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

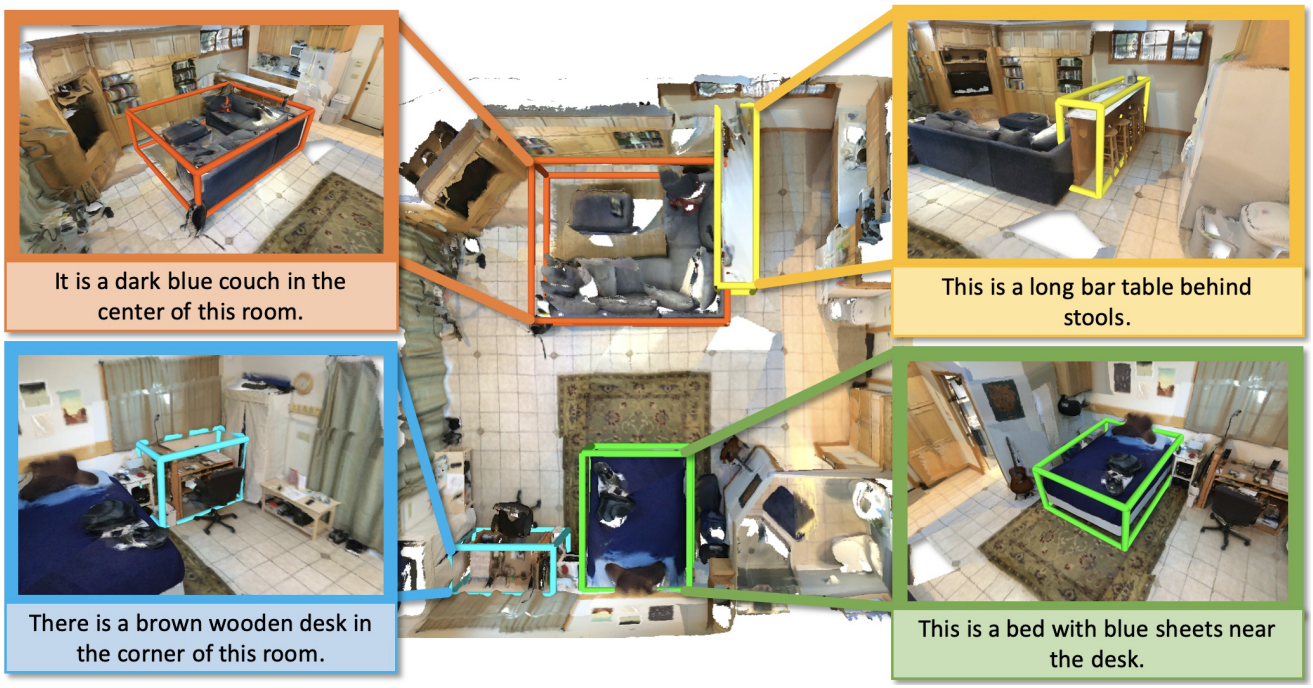

Figure 1: ScanRefer takes as input a 3D reconstructed scene and a free-form language description referring to an object in that scene; the output is a prediction of the bounding box for the target object. The object appearance and spatial localization information is inferred from the description and further utilized for localizing the referred object in the 3D scene.

Abstract

We introduce the new 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****, where the core idea is to learn a fused descriptor from 3D object proposals and encoded sentence embeddings. This learned descriptor then correlates the language expressions with the underlying geometric features of the 3D scan and facilitates the regression of the 3D bounding box of the target object. In order to train and benchmark our method, we introduce a new ScanRefer dataset, containing* $46,173$ *descriptions of* $9,943$ *objects from* 703 *ScanNet [8] scenes. ScanRefer is the first large-scale effort to perform object localization via natural language expression directly in 3D.*

1. Introduction

The deep learning revolution has enabled great progress in localizing objects in a given visual context given a natural language description referring to the object. Seminal methods for exploring visual grounding and referring expression resolution tasks have drawn attention in this research area [20, 19, 35]. Datasets such as ReferIt [26], RefCOCO [65], and Flickr30K Entities [42] are a key reason for this groundbreaking progress. However, at the same time they are also restricted to the 2D image domain. That is, bounding boxes and pixel segments representing objects are inherently limited to the context of a single RGB image and are unable to capture the full 3D extent of the objects. Knowing the 3D location and spatial extent of an object is critical for agents that need to interact with objects. For instance, if you ask a robot to bring "the black stool next to the fridge", it is important for the robot to understand the underlying spatial layout of the environments in order to

successfully navigate within and interact with the environment.

In this work, we ground natural language expressions in 3D scenes. Specifically, we localize objects in 3D point clouds given natural language descriptions referring to the objects. To this end, we introduce an architecture that jointly learns to propose 3D bounding boxes for objects in an input point cloud and match the boxes against input descriptions.

To train and evaluate our architecture, we collect a new dataset: ScanRefer which augments RGB-D scans in ScanNet [8] with natural language descriptions. Concretely, we hire human annotators to describe objects in the reconstructed 3D environments, as well as their spatial context. This way, we annotate all unique scenes in ScanNet with free-form natural language descriptions. In total, we acquire $46,173$ descriptions of $9,943$ 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 the 3D vision community.

In summary, our main contributions are as follows:

- We introduce the ScanRefer dataset containing $46,173$ human-written free-form descriptions of $9,943$ objects in 3D scans.
- We propose a data-driven method for localizing 3D target objects in a point cloud using natural language expressions.

2. Related Work

2.1. Grounding Referring Expressions in Images

There has been a plethora of successful work connecting images to natural language descriptions across tasks such as image captioning [25, 24, 55, 60], text-to-image retrieval [56, 22], and visual grounding [20, 35, 67]. The task of visual grounding (also known as referring expression comprehension or phrase localization) is to localize a region described by a given referring expression, the query. To address this task, many methods [20, 51, 56, 57, 42, 43, 35, 65, 66, 67, 10] have been proposed. A common strategy of these works is to follow a two-stage pipeline, where in the first stage an object detector, either unsupervised [69] or pretrained [50], is used to propose regions of interest, and in the second stage the regions are ranked by similarity to the query, with the highest scoring region provided as the final output. In contrast, other methods [19, 39, 63] address the referring expression task with a single stage proposing end-to-end approaches, and some [19, 32, 30, 36, 64, 4] produce a segmentation mask for the object description. Another line of research investigates how to improve visual grounding using more advanced methods such as incorporating syntax [33, 15], using graph attention networks [58, 62], speaker-listener models [35, 66], as well as weakly supervised methods [59, 68, 9] and zero-shot settings for unseen nouns [52].

Despite the impressive work in visual grounding, these methods all operate on 2D image datasets [42, 26, 65]. A recent dataset [37] integrates the RGB-D images but it still lacks the complete 3D context beyond a single image. Qi et al. [48] proposes to study referring expressions in an embodied setting, where semantic annotations are project from 3D to 2D bounding boxes on images observed by an agent. In our work, we go beyond individual images. Our core 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.

