LanguageRefer: Spatial-Language Model for 3D Visual Grounding

Anonymous Author(s)
Affiliation
Address
email

Abstract: To realize robots that can understand human instructions and perform meaningful tasks in the near future, it is important to develop learned models that can understand referential language to identify common objects in real-world 3D scenes. In this paper, we develop a spatial-language model for a 3D visual grounding problem. Specifically, given a reconstructed 3D scene in the form of a point cloud with 3D bounding boxes of potential object candidates, and a language utterance referring to a target object in the scene, our model identifies the target object from a set of potential candidates. Our spatial-language model uses a transformer-based architecture that combines spatial embedding from bounding-box with a finetuned language embedding from DistilBert [1] and reasons among the objects in the 3D scene to find the target object. We show that our model performs competitively on visio-linguistic datasets proposed by ReferIt3D [2]. We provide additional analysis of performance in spatial reasoning tasks decoupled from perception noise, the effect of view-dependent utterances in terms of accuracy, and view-point annotations for potential robotics applications.

Keywords: Referring task, Language model, 3D visual grounding, 3D Navigation

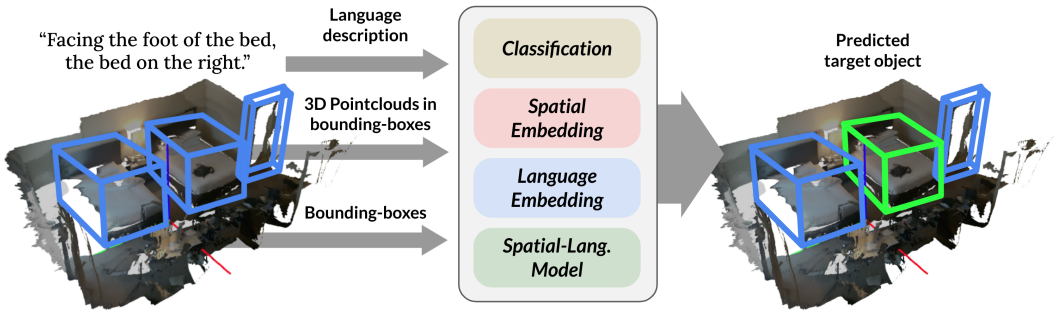


Figure 1: **Simplified overview of LanguageRefer. LanguageRefer** is a model takes 3D pointcloud of a scene, bounding-boxes of objects in the scene, and grounding language description of a single object in the scene and refer the described object. It has four modules: classification, spatial embedding, language embedding, and spatial-language model.

1 Introduction

For a robot to communicate seamlessly with humans to perform meaningful tasks in indoor environments, it should understand the natural language utterances and ground it to the elements in the real world. Recently, several advancements have been made to combine language with visual elements resulting methods for tasks such as visual question and answering (VQA) involving spatio-temporal reasoning tasks [3, 4, 5], embodied QA [6], and pre-training for visual recognition tasks with language descriptions [7]. Embodied agents are able to follow visually grounded language instructions

Submitted to the 5th Conference on Robot Learning (CoRL 2021). Do not distribute.

to perform embodied tasks [8, 9]. To enable a real robot to perform these intelligent tasks, it is critical to use 3D representation from raw sensor data. ReferIt3D [2] proposes a benchmarking dataset of language utterances referring to objects in 3D scenes from ScanNet [10] dataset. Our long-term goal is to enable robot to visual navigate in an indoor environment based on referential instructions. In this paper we take a step towards it by exploiting the ReferIt3D dataset to learn a model that can identify the 3D object referred in a language utterance.

Referential language to identify an object in real-world 3D scenes is a challenging problem. Consider an example utterance - *"Facing the foot on the bed, the bed on the right"* along with a 3D scene as shown in Figure 1. It is easier for a human to follow the clues in the language, infer the point of view of the speaker, locate all the elements referred to, and spatially reason to locate the *bed* in the scene despite of two instances of *beds*. However, view-point prediction, object identification, and spatial reasoning are still open-ended research problems in robotics and vision. The 3D reference task as proposed by ReferIt3D is challenging because of four reasons: a) reconstructed 3D scenes of the real world in ScanNet dataset are noisy and lacks fine details in comparison to 2D images or rendered 3D scenes, b) fine-grained class labels and expressions in natural language utterances are not exactly matched and diverse, c) view-dependent utterances often require guessing the original view-points which hinders model to properly learn spatial concepts, and d) combined complexity of multiple challenges makes it hard to analyze what model learns or understands.

We are inspired by the success of the methods [11, 12] on CLEVR and CLEVRER domains in spatial reasoning task and hypothesize that in the 3D reference task, if the spatial-reasoning task is decoupled from the perceptual task of identifying the objects in the 3D scene, we can achieve better performance and clearly track the contribution of perception noise in performance. More specifically, instead of developing integrated multi-modal perception system, we assume a pre-trained instance classification model or ground-truth classes for the objects that can inform the spatial reasoning task. We focus on how the language model with spatial information can handle the referring task. The ReferIt3D dataset includes two sub-datasets containing natural and synthetic language utterances, namely Nr3D and Sr3D respectively. In our experiments, our model is able to achieved comparable scores on Nr3D and Sr3D with predicted instance class labels. We observe high accuracy with ground-truth labels in Sr3D which indicates that the model has better understanding from template-based language data. Our pipeline is modular with multiple models (perceptual model, spatial embedding model, pre-trained language embedding model, and spatial-language model). This modular approach gives flexibility and can adapt to environments with different appearance, and object entities.

