Abstract

Understanding 3D scenes from multi-view inputs has been proven to alleviate the view discrepancy issue in 3D visual grounding. However, existing methods normally neglect the view cues embedded in the text modality and fail to weigh the relative importance of different views. In this paper, we propose ViewRefer, a multi-view framework for 3D visual grounding exploring how to grasp the view knowledge from both text and 3D modalities. For the text branch, ViewRefer leverages the diverse linguistic knowledge of large-scale language models, e.g., GPT, to expand a single grounding text to multiple geometry-consistent descriptions. Meanwhile, in the 3D modality, a transformer fusion module with inter-view attention is introduced to boost the interaction of objects across views. On top of that, we further present a set of learnable multi-view prototypes, which memorize scene-agnostic knowledge for different views, and enhance the framework from two perspectives: a view-guided attention module for more robust text features, and a view-guided scoring strategy during the final prediction. With our designed paradigm, ViewRefer achieves superior performance on three benchmarks and surpasses the second-best by +2.8%, +1.5%, and +1.35% on Sr3D, Nr3D, and ScanRefer.

1. Introduction

The aim of visual grounding is to ascertain the precise location of an object from an image or a 3D scene according to given query texts. The field of 2D visual grounding [24, 44, 28] has undergone significant advances recently. Meanwhile, the developments in embodied agent [36, 9], vision-language navigation [62, 40], and autonomous driving [14] have also stimulated increasing attention on 3D visual grounding [1, 6].

Inspired by the strategies in 2D counterparts, most 3D visual grounding methods adopt a two-stage pipeline, which first detect all object proposals in the scene and then ground the target ones. Unlike 2D images with fixed object positions, the large-scale 3D scenes consist of irregular-distributed point clouds with intricate spatial information, which is view-invariant and causes more challenges. As discussed in previous works [35, 22], one urging issue is view discrepancy, caused by the uncertain perspective between the intelligent agent (model) and the commander (grounding text giver). Given relative positions between objects, the text descriptions are supposed to change according to different viewpoints, e.g., when turning the view from “facing” into “back to”, the “right” chair should be rectified as a
“left” one. Unfortunately, the public available datasets [1, 6] for 3D visual grounding only provide one text query corresponding to point clouds of uncertain viewpoints.

To alleviate such potential misalignment, existing methods either manually align the 3D scenes to the paired texts [35], or simultaneously feed multiple views into the network for better view robustness [22]. However, these methods have two major limitations. Firstly, they only focus on solving the view dependence issue from the 3D modality, while neglecting the lack of view cues within text input. Secondly, for the multi-view input, they introduce no specifically designed modules to capture the view knowledge, which is yet significant to discriminate the relative importance of each view. Therefore, we ask: Can we explicitly grasp the view knowledge from both text and 3D modalities to further boost the 3D grounding performance?

To this end, we propose ViewRefer, a multi-view framework for 3D visual grounding, which captures sufficient view cues from both text and 3D modalities to understand the spatial inter-object relation. Our overall paradigm is shown in Figure 1. For the text modality, we leverage large-scale language models (LLMs) to expand the input grounding text with view-related descriptions. Such LLM-expanded texts can capture sufficient multi-view semantics inherited from LLMs’ linguistic knowledge and perform better grounding performance for the target objects. For the 3D modality, we take as input the multi-view 3D scenes and adopt a fusion transformer for 3D-text feature interactions. In each block, we apply different attention mechanisms to exchange information across modalities, views, and objects, contributing to thorough multi-modal fusion. On top of that, we further introduce a set of learnable multi-view prototypes, which aims to capture the inter-view knowledge during training. The guidance of prototypes lies in two aspects. The first complements input text features with adaptive multi-view semantics, the second refines the final output by weighing the importance of different views. Both of them provide high-level guidance for multi-view visual grounding in ViewRefer.

To demonstrate the effectiveness of our approach, we evaluate its performance on three commonly used benchmarks, i.e., Sr3D [1], Nr3D [1] and ScanRefer [6], where ViewRefer consistently achieves superior performance, surpassing the second-best by +2.8%, +1.5%, +1.35%, respectively. The main contributions of our paper are summarized as follows:

- We propose ViewRefer, a multi-view framework for 3D visual grounding, which grasps view knowledge to alleviate the challenging view discrepancy issue.

