3DRefTransformer: Fine-Grained Object Identification in Real-World Scenes Using Natural Language

Ahmed Abdelreheem, Ujjwal Upadhyay, Ivan Skorokhodov,
Rawan Al Yahya, Jun Chen, Mohamed Elhoseiny
King Abdullah University of Science and Technology (KAUST), Saudi Arabia
{ahmed.abdelreheem, ujjwal.upadhyay, ivan.skorokhodov, rawan.yahya, jun.chen, mohamed.elhoseiny}@kaust.edu.sa

Abstract

In this paper, we study fine-grained 3D object identification in real-world scenes described by a textual query. The task aims to discriminatively understand an instance of a particular 3D object described by natural language utterances among other instances of 3D objects of the same class appearing in a visual scene. We introduce the 3DRefTransformer net, a transformer-based neural network that identifies 3D objects described by linguistic utterances in real-world scenes. The network's input is 3D object segmented point cloud images representing a real-world scene and a language utterance that refers to one of the scene objects. The goal is to identify the referred object. Compared to the state-of-the-art models that are mostly based on graph convolutions and LSTMs, our 3DRefTransformer net offers two key advantages. First, it is an end-to-end transformer model that operates both on language and 3D visual objects. Second, it has a natural ability to ground textual terms in the utterance to the learning representation of 3D objects in the scene. We further incorporate object pairwise spatial relation loss and contrastive learning during model training. We show in our experiments that our model improves the performance upon the current SOTA significantly on Referit3D Nr3D and Sr3D datasets. Code and Models will be made publicly available at `https://vision-cair.github.io/3dreftransformer/`.

1. Introduction

In recent years, A lot of interest has been demonstrated in connecting vision and language. Improving the performance of vision and language-based tasks is essential for a wide range of applications ranging from 2D image related applications like image captioning [19, 35, 38, 25, 2], text-to-image generation [13, 33, 9], and visual question answering [3, 30] to more complex tasks in robotics like language guided navigation [20, 22, 10].

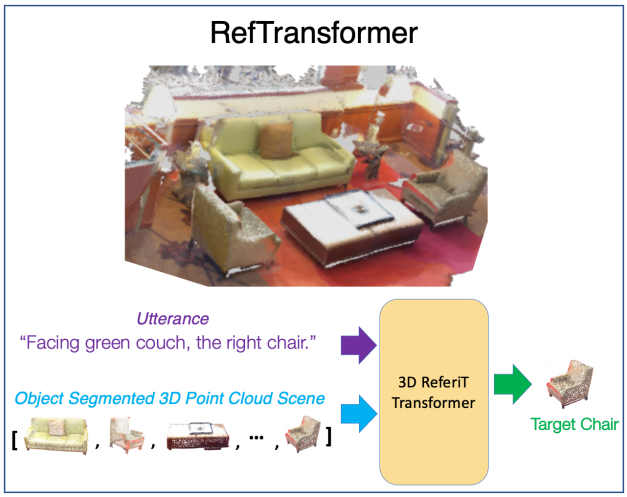

Figure 1. Our model takes a 3D point cloud visual input of M objects. These objects includes target object; objects of the same class as the target object, i.e distractors, and objects of another class as target object. Simultaneously, the model takes a language input of an utterance describing the target object. 3DRefTransformer Net distinguishes target object from other objects in the scene.

The task of visual grounding has attracted a lot of attention [16, 14, 24, 8]. The goal of visual grounding is to localize an object in an image given its natural language description. However, the majority of work focuses on forming a relationship between language and 2D images, which despite commendable progress, fails to capture 3D reality. Additionally, there is a lack of research identifying fine-grained 3D objects, despite numerous applications, such as autonomous driving, and navigation [37, 31].

Solving vision and language tasks where we operate directly on 3D visual representations (like 3D point clouds) has seen considerable interest in recent years. The reason behind this interest is the abundance of 3D sensors like LiDARs, and it would be more plausible to operate directly on

the 3D modality. However, modeling language to operate on real 3D scenes is significantly less explored, and despite its importance, it lacks large-scale benchmarks compared to other 3D problems. Tasks like object identification or localization in a 3D point cloud scene using natural language started to grab attention in the community. Referit3D [1] and ScanRefer [5] works provided datasets suited to these kinds of tasks. In Referit3D, the authors proposed solving the object identification task using DGCNN networks [36]. They assume that the input is object-segmented point clouds and assume that the 3D object bounding box proposals are provided. While the authors in ScanRefer [5] remove this assumption and the input to their proposed model is just the complete 3D scene point cloud. There is a follow-up by [18], where they replaced the 3D object proposals with instance segmentation masks, where each semantic mask represents a 3D object in the scene.