Another aspect of the 3D reference task is the view-point prediction. The ReferIt3D dataset contains utterances that can be grouped into view-independent (VI) and view-dependent (VD). An example of VI utterance is *"The lamp closer to the white armchair"*, and VD utterance is *"The lamp on the right in between the beds"*. The VD utterance requires view-point prediction. This is crucial for an application in robotics where the agent has to infer view-point with respect to which the speaker is referring to. In the ReferIt3D dataset, some of VD utterances do not have information to guess the valid orientation of the agent (who utters the language description). It prevents a model in understanding the significance of spatial relationships such as 'left of.' This is due to annotation process of ReferIt3D datasets where both a speaker and a listener can freely rotate the scene to infer this ill-defined orientation in the utterance. Human annotators who are aware of spatial concepts tend to validate arbitrary orientations. However, a data-driven model will suffer from degraded learning when it is not capable of doing verification just like annotators. Viewpoint-free annotations may not well fit to most robotics applications; but still it is important to predict or verify whether the given statement and the viewpoint match. For better training and investigation of the agent in view-dependencies, we provide an extra collection of annotation of orientations for VD utterances and compare the models with and without the view-point correction from annotations.

Overall, the contributions of the this paper include: 1) a novel transformer based spatial-language model in a modular pipeline for better understanding of spatial relationships in a 3D visual grounding task. We show that our model achieves comparable performance with the state-of-the-art methods on Nr3D and Sr3D datasets, 2) we provide analysis of the ReferIt3D dataset with viewpoint orientation annotations to remove potential artifacts from implicit orientations, 3) ablation and additional experiments with ground-truth classes which decouple the impact of the perception noise in spatial reasoning task.

2 Related Work

2.1 Vision-and-Language Navigation (VLN) and Robot Navigation

Vision-and-language navigation (VLN) has been extensively studied and achieved remarkable progress ([13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23]). Given a language instruction in the simulation environment, the goal is to make an agent reach the desired node on the pre-defined traversal graph from images. A few extensions were proposed in the literature. ALFRED [8] proposes an extended VLN task that grounds a sequence of sub-tasks to achieve a higher-level task in AI2Thor environment [9]. In ALFRED, navigation sequences have (implicit) goals which are often close to objects of interest in next sub-tasks. For instance, the agent is asked to move in front of the sink because it is going to clean the cup in the sink. With high-level semantic tasks, object-centric spatial understanding is more of importance.

In another direction, recent approaches have relaxed the constraint of discrete traversal in VLN into continuous space ([24, 25, 26, 27, 28]). Here, the agent needs to deal with higher task complexity involving time and space. Thus expanding the space representation to 3D can be an effective solution. In context of visual language navigation, we consider the 3D visual grounding task as a proxy of 3D indoor navigation with fully observability assumption and goal-oriented language descriptions. Especially, our approach focuses on understanding spatial relationships among objects which plays key role in VLN and robot navigation.

2.2 2D and 3D Visual Grounding

2D visual grounding is a task to localize an object or region in the image given a language description about the object or the region ([29, 30, 31]). Most methods are two-stage approaches where they first generate proposals and choose the grounded proposal by comparing the proposals to the language description ([32, 33, 34, 35, 36]).

3D visual grounding is to localize a 3D bounding box from the pointcloud of a scene given a language description. Recently, ReferIt3D [2] and ScanRefer [37] were proposed datasets for 3D visual grounding task with language annotation on ScanNet [10] dataset. Most 3D grounding approaches are ([2, 37, 38, 39, 40]) following two-stage scheme as many approaches do in 2D visual grounding task; first, propose multiple bounding boxes or use the ground-truth bounding boxes and second, features from proposals are combined or compared with features from language description. InstanceRefer [38] extracts a few attribute features both from pointclouds and utterance and compare them to select the best matching object. FFL-3DOG [39] does the feature matching from language and the pointclouds in guidance with a language and a visual scene graph. These methods rely on specific designs such as bird-eye-view mapping with different types of operations, intensive language preprocessing for graph generation. In contrast, our approach leverages the language embedding space[1]; we have marginal cost of object and language embedding fusion and minimal manual design. Decoupled perception module makes our approach modular and transferable to different data.

3 Problem Statement & Methodology

Given a 3D scene S with a list of objects $(O_1, \cdots, O_M)$ and a language utterance U, the problem is to predict the target object $O_{\mathcal{T}}, \mathcal{T} \in \mathcal{I}_M = \{1, \cdots, M\}$ referred to in the language. A single object O_i comprises a bounding-box $B_i \in \mathcal{B} = \mathbb{R}^6$ and corresponding pointcloud $P_i \in \mathcal{P} = \mathbb{R}^{N_i \times 6}$ (xyz positions and RGB values) within the bounding-box with N_i number of points.