2.2. Object Detection in 3D

With the recent growth of popularity of 3D scene understanding in computer vision, a variety of object detection methods have been proposed to the 3D domain. Many of these works operate on volumetric grids [17, 23, 29, 38, 11] and have achieved great success on several 3D RGB-D datasets [54, 8, 2]. As an alternative to regular grids, point-based methods, such as PointNet [45] or PointNet++ [47], have been used as backbones for 3D detection and/or object instance segmentation [61, 12]. Very recently, Qi et al. [46] introduced an object detection scheme based on Hough Voting [18] which differs most from other metric or anchor-based methods and is specifically tailored to point-based representations. Our approach extracts geometric features in a similar fashion, but our architecture further backprojects 2D feature information since color signal is critical for describing 3D objects with natural language.

2.3. 3D Vision Language

Despite the fact that vision and language research has been gaining popularity in image domains (e.g., image captioning [24, 55, 60, 34], image-text matching [13, 28, 31, 21, 14], and text-to-image generation [49, 14, 53]), only very few researchers focus on the joint domain of vision and language in 3D. For instance, the work by Chen et al. [5] learns a joint embedding of 3D shapes from ShapeNet [3] and corresponding free-form natural language descriptions. Achlioptas et al. [1] disambiguates between different objects using language. Recent work by Prabhudesai et al. [44] has started to investigate grounding of language to 3D by identifying 3D bounding boxes of target objects for simple arrangements of primitive shapes of different colors. Instead of focusing on isolated objects, our work considers large 3D real-world environments obtained by RGB-D reconstructions that are typical in semantic 3D scene understanding.

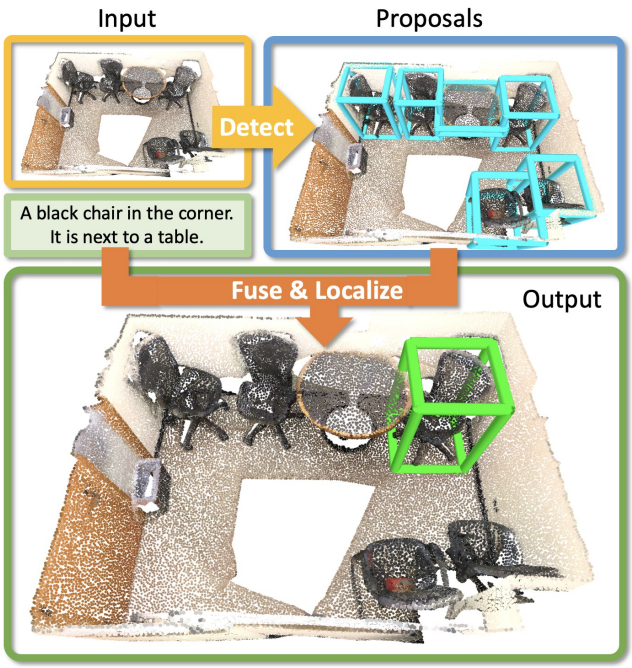

Figure 2: Overview of 3D object localization using natural language task: our ScanRefer takes as input a point cloud representing a reconstructed 3D scene and a free-form description referring uniquely to a specific object in that scene, and predicts the most likely object bounding box as the final output.

3. Task

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

4. Dataset

We build our ScanRefer dataset based on ScanNet [8] which is composed of 1513 RGB-D scans taken in 707 unique indoor environments. We provide around 5 unique descriptions for each object in each scene, focusing on complete coverage of all objects that are present in the reconstruction.

4.1. Data Collection

We develop a 3D web-based annotation interface using WebGL which we deployed on Amazon Mechanical Turk (AMT) to collect object descriptions in the ScanNet scenes. The annotation pipeline consists of two stages, first description collection (see Fig. 3), followed by verification (see Fig. 4). This is similar in spirit to the ReferItGame [26], except we separate the writing of the description and the selection of referred objects into two phases. We select the set of objects to annotate from ScanNet by restricting the object categories to indoor furnitures and excluding structural objects such as "Floor" and "Wall" (see supplementary material for more details). We manually verify the reconstructed objects and filter out objects whose reconstructions are too incomplete and/or hard to identify.

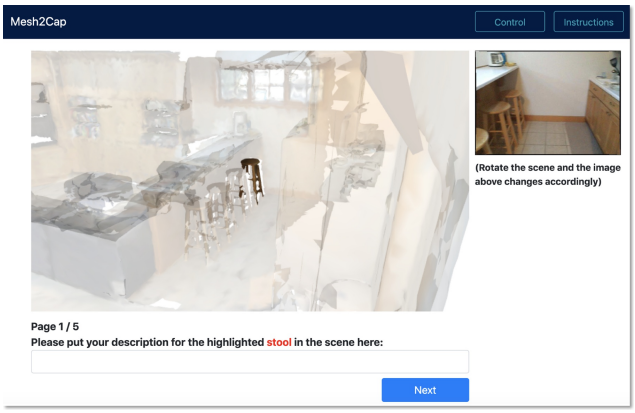

Figure 3: Our web-based annotation interface: annotators are requested to describe a batch of 5 target objects. The viewpoint can be adjusted by the user while the image on the right is chosen based on the camera view.

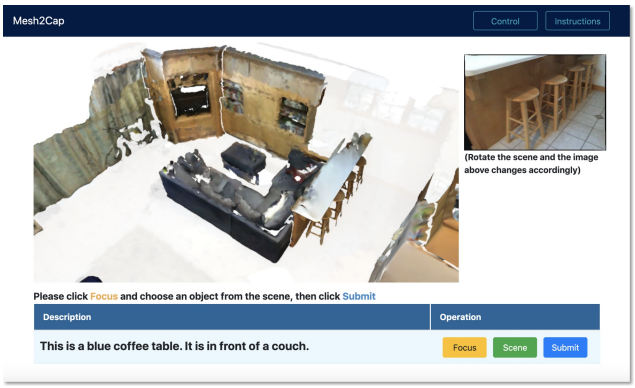

Figure 4: Our web-based verification interface: verifiers are asked to select objects that match the provided descriptions from the collection step. The ambiguous descriptions, which can be used to match multiple objects in the scene, are excluded from the final dataset.

4.1.1 Annotation

To collect object descriptions, we present a 3D web-based UI that shows the object in context (see Fig. 3). We show the workers a visualization for the scene with all objects other than the target object faded out and an image gallery on the side to compensate for incomplete details in the recon-

structions. Annotators are provided with a random initial viewpoint that includes the target object and camera controls that they can use to adjust the camera view to better examine the target object. We ask the annotator to describe the appearance of the target object as well as its spatial location within the scene and relation to other objects. To ensure the descriptions are informative, we require the annotator to provide at least two full sentences. The description collection tasks are batched and randomized so that each object is described by five different workers.

4.1.2 Verification

To ensure the quality of the descriptions, 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 scene that match the description (see Fig. 4). In addition, verifiers are asked to correct any spelling and wording issues, and provide a corrected description when necessary. In the end, we filter out in total 2,823 invalid descriptions that do not match the target objects and fix writing issues for 2,129 problematic descriptions.

4.2. Dataset Statistics

In total, we collected 46,173 descriptions for 703 ScanNet scenes[1]. On average, there are 14.14 objects, 65.68 descriptions per scene, and 4.91 descriptions per object after filtering. 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. Tab. 1 shows the details of the dataset statistics.

Number of descriptions	46,173
Number of scenes	703
Number of objects	9,943
Number of objects per scene	14.14
Number of descriptions per scene	65.68
Number of descriptions per object	4.64
Size of vocabulary	7,378
Min/max/average length of descriptions	4/117/17.91

Table 1: ScanRefer dataset statistics.