In this work, we mainly focus on the fine-grained object identification problem. We address the Referit3D task, choosing or identifying the target object described by one of its language descriptions (called utterances) that describe this target object in the 3D scene. Each utterance in Referit3D, Nr3D, and Sr3D datasets uniquely describes an object in the 3D scene, where each scene has at least one object of the class of interest among other objects. Such complex tasks require the model to reason about how the scene objects are arranged with respect to each other in the scene. To that end, We propose a transformer-based model that achieved the state of the art performance with significant improvement. Adopting off-the-shelf transformer models from the 2D domain can be limiting. Transformer models require a huge amount of data to perform well. For example, LXMERT [32], a transformer model for visual and language cross-modality encoder representation, is pre-trained on a set of large-scale datasets in terms of million image and text pairs. To sum up, our contributions in this paper are the following:

- We proposed 3DRefTransformer, a transformer-based network that contains uni-modal and cross-modal attention mechanisms. The model inherits a natural ability to ground the textual terms to their corresponding objects. However, due to the scarcity of data in the 3D domain, adopting transformer models from the 2D domain remains difficult.
- To alleviate the scarcity of (vision/language) datasets in a 3D domain, we propose a pairwise object spatial relation prediction loss and also adopt a contrastive learning approach to improve the performance of the model.
- To incorporate the spatial relation loss, We generate scene graphs for ScanNet [7] dataset in a synthetic way and show that it can improve the performance.

2. Related Work

Point Cloud Processing Methods. Recent works have proposed point-based methods for point cloud feature representation that can be used for 3D shape classification and semantic segmentation tasks. PointNet [27] and PointNet++ [28] are point-wise based methods. PointNet incorporates symmetric operations and shared MLP layers to generate global feature representation for the point cloud. Instead, PointNet++ uses a hierarchical feature representation of local regions to describe a point cloud. Guo et al. [11] are the first to incorporate transformer encoders to process point clouds. Since self-attention is permutation invariant to input sequences, the authors find the transformer well suited for point cloud processing.

2D Vision and Language. Significant progress has been in 2D vision, and natural language achieving strong performance on tasks ranging from image captioning [19, 35, 38, 25, 2], visual grounding [17], text-to-image generation, and visual question answering [23, 3, 30]. Recent methods tackling these tasks are based on transformer networks (e.g., [34, 32, 21, 12, 29]), demonstrating state-of-the-art performance in a variety of tasks. What drove their success in most of these methods in 2D vision and NLP is the availability of large-scale datasets. However, in natural 3D world scenes, it is challenging to incorporate transformer-based methods trivially. Hence, proper design of the 3D visiolinguistic extension in our case is critical for good results.

Language Guided 3D Object Localization. Vision and language in 3D domain have recently gained a lot of attention, and it is still considered a not well-explored area, unlike the 2D domain. Chen et al. introduced the task of localizing an object in a 3D scene using natural language [5]. The authors in [5] introduce ScanRefer dataset for such task. The input to their proposed model is a language description and a point cloud of a 3D scene. Their proposed model was based on a previous work [26], where the network first generates 3D region proposals (which are candidate objects), then the proposals' visual features are concatenated with the language description features to compute a score for each proposal. The proposal with the highest score is the output prediction. Concurrently, Achlioptas et al. [1], propose another two datasets Referit3D Nr3D and Sr3D. These two datasets can be used to distinguish an object in a 3D scene from the others (including objects with the same class type). Chen et al. [6] introduced a captioning model for 3D objects in point cloud scenes using the ScanRefer dataset.

3. Method

Given a 3D real-world scene S represented as an RGB-colored point cloud, it is segmented into a set of M objects

O_i such that $S = \{O_1, ..., O_M\}$. Also, an utterance U is provided, and it uniquely describes a target object in the scene S. The goal of the model is to identify the referred target object $O_T \in S$ by the utterance U. For the model to solve this task, it should first classify the 3D objects in the scene. Then it has to learn how the objects are spatially arranged in the scene with respect to each other, i.e., context and spatial relation information is crucial for solving the task successfully.

In this paper, we investigate the effect of using a self-attention transformer encoder on the model performance. In the following subsections, we discuss our network's architecture that achieves state-of-the-art performance on Referit3D benchmark, outperforming existing methods by 1.7% and 2.0% on Nr3D and Sr3D datasets, respectively.

3.1. Input Embeddings

Object-level Embedding. To be closer to Referit3D [1] model, we choose PointNet++ [28] for point clouds encoding. For each object point cloud o_i in the input scene S having M objects where $S = \{o_1, ..., o_M\}$ and $i = 1...M$, we encode its point cloud as a feature vector f_i where $F = (f_1, ..., f_M)$. We then use F as the objects' point cloud embeddings as input to the object transformer mentioned in Section. 3.2. Since the point cloud by nature has the position information, we don't use the positional embedding that is necessary in case of language embeddings.

Word-level Embedding. To embed the utterance U, we generate a sequence of word embedding vectors using a trainable embedding layer. We then sum each embedding vector with its positional embedding as in [34]. The resulting word embedding vectors W are passed to the language transformer discussed in Section. 3.2.

3.2. A Transformer Encoder for Each Modality

Our proposed model employs a transformer architecture for each modality, i.e., for both visual and textual data. In contrast to several contemporary approaches that also fuse different modalities with transformer blocks [32, 21, 15], firstly, we allow the tokens of each of the two modalities to self attend only on themselves in a separate transformer encoder. Then only we allow the enhanced token representations coming out from both uni-modal transformer encoders to do cross-modal attention. Using a separate transformer encoder for each modality enables the following: In the visual transformer, each visual object first gets a better scene-aware visual representation of itself without dissipating attention weights on textual tokens. Also, it is more suitable to apply some of the auxiliary losses (i.e., Spatial Relation Loss 3.7) on the uni-modal representations than after the multi-modal transformer. The same idea applies to the utterance sequence as well. We encode the utterance words W using a transformer encoder to get an output sequence $\tilde{W}$. Both uni-modal transformers have the same number of layers L. The feature in each output sequence in both transformers has the same dimension d_h.

