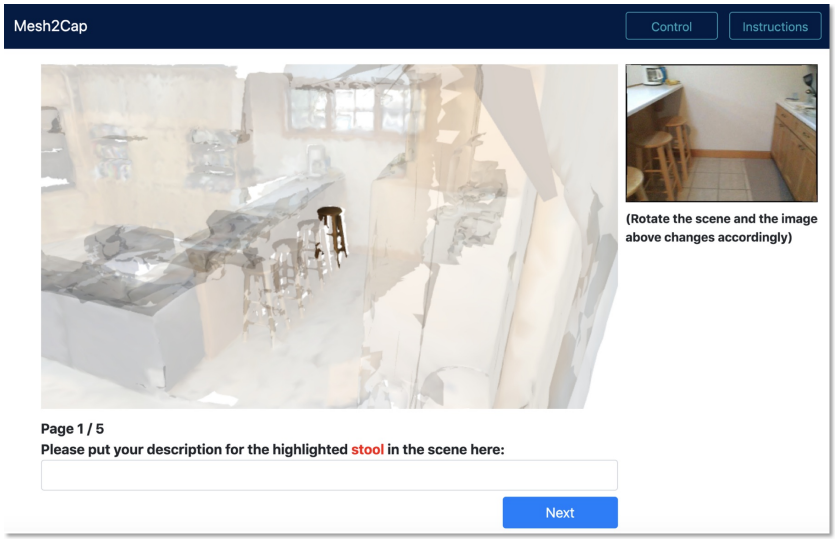


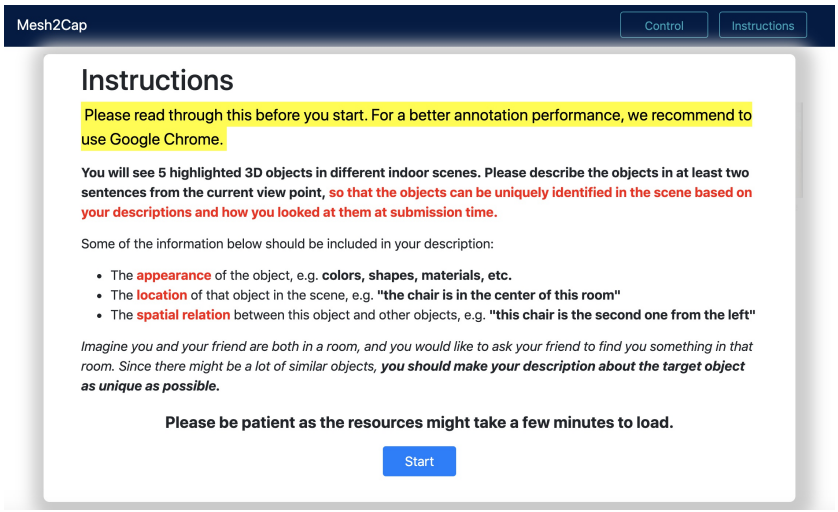
Fig. 4: Average and maximum numbers of objects in each category per scene in the ScanRefer dataset. For each category, we only consider scenes that contains the corresponding objects.

– 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 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. 6a. To ensure that the descriptions 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 descrip-



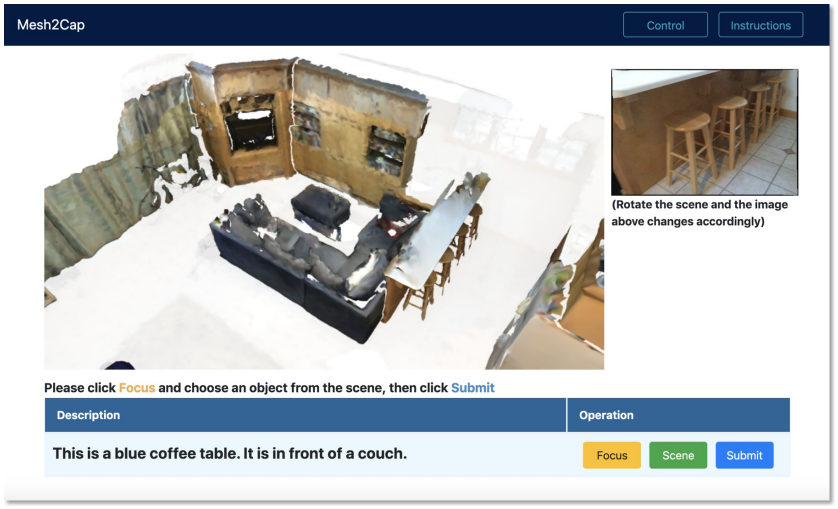
(a) Annotation interface for Amazon Mechanical Turk workers used to create the ScanRefer dataset.