5. Method

We propose an end-to-end approach on point clouds to address the localization task (see Fig. 5). Our architecture consists of two main modules: 1) detection & encoding; 2) fusion & localization. The detection & encoding module encodes the input point cloud and description, and outputs the object proposals and the language embedding, which are then fed into the fusion module to mask out invalid object proposals and produce the fused features. Finally, the localization module chooses the most likely object proposal as the final output.

[1] 4 scenes are excluded due to lack of objects to describe

5.1. Data Representations

Point clouds We randomly sample 40,000 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 the point features such as colors and normals. We use the point coordinates p_i as the only geometrical information. Since descriptions of objects refer to attributes such as color and texture, we incorporate visual appearance by adapting the feature projection scheme in Dai and Nießner [7] to project multi-view image features $c_i \in \mathcal{R}^{128}$ to the point cloud. The image features are extracted using a pre-trained ENet [40]. Following Qi et al. [46], we also append the height of the point from the ground and normals to the point features. The final point cloud data is prepared offline as $\mathcal{P}' = \{(p_i, c_i)\} \in \mathcal{R}^{40{,}000 \times 135}$.

Descriptions We tokenize the input description with SpaCy [16] and map the tokens to 300-dimensional word embedding vectors $\mathcal{W} = \{w_j\}$ using pretrained GLoVE word embeddings [41].

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++ [47] backbone and the voting module in Qi et al. [46] to process the point cloud input and aggregate all potential object candidates to individual clusters. The output from the detection module is a set of point clusters $\mathcal{O} \in \mathcal{R}^{m \times 128}$ representing all object proposals with enriched point features, where m represents the upper bound of the number of proposals. Next, the proposal module takes in the point clusters and further processes those clusters to predict the objectness mask $\mathcal{M} \in \mathcal{R}^m$ and regress the bounding boxes $\mathcal{B} \in \mathcal{R}^{m \times 6}$ for all m proposals, where each $\mathcal{B}_i$ consists of the coordinates of the box center and the size residuals in all 3 dimensions.

Encoding The sequences of word embedding vectors representing the input description are fed into a GRU cell [6] to aggregate the textual information. We take the final hidden state of the GRU cell as the final language embedding.

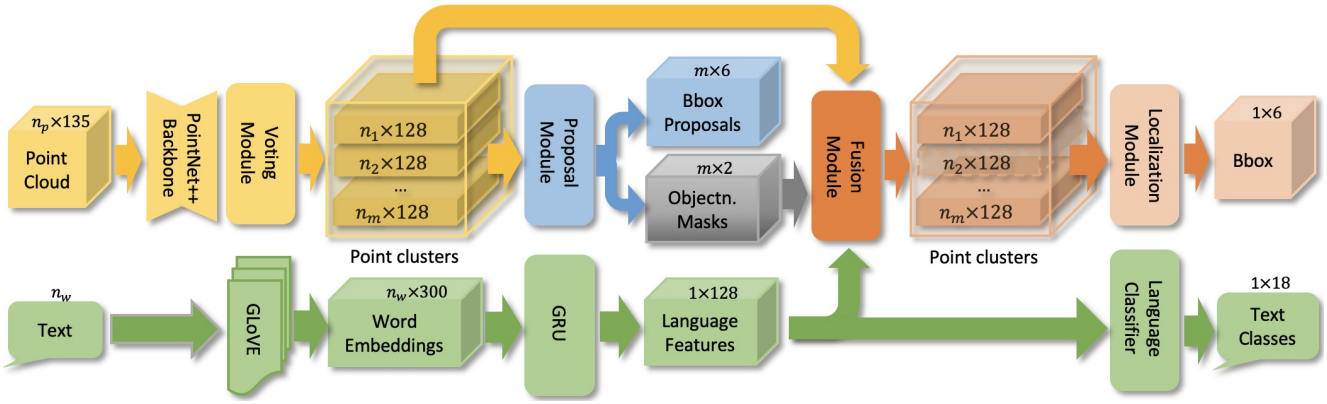

Figure 5: ReferNet architecture: object localization is conditioned on language input. The PointNet++ [47] backbone takes as input a point cloud and aggregates it to high-level point feature maps, which are then clustered and fused as object proposals by a voting module similar to Qi et al. [46]. Object proposals are masked by the objectness predictions, and then fused with the sentence embedding of the input descriptions, which is obtained by a GLoVE [41] + GRU [6] 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 most likely object proposal for the input description.

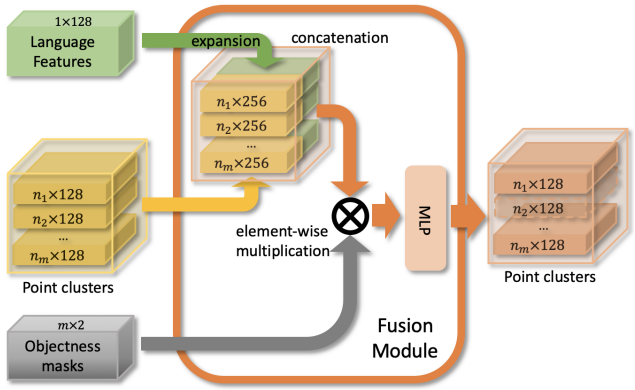

Figure 6: The fusion module takes as input the aggregated point clusters, the language embeddings, and the predicted objectness masks. It first concatenates the point clusters with the expanded language features as the raw fused features, of which the invalid ones will be masked out by the predicted objectness masks. Finally, a multi-layer perceptron takes in the raw fused features and outputs the final fused multimodal point features.

Fusion The outputs from the previous detection and encoding modules are fed into the fusion module (see Fig. 6) to integrate the point features together with the language embeddings. Specifically, each feature vector $o_i \in \mathcal{R}^{128}$ in the point cluster $\mathcal{O}$ is concatenated with the language embedding $e_i \in \mathcal{R}^{128}$ as the extended feature vector, which is then masked by the predicted objectness mask $m_i \in \{0, 1\}$ and fused by a multi-layer perceptron as the the final fused cluster features $o'_i \in \mathcal{R}^{128}$.

Localization Point clusters with fused cluster features $\mathcal{O}' = \{o'_i\}$ are processed by a single layer perceptron to produce the raw scores, which are then squashed into the interval of $[0, 1]$ using a softmax function. The object proposal with the highest score is the final output.

5.3. Loss Function

Reference loss For the produced score $s_i \in [0, 1]$ for object proposal o_i among all m proposals, our target is constructed as $T = \{t_i\}, i = 1, ..., m$, where $t_i \in \{0, 1\}$ represents if the proposal matches the ground truth object. Note that the detection module may produce overlapping proposals for a single object, so the ground truth vector is not necessarily an one-hot vector. We then formulate our reference loss using a binary cross-entropy function as:

$$\mathcal{L}_{ref} = -\sum_{i=1}^{M} [w_{neg}(1 - t_i)\log(1 - s_i) + w_{pos} t_i \log(s_i)] \tag{1}$$

where w_{neg} and w_{pos} are the weights for negative and positive samples. In practice, we set those two weights to 0.01 and 1, respectively.

Object detection loss We follow the same detection loss as introduced in Qi et al. [46] to handle object proposals. The detection loss is marked as $\mathcal{L}_{det}$.

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 classes and exclude the label "Floor" and "Wall" to match our dataset settings. The language to object classification loss $\mathcal{L}_{cls}$ here is a multi-class cross-entropy loss.