- For the text and 3D modalities, we respectively introduce LLM-expanded grounding texts and a fusion transformer for capturing multi-view information.

- We present multi-view prototypes to provide high-level guidance to our framework, which contributes to superior 3D grounding performance.

2. Related Work

3D Visual Grounding. 3D Visual Grounding task aims to locate the targeted object in a 3D scene according to a natural language expression. As the baselines, Scanrefer [6] and Referit3D [1] first propose datasets for 3D visual grounding that contain object-annotation pairs on ScanNet [7]. Most recent works [15, 47, 59, 35, 45, 22, 42] adopt a two-stage pipeline that leveraging ground truth or a 3D object detector [26, 23] to obtain the object proposals, utilizing text and 3D encoder [31, 32, 52, 54] to extract features, and then grounding the target one after the feature fusion. Among them, InstanceRefer [47] simplifies the task by treating it as an instance-matching problem. LanguageRefer [35] transforms the multi-modal task into a language modeling problem by replacing the 3D features with predicted object labels. SAT [45] utilizes extra 2D semantics to enhance the multi-modal alignment. MVT [22] first attempts to alleviate the view discrepancy issue by starting from 3D modality to build a view-robust multi-modal representation. Different from the two-stage methods, 3D-SPS [30] treats the 3D visual grounding task as a keypoint selection problem, and first proposes a single-stage network to bridge the gap between detection and matching. Some of the works above point out the crucial view discrepancy issue in 3D visual grounding and propose some preliminary designs to address it. Unlike prior works that only focus on solving the issue from the 3D modality, we start by grasping view knowledge from both text and 3D modalities to address the challenging view discrepancy problem.

Multi-view Learning in 3D. Some recent studies [38, 3, 37, 11, 19, 18] in 3D vision concentrate on improving representation learning by generating 2D renderings from 3D under multiple viewpoints. MVCNN [38] generates a large number of 2D images, and simply encodes them for 3D shape classification. PointCLIP [50] transfers 2D knowledge into the 3D domain via multi-view projection for zero-shot learning. SimpleView [11] proposes an effective multi-view framework for shape classification in 3D point cloud learning. Following the framework of [49], I2P-MAE [53] utilizes projected multi-view 2D depth maps to guide 3D point cloud pre-training. DETR3D [41], PETR [29], and related methods [17, 20] conduct 3D object detection based on multi-view images. These works have shown the effectiveness of multi-view representations in enhancing the performance and robustness of various 3D tasks. For our 3D visual grounding task, we introduce ViewRefer to effectively grasp the multi-view cues from multi-modal input data.
Multi-modality Learning in 3D. Multi-modality learning aims at learning from signals of multiple modalities at the same time, which obtains more robust performance than single-modality learning. Recently, multiple multi-modality networks in 2D vision [34, 10, 57, 60, 61, 13, 27, 56, 33, 48, 51, 46, 39] show effective performance. Inspired by them, PointCLIP V1 [50] and V2 [63] introduce multi-modality into 3D by adopting CLIP’s pre-trained knowledge. What’s more, [58, 55] conduct point-voxel joint-training design, and CrossPoint [2] proposes an image-point contrastive learning network. Also, via filter inflation, Image2point [43] utilizes pre-trained 2D knowledge for point cloud understanding. With the emergence of MAE [16], some works [53, 12, 5] combine multi-modality learning with MAE-based pre-training paradigm and achieve great representation capabilities. For 3D visual grounding, a nature multi-modality task in 3D, ViewRefer imports specific designs for both single-modal feature extraction and multi-modal fusion for precise cross-modal grounding.

3. Method

In this section, we illustrate the details of ViewRefer for 3D visual grounding. We first present our overall pipeline in Section 3.1. Then, in Section 3.2 and 3.3, we respectively elaborate on the proposed designs for text and 3D modalities, i.e., LLM-expanded grounding texts and fusion transformer. Finally, in Section 3.4, we introduce the multi-view prototypes into our framework.