We propose a language model based approach, **LanguageRefer**, to solve a 3D visual grounding task. Our model focuses on understanding spatial relationship between objects from language description and 3D bounding-box information due to 1) high dependency on spatial relationship description in language and 2) holistic nature of spatial relationship understanding different from unary attributes such as color and shape. As shown in the Figure. 2, we use two-stage approach: 1) determine the

[1] SAT [40] has been released at a similar time of ours. Their approach uses Transformer models but it needs a full training of two Bert models for fusing multiple embedding space and modifying word embedding space from the third layer.

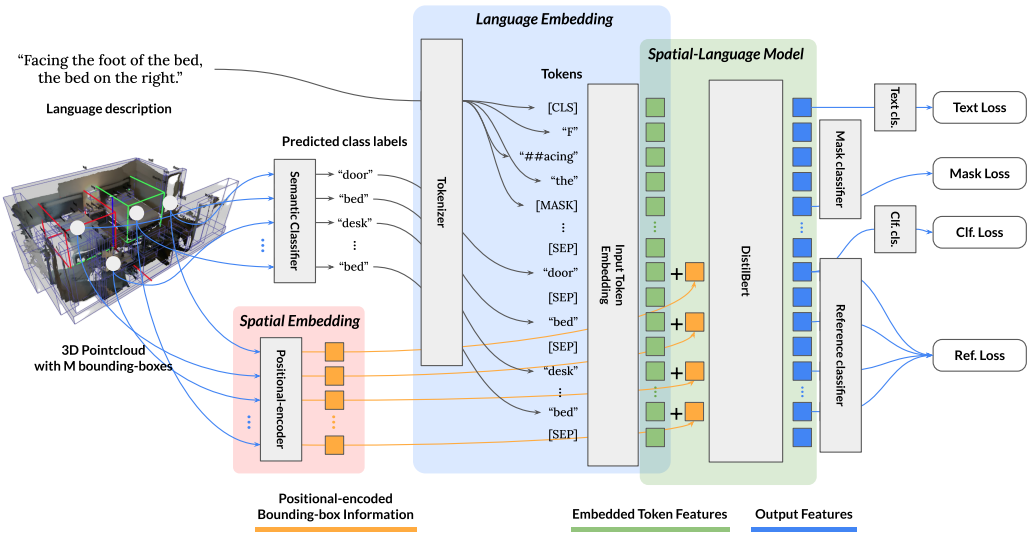


Figure 2: **Detailed overview of LanguageRefer.** A semantic classifier predicts class labels from 3D pointcloud in each bounding box (using color and xyz positions). The language description or utterance (e.g. "*Facing the foot of the bed, the bed on the right*") is transformed into a sequence of tokens along with a sequence of predicted class labels. The input token embedding in DistilBert [1] converts the tokens into embedded features vectors (green squares). Bounding-box position and size information is positional-encoded to form encoded vectors using technique from [41] (orange squares) and are added to the corresponding embedded feature vectors (green squares). After the addition, the modified features are processed by our reference model based on DistilBert and fed to multiple tasks. The main task is a reference task which is to choose the referred object from the object features. The text classification task takes the output feature from the [CLS] token and predicts the target object's class label (e.g. *bed*). The instance classification task is a binary classification of determining whether the given object feature belongs to the target class. Lastly, the masking task is to recover the original token from a randomly replaced token in the utterance which is commonly used in language modeling.

class labels of objects in the scene, 2) determine the referred object based on inputs and predictions with spatial-language embedding. In the following subsections we will describe these steps in detail.

Semantic Classification Model and Tokenization: For semantic classification of the pointcloud in a bounding-box, we employed PointNet++ [42] and it achieved 69% accuracy on average in the test dataset. In training and inference, we use a sampled pointcloud $P'_i \in \mathbb{R}^{1024\times 6}$ and PointNet++ predicts the semantic label $\hat{l} \in \mathcal{L}$ where $\mathcal{L}$ is a set of class labels in text. Each scene has pairs of predicted class labels and a bounding-box values $((\hat{l}_i, B_i) : \hat{l}_i \in \mathcal{L}, B_i \in \mathcal{B})$ from objects. Predicted labels are concatenated to the utterance U with a separator [SEP], then split into a list of indices of tokens by a tokenizer: U becomes $(u_1, \cdots, u_t)$ and each predicted class label $\hat{l}_i$ becomes $(o^i_1, \cdots, o^i_{n_1})$ where each token index is in $\mathcal{I}_D$ and D is the size of the dictionary.