Final loss The final loss is a linear combination of the reference loss, object detection loss and the language to object

	unique		multiple		overall	
	P@0.25	P@0.5	P@0.25	P@0.5	P@0.25	P@0.5
OracleCatRand	100	100	18.09	17.84	29.99	29.76
PointRefNet [47]	16.83	12.85	7.82	4.71	9.16	5.92
VoteNetRand [46]	40.88	23.04	6.83	3.35	11.90	6.28
SCRC [20]	24.03	9.22	17.77	5.97	18.70	6.45
Ours (xyz)	45.54	29.37	19.76	11.73	24.02	14.64
Ours (xyz+rgb)	45.63	30.35	21.18	12.67	25.22	15.59
Ours (xyz+rgb+normals)	47.57	31.55	20.64	12.31	25.09	15.49
Ours (xyz+multiview)	50.78	33.37	18.71	11.22	24.01	14.89
Ours (xyz+multiview+normals)	50.67	32.29	20.75	12.35	25.69	15.65
Ours (xyz+lobjcls)	51.97	35.43	21.30	12.35	26.38	16.17
Ours (xyz+rgb+lobjcls)	53.75	37.47	21.03	12.83	26.44	16.90
Ours (xyz+rgb+normals+lobjcls)	54.96	35.24	22.27	13.55	27.67	17.13
Ours (xyz+multiview+lobjcls)	**56.68**	33.78	21.35	**14.54**	27.19	17.72
Ours (xyz+multiview+normals+lobjcls)	55.09	**37.66**	**22.93**	13.94	**28.25**	**17.85**

Table 2: Comparison of the localization results in percentage obtained by the ScanRefer and the baseline models. We also report the scores on the "unique" and "multiple" subsets, where the target object is the unique one in the scene and there are multiple objects similar to the target, respectively. Our method outperforms the baselines on all metrics by a large margin.

classification loss:

$$\mathcal{L} = \alpha\mathcal{L}_{ref} + \beta\mathcal{L}_{det} + \gamma\mathcal{L}_{cls} \quad (2)$$

where α, β and γ are the weights for the individual loss terms and are set to 0.1, 10, and 1 in our experiments.

5.4. Training and Inference

Training During the 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 use a softmax function to compress the raw scores from the localization module to $[0, 1]$. The higher the predicted score is, the more likely the proposal will be chosen as output. In addition, to filter out invalid object proposals which stand for false detections, we utilize the predicted objectness mask to ensure that only the 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 will be 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.

5.5. Implementation Details

We implement our architecture using PyTorch and train the model end-to-end using the ADAM optimizer [27] with a learning rate of $1e{-}3$. We train the model for roughly $130,000$ iterations until it fully converges. 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.

6. Experiments

Train/Val Split. We adopt the official ScanNet [8] split from which we construct a training set with 36,665 samples and a validation set with 9,508 samples. The split ensures disjoint train/val scenes (results are reported on val).

Metric. To evaluate the performance of our method, we measure precision for thresholded intersection over union (IoU) between the predicted and ground truth bounding box. Similar to work with 2D images, we use precision@IoU k as our metric, where the threshold value k is set to 0.25 and 0.5 in our experiments.

Baselines:
SCRC A 2D image baseline for visual grounding by extending SCRC [20] to 3D using back-projection. Since the method operates on a single frame, we retrieve the closest frames from the ScanNet frames using the camera pose to construct the training set. At inference time, we feed in every 20th frame and predict the target bounding boxes in each frame. Finally, we select the bounding box with the highest similarity score from the bounding box candidates and project them to 3D using the corresponding depth map.
PointRefNet We modify the PointNet++ [47] semantic segmentation pipeline to include language descriptions by feeding in the sentence embeddings from a GLoVE [41] +

Figure 7: Qualitative results from baseline methods and ScanRefer. Predictions are marked green if they have an IoU score higher than 0.5, otherwise they are marked red.

GRU [6] encoder, and fusing them with the point feature maps.

VoteNetRand From predicted object proposals, we select one of the bounding box proposals predicted by our architecture with the correct semantic class label.

OracleCatRand To illustrate the difficulty of our task, we use an oracle with ground truth bounding boxes of objects, and select a random box that matches the object category.

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 1,572 and 7,936 samples, respectively. The "unique" subset contains samples where only one unique object from a certain category matches the description,

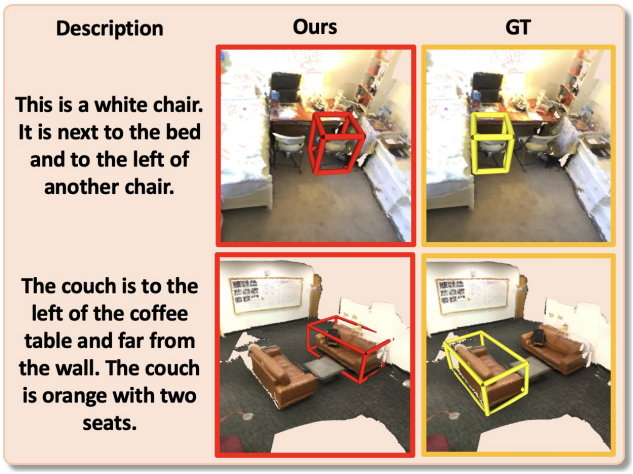

Figure 8: Limitation: sometimes ScanRefer fails to disambiguate objects of similar appearance in a scene.

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. 2).

6.2. Quantitative Analysis

As Tab. 2 shows, our method equipped with a language-to-object classifier and fed with spatial coordinates and multi-view features outperforms the baselines. Notably, the PointRefNet baseline performs better on the "unique" subset where the prediction relies more on identifying the object category, while it has lower performance on the "multiple" split where it cannot distinguish between multiple objects (see Fig. 7). With 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 the object detection component can still 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 SCRC which is limited by the view of a single frame.

6.3. Qualitative Analysis

Fig. 7 shows results produced by SCRC, PointRefNet, 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, PointRefNet fails to predict correct bounding boxes since it does not identify object instances, while SCRC is limited by the single view and hence cannot produce accurate bounding boxes in 3D space. Some failure cases of our method are displayed in Fig. 8, indicating that our architecture cannot handle all spatial relations to distinguish between ambiguous objects.

6.4. Ablation Studies

Does a language-based object classifier help? To show the effectiveness of the extra supervision on input descriptions, we conduct an ablation experiment with the language to object classifier (+lobjcls) and without. As Tab. 2 shows, 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 its sibling model trained with only geometry (xyz+lobjcls) and the one trained with additional RGB values from the reconstructed meshes (xyz+rgb+lobjcls). As shown in Tab. 2, ScanRefer trained with geometry and pre-processed multi-view image features outperforms the other two models. Notably, the performance of models trained with extra color information experiences an immediate increase when compared to the one trained only with geometry.

Do other features help? We include normals from the ScanNet meshes to the input point cloud features and compare performance against the networks trained without them. As Tab. 2 indicates, the additional 3D information can improve the scores and 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 first introduce the ScanRefer dataset which contains 46,173 unique descriptions for 9,943 objects from 703 ScanNet [8] scenes. We further 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. Our method outperforms state-of-the-art baseline methods by a large margin. 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. 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*. Finally, we thank Angela Dai for the video voice-over.