3.1. Overall Pipeline

The whole framework of ViewRefer is shown in Figure 2. Given the point cloud of a 3D scene, we first rotate its coordinates as $N$ different views and encode the $N$-view features $F_v \in \mathbb{R}^{N \times K \times D}$ following [22], where $K$ and $D$ denote object number and feature dimension, respectively. Meanwhile, for the input text, we propose to feed it into the pre-trained large-scale language models (LLMs) and obtain $M$ expanded texts with view cues, denoted as $T$ (Section 3.2). Then, we adopt BERT [8] as the text encoder to extract the LLM-expanded text features as $F_t \in \mathbb{R}^{M \times L \times D}$, where $L$ denotes the max sequence length.

On top of that, the $N$-view 3D features and $M$ view-guided text features are fed into fusion transformer with cascaded blocks for multi-modal interactions (Section 3.3). Each block sequentially contains the layers for intra-view self-attention, multi-model cross-attention, and inter-view self-attention. By this fusion transformer, the multi-view 3D features can sufficiently interact with each other and incorporate grounding information from the expanded text features. After this, the fused features, denoted as $F_{vt}$, are passed into a prediction head for multi-view grounding logits, which are finally aggregated across views as the output.

To better grasp the latent view cues, we propose a set of multi-view prototypes learnable during training, and leverage them to assist the grounding from two aspects (Section 3.4). First, we leverage the prototypes to produce view-guided scoring.
guided context $F_q$ to the text domain, which is then combined with text features as $F_t'$ before feeding into the fusion transformer. Second, the prototypes are also utilized for weighing the importance of different views. By a view-guided scoring strategy, we further inject multi-view knowledge into the final logits for better grounding results.

We follow previous works [35, 22] to adopt three losses on ViewRefer, which are cross-entropy loss $L_{ref}$ on the grounding predictions, $L_{text}$ of text classification on the text encoder, and $L_{3D}$ of 3D shape classification upon the object encoder. The whole loss function is formulated as

$$L = L_{ref} + \beta \cdot L_{text} + \gamma \cdot L_{3D},$$

where $\beta$ and $\gamma$ denote the weights for $L_{text}$ and $L_{3D}$. Both of them are set as 0.5 in our method.

### 3.2. LLM-expanded Grounding Texts

To fully exploit view knowledge within the text modality, we propose to utilize the abundant linguistic knowledge in LLMs, e.g., GPT-3 [4], to expand the original simple grounding texts. As discussed in [1], the 3D-text data in 3D visual grounding task can be divided into view-dependent and view-independent pairs, depending on whether the expression is constrained to the speaker’s perspective. For both of them, we construct general command templates fed into the GPT-3 to generate $M$ LLM-expanded texts. Further, considering their different semantic compositions, we adopt specific expanding strategies for the two categories.

**View-Dependent Text.** The description in such texts is strongly related to speaker’s viewpoint. Therefore, for an input view-dependent text, we first generate its synonymous sentence that refers to the same target but under different speaker’s views. Specifically, we create a view-related phrase dictionary, the keys of which include phrases for orientation like “looking at”, and phrases describing positional relationships like “left”. The corresponding values to the keys are the opposite phrases of them, e.g., “looking at - with back to”, and “left - right”. Then, we construct a command template as the input for GPT-3, which includes the input text and aforementioned view-related phrases in the dictionary. Such general template is designed as

```
"Rephrase the sentence of '[TEXT]' to the opposite perspective, which contains phrases '"[PHRASES]': "
```

where [TEXT] and [PHRASES] are replaced by the input text and view-related phrases, respectively. For example, given the initial text of “Facing the front of the couch, pick the table that is to the right of the couch”, we complete the command as

```
"Rephrase the sentence of 'Facing the front of the couch, pick the table that is to the right of the couch' to the opposite perspective, which contains phrase 'with back to', 'left': "
```