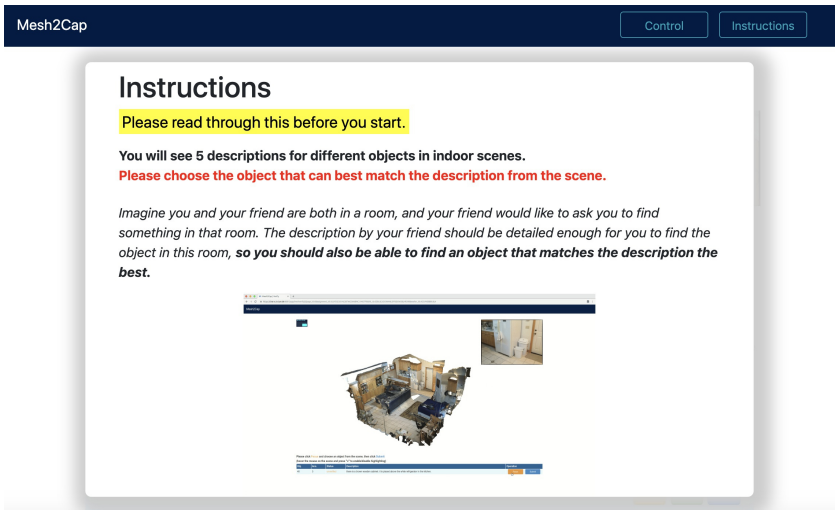
(b) Annotation instructions shown to the Amazon Mechanical Turk workers.

Fig. 5: (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.

tions to our database. To guide the trained verifiers, we provide the verification instructions as shown in Fig. 6b.



(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. 6: (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.

B Additional Implementation Details

B.1 Fusion Module

Fig. 7 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

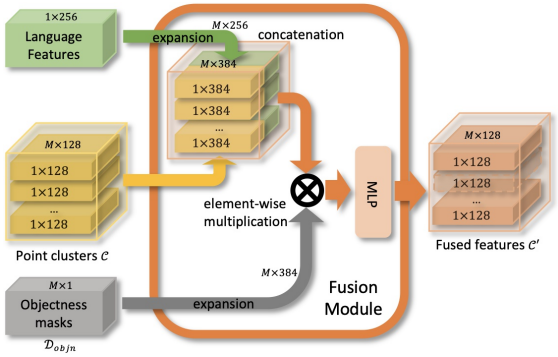


Fig. 7: The fusion module takes as input the aggregated point clusters, the language embeddings, and the predicted objectness masks. It first concatenates the point clusters with the expanded language features as the raw fused features, of which the invalid ones will be masked out by the predicted objectness 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 3: Object detection results measured using mean average precision (mAP) at IOU of 0.5 for the 18 difference classes for [a] VoteNet [2], [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]).

expanded language embedding $E = e' \in \mathcal{R}^{M\times 256}$, then multiply the expanded objectness mask $D'_{objn} \in \mathcal{R}^{M\times 384}$ to filter out invalid object proposals. A multi-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. 3). We also compare against the the object detection results of VoteNet [2]. 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 [1] 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. 3, 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 [2] detection backbone, we conduct experiments to measure the training and evaluation variance across multiple runs. As shown in Tab. 4 and Tab. 5, due to random sampling, there is a stddev of 0.40 across training runs and a stddev of 0.20 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	38.27	16.81	20.97
4	39.19	15.89	20.41
8	37.56	16.38	20.49
standard deviation	0.65	0.65	0.40
mean	38.34	16.36	20.62

Table 4: 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 0.40.

random seed	unique Acc@0.5	multiple Acc@0.5	overall Acc@0.5
42	39.95	18.17	22.39
2	40.27	18.02	22.34
4	39.78	17.62	21.92
8	39.46	17.97	22.14
16	41.14	17.50	22.09
32	40.49	17.77	22.18
64	40.54	18.18	22.52
128	40.27	17.63	22.02
256	40.76	17.96	22.38
512	38.64	17.97	22.98
standard deviation	0.71	0.24	0.20
mean	40.13	17.88	22.20

Table 5: Variance between evaluation runs due to the random sampling of points in the VoteNet [2]. 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 0.20.