3.3. Positional Embedding For Object Features

PointNet++ feature representation carries some information about where the object is. In addition to that, we investigate whether the positional information in PointNet++ features is sufficient by itself or not. We implemented two different types of positional embedding.

Absolute Positional Embedding. For each object in O, we encode its bounding box centers and scales using MLP, and we add this positional embedding to the object features.

$$P_{o_i} = f_i + MLP(bbox\_center_{o_i}, bbox\_dim_{o_i}) \quad (1)$$

Relative Positional Embedding. We compute the pair-wise distances between the bounding boxes centers of the scene objects. We incorporate them in a different way than the absolute positional embedding. Instead, we pass each pair-wise through an MLP[3,32,1] and we add the result value to the computed self-attention weights before doing the softmax in the self-attention layer. Consider the input for a self attention layer in the object transformer: queries Q of size Mxd_f, keys K of size Mxd_f, and values V and the pair-wise object distances are the matrix P of dimensions MxM. Then the output of the self-attention layer is:

$$A = \frac{QK^T + P}{\sqrt{d_f}} \quad (2)$$

3.4. Multi-modal (MM) Transformer Encoder)

After we get enhanced features from each modality transformer encoder layer, we use a multi-modal transformer to further encode both modalities together. The input sequence to this MM transformer is the union of output sequences from both uni-modal transformers $Z = \tilde{F} \cup \tilde{W}$. To provide better interaction between the object features and the language features, We allow all the input tokens in Z to self-attend on each other. This type of MM transformers showed outstanding performance in the TextVQA problem as proposed in [15]. This kind of attention allows contrasting each object with each word in the utterance. The MM transformer is a stack of L self-attention layers. After passing the input sequence Z, We obtain $\tilde{Z}$ that consists of the final feature representations for each object point cloud and language word.

The reason behind using multi-modal transformer was to form a relation between the two modalities in way that we can interpret different language descriptions over all the objects present in the scenes based on the adjectives and location-identifying words.

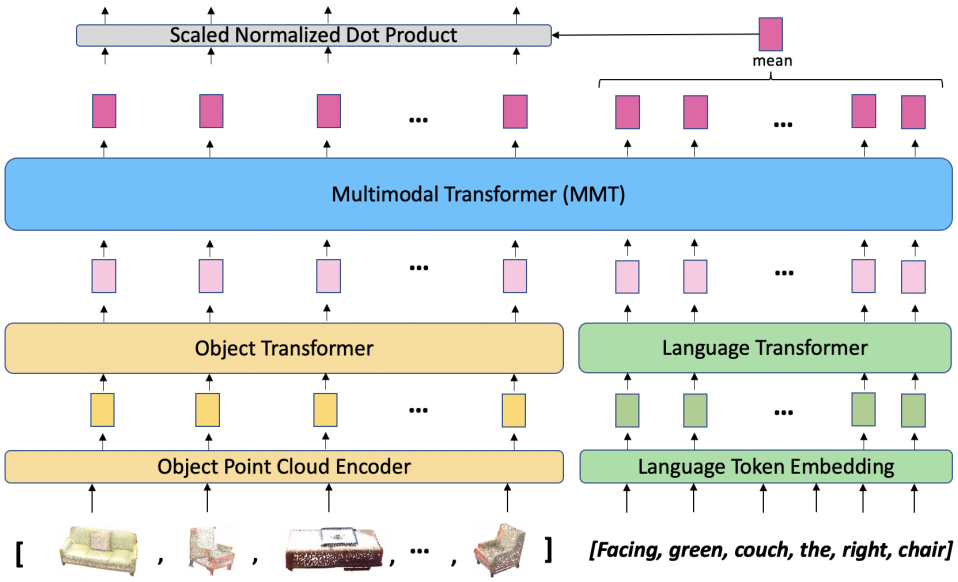

Figure 2. Our 3DRefTransformer proposed model: The model takes two input sequences: a visual sequence of 3D point clouds of M objects and a sequence of the utterance words. Each object's point cloud is encoded using PointNet++ then passes through an object transformer where each object feature self-attends on other objects in the scene to encode its context. For each word in the input sequence, we embed it using a trainable embedding layer. Then it is passed through a language transformer, where a linguistic self-attention occurs. Both transformers' output goes through a multi-modal transformer layer that applies self-attention between the linguistic and visual features. Finally, to identify the target object, we apply scaled normalized cosine similarity to compute the logits.

3.5. Contrastive Learning

We incorporated contrastive learning signals during training. Contrastive learning losses help the model better distinguish between similar language utterances and objects. For example, a (positive utterance, target object) pair should be away from other (negative utterance, distractor object) pairs. To implement such training signals, we do it in two ways:

Target object to negative utterances C_{OU}. We do a scaled cosine similarity between the target object feature and a vector containing the positive utterance in addition to sampled negative utterances in the current mini-batch. The similarity between the positive utterance and the target object should be the maximum.

$$L(\mathbf{y}_i, y_i) = \frac{1}{N_U} \sum_{i=1} -\log\left(p_i = \frac{e^{\mathbf{u}_i^\top \mathbf{o}_i}}{\sum_{j=1}^{K} e^{\mathbf{u}_i^\top \mathbf{o}_j}}\right) \tag{3}$$

Where N is number of objects in the 3D scene, and y_i is a one-hot encoding that represents the target object label.