By this command, we obtain GPT-3’s response as “With back to the front of the couch, pick the table that is on the left side of the couch”. In this way, the rephrased description contains opposite-view information. Together with the original single text, we enrich the text modality with more view-dependent semantics. To further unleash LLMs’ linguistic knowledge, we respectively rephrase the input sentence and its opposite-view synonym for $(M - 2)/2$ times with a command template like

```
"Rephrase the sentence of '[TEXT]': "
```

where [TEXT] is replaced by the input text or its opposite-view synonym. Additionally, we also provide pre-defined replaced pairs as input prompts for rephrasing, e.g., ‘near’-‘next to’, ‘beside’. Through this expanding strategy, we enrich the initial single text into a set of $M$ LLM-expanded texts, which contain abundant multi-view cues and effectively alleviate the discrepancy issue.

**View-Independent Text.** The description in view-independent texts is irrelevant to speaker’s viewpoint. Thus, we adopt no opposite-view enrichment on these view-independent inputs, and directly rephrase them for

![View-guided Scoring Strategy](image_url)
first randomly initialize a learnable context \( Q \) to incorporate view-guided contexts into the textual view-guided Textual Context. We first utilize the prototypes as ‘key’ and ‘value’, and propose a view-guided 3D-text view-consistent knowledge from a common multi-modal space, which serve as high-level guidance for the grounding process from the following two aspects. Firstly, an intra-view attention layer is adopted for the \( N \)-view features \( F_v \), which independently interacts \( K \) object features within each view. Secondly, we utilize a multi-modal cross-attention layer referring to [22] to fuse textual information from \( F_v \) into the multi-view features. We formulate them as

\[
F_v = F_v + \text{Intra-Attn}(F_v),
\]

\[
F_{vt} = F_v + \text{Cross-Attn}(F_v, F'_t),
\]

where \( F_{vt} \in \mathbb{R}^{N \times K \times D} \) denotes the multi-modal features.

After that, to boost the cross-view communication, we subsequently introduce an inter-view attention layer. Orthogonal to intra-view attention, this layer calculates attention weights across different views for the same object, formulated as

\[
F_{vt} = \left( F'_{vt} + \text{Inter-Attn}(F'_{vt}) \right)^T.
\]

With such interaction, the network can better capture the inter-view differences from one object, and focus on more informative views to encode multi-modal features \( F_{vt} \).

3.4. Multi-view Prototypes

Besides our LLM-expanded texts and fusion transformer, we further introduce a set of multi-view prototypes to conduct view-guided 3D visual grounding in ViewRefer. The prototypes are randomly initialized embeddings and represented as \( \text{Proto} \in \mathbb{R}^{N \times D} \), where \( N, D \) denote the view number and feature dimension. They aim to memorize the 3D-text view-consistent knowledge from a common multi-modal space, which serve as high-level guidance for the grounding process from the following two aspects.

View-guided Textual Context. We first utilize the prototypes to incorporate view-guided contexts into the textual features \( F_t \). As shown in the bottom right of Figure 2, we first randomly initialize a learnable context \( Q \in \mathbb{R}^{M \times D} \), which serves as ‘query’ to extract 3D-text multi-view information from the prototypes \( \text{Proto} \). Then, we regard the prototypes as ‘key’ and ‘value’, and propose a view-guided attention module with residual connections for feature interaction. We utilize one cross-attention layer in the module, formulated as

\[
F_q = Q + \text{VG-Attn}(\text{Proto}, Q),
\]

where \( \text{VG-Attn} \) denotes the view-guided attention module. After this, the query features \( F_q \) have adaptively aggregate informative multi-view cues and are element-wisely added to the input text features \( F_t \) as

\[
F'_t = F_t + \alpha \cdot F_q,
\]

where \( \alpha \) denotes a frozen balance factor weighing the importance of view-guided context, which is initialized before training. From the interaction with prototypes, the text features \( F'_t \) become view-aware and perform better multi-modal fusion in the subsequent transformer.