Language Model and Token Embedding Generation: Our model uses pre-trained language model, DistilBert, for the reference task. Transformers consider relationship among all pairs of elements through attention and it can be effectively leveraged in understanding spatial relationships between objects as discrete entities. In our formulation, predicted class labels are considered as sentences so they are concatenated to the utterance with separation by [SEP]. Therefore, the final sequence of token indices would be V = ([CLS], $u_1, \cdots, u_t$, [SEP], $o^1_1, \cdots, o^1_{n_1}$, [SEP], $\cdots$, [SEP], $o^M_1, \cdots, o^M_{n_M}$, [SEP]). The token index sequence complies with the specification of DistilBert but with the violation of the number of sentences. Then we transform V into token embedding sequence $W = (w_1, \cdots, w_T), w_i \in \mathbb{R}^{768}$ by the word embedding in DistilBert. For concise notation, we define the indices of utterance tokens in V or W as a mask $M_U = (2, \cdots, t+1)$ and the indices of first tokens from objects $(o^1_1, \cdots, o^M_1)$ as a mask M_O. We also define mask operator $[\cdot]$ to manipulate specific elements in the sequence. For instance, $W[M_U] = 0$ will empty all the utterance embeddings in W.

Spatial-Language Model and Spatial Embedding: In order to combine spatial information from raw bounding-box value to the token embedding W, we employ positional encoding PE$(\cdot)$ to transform bounding-box vector $B_i \in \mathcal{B}$ to $b_i = \text{PE}(B_i) \in \mathbb{R}^{768}$ and they are added to $W[M_O]$. Then the token embedding W combined with spatial information is finally transformed into the output embedding $X = (x_1, \cdots, x_T), x_i \in \mathbb{R}^{768}$ by the reference model finetuned from the pre-trained DistilBert. The final reference task is done by the reference classifier from $X[M_O]$ as explained in the following subsection.

Loss Functions: In training, we use four tasks for training and corresponding loss values. First, we use the reference loss, $\mathcal{L}_{ref}$, following the original proposal in [2]. We ask the model to choose one object as the target instance among M candidates. We collect scalar values from $X[M_O]$ by a linear layer and take the argmax on those values to choose the target instance.

Second, we add a binary target classification loss $\mathcal{L}_{tar}$ on $X[M_O]$ to determine whether given an object belongs to the target class.

Third task is a target classification from utterance, $\mathcal{L}_{text}$. Given tokens from utterance, $X[M_U]$, it should be able to predict the target class label $l \in \mathcal{L}$.

Last, we employ mask loss from language model pre-training, $\mathcal{L}_{mask}$. We randomly replace the tokens from nouns in the utterance with probability of 15 %. The noun token is replaced by [MASK] with chance of 80 %, by a random token with chance of 10 %, or remain the same with probability of 10 %. Then the model is asked to recover the original token index. We expect the model to fill in the replaced tokens in the utterance by understanding the relationship between objects. We use cross entropy loss for all tasks. We compute the final loss as follows:

$$\mathcal{L} = \mathcal{L}_{ref} + 0.5\mathcal{L}_{clf} + 0.5\mathcal{L}_{text} + 0.5\mathcal{L}_{mask}. \tag{1}$$

At inference, we followed the approach of InstanceRefer [38] to filter out objects that do not belong to the predicted target class. Further, we use top-k predictions for instance class labels in the filtering approach to prevent the true target instance from getting rid of. An accuracy of the target classification model was 94%. For masking loss computation, we extracted nouns in the utterance. We used flair [43] for part-of-speech (POS) tagging.

4 Experiments

4.1 Dataset

We evaluated our model on the referring task from ReferIt3D [2]. Our model was evaluated with two datasets Nr3D (Natural Reference in 3D) and Sr3D (Spatial Reference in 3D, contains spatial descriptions only). Both datasets augment ScanNet [10], a reconstructed 3D indoor scene dataset, with language descriptions. Nr3D has 41,503 natural language utterances and Sr3D contains 83,572 template-based utterances on 707 scenes following the official splits of ScanNet [10]. The datasets have 76 target classes and they are design to have multiple same-class distractors in the scene.

4.2 Experiment Settings

ReferIt3D [2] provides the ground-truth bounding-boxes of objects in the scene, pointcloud of the scene, utterances and corresponding target objects. We measure the accuracy of the model by comparing the selected object from M candidates to the ground-truth target object.

We trained models with learning rate of 0.0001 with AdamW, warm-up and linear scheduling. We initialize our model with a pre-trained model of distilbert-base-cased [1] from huggingface implementation [44].

We compared the performance of our model with state-of-the-art methods on ReferIt3D ([2, 40, 37, 38, 39]). Since SAT [40] uses extra 2D image dataset in their training, we separate non-SAT, their baseline model that is not trained with the extra dataset, from SAT.