References

[1] Panos Achlioptas, Judy Fan, Robert XD Hawkins, Noah D Goodman, and Leonidas J Guibas. ShapeGlot: Learning language for shape differentiation. In *Proc. International Conference on Computer Vision (ICCV)*, 2019.

[2] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3D: Learning from RGB-D data in indoor environments. In *Proceedings of the International Conference on 3D Vision (3DV)*, 2017.

[3] Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. ShapeNet: An information-rich 3D model repository. *arXiv preprint arXiv:1512.03012*, 2015.

[4] Ding-Jie Chen, Songhao Jia, Yi-Chen Lo, Hwann-Tzong Chen, and Tyng-Luh Liu. See-through-text grouping for referring image segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7454–7463, 2019.

[5] Kevin Chen, Christopher B Choy, Manolis Savva, Angel X Chang, Thomas Funkhouser, and Silvio Savarese. Text2Shape: Generating shapes from natural language by learning joint embeddings. In *Proc. Asian Conference on Computer Vision (ACCV)*, 2018.

[6] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[7] Angela Dai and Matthias Nießner. 3DMV: Joint 3D-multi-view prediction for 3D semantic scene segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 452–468, 2018.

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

[9] Samyak Datta, Karan Sikka, Anirban Roy, Karuna Ahuja, Devi Parikh, and Ajay Divakaran. Align2Ground: Weakly supervised phrase grounding guided by image-caption alignment. In *Proceedings of the IEEE International Conference on Computer Vision*, 2019.

[10] Pelin Dogan, Leonid Sigal, and Markus Gross. Neural sequential phrase grounding (SeqGROUND). In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4175–4184, 2019.

[11] Cathrin Elich, Francis Engelmann, Jonas Schult, Theodora Kontogianni, and Bastian Leibe. 3D-BEVIS: Birds-eye-view instance segmentation. *arXiv preprint arXiv:1904.02199*, 2019.

[12] Francis Engelmann, Theodora Kontogianni, and Bastian Leibe. Dilated point convolutions: On the receptive field of point convolutions. *arXiv preprint arXiv:1907.12046*, 2019.

[13] Fangxiang Feng, Xiaojie Wang, and Ruifan Li. Cross-modal retrieval with correspondence autoencoder. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 7–16. ACM, 2014.

[14] Jiuxiang Gu, Jianfei Cai, Shafiq R Joty, Li Niu, and Gang Wang. 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*, pages 7181–7189, 2018.

[15] Richang Hong, Daqing Liu, Xiaoyu Mo, Xiangnan He, and Hanwang Zhang. Learning to compose and reason with language tree structures for visual grounding. *IEEE transactions on pattern analysis and machine intelligence*, 2019.

[16] Matthew Honnibal and Ines Montani. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear, 2017.

[17] Ji Hou, Angela Dai, and Matthias Nießner. 3D-SIS: 3D semantic instance segmentation of RGB-D scans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4421–4430, 2019.

[18] Paul VC Hough. Machine analysis of bubble chamber pictures. In *Conf. Proc.*, volume 590914, pages 554–558, 1959.

[19] Ronghang Hu, Marcus Rohrbach, and Trevor Darrell. Segmentation from natural language expressions. In *European Conference on Computer Vision*, pages 108–124. Springer, 2016.

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

[21] Yan Huang, Wei Wang, and Liang Wang. Instance-aware image and sentence matching with selective multimodal LSTM. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2310–2318, 2017.

[22] Yan Huang, Qi Wu, Chunfeng Song, and Liang Wang. Learning semantic concepts and order for image and sentence matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6163–6171, 2018.

[23] Hou Ji, Angela Dai, and Matthias Nießner. 3D-SIC: 3D semantic instance completion for RGB-D scans. In *arXiv preprint arXiv:1904.12012*, 2019.

[24] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3128–3137, 2015.

[25] Andrej Karpathy, Armand Joulin, and Li Fei-Fei. Deep fragment embeddings for bidirectional image sentence mapping. In *Advances in neural information processing systems*, pages 1889–1897, 2014.

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

[27] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[28] Ryan Kiros, Ruslan Salakhutdinov, and Richard S Zemel. Unifying visual-semantic embeddings with multimodal neural language models. *arXiv preprint arXiv:1411.2539*, 2014.

[29] Jean Lahoud, Bernard Ghanem, Marc Pollefeys, and Martin R Oswald. 3D instance segmentation via multi-task metric learning. *arXiv preprint arXiv:1906.08650*, 2019.

[30] Ruiyu Li, Kaican Li, Yi-Chun Kuo, Michelle Shu, Xiaojuan Qi, Xiaoyong Shen, and Jiaya Jia. Referring image segmentation via recurrent refinement networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5745–5753, 2018.

[31] Shuang Li, Tong Xiao, Hongsheng Li, Wei Yang, and Xiaogang Wang. Identity-aware textual-visual matching with latent co-attention. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1890–1899, 2017.

[32] Chenxi Liu, Zhe Lin, Xiaohui Shen, Jimei Yang, Xin Lu, and Alan Yuille. Recurrent multimodal interaction for referring image segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1271–1280, 2017.

[33] Daqing Liu, Hanwang Zhang, Feng Wu, and Zheng-Jun Zha. Learning to assemble neural module tree networks for visual grounding. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4673–4682, 2019.

[34] Jiasen Lu, Caiming Xiong, Devi Parikh, and Richard Socher. 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*, pages 375–383, 2017.

[35] Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L Yuille, and Kevin Murphy. Generation and comprehension of unambiguous object descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 11–20, 2016.

[36] Edgar Margffoy-Tuay, Juan C Pérez, Emilio Botero, and Pablo Arbeláez. Dynamic multimodal instance segmentation guided by natural language queries. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 630–645, 2018.

[37] Cecilia Mauceri, Martha Palmer, and Christoffer Heckman. SUN-Spot: An RGB-D dataset with spatial referring expressions. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 0–0, 2019.

[38] Gaku Narita, Takashi Seno, Tomoya Ishikawa, and Yohsuke Kaji. PanopticFusion: Online volumetric semantic mapping at the level of stuff and things. *arXiv preprint arXiv:1903.01177*, 2019.

[39] Anh Nguyen, Thanh-Toan Do, Ian Reid, Darwin G Caldwell, and Nikos G Tsagarakis. Object captioning and retrieval with natural language. *arXiv preprint arXiv:1803.06152*, 2018.

[40] Adam Paszke, Abhishek Chaurasia, Sangpil Kim, and Eugenio Culurciello. ENet: A deep neural network architecture for real-time semantic segmentation. *arXiv preprint arXiv:1606.02147*, 2016.

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

[42] Bryan A Plummer, Liwei Wang, Chris M Cervantes, Juan C Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *Proceedings of the IEEE international conference on computer vision*, pages 2641–2649, 2015.

[43] Bryan A Plummer, Paige Kordas, M Hadi Kiapour, Shuai Zheng, Robinson Piramuthu, and Svetlana Lazebnik. Conditional image-text embedding networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 249–264, 2018.

[44] Mihir Prabhudesai, Hsiao-Yu Fish Tung, Syed Ashar Javed, Maximilian Sieb, Adam W Harley, and Katerina Fragkiadaki. Embodied language grounding with implicit 3D visual feature representations. *arXiv preprint arXiv:1910.01210*, 2019.