Positive utterance to negative objects C_{UO}. Similarly, we sample from the current mini-batch object features that are not referred to by the positive utterance, and we do scaled dot product between the positive utterance and a vector containing the target object feature as well as the sampled negative object features in the current mini-batch. The similarity between the positive utterance and the target object should be the maximum.

$$L(\mathbf{u}_i, y_i) = \frac{1}{N} \sum_{i=1}^{N} -\log\left(p_i = \frac{e^{\mathbf{u}_i^\top \mathbf{o}_i}}{\sum_{j=1}^{K} e^{\mathbf{u}_i^\top \mathbf{o}_j}}\right) \tag{4}$$

Where N is number of objects in the 3D scene, and y_i is a one-hot encoding that represents the target object label.

3.6. Output: Object Identification

To identify the target object, we further used scaled cosine distance to compute the similarity between each object feature in $\tilde{Z}$ and a single feature $\tilde{U}$ that represents the whole utterance, where $\tilde{U}$ is the mean of all words features in $\tilde{Z}$. The scale factor η is used as a temperature factor. We apply such an operation as we want the target object visual feature to be as close as possible to the utterance language feature.

$$Y = \eta \tilde{Z} \tilde{\mathbf{U}}^\top \tag{5}$$

3.7. Training Objectives

In our experiments, we train all models from scratch in an end-to-end manner. The model was trained on 4 objective functions. The first loss L_{ref} is the target object referential cross-entropy loss, which maintains that the model has a high similarity between the utterance and the target

object features. The second term L_c is object classification cross-entropy loss, which operates directly on the PointNet++ [28] features. The third loss L_t is a classification cross-entropy loss applied on the utterance feature forcing the language representation to learn the referred class label.

Spatial Relation Prediction. Given the input objects sequence O, the task of the object transformer encoder is to predict the spatial relationship for some annotated object pairs (o_i, o_j). The goal of this task is to encourage the object transformer to understand the spatial relationship between the objects in the 3D scene. Having spatially aware transformers, we allow better performance in the main referential task. To achieve this, we synthesized some scenes to have the object transformer better understand the importance of spatial relations between various objects present in the scene. Synthesis of this scene was a randomized process where we place objects in the scene with one of the pre-defined spatial configurations relative to already present objects. In total, we have eight unique spatial relations $R_{spa} = \{left, right, front, back, above, below, closest, farthest\}$. During training, we pass at most 200 random sampled tuples where each tuple can be represented as $(o_i, o_j, r^{i,j}_{spa})$. o_i and o_j are two different objects present in the same scene S and we apply the cross-entropy classification loss on the concatenated object features representations improved by the object transformer.

$$L_{spa} = crossentropy(MLP([o_i, o_j]), r^{i,j}_{spa}) \quad (6)$$

To sum up, in Equation. 7, we show the weighted sum of the four loss terms used during training. Empirically, we set $\gamma = 3.3$ using the performance on the validation set.

$$L = L_{ref} + \frac{1}{2}(L_{C_{UO}} + L_{C_{OU}}) + \frac{1}{2}(L_c + L_t) + \gamma L_{spa} \quad (7)$$

4. Experiments

4.1. Datasets

We focus mainly on Referit3D problem that emphasizes on distinguishing a target object among its same-class distractors and other objects in a 3D scene. We evaluate our model on the Referit3D Nr3D and Sr3D datasets and compare it with the state-of-the-art models.

Nr3D dataset. It consists of 41.5k human utterances covering 76 fine-grained object classes. Each utterance uniquely describes a target object in a 3D ScanNet Scene.

Sr3D dataset. It consists of 83.5k template-based synthesized utterances.

Train/Val Splits. We use the same train/val splits as in ReferIt3D[1].

4.2. Implementation Details

Each transformer module in our model consists of two transformer layers. While positional embedding is essential when encoding sequential linguistic inputs using a transformer model, it is not necessary for point cloud representation. Point cloud representation of 3D objects is naturally defined as a set where each feature f is a vector holding the spatial position of the object. Hence, we do not need to apply positional encoding for point cloud objects. In section 4.3, we ablate the performance when using different positional embeddings.

Following [1], for each training example (utterance, ScanNet [7] scene objects), we consider at most 52 objects per scene during training. During testing, we consider at most 89 objects per scene. In both training and testing, we sample 1024 points from every object's point cloud.

The language random embedding for the input utterance has a dimensionality of 128. Similarly, the dimension of the hidden features, the language token embedding (PointNet++), and the objects' feature dimension are all set to 128. Empirically, the scale factor in the normalized scaled cosine similarity is 3.3. We refer the reader to the supplementary material for further ablation on trying different values of this scale factor.

Optimizer. During training, we used an Nvidia V100 GPU with a batch size of 16 and a base learning rate of 0.0005 with an ADAM optimizer for 60 epochs. We incorporated a learning rate scheduler where the learning rate is reduced to 0.65 of its value when the validation accuracy does not increase for five consecutive epochs. For all the transformer encoders, we used the rezero normalization trick mentioned in [4] as it was empirically better than layernorm.