4.3 Evaluation on ReferIt3D [2]

Table 1 shows the accuracy on Nr3D and Sr3D. Our model outperformed other models (without extra training dataset) on both Nr3D and Sr3D. Our model achieved 43.9% in Nr3D with $+8.3\%$

Dataset	Method	Overall	Easy	Hard	View-dep.	View-indep.
Nr3D	ReferIt3D [2]	35.6 %	43.6 %	27.9 %	32.5 %	37.1 %
	ScanRefer [37]	34.2 %	41.0 %	23.5 %	29.9 %	35.4 %
	InstanceRefer [38]	38.8 %	46.0 %	31.8 %	34.5 %	41.9 %
	FFL-3DOG [39]	41.7 %	48.2 %	35.0 %	37.1 %	44.7 %
	non-SAT [40]	37.7 %	44.5 %	31.2 %	34.1 %	39.5 %
	Ours	**43.9** %	**51.0** %	**36.6** %	**41.7** %	**45.0** %
Nr3D w/ 2D images	SAT [40]	49.2 %	56.3 %	42.4 %	46.9 %	50.4 %
Sr3D	ReferIt3D [2]	40.8 %	44.7 %	31.5 %	39.2 %	40.8 %
	InstanceRefer [38]	48.0 %	51.1 %	40.5 %	45.4 %	48.1 %
	non-SAT [40]	47.4 %	N/A	N/A	N/A	N/A
	Ours	**56.0** %	**58.9** %	**49.3** %	**49.2** %	**56.3** %
Sr3D w/ 2D images	SAT [40]	57.9 %	61.2 %	50.0 %	49.2 %	58.3 %

Table 1: **Accuracy on ReferIt3D [2].** Our model outperformed state-of-the-art models in both Nr3D and Sr3D except for models with additional training data (SAT with 2D images). The average performance gap between ours and other models on Sr3D (10.6%) is larger than that on Nr3D (6.3%) since our model uses only spatial reasoning for the reference task.

improvement over the baseline method of ReferIt3D [2] and 2.2% increase from the best accuracy from other models in Nr3D. In Sr3D, our model achieved 56.0% which is 15.2% and 8.0% increase from the baseline and the best accuracy in the Sr3D section, respectively. It is also comparable to the accuracy of SAT (with 1.9% difference) which was trained with additional 2D image training data. It proves that our model is able to accurately reason spatial relationship from the spatial-language embedding and outperforms in both datasets despite the loss of information in appearance. In addition, less diverse language expression and utterances only about spatial reasoning can explain outperform in Sr3D.

4.4 Evaluation with Ground-Truth Class Labels

We further investigate the model's ability in spatial reasoning by removing noise in class labels. It was done by replacing predicted class labels by ground-truth class labels and it was possible due to the explicit usage of class labels in our model.

Table 2 shows the result with and without ground-truth class labels. The first and the second column demonstrates the type of dataset in training and evaluation, respectively. For instance, the second row shows the evaluation result of the model trained in Nr3D dataset with ground-truth class labels and evaluated in Nr3D dataset but with predicted class labels. When we train and evaluate our model with ground-truth labels in Nr3D and Sr3D (row 4 and 8), we get accuracy of 54.3% and 91.1%, respectively. In comparison to the accuracy of models trained and evaluated with noisy labels, improvement is 10.4% and 35.1% for each dataset. When we use the models trained with noisy labels and evaluate them with ground-truth labels, they get 9.7% and 24.2% performance increase. We believe multiple reasons may cause the difference in performance gaps on Nr3D and Sr3D; diversity in natural language dataset, higher portion of view-dependent utterances or view-dependent utterances without hint of the orientation, description other than spatial relationships such as color or appearance.

In addition, we examine the accuracy in switched dataset; train a model in Nr3D and evaluate in Sr3D, and vice versa. Last two rows in Table 2 show the results of two switched evaluation results: 40.0% and 37.6% on Sr3D and Nr3D. Accuracy of 37.6% on Nr3D from the model trained on Sr3D is comparable to accuracy of non-SAT [40] and InstanceRefer [38]. It shows generalizability of spatial reasoning in our model and potential in transfer to different perception modules and datasets.

4.5 Ablation on Loss terms

We have trained models with different combinations of loss terms and Table 3 shows the results. We found that only the binary classification loss shows its effect clearly ($+3.5\%$ from Ref. to Ref.-Clf., $+3.9\%$ from Ref.-Mask to Ref.-Mask-Clf., $+1.0\%$ from Ref.-Text to Ref.-Text-Clf.). The mask

Dataset		Overall	Easy	Hard	View-dep.	View-indep.
Training	Evaluation					
Nr3D-pred	Nr3D-pred	43.9 %	51.0 %	36.6 %	41.7 %	45.0 %
Nr3D-gt	Nr3D-pred	41.4 %	48.0 %	34.6 %	38.1 %	43.0 %
Nr3D-pred	Nr3D-gt	53.6 %	64.4 %	42.5 %	50.7 %	55.0 %
Nr3D-gt	Nr3D-gt	54.3 %	65.5 %	42.8 %	49.1 %	56.8 %
Sr3D-pred	Sr3D-pred	56.0 %	58.9 %	49.3 %	49.2 %	56.3 %
Sr3D-gt	Sr3D-pred	52.1 %	53.8 %	48.2 %	43.9 %	52.5 %
Sr3D-pred	Sr3D-gt	80.2 %	83.2 %	73.1 %	62.5 %	81.0 %
Sr3D-gt	Sr3D-gt	91.1 %	93.1 %	86.2 %	67.0 %	92.1 %
Nr3D-pred	Sr3D-pred	40.0 %	43.6 %	31.5 %	41.2 %	39.9 %
Sr3D-pred	Nr3D-pred	37.6 %	45.3 %	29.6 %	34.5 %	39.0 %