[45] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. PointNet: Deep learning on point sets for 3D classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 652–660, 2017.

[46] Charles R Qi, Or Litany, Kaiming He, and Leonidas J Guibas. Deep hough voting for 3D object detection in point clouds. In *Proceedings of the IEEE International Conference on Computer Vision*, 2019.

[47] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. PointNet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in neural information processing systems*, pages 5099–5108, 2017.

[48] Yuankai Qi, Qi Wu, Peter Anderson, Marco Liu, Chunhua Shen, and Anton van den Hengel. RERERE: Remote embodied referring expressions in real indoor environments. *arXiv preprint arXiv:1904.10151*, 2019.

[49] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. *arXiv preprint arXiv:1605.05396*, 2016.

[50] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[51] Anna Rohrbach, Marcus Rohrbach, Ronghang Hu, Trevor Darrell, and Bernt Schiele. Grounding of textual phrases in images by reconstruction. In *European Conference on Computer Vision*, pages 817–834. Springer, 2016.

[52] Arka Sadhu, Kan Chen, and Ram Nevatia. Zero-shot grounding of objects from natural language queries. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4694–4703, 2019.

[53] Shikhar Sharma, Dendi Suhubdy, Vincent Michalski, Samira Ebrahimi Kahou, and Yoshua Bengio. ChatPainter: Improving text to image generation using dialogue. *arXiv preprint arXiv:1802.08216*, 2018.

[54] Shuran Song, Samuel P Lichtenberg, and Jianxiong Xiao. SUN RGB-D: A RGB-D scene understanding benchmark suite. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 567–576, 2015.

[55] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3156–3164, 2015.

[56] Liwei Wang, Yin Li, and Svetlana Lazebnik. Learning deep structure-preserving image-text embeddings. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5005–5013, 2016.

[57] Liwei Wang, Yin Li, Jing Huang, and Svetlana Lazebnik. Learning two-branch neural networks for image-text matching tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(2):394–407, 2018.

[58] Peng Wang, Qi Wu, Jiewei Cao, Chunhua Shen, Lianli Gao, and Anton van den Hengel. Neighbourhood watch: Referring expression comprehension via language-guided graph attention networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1960–1968, 2019.

[59] Fanyi Xiao, Leonid Sigal, and Yong Jae Lee. Weakly-supervised visual grounding of phrases with linguistic structures. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5945–5954, 2017.

[60] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057, 2015.

[61] Bo Yang, Jianan Wang, Ronald Clark, Qingyong Hu, Sen Wang, Andrew Markham, and Niki Trigoni. Learning object bounding boxes for 3D instance segmentation on point clouds. *arXiv preprint arXiv:1906.01140*, 2019.

[62] Sibei Yang, Guanbin Li, and Yizhou Yu. Dynamic graph attention for referring expression comprehension. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4644–4653, 2019.

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

[64] Linwei Ye, Mrigank Rochan, Zhi Liu, and Yang Wang. Cross-modal self-attention network for referring image segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10502–10511, 2019.

[65] Licheng Yu, Patrick Poirson, Shan Yang, Alexander C Berg, and Tamara L Berg. Modeling context in referring expressions. In *European Conference on Computer Vision*, pages 69–85. Springer, 2016.

[66] Licheng Yu, Hao Tan, Mohit Bansal, and Tamara L Berg. A joint speaker-listener-reinforcer model for referring expressions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7282–7290, 2017.

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

[68] Fang Zhao, Jianshu Li, Jian Zhao, and Jiashi Feng. Weakly supervised phrase localization with multi-scale anchored transformer network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5696–5705, 2018.

[69] C Lawrence Zitnick and Piotr Dollár. Edge boxes: Locating object proposals from edges. In *European conference on computer vision*, pages 391–405. Springer, 2014.

Supplementary Material

A. Dataset

A.1. Statistics

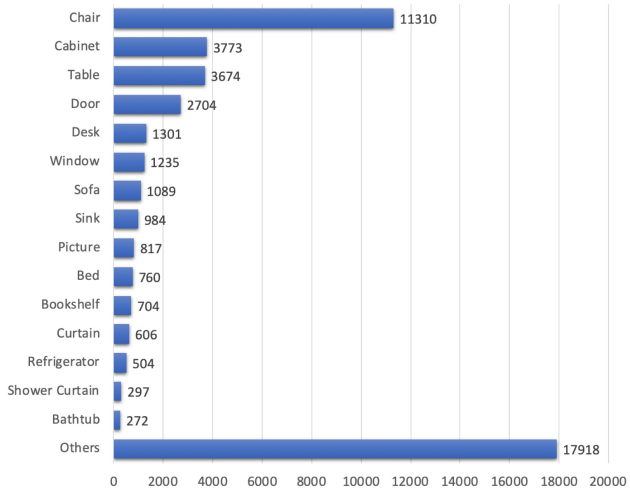

Figure 9: Distribution of categories of annotated objects in the ScanRefer dataset.

We present the distribution of categories of the ScanRefer dataset in Fig. 9. 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 [8]. In total, we annnotate 9,976 objects from 265 categories from ScanNet [8]. 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. 10 shows the distribution of finer-grained objects in the category "Others". For each of the 18 coarse categories, Fig. 11 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.

A.2. Collection Details

In this section, we provide more details of the data annotation and verification processes of ScanRefer.

A.2.1 Annotation

We deploy our web-based annotation application on Amazon Mechanical Turk (AMT) to collect object descriptions in the reconstructed RGB-D scans. To ensure that the initial descriptions are written in proper English, we restrict

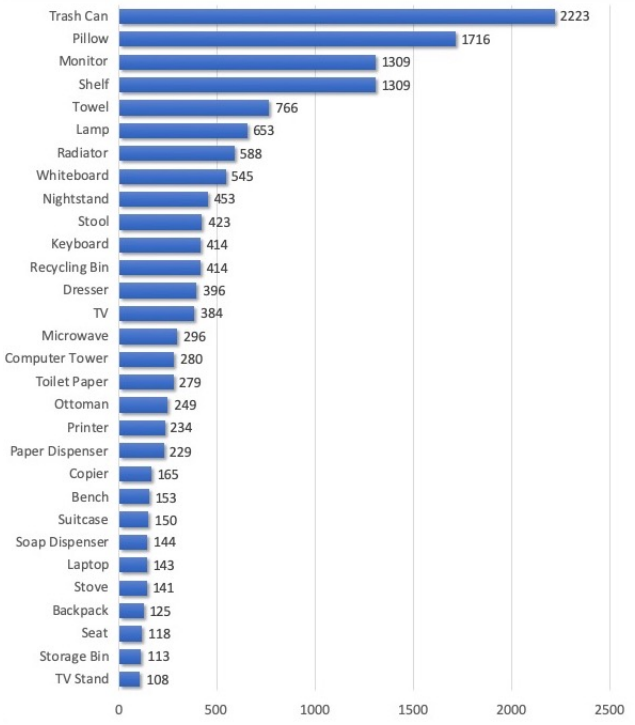

Figure 10: Distribution of the top 30 categories in the "Others" category of the ScanRefer dataset.