4.3. Ablation Studies

We compare our proposed model to the current SOTA methods. In Table. 1, we show that our model outperforms other approaches. Our model significantly outperforms the current state-of-the-art. In Referit3DNet [1], their proposed model uses DGCNN [36] network, which struggles to understand the spatial relations of objects in the scene without some additional guidance. DGCNN also struggles to contrast between different objects due to the following reason. At each DGCNN [36] graph layer, the neighborhood is computed dynamically based on the top K nearest neighbors in the latent space. Hence, most of the time, the neighborhood of each object feature in the first graph layers will be its same-class distractors features. This behavior will make it difficult for the later graph networks to understand the context between the objects in the scene.

We show that our model outperforms other models in hard contexts. Hard contexts mean that the scene has more than one object of the same class as the target object. Also,

Dataset	Method	*Accuracy*				
		Overall	*VD*	*VID*	*Easy*	*Hard*
Nr3D	Referit3DNet [1]	35.6% ± 0.7%	32.5% ± 0.7%	37.1% ± 0.8%	43.6% ± 0.8%	27.9% ± 0.7%
Nr3D	TGNN [18]	37.3% ± 0.3%	**35.8% ± 0.2%**	38.0% ± 0.3%	44.2% ± 0.4%	30.6% ± 0.2%
Nr3D	Ours w/o L_{Con}	38.2% ± 0.2%	33.3% ± 0.3%	40.5% ± 0.2%	46.0% ± 0.5%	30.6% ± 0.3%
Nr3D	**Ours**	**39.0% ± 0.2%**	34.7% ± 0.3%	**41.2% ± 0.4%**	**46.4% ± 0.4%**	**32.0% ± 0.3%**
Sr3D	Referit3DNet [1]	40.8% ± 0.2%	39.2% ± 1.0%	40.8% ± 0.1%	44.7% ± 0.1%	31.5% ± 0.7%
Sr3D	TGNN [18]	45.0% ± 0.2%	**45.8% ± 1.1%**	45.0% ± 0.2%	48.5% ± 0.2%	36.9% ± 0.5%
Sr3D	**Ours**	**47.0% ± 0.2%**	44.3% ± 0.3%	**47.1% ± 0.2%**	**50.7% ± 0.1%**	**38.3% ± 0.5%**

Table 1. Comparison with ReferIt3D/TGNN on the accuracy of referential object identification task.

Positional Embedding			*Accuracy*				
PNet++	Absolute	Relative	*Overall*	*VD*	*VID*	*Easy*	*Hard*
✓			38.2% ± 0.2%	33.3% ± 0.3%	40.5% ± 0.2%	**46.0% ± 0.5%**	30.6% ± 0.3%
✓		✓	38.0% ± 0.2%	32.8% ± 0.3%	40.5% ± 0.3%	45.6% ± 0.4%	30.6% ± 0.4%
✓	✓		**38.7% ± 0.3%**	33.9% ± 0.3%	**41.0% ± 0.3%**	45.8% ± 0.4%	**31.8% ± 0.3%**
✓	✓	✓	38.3% ± 0.4%	**34.5% ± 0.3%**	40.2% ± 0.5%	45.2% ± 0.3%	31.7% ± 0.5%

Table 2. The effect of adding absolute or relative positional embeddings to the object PointNet++ features on a model variant (trained without contrastive learning losses). The results suggests that PointNet++ conveys the positional information, However, adding explicitly absolute positional embeddings slightly improves the performance.

our model outperforms the others in view independent contexts. View independent contexts mean that in order to find the target object, you do not need to be looking from a certain view. For example, "Facing the couch, the nightstand on your left". This performance in hard and view independent contexts suggests that our model is more capable of understanding the spatial relationships between objects and is better at contrasting hard distractors of the same class as the target object.

Contrastive Training The available datasets' size is still too small compared to the large-scale datasets in the 2D domain. To alleviate such limited availability of the data, we investigate the effect of contrastive learning. Table. 1 shows the gain in performance in every type of context by incorporating the contrastive loss. The most significant improvement is in the hard contexts. By moving away from the feature representation of both the target visual object and target utterance from their distractor objects and negative utterances, the model learns how to better determine the target object among its distractors.

Object Positional Embeddings We investigate the effect of incorporating different positional embeddings to objects' point cloud features. We want to see how much PointNet++ can convey such positional information. We tried different experiments; (a) we do not add any extra positional information to the object features, (b) adding relative positional embeddings as mentioned in section **mention section**, (c) adding absolute positional embeddings, and (d) adding both relative and absolute positional embeddings. In table. 2, we find that PointNet features carry positional information by themselves, and adding relative positional embedding hurt the performance a bit, especially in the view-dependent scenes. This finding may appear counterintuitive. However, we argue that self-attention can find more representative positional embedding without explicitly adding basic relative positional embeddings.

Layers in MMT	*Accuracy*
$L_{MMT} = 2$	**39.0% ± 0.2%**
$L_{MMT} = 3$	38.4% ± 0.3%
$L_{MMT} = 4$	37.9% ± 0.3%

Table 3. 3D RefTransformer Performance with different number of layers in the multi-modal transformer. Using two encoder layers gave the best results.