Table 2: **Ablation with ground-truth class labels.** First and second column shows types of data used in training and evaluation. High accuracy (91.1% at row 8) of the model both trained and evaluated with ground-truth class labels on Sr3D shows the model's ability in spatial reasoning besides the perception noise. Accuracy gaps of ground-truth and predicted class labels on Nr3D and Sr3D (9.7%, 24.2%) indirectly tell us complexity of the language and information loss by classification. Transferring the model trained with Sr3D to Nr3D evaluation reported a comparable number (37.6% at row 10) to those from other methods. Our model can be easily plugged with different classification models or datasets.

loss was not effective and the text classification loss degraded the accuracy. One hypothesis on this is that text classification loss is not helping the model to transform the initial language embedding to spatial-language embedding. The model may not require strong language constraint because it already has a good initial language embedding and the text classification loss is not useful in shaping spatial-language embedding space because it is independent to spatial configuration of the scene. Due to the performance drop, we choose our model without the text classification loss.

Ref.	Clf.	Mask	Text	Overall	Easy	Hard	View-dep.	View-indep.
✓	-	-	-	40.6 %	48.2 %	32.8 %	38.5 %	41.6 %
✓	✓	-	-	44.1 %	51.3 %	36.7 %	41.0 %	45.6 %
✓	-	✓	-	40.0 %	47.2 %	32.5 %	36.8 %	41.5 %
✓	-	-	✓	40.0 %	48.7 %	30.8 %	37.8 %	40.9 %
✓	✓	✓	-	43.9 %	51.0 %	36.6 %	41.7 %	45.0 %
✓	✓	-	✓	41.0 %	49.7 %	31.9 %	39.3 %	41.8 %
✓	✓	✓	✓	40.0 %	48.2 %	31.5 %	38.9 %	40.6 %

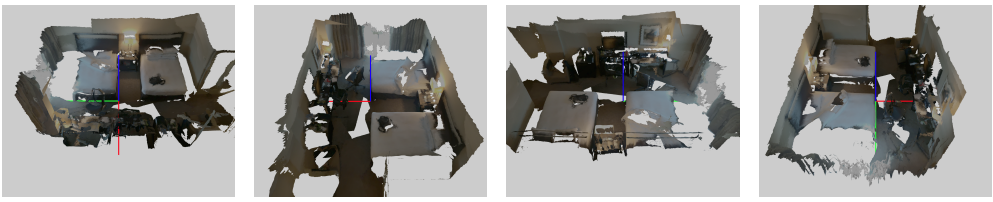
Table 3: **Ablation of loss terms on Nr3D.** The classification loss was effective, the mask loss did not affect the accuracy much, and the text loss degraded the accuracy. We choose models without text losses.

Correction		Overall	Easy	Hard	View-dep.	View-indep.
Training	Evaluation					
✓	-	43.5 %	50.7 %	36.0 %	37.0 %	46.6 %
✓	✓	49.0 %	56.0 %	41.8 %	54.4 %	46.4 %
-	✓	43.1 %	50.3 %	35.6 %	40.4 %	44.4 %
-	-	43.9 %	51.0 %	36.6 %	41.7 %	45.0 %

Table 4: **Comparison of accuracy with and without corrected orientations.**

4.6 View-point Annotation

View-dependent utterances without information about original view-point make the reference task in ReferIt3D [2] more challenging. For instance, utterances such as "*The door is wood with the*



(a) An example of the standard orientation 1. (b) An example of the standard orientation 2. (c) An example of the standard orientation 3. (d) An example of the standard orientation 4.

Figure 3: **Examples of standard orientations for view-point annotation (a-d).** We assume that the robot is always inside of the room except for the cases specified by utterances.

handle on the left side." assume specific orientations of the agent and it is impossible to recover the true orientation without knowing the referred object, not like view-dependent utterances with explicit view-point information such as "*Facing the foot of the bed.*" However, the original dataset of ReferIt3D [2] does not distinguish the utterances without orientation information from those with the orientation information. Therefore, we split the view-dependent (VD) utterance category into two subcategories,VD-explicit and VD-implicit, where VD-explicit has explicit view-point information in the utterance. Then we collected orientations that make the utterances valid from human annotators. We set four standard orientation assuming the agent is in the room (around the center of the scene) and ask the annotators to select all of orientations that can be considered valid from utterance. Figure 3 shows examples of four orientations. With the assumption of the agent being inside of the room, we found that four orientations are good enough to recover the original view-points of the speakers. In total, 12,680 view-dependent utterances of the Nr3D dataset were annotated. From those, 5,942 utterances are classified as VD-expclit. For train and test split, 10,206 and 2,474 utterances were annotated.