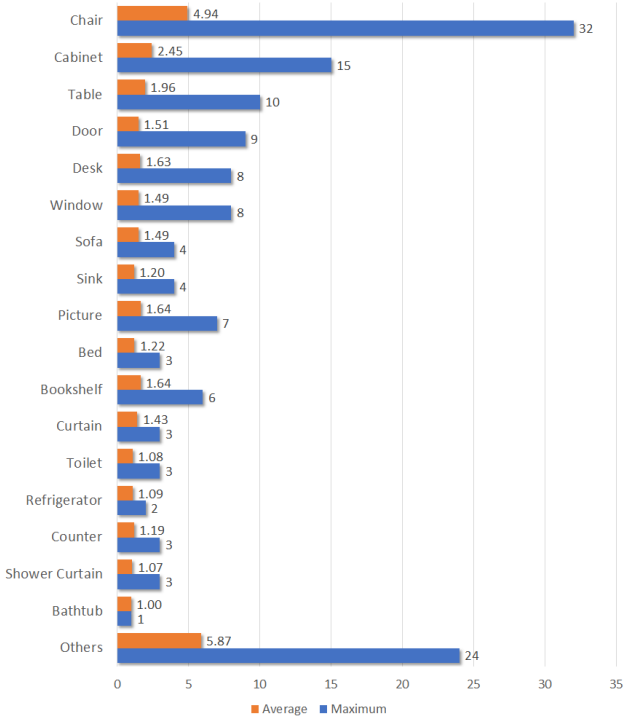

Figure 11: Average and maximum numbers of objects in each category per scene in the ScanRefer dataset. For each category, we only consider scenes that contains the corresponding objects.

the workers to be from the United States, the United Kingdom, Canada or 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. 12. 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".

A.2.2 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. 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 descriptions to our database. To guide the trained verifiers, we provide the verification instructions as shown in Fig. 13.

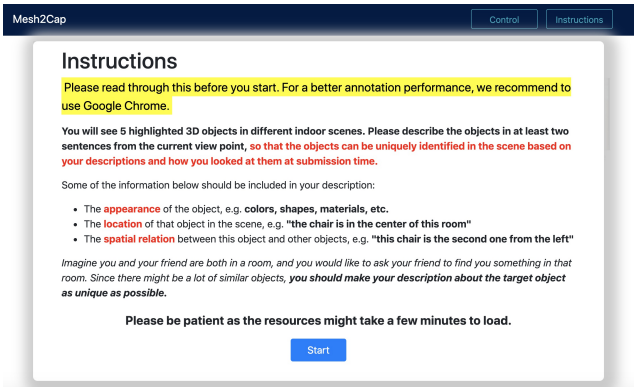

Figure 12: Screenshot of the instructions for the Amazon Mechanical Turk workers before providing descriptions for objects.

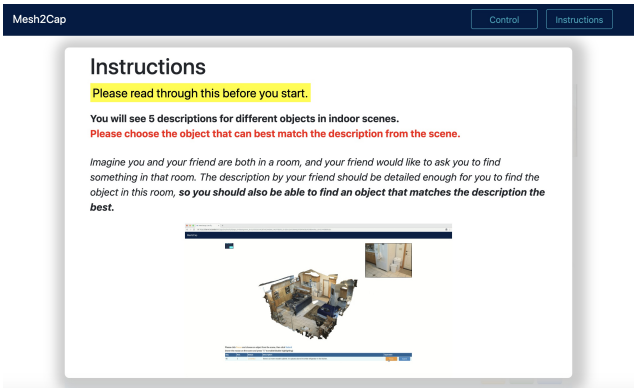

Figure 13: Screenshot of the instructions that the trained verifiers have to go through before starting the verification.

B. Object Detection Results

Tab. 3 shows the object detection results from Qi et al. [46] and all ablations of our architectures. We apply the mean average precision (mAP) thresholded by IoU value 0.5 as our evaluation metric. All values in Tab. 3 are in percentage. We exclude structural objects such as "Floor" and "Wall". We group all categories which are not in the ScanNet benchmark categories [8] including "Otherfurnitures", "Otherstructure" and "Otherprop" into the "Others" category in our evaluation. Note that the "Others" category in our evaluation includes more types of objects, such as "Pillow" and "Keyboard", than the "Otherfurniture" category of the ScanNet benchmark.

As shown in Tab. 3, including point normals as extra point features (rows [3,5]) in training increases the detection results when compared to the models trained without the normals (rows [2,4]). Also, training with extracted high-level color features from the multi-view images (rows [4,5]) also produces better detection results compared with the results from models trained with just the raw RGB values (rows [2,3]). Note that the networks equipped with the language-based object classifier (rows [6-10]) fail to produce better detections than the ones without the extra language classifier module (rows [1-5]). This is to be expected as the language provides information to help differentiate between objects of the same category, but does not contain any information for a better bounding box prediction for object detection.

C. Additional Qualitative Analysis

We present additional examples of localization results by our method for further qualitative analysis. As shown in Fig. 14, our method can directly incorporate the immediate detections and further produce the bounding boxes for the target objects by the learned fused features. Our method can not only extract the appearance and location information from the input description, but also produce accurate localization bounding boxes for the targets. There are also challenging cases where our method fails to localize the target object. As Fig. 15 shows, our method is sometimes limited by inaccurate detections, especially for small objects such as the pictures (5th row). This indicates that the object detection submodule can still be improved. Also, our method could not handle all spatial relations in the input descriptions. For instance, for comparative phrases (e.g., "leftmost" or "rightmost") or counting (e.g., "the second one from the left"), the models fails to pick out the correct object, as shown in Fig. 16.

	cab.	bed	chair	sofa	tabl.	door	wind.	bkshf.	pic.	cntr.	desk	curt.	fridg.	showr.	toil.	sink	bath.	others	mAP
[0]	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
[1]	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
[2]	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
[3]	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
[4]	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
[5]	**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**
[6]	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
[7]	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
[8]	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
[9]	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
[10]	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: [0] VoteNet [46], [1] Ours (xyz), [2] Ours (xyz+rgb), [3] Ours (xyz+rgb+normals), [4] Ours (xyz+multiview), [5] Ours (xyz+multiview+normals), [6] Ours (xyz+lobjcls), [7] Ours (xyz+rgb+lobjcls), [8] Ours (xyz+rgb+normals+lobjcls), [9] Ours (xyz+multiview+lobjcls), [10] Ours (xyz+multiview+normals+lobjcls). Training with the point normals (compare rows [3,5] to rows [2,4]) and multiview features (compare rows [4,5] to rows [2,3]) clearly leads to a better performance. As expected, the models equipped with language-based object classifier (rows [6-10]) does not results in better object detection compared to the models without such a module (rows [1-5]).