Spatial Relation Loss To show the effect of introducing this loss term, we carried two experiments on a variant of our model with/without using the spatial loss. As shown in Table. 4. The model performs better by encouraging to learn such object pair-wise relations, especially in the hard and view-dependent contexts. Such contexts require a better understanding of how same-class distractor objects are placed in the scene. On the other hand, the spatial relations data used as ground truth are generated using code, this data is noisy and not covering all the possible relations, and we believe it is why it did not cause a significant boost in the model performance.

Number of layers in the multi-modal transformer. We present the effect of changing the number of layers in the multi-modal transformer on the overall performance in Ta-

	Overall	*VD*	*VID*	*Easy*	*Hard*
w/o L_{spa}	37.5% ± 0.2%	33.1% ± 0.2%	39.6% ± 0.4%	39.9% ± 0.3%	29.5% ± 0.3%
w/ L_{spa}	**38.2% ± 0.2%**	**33.3% ± 0.3%**	**40.5% ± 0.2%**	**46.0% ± 0.5%**	**30.6% ± 0.3%**

Table 4. Contrast between using spatial loss or without spatial loss: Using the spatial loss helps especially in hard contexts and view independent contexts.

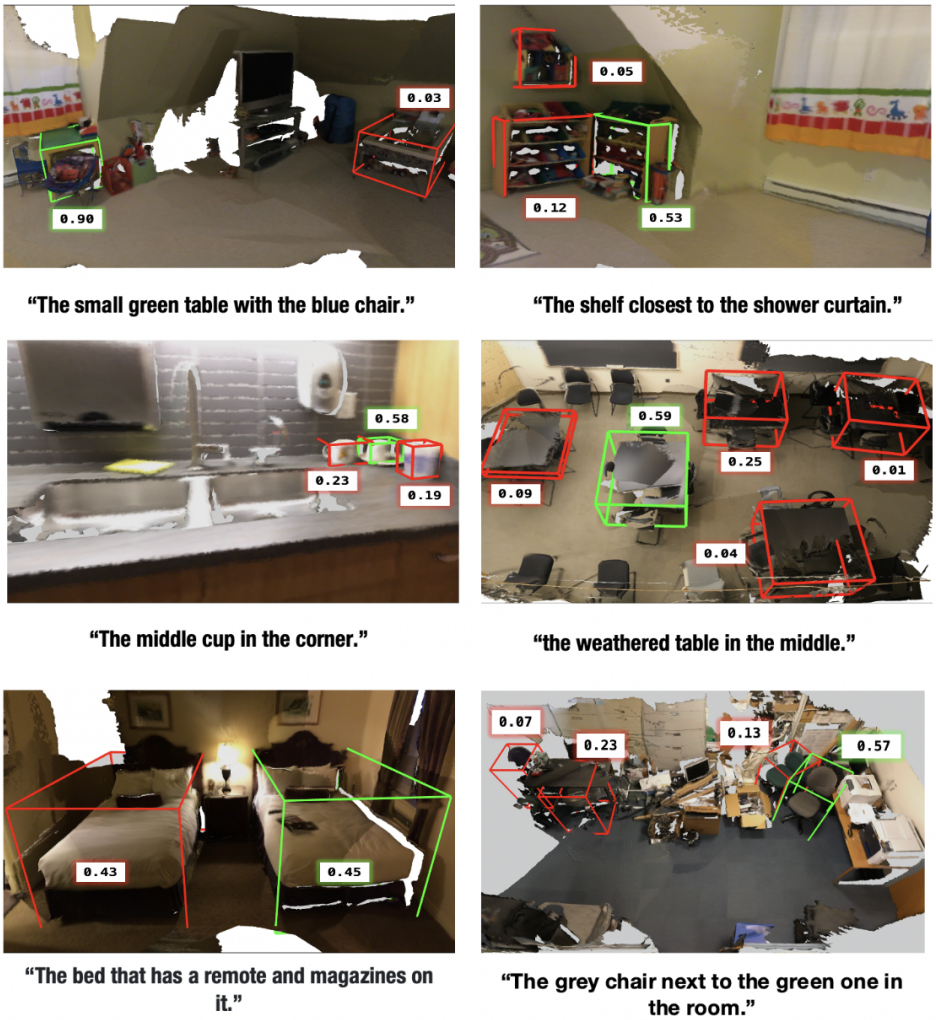

Figure 3. Successful qualitative examples from our best performing method. Green means the prediction and ground truth. Red represents same-class distractor objects. Our method outperforms other methods in the hard contexts.

ble 3. We can observe that when adding any extra layer, the performance drops. We argue that the reason behind this is the small size of the dataset currently available.

4.4. Qualitative Results

We have provided some qualitative examples in Figure. 3 from our best performing model. We observe that our model can successfully and accurately describe the spatial relations among different objects from the scene. It can also distinguish between the objects. For example, "The middle cup in the corner". It could distinguish the referring cup from other similar cups and point out its position. We also demonstrate the object grounding examples in Figure 5. We show an example of our method's grounding ability. The query text is "The lamp next to the window", and our model can locate it with high confidence while maintaining attention on relevant words in the utterance. In Figure. 4, we show some challenging examples predicted successfully by our methods, compared to Referit3DNet predictions which are less accurate. For more qualitative results, please refer to the supplementary.