From the orientation for view-dependent utterances, we revised the dataset with the corrected orientation. For view-dependent utterances, we rotate the scene with respect to the annotation in order to make all the scenes remain valid at the canonical orientation. For view-independent utterances, we randomly rotate the scenes since they are valid in any directions. Table 4 shows the accuracy values from models trained with and without corrected orientations. At inference, we evaluated each model with and without corrected orientations as well. The second row shows the accuracy of the model which is trained and evaluated with corrected orientations. It overall accuracy was improved by $+5.1\%$ from the best model without the correction. Note that the improvement on view independent utterances was marginal ($+1.4\%$), but the improvement on view dependent utterances was significant ($+12.7\%$). Also the model trained with corrected orientations achieved comparable accuracy to the best model even without corrections at inference ($+0.4\%$, -0.4%). It implies the correction helped the model to correctly interpret the view-dependent scene when the orientation is consistently aligned without introducing unwanted bias.

In the supplementary video material, we illustrate how a robot can use our LanguageRefer model to predict a referred target object from a language utterance and navigate towards it.

5 Conclusion

We have proposed LanguageRefer, a spatial-language model for 3D visual grounding in a referring task. LanguageRefer combines language embeddings from the utterance and class labels with positional-encoded spatial information for efficient learning the spatial-language embedding space without different modules for individual modality. Experimental results show that LanguageRefer outperformed state-of-the-art models on ReferIt3D without additional training data. Analysis and ablations demonstrate effects of 1) noisy class labels, 2) arbitrary viewpoints in view-dependent utterances, 3) and transfer of model to different datasets for future robotics applications.

References

[1] V. Sanh, L. Debut, J. Chaumond, and T. Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

[2] P. Achlioptas, A. Abdelreheem, F. Xia, M. Elhoseiny, and L. Guibas. Referit3d: Neural listeners for fine-grained 3d object identification in real-world scenes. *16th European Conference on Computer Vision (ECCV)*, 2020.

[3] J. Johnson, B. Hariharan, L. van der Maaten, L. Fei-Fei, C. L. Zitnick, and R. Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *CVPR*, 2017.

[4] K. Yi, C. Gan, Y. Li, P. Kohli, J. Wu, A. Torralba, and J. B. Tenenbaum. CLEVRER: collision events for video representation and reasoning. In *ICLR*, 2020.

[5] R. Girdhar and D. Ramanan. CATER: A diagnostic dataset for Compositional Actions and TEmporal Reasoning. In *ICLR*, 2020.

[6] A. Das, S. Datta, G. Gkioxari, S. Lee, D. Parikh, and D. Batra. Embodied question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–10, 2018.

[7] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*, 2021.

[8] M. Shridhar, J. Thomason, D. Gordon, Y. Bisk, W. Han, R. Mottaghi, L. Zettlemoyer, and D. Fox. Alfred: A benchmark for interpreting grounded instructions for everyday tasks. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10737–10746, 2020.

[9] E. Kolve, R. Mottaghi, W. Han, E. VanderBilt, L. Weihs, A. Herrasti, D. Gordon, Y. Zhu, A. Gupta, and A. Farhadi. AI2-THOR: An Interactive 3D Environment for Visual AI. *arXiv*, 2017.

[10] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5828–5839, 2017.

[11] D. Ding, F. Hill, A. Santoro, and M. Botvinick. Object-based attention for spatio-temporal reasoning: Outperforming neuro-symbolic models with flexible distributed architectures. *arXiv preprint arXiv:2012.08508*, 2020.

[12] A. Kamath, M. Singh, Y. LeCun, I. Misra, G. Synnaeve, and N. Carion. Mdetr–modulated detection for end-to-end multi-modal understanding. *arXiv preprint arXiv:2104.12763*, 2021.

[13] P. Anderson, Q. Wu, D. Teney, J. Bruce, M. Johnson, N. Sünderhauf, I. Reid, S. Gould, and A. V. Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3674–3683, 2018.

[14] A. Ku, P. Anderson, R. Patel, E. Ie, and J. Baldridge. Room-across-room: Multilingual vision-and-language navigation with dense spatiotemporal grounding. In *EMNLP*, 2020.

[15] Y. Qi, Q. Wu, P. Anderson, X. Wang, W. Wang, C. Shen, and A. V. Hengel. Reverie: Remote embodied visual referring expression in real indoor environments. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9979–9988, 2020.

[16] K. Chen, J. K. Chen, J. Chuang, M. V'azquez, and S. Savarese. Topological planning with transformers for vision-and-language navigation. *ArXiv*, abs/2012.05292, 2020.

[17] M. Savva, A. Kadian, O. Maksymets, Y. Zhao, E. Wijmans, B. Jain, J. Straub, J. Liu, V. Koltun, J. Malik, D. Parikh, and D. Batra. Habitat: A platform for embodied ai research. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9338–9346, 2019.

[18] C.-Y. Ma, J. Lu, Z. Wu, G. Al-Regib, Z. Kira, R. Socher, and C. Xiong. Self-monitoring navigation agent via auxiliary progress estimation. *ArXiv*, abs/1901.03035, 2019.

[19] L. Ke, X. Li, Y. Bisk, A. Holtzman, Z. Gan, J. Liu, J. Gao, Y. Choi, and S. Srinivasa. Tactical rewind: Self-correction via backtracking in vision-and-language navigation. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6734–6742, 2019.