View-guided Scoring Strategy. Furthermore, we leverage the multi-view prototypes to conduct a scoring for the predicted logits after the fusion transformer. As shown in Figure 3, we calculate \( N \)-view scores for the multi-modal features \( F_{vt} \), which assess the relative importance of each view in the final prediction process. We first apply a pooling operation to \( F_{vt} \) along the dimension of intra-view objects, which aggregates the global features of different views. We then adopt linear layers to project the prototypes and multi-modal features into a unified embedding space. Upon that, we calculate the multi-view scores by matrix multiplication of each \( N \)-view pair as

\[
S = \text{MatMul} \left( \text{Proto}, \text{Pooling}(F_{vt}) \right),
\]

where \( S \in \mathbb{R}^{N \times 1} \) denotes the score for each \( N \) view. After this, the multi-view scores serve as the weights to aggregate predicted logits among different views. Such scoring can adaptively constrain the prediction of those views inconsistent with the input text, and emphasize the informative views for accurate 3D visual grounding.

4. Experiments

In this section, we evaluate the performance of ViewRefer on three commonly used benchmarks, i.e., Sr3D [1], Nr3D [1], and ScanRefer [6].

4.1. Datasets

Sr3D and Nr3D. The Sr3D dataset [1] contains 83,572 template-based utterances leveraging spatial relationships among objects to localize a referred object in a 3D scene. The scenes are constrained to have no more than six distractors, i.e., objects belonging to the same category as the target. The Nr3D dataset [1] provides annotations for the
we utilize the ground truth object proposals for Nr3D and

<table>
<thead>
<tr>
<th>Method</th>
<th>Sr3D</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th>Nr3D</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>ReferIt3D [1]</td>
<td>40.8%</td>
<td>44.7%</td>
<td>31.5%</td>
<td>39.2%</td>
<td>40.8%</td>
<td>35.6%</td>
<td>43.6%</td>
<td>27.9%</td>
<td>32.5%</td>
</tr>
<tr>
<td>TGNN [21]</td>
<td>45.0%</td>
<td>48.5%</td>
<td>36.9%</td>
<td>45.8%</td>
<td>45.0%</td>
<td>37.3%</td>
<td>44.2%</td>
<td>30.6%</td>
<td>35.8%</td>
</tr>
<tr>
<td>InstanceRefer [47]</td>
<td>48.0%</td>
<td>51.1%</td>
<td>40.5%</td>
<td>45.4%</td>
<td>48.1%</td>
<td>38.8%</td>
<td>46.0%</td>
<td>31.8%</td>
<td>34.5%</td>
</tr>
<tr>
<td>3DVG-Transformer [59]</td>
<td>51.4%</td>
<td>54.2%</td>
<td>44.9%</td>
<td>44.6%</td>
<td>51.7%</td>
<td>40.8%</td>
<td>48.5%</td>
<td>34.8%</td>
<td>34.8%</td>
</tr>
<tr>
<td>LanguageRefer [35]</td>
<td>56.0%</td>
<td>58.9%</td>
<td>49.3%</td>
<td>49.2%</td>
<td>56.3%</td>
<td>43.9%</td>
<td>51.0%</td>
<td>36.6%</td>
<td>41.7%</td>
</tr>
<tr>
<td>TransRefer3D [15]</td>
<td>57.4%</td>
<td>60.5%</td>
<td>50.2%</td>
<td>49.9%</td>
<td>57.7%</td>
<td>42.1%</td>
<td>48.5%</td>
<td>36.0%</td>
<td>36.5%</td>
</tr>
<tr>
<td>SAT [45]</td>
<td>57.9%</td>
<td>61.2%</td>
<td>50.0%</td>
<td>49.2%</td>
<td>58.3%</td>
<td>49.2%</td>
<td>56.3%</td>
<td>42.4%</td>
<td>46.9%</td>
</tr>
<tr>
<td>MVT [22]</td>
<td>64.5%</td>
<td>66.9%</td>
<td>58.8%</td>
<td>58.4%</td>
<td>64.7%</td>
<td>55.1%</td>
<td>61.3%</td>
<td>49.1%</td>
<td>54.3%</td>
</tr>
</tbody>
</table>

Table 1: Performance of ViewRefer on Sr3D and Nr3D. We report the performance on the overall dataset and all its splits. ‘*’ denotes our implementation results.