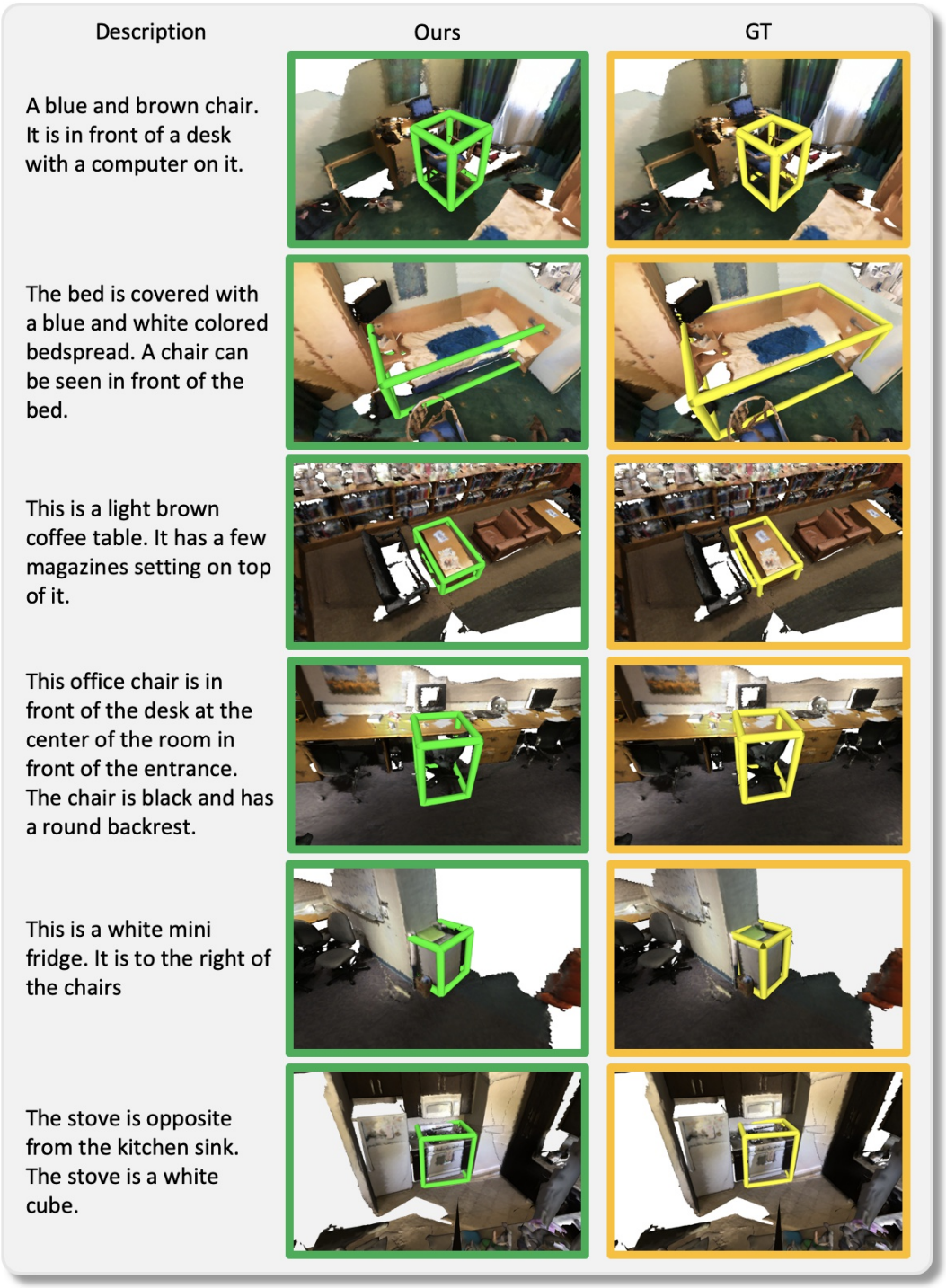

Figure 14: Additional qualitative analysis. Our method is capable of localizing the target object in a 3D indoor scene with the help of the free-form description.

Figure 15: The performance of our method is limited by the results of the object detection. Inaccurate detections for small and thin objects, e.g., a picture, could lead to failure cases of localization.

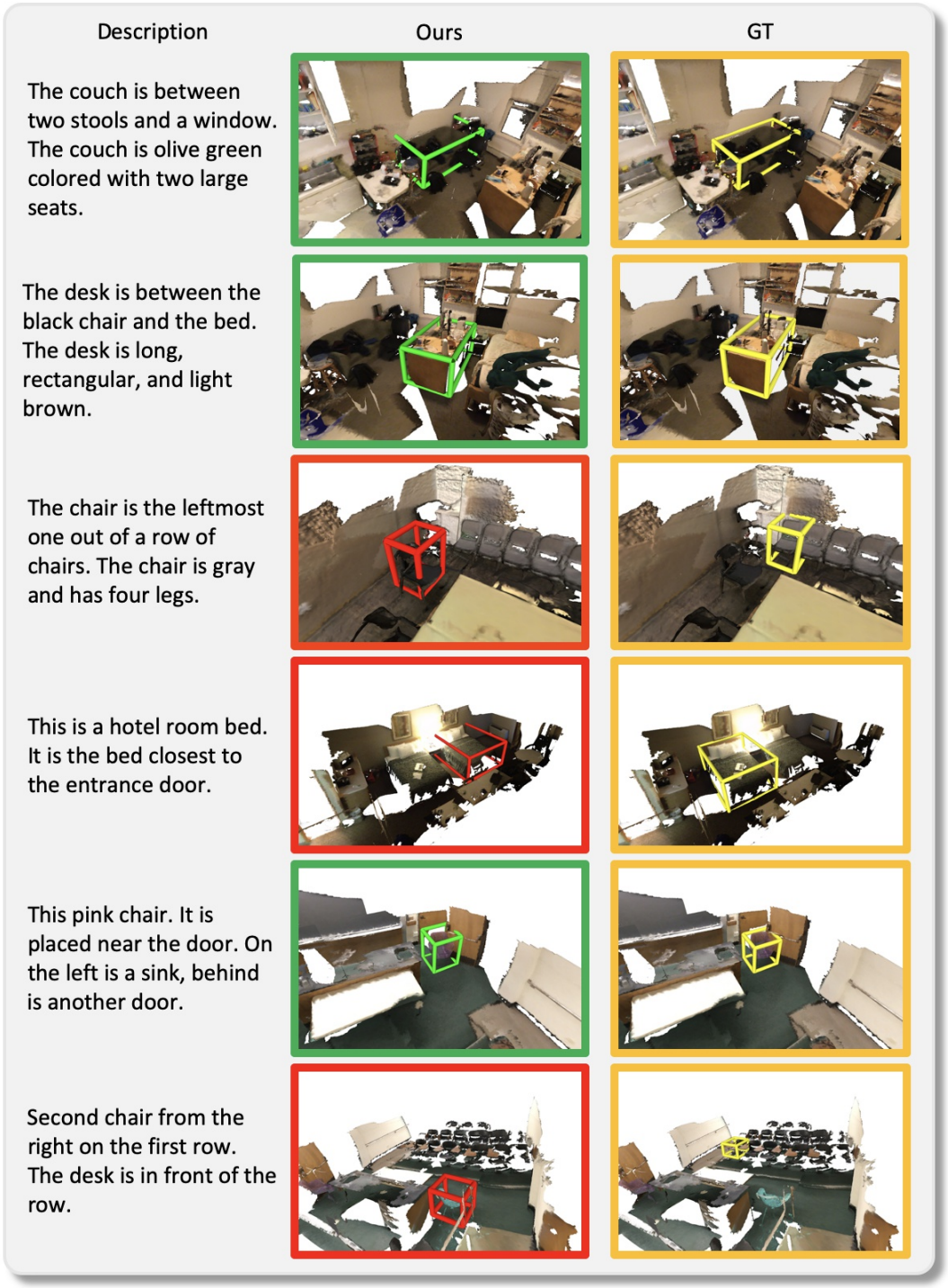

Figure 16: Localizing objects which are not unique in the scenes could be challenging. Our method fails to distinguish the correct target among several similar objects, e.g. finding a target chair in a classroom which are full of identical chairs.