C.3 Additional Ablation Study

In Tab. 6, 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. 6 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)	50.34	31.90	23.09	14.51	28.37	17.88
Ours (object names)	57.63	36.52	26.43	16.61	32.48	20.47
Ours (first sentences)	60.08	38.08	27.55	17.32	33.86	21.34
Ours (whole descriptions)	**63.04**	**39.95**	**28.91**	**18.17**	**35.53**	**22.39**

Table 6: 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. 6 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 [2]+GRU and VoteNetBest with out method We show more qualitative results in Fig. 8 to display the difference in performance between these three methods. As shown in the first column in Fig. 8, using a pretrained VoteNet [2] 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. 9) and "multiple" (Fig. 10 & Fig. 11) subsets. As shown in Fig. 9, 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. 8: Additional qualitative analysis comparing our method with VoteNet [2]+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. 10 and 11 show that our method is able to localize objects correctly (Fig. 10 rows 1,5, Fig. 11 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. 10 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. 10 rows 3-4,6). For instance, for comparative

Fig. 9: 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. 10 rows 4).

Fig. 10: 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).

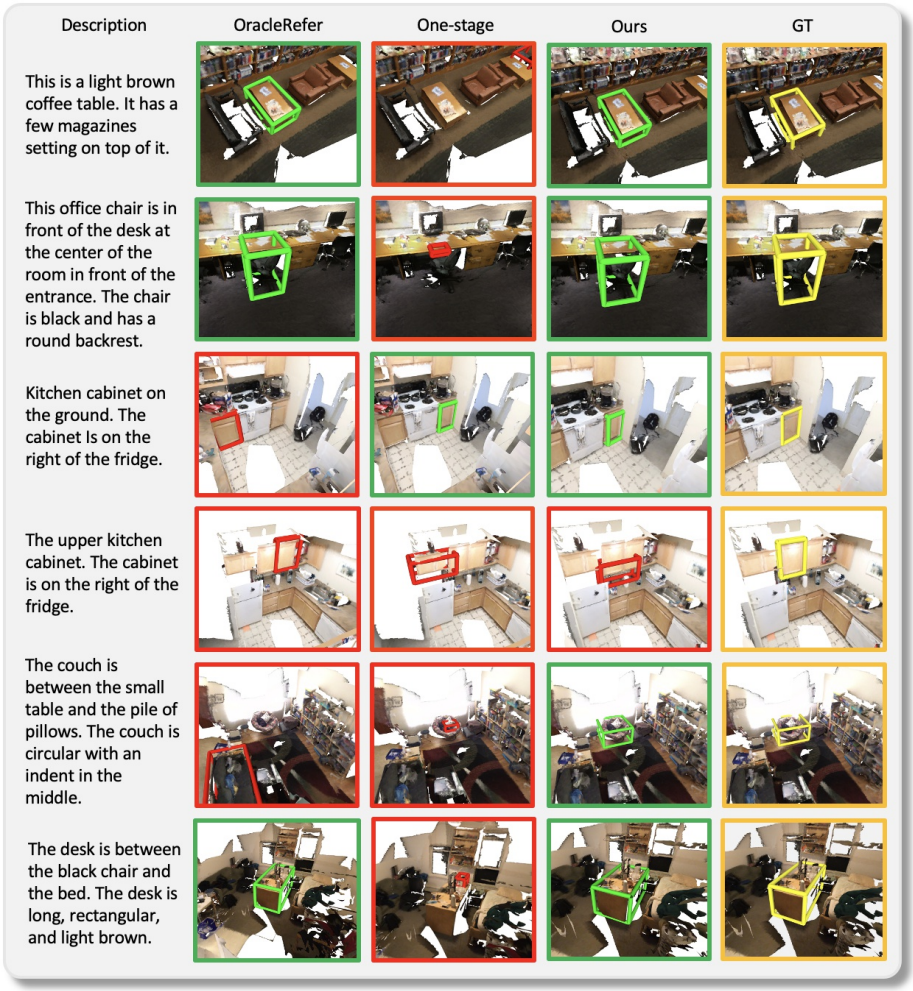


Fig. 11: Additional qualitative analysis for the "multiple" subset where there are multiple objects with the same category as the target objects. While our methods can correctly localize the target object in some cases (rows 1-3,5-6), it can fail due to the limited accuracy of the object detector and difficulty handling spatial relations (rows 4).

Bibliography

[1] 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)

[2] 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)