[20] P. Anderson, A. X. Chang, D. S. Chaplot, A. Dosovitskiy, S. Gupta, V. Koltun, J. Kosecka, J. Malik, R. Mottaghi, M. Savva, and A. Zamir. On evaluation of embodied navigation agents. *ArXiv*, abs/1807.06757, 2018.

[21] D. Fried, R. Hu, V. Cirik, A. Rohrbach, J. Andreas, L.-P. Morency, T. Berg-Kirkpatrick, K. Saenko, D. Klein, and T. Darrell. Speaker-follower models for vision-and-language navigation. In *NeurIPS*, 2018.

[22] Y. Hong, Q. Wu, Y. Qi, C. Rodriguez-Opazo, and S. Gould. A recurrent vision-and-language bert for navigation. *ArXiv*, abs/2011.13922, 2020.

[23] A. Shrivastava, K. Gopalakrishnan, Y. Liu, R. Piramuthu, G. Tur, D. Parikh, and D. Hakkani-Tur. Visitron: Visual semantics-aligned interactively trained object-navigator. *ArXiv*, abs/2105.11589, 2021.

[24] P. Anderson, A. Shrivastava, J. Truong, A. Majumdar, D. Parikh, D. Batra, and S. Lee. Sim-to-real transfer for vision-and-language navigation. *arXiv preprint arXiv:2011.03807*, 2020.

[25] J. Krantz, E. Wijmans, A. Majumdar, D. Batra, and S. Lee. Beyond the nav-graph: Vision-and-language navigation in continuous environments. In *European Conference on Computer Vision*, pages 104–120. Springer, 2020.

[26] V. Blukis, R. A. Knepper, and Y. Artzi. Few-shot object grounding and mapping for natural language robot instruction following. *ArXiv*, abs/2011.07384, 2020.

[27] J. Roh, C. Paxton, A. Pronobis, A. Farhadi, and D. Fox. Conditional driving from natural language instructions. *ArXiv*, abs/1910.07615, 2019.

[28] M. Z. Irshad, C.-Y. Ma, and Z. Kira. Hierarchical cross-modal agent for robotics vision-and-language navigation. *ArXiv*, abs/2104.10674, 2021.

[29] B. A. Plummer, L. Wang, C. M. Cervantes, J. C. Caicedo, J. Hockenmaier, and S. Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. *International Journal of Computer Vision*, 123:74–93, 2015.

[30] S. Kazemzadeh, V. Ordonez, M. Matten, and T. L. Berg. Referitgame: Referring to objects in photographs of natural scenes. In *EMNLP*, 2014.

[31] J. Mao, J. Huang, A. Toshev, O.-M. Camburu, A. Yuille, and K. Murphy. Generation and comprehension of unambiguous object descriptions. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11–20, 2016.

[32] L. Yu, Z. L. Lin, X. Shen, J. Yang, X. Lu, M. Bansal, and T. L. Berg. Mattnet: Modular attention network for referring expression comprehension. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1307–1315, 2018.

[33] S. Yang, G. Li, and Y. Yu. Dynamic graph attention for referring expression comprehension. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4643–4652, 2019.

[34] L. Yu, P. Poirson, S. Yang, A. Berg, and T. L. Berg. Modeling context in referring expressions. *ArXiv*, abs/1608.00272, 2016.

[35] Y. Liu, B. Wan, X.-D. Zhu, and X. He. Learning cross-modal context graph for visual grounding. *ArXiv*, abs/1911.09042, 2020.

[36] L. Ye, M. Rochan, Z. Liu, and Y. Wang. Cross-modal self-attention network for referring image segmentation, 2019.

[37] D. Z. Chen, A. X. Chang, and M. Nießner. Scanrefer: 3d object localization in rgb-d scans using natural language. *16th European Conference on Computer Vision (ECCV)*, 2020.

[38] Z. Yuan, X. Yan, Y. Liao, R. Zhang, Z. Li, and S. Cui. Instancerefer: Cooperative holistic understanding for visual grounding on point clouds through instance multi-level contextual referring, 2021.

[39] M. Feng, Z. Li, Q. Li, L. Zhang, X. Zhang, G. Zhu, H. Zhang, Y. Wang, and A. Mian. Free-form description guided 3d visual graph network for object grounding in point cloud, 2021.

[40] Z. Yang, S. Zhang, L. Wang, and J. Luo. SAT: 2d semantics assisted training for 3d visual grounding. *CoRR*, abs/2105.11450, 2021. URL https://arxiv.org/abs/2105.11450.

[41] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.

[42] C. R. Qi, L. Yi, H. Su, and L. J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *NIPS*, 2017.

[43] A. Akbik, T. Bergmann, D. Blythe, K. Rasul, S. Schweter, and R. Vollgraf. Flair: An easy-to-use framework for state-of-the-art nlp. In *NAACL 2019, 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 54–59, 2019.

[44] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, and A. M. Rush. Huggingface's transformers: State-of-the-art natural language processing, 2020.