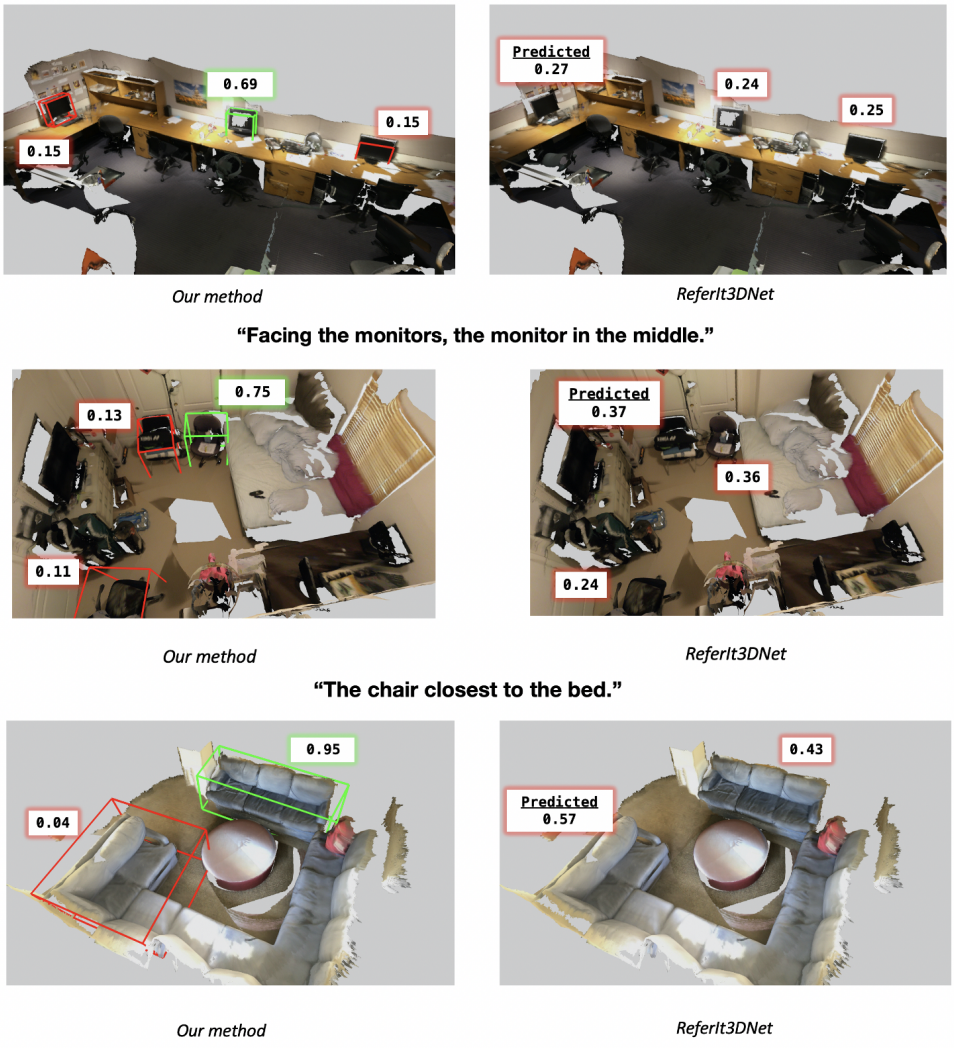

Figure 4. Comparing some examples between Referit3DNet (on the right) and our proposed method (on the left). These examples Referit3DNet failed to identify the target object while our proposed method successfully identifies the target object.

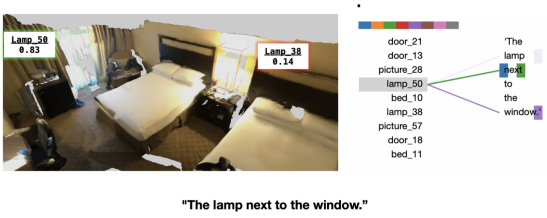

Figure 5. Example illustration: we show how target object visual feature (Lamp 50 in this example) attends on the utterance word tokens. Our model can successfully distinguish which lamp that the current utterance is referring to.

5. Conclusion

Identifying fine-grained 3D objects in real-world scenes based on their textual query descriptions is a challenging task that requires building both context-aware visual representations that are robust to distractions and fuse them with the textual representation of a query. In this paper, we presented 3DRefTransformer model, which employs Transformer blocks to better encode context information into visual representations for better discrimination between a target 3D object and possible distractors. Our empirical results demonstrate that the proposed architecture shows significant improvement over the baselines in the overall performance. However, we are still far below the level of human performance, and our qualitative analysis shows that the model can be confused in relatively simple cases. We also did several ablation experiments and demonstrated that self-attention alone is insufficient to obtain competitive performance on this problem.

References

[1] Panos Achlioptas, Ahmed Abdelreheem, Fei Xia, Mohamed Elhoseiny, and Leonidas Guibas. Referit3d: Neural listeners for fine-grained 3d object identification in real-world scenes. *16th European Conference on Computer Vision (ECCV)*, 2020.