<table>
<thead>
<tr>
<th>Method</th>
<th>Multiple</th>
<th>Overall</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Acc@0.25</td>
<td>Acc@0.50</td>
</tr>
<tr>
<td>ReferIt3D [1]</td>
<td>21.03%</td>
<td>12.83%</td>
</tr>
<tr>
<td>ScanRefer [6]</td>
<td>30.63%</td>
<td>19.75%</td>
</tr>
<tr>
<td>TGNN [21]</td>
<td>27.01%</td>
<td>21.88%</td>
</tr>
<tr>
<td>InstanceRefer [47]</td>
<td>31.27%</td>
<td>24.77%</td>
</tr>
<tr>
<td>MVT [22]</td>
<td>31.92%</td>
<td>25.26%</td>
</tr>
<tr>
<td>MVT* [22]</td>
<td>31.46%</td>
<td>24.85%</td>
</tr>
<tr>
<td>ViewRefer</td>
<td>33.08%</td>
<td>26.50%</td>
</tr>
</tbody>
</table>

Table 2: Performance of ViewRefer on ScanRefer. We report the Acc@0.25 and Acc@0.50 on ScanRefer dataset. ‘*’ denotes our implementation results.

ScanNet [7], an indoor 3D scene dataset, comprising 45,503 natural and free-form utterances. The dataset includes 707 distinct indoor scenes with target objects belonging to 76 fine-grained categories. Two distinct data splits are employed in Sr3D and Nr3D, namely the “Easy” and “Hard” splits that differ based on the number of distractors in the scene, and the “View-dependent” and “View-independent” splits that vary based on whether the referring expression relies on the speaker’s viewpoint.


4.2. Experimental Settings

Evaluation Metrics. Following existing works [22, 47], we utilize the ground truth object proposals for Nr3D and

Sr3D [1] datasets, while adopting detector-generated object proposals via pre-trained detector, PointGroup [23], for ScanRefer [6] dataset. As for the evaluation metrics, we follow previous works [22, 47] to measure the networks via the accuracy of the target predictions for Nr3D and Sr3D, and the Acc@mIoU metric for ScanRefer where, $m \in \{0.25, 0.50\}$. The Acc@mIoU measures the proportion of text input with the predicted bounding box overlapping the ground truth box by intersection over the union (IoU) higher than $m$.

Implementation Details. For Sr3D and Nr3D datasets [1], we train the network for 100 epochs with a batch size of 24. We utilize AdamW [25] as the optimizer with an initial learning rate of $5 \times 10^{-5}$ for the fusion transformer, $5 \times 10^{-4}$ for other modules, and weight decay as $1 \times 10^{-3}$. After 40 epochs, we decline the learning rate by multiplying 0.65 per 10 epochs. For ScanRefer [6], we set the max epoch number as 30, and batch size as 32. We adopt AdamW [25] as the optimizer with an initial learning rate of $5 \times 10^{-4}$ with no weight decay. As for the structure of the network, we set both the view number $N$ and the generated text number $M$ as 4. We utilize 3 layers for the text encoder and 4 transformer blocks for the fusion transformer with 8 attention heads. For data augmentation, on Sr3D and Nr3D, we conduct translation on each 3D object and random rotation on 3D scenes during training. Additionally, we adopt the flip operation on Sr3D dataset. We utilize the same data augmentation with previous works [22, 47] on ScanRefer dataset. As for the pooling operation in the view-guided scoring strategy, we adopt sum pooling for Sr3D and ScanRefer datasets, and the summation of max and average pooling for Nr3D.

1We reproduce the results based on the official open-source code [22] on a single NVIDIA A100 GPU.
Facing the front of the toilet, choose the toilet paper that is on the back of it.

Looking at the front of the armchair, pick the office chair that is on the right of the armchair.

Looking at the front of the piano, choose the doors that are on the left side of it.

Find the shelf that is in front of the bookshelf.

Facing the front of the toilet, select the curtain that is to the right of it.