[2] Harsh Agrawal, Karan Desai, Yufei Wang, Xinlei Chen, Rishabh Jain, Mark Johnson, Dhruv Batra, Devi Parikh, Stefan Lee, and Peter Anderson. nocaps: novel object captioning at scale. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 8948–8957, 2019.

[3] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2425–2433, 2015.

[4] Thomas Bachlechner, Huanru Henry Majumder, Bodhisattwa Prasad Mao, Garrison W. Cottrell, and Julian McAuley. Rezero is all you need: Fast convergence at large depth. In *arXiv*, 2020.

[5] Dave Zhenyu Chen, Angel X Chang, and Matthias Nießner. Scanrefer: 3d object localization in rgb-d scans using natural language. *arXiv preprint arXiv:1912.08830*, 2019.

[6] Dave Zhenyu Chen, Ali Gholami, Matthias Nießner, and Angel X. Chang. Scan2cap: Context-aware dense captioning in rgb-d scans, 2020.

[7] 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), IEEE*, 2017.

[8] Chaorui Deng, Qi Wu, Qingyao Wu, Fuyuan Hu, Fan Lyu, and Mingkui Tan. Visual grounding via accumulated attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7746–7755, 2018.

[9] Alaaeldin El-Nouby, Shikhar Sharma, Hannes Schulz, Devon Hjelm, Layla El Asri, Samira Ebrahimi Kahou, Yoshua Bengio, and Graham W. Taylor. Tell, draw, and repeat: Generating and modifying images based on continual linguistic instruction, 2019.

[10] Daniel Fried, Ronghang Hu, Volkan Cirik, Anna Rohrbach, Jacob Andreas, Louis-Philippe Morency, Taylor Berg-Kirkpatrick, Kate Saenko, Dan Klein, and Trevor Darrell. Speaker-follower models for vision-and-language navigation. In *Advances in Neural Information Processing Systems*, pages 3314–3325, 2018.

[11] Meng-Hao Guo, Jun-Xiong Cai, Zheng-Ning Liu, Tai-Jiang Mu, Ralph R. Martin, and Shi-Min Hu. Pct: Point cloud transformer, 2021.

[12] Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. Deberta: Decoding-enhanced bert with disentangled attention, 2020.

[13] Tobias Hinz, Stefan Heinrich, and Stefan Wermter. Semantic object accuracy for generative text-to-image synthesis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, page 1–1, 2020.

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

[15] Ronghang Hu, Amanpreet Singh, Trevor Darrell, and Marcus Rohrbach. Iterative answer prediction with pointer-augmented multimodal transformers for textvqa. *CoRR*, abs/1911.06258, 2019.

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

[17] R. Hu, H. Xu, M. Rohrbach, J. Feng, K. Saenko, and T. Darrell. Natural language object retrieval. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4555–4564, 2016.

[18] Pin-Hao Huang, Han-Hung Lee, Hwann-Tzong Chen, and Tyng-Luh Liu. Text-guided graph neural networks for referring 3d instance segmentation. 2021.

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

[20] Jacob Krantz, Erik Wijmans, Arjun Majumdar, Dhruv Batra, and Stefan Lee. Beyond the nav-graph: Vision-and-language navigation in continuous environments. *arXiv preprint arXiv:2004.02857*, 2020.

[21] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *Advances in Neural Information Processing Systems*, pages 13–23, 2019.

[22] Chih-Yao Ma, Zuxuan Wu, Ghassan AlRegib, Caiming Xiong, and Zsolt Kira. The regretful agent: Heuristic-aided navigation through progress estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6732–6740, 2019.

[23] Mateusz Malinowski and Mario Fritz. A multi-world approach to question answering about real-world scenes based on uncertain input. In *Advances in neural information processing systems*, pages 1682–1690, 2014.

[24] 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.

[25] Junhua Mao, Wei Xu, Yi Yang, Jiang Wang, Zhiheng Huang, and Alan Yuille. Deep captioning with multimodal recurrent neural networks (m-rnn). *arXiv preprint arXiv:1412.6632*, 2014.

[26] 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.

[27] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation, 2017.

[28] 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.

[29] Di Qi, Lin Su, Jia Song, Edward Cui, Taroon Bharti, and Arun Sacheti. Imagebert: Cross-modal pre-training with large-scale weak-supervised image-text data, 2020.

[30] Mengye Ren, Ryan Kiros, and Richard Zemel. Exploring models and data for image question answering. In *Advances in neural information processing systems*, pages 2953–2961, 2015.

[31] Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, et al. Habitat: A platform for embodied ai research. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9339–9347, 2019.

[32] Hao Tan and Mohit Bansal. Lxmert: Learning cross-modality encoder representations from transformers. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5103–5114, 2019.

[33] Ming Tao, Hao Tang, Songsong Wu, Nicu Sebe, Fei Wu, and Xiao-Yuan Jing. Df-gan: Deep fusion generative adversarial networks for text-to-image synthesis, 2020.

[34] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[35] 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.

[36] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E. Sarma, Michael M. Bronstein, and Justin M. Solomon. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics (TOG)*, 2019.

[37] Fei Xia, Amir R Zamir, Zhiyang He, Alexander Sax, Jitendra Malik, and Silvio Savarese. Gibson env: Real-world perception for embodied agents. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9068–9079, 2018.

[38] 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.