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

Figure 4: Visualization of the 3D Visual Grounding Results. For the presented 3D scenes, we utilize green, red, blue, and orange boxes to represent the ground truth, incorrect predictions, correct predictions, and view-dependent objects, respectively.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>22.4%</td>
<td>23.9%</td>
<td>18.7%</td>
<td>26.9%</td>
<td>22.2%</td>
</tr>
<tr>
<td>✓</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>61.5%</td>
<td>64.2%</td>
<td>55.2%</td>
<td>50.6%</td>
<td>61.9%</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>64.4%</td>
<td>67.0%</td>
<td>58.6%</td>
<td>51.5%</td>
<td>65.0%</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>-</td>
<td>-</td>
<td>65.0%</td>
<td>67.2%</td>
<td>59.9%</td>
<td>51.7%</td>
<td>65.6%</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>66.0%</td>
<td>68.4%</td>
<td>60.4%</td>
<td>51.9%</td>
<td>66.4%</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>66.5%</td>
<td>68.7%</td>
<td>60.8%</td>
<td>52.1%</td>
<td>66.9%</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>67.0%</td>
<td>68.9%</td>
<td>62.1%</td>
<td>52.2%</td>
<td>67.7%</td>
</tr>
</tbody>
</table>

Table 3: Ablation Study on Different Components for Grasping Multi-view Knowledge on Sr3D. ‘VG. Score.’ and ‘VG. Cont.’ denote the view-guided scoring strategy and view-guided textual context, respectively.

4.3. Quantitative Analysis

Performance on Sr3D. In Table 1, we report the performance of ViewRefer on Sr3D dataset [1] for 3D visual grounding. Our ViewRefer outperforms prior works in all splits, surpassing the state-of-the-art method [22] on the overall accuracy by +2.8%, which demonstrates the effectiveness of the view knowledge grasped by ViewRefer. Also, the significant amplification of +5.1% on the challenging ‘Hard’ split shows the effectiveness of our ViewRefer in 3D scenes with complex spatial relations between objects, which further indicates our superior understanding of spatial relations.

Performance on Nr3D. In Table 1, we evaluate ViewRefer on Nr3D dataset, where ViewRefer exceeds best competitor [22] by an overall improvement of +1.5%, and +2.1% on “view-dependent” split, indicating the superiority of ViewRefer on data with intricate view-dependent natural referential utterances, which further demonstrates the effectiveness of our view-guided designs.

Results on ScanRefer. We also evaluate ViewRefer on ScanRefer [6] and report the Acc@nIoU performance of ViewRefer in Table 2. Clearly, compared with prior works, our ViewRefer achieves the highest scores on all metrics. This well illustrates the superior view-understanding capability of our multi-view training framework.
4.4. Qualitative Analysis

3D Visual Grounding Results. In Figure 4, we select some view-dependent cases from Sr3D [1] and visualize the grounding truth boxes, predictions of MVT [22], and predictions of ViewRefer in each column from top to bottom. We emphasize the view-dependent objects in orange boxes with highlights for view alignment. As shown in (a), MVT fails to ensure the correct view as the text input describes, and grounds to wrong object with the same category, while ViewRefer successfully predicts the target under the correct perspective. From the predictions in (b) and (c), we find although ViewRefer and MVT both understand the correct views, only ViewRefer grounds the right target, which shows that with the view-guided designs, our network gradually captures the view knowledge as the network deepens.

Failure Cases. In Figure 6, we visualize the failure cases of our ViewRefer. We observe that the failure cases can be grouped into two parts. One part, like the first two columns, contains overmuch distractors that have the same class as the target object, which critically confuses the model. Also, as shown in the last two columns, the other part’s difficulty is the long indigestible ground texts containing too complex grammatical structures for the model to grasp the semantic information. In the future, we will focus on the solutions for these kinds of cases.

Interpretability of View-guided Scoring Strategy. To demonstrate the interpretability of the proposed view-guided scoring strategy, we utilize the multi-view prototypes and output features from the fusion transformer blocks, to calculate and analyze the intermediary multi-view scores. In Figure 5, we present the scores calculated from the first and last blocks of view-dependent cases, where we mark the correct view with “✓”. As shown from the last blocks’ scores, with the proposed scoring strategy, ViewRefer can effectively assess the relative importance of each view for exact visual grounding. Also, we observe the upward trend of the scores of correct view, which shows that with the view-guided designs, our network gradually captures the view knowledge as the network deepens.

5. Ablation Study

To explore the effectiveness of each design in ViewRefer, we conduct extensive ablation study on Sr3D [1] dataset and evaluate the overall accuracy on 3D visual grounding. In Table 3, we report the results of different combinations of proposed designs. In Table 4, we explore the positions of inter-view attention and different view numbers.
Looking at the front of the bed, pick the backpack that is on the right side of the bed.

Looking at the front of the desk, choose the picture that is to the left of it.

The correct window is the smaller square one when standing in the middle of the room facing the bookshelves it is on the right.

As you walk towards the table with window facing you, it is the chair closest to you on the left hand side.

Figure 6: Visualization of Failure Cases. The green boxes and the red boxes represent the ground truth objects and the predictions of ViewRefer, respectively. The blue boxes are the distractors.

<table>
<thead>
<tr>
<th>Position of Inter-view Attn.</th>
<th>View Number</th>
<th>Overall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Before</td>
<td>Between</td>
<td>After</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>-</td>
</tr>
<tr>
<td>✓</td>
<td>-</td>
<td>✓</td>
</tr>
<tr>
<td>✓</td>
<td>-</td>
<td>✓</td>
</tr>
<tr>
<td>✓</td>
<td>-</td>
<td>✓</td>
</tr>
</tbody>
</table>

Table 4: Ablation Study on Inter-view Attention and View Number. The ‘Before’, ‘Between’, ‘After’ denote block structure of putting the “inter-view attention” layer before, between, and after the “intra-view attention” layer and “multi-modal cross-attention” layer, respectively.

Effectiveness of Each Component. Based on the baseline network, we conduct ablation studies by adding the designed modules one by one until the final structure of ViewRefer. In Table 3, the first two rows in light gray are baselines with basic fusion modules as described in [22]. As reported, the import of each major component benefits the object location separately.

Fusion Transformer and View Number. In Table 4, we conduct ablation studies on the structures of fusion transformer and 3D view number. For the fusion transformer, we evaluate ViewRefer with three different designs within the fusion block, which are placing the “inter-view attention” layer before, between, and after the “intra-view attention” layer and “multi-modal cross-attention” layer, respectively. We also explore the influence of 3D view number in Table 4. As shown, the fusion transformer with inter-view attention on the last position, and the view number of 4 perform the best. This is because, after the internal interaction among objects within each view, and the fusion between 3D and text, the 3D feature with textual and inner spatial information can boost the cross-view interaction in the inter-view attention layer. Note that subjected to the requirement of frozen view number in the view-guided scoring strategy, we do not evaluate ViewRefer under the setting of adopting different view numbers for training and test phases.

6. Conclusion

We propose ViewRefer, a multi-view framework for 3D visual grounding that grasps view knowledge from both text and 3D modalities to alleviate the challenging view discrepancy issue. In the text modality, we introduce LLM-expanded grounding texts to utilize the diverse linguistic knowledge of large-scale language models for view understanding. For 3D modality, we propose a fusion transformer with inter-view attention to grasp rich multi-view knowledge. Furthermore, with a set of learnable multi-view prototypes, we enhance ViewRefer from two perspectives, view-guided textual context and a view-guided scoring strategy, which further enhances view-guided multi-modal fusion for exact grounding. Extensive experiments are conducted to demonstrate the superiority of ViewRefer on 3D visual grounding. We expect ViewRefer can inspire more works to further explore the possibility of grasping view knowledge to benefit view understanding in 3D visual grounding.

Acknowledgements This work is partially supported by the Shanghai AI Laboratory, National Key R&D Program of China (2022ZD0160100) and the National Natural Science Foundation of China (62106183).
References


[16] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16000–16009, 2022. 3


[48] Renrui Zhang, Hanjun Deng, Bohao Li, Wei Zhang, Hao Dong, Hongsheng Li, Peng Gao, and Yu Qiao. Collaboration


