Supplementary Materials of Data-Efficient 3D Visual Grounding via Order-Aware Referring

Tung-Yu Wu[1*] Sheng-Yu Huang[1*] Yu-Chiang Frank Wang[1,2]
[1]Department of Electrical Engineering, National Taiwan University
[2]NVIDIA
{b08901133, f08942095}@ntu.edu.tw, frankwang@nvidia.com

A. More Details of Vigor

This section provides more details regarding model implementations and experimental setups. In particular, we first demonstrate the pseudocode of order-aware sample synthesis for pre-training in Sec. 3.4 and the complete training pipeline in Sec. 3.5. Then, we elaborate on the hyperparameters of Vigor and implementations of baselines for experiments in Sec. 4.

A.1. Pre-Training Sample Synthesis and Training Pipeline

This section provides the pseudocode of synthesizing order-aware samples for Vigor's pre-training in Sec. 3.4 and the complete training pipeline in Sec. 3.5 in our main paper, respectively. Specifically, Algorithm A1 demonstrates the pipeline to synthesize an order-aware pre-training sample given object proposals P and predicted object labels L. On the other hand, Algorithm A2 illustrates the complete pipeline to train Vigor with synthesized samples and natural-description samples, such as NR3D and ScanRefer.

Algorithm A1 Order-Aware Sample synthesis for Vigor Pre-training

```
Input: $P$ and $L$
Hyperparameters: $B$
Output: $D^{aug}$, $O^{aug}_{1:B}$, and $p^{aug}_{1:B}$

randomly sample and arrange $\{l^{aug}_1, \cdots, l^{aug}_B\}$ from $L$.
extract class names of $\{l^{aug}_1, \cdots, l^{aug}_B\}$ as $O^{aug}_{1:B} = \{O^{aug}_1, \ldots, O^{aug}_B\}$.
$D^{aug}$ = "There is a $\{O^{aug}_1\}$ in the room, find the $\{O^{aug}_2\}$ farthest to it, and then find the $\{O^{aug}_3\}$ farthest to that $\{O^{aug}_2\}$, $\{\ldots\}$, finally you can see the $\{O^{aug}_B\}$ farthest to that $\{O^{aug}_{B-1}\}$."
get $p^{aug}_1$ by randomly removing objects in $P$ with class of $O^{aug}_1$ until only one of them is left.
initialize the anchor/target objects set as $\{p^{aug}_1\}$.
for $i = 2, 3, \ldots, B$ do
    for all objects in $P$ with the class of $O^{aug}_i$, find the one farthest from $p^{aug}_{i-1}$ to be $p^{aug}_i$.
    append $p^{aug}_i$ to $\{p^{aug}_1, \ldots, p^{aug}_{i-1}\}$.
end for
$p^{aug}_{1:B} \leftarrow \{p^{aug}_1, \cdots, p^{aug}_B\}$
return $D^{aug}$, $O^{aug}_{1:B}$, and $p^{aug}_{1:B}$
```

A.2. Implementation and Details and Experimental Settings

For both datasets, we use PyTorch [14] library to implement Vigor. We train Vigor using Adam [11] optimizer with a single NVIDIA Tesla V100 GPU.

To conduct batch-wise training with a fixed number of Object-Referring blocks, we set the length of referential order B to be 4, i.e., we trim the original referential order from the front if its length exceeds 4 and pad it if its length is lower than 4. We adopt BERT [4] as the text encoder to extract T and Pointnet++ [15] as the visual encoder to acquire F_1. We sample I=1024 points for each object proposal in the scene. Object proposal number K and token number $|D|$ are sample-dependent. Object feature dimension d_i for i-th Object-Referring block is set to 768, aligning with BERT's 768 dimension, to conduct cross-attention.

A.2.1 NR3D

For NR3D, we use pre-trained Pointnet++ to classify all object proposals as object labels L following BUTD-DETR [9]. We warm-up Vigor for 15k steps on ScanNet scene point cloud and our augmented samples in Sec. 3.4 and continue on real-world data pairs in NR3D (around 1.2k, 12k, and 120k steps for 1%, 10%, and 100% data, respectively) using a batch size of 24. With one NVIDIA

[*] Equal Contribution

Algorithm A2 Vigor Training Pipeline

```
Input: scene-description paired training samples ${\{C_1, D_1\}, \{C_2, D_2\}, \dots}$
Hyperparameters: pre-training step $S_p$, official training step $S_o$, and $B$

initialize Vigor's model weights $\phi$.
$L_{pre} \leftarrow \{\mathcal{L}_{text}, \mathcal{L}_{mask}, \mathcal{L}_{ref}, \mathcal{L}_{crd}\}$
for $i = 2, 3, \dots, S_p$ do
    sample a scene point cloud $C$ from paired training samples.
    acquire $P$ and $L$ of $C$.
    use A1 with $\{P, L, B\}$ as inputs to synthesize $\{D^{aug}, O^{aug}_{1:B}, p^{aug}_{1:B}\}$.
    use $\{P, L, D^{aug}, O^{aug}_{1:B}, p^{aug}_{1:B}\}$ to update $\phi$ with $L_{pre}$.
end for
$L_{train} \leftarrow \{\mathcal{L}_{text}, \mathcal{L}_{mask}, \mathcal{L}_{ref}\}$
for $i = 2, 3, \dots, S_o$ do
    sample a data point $\{C, D\}$ in the training samples
    acquire $P$ and $L$ of $C$.
    apply LLM to acquire $O_{1:B}$.
    use $\{P, L, D, O_{1:B}\}$ to update $\phi$ with $L_{train}$.
end for
```

Tesla V100 GPU, the warm-up takes around 3 hours and the NR3D real-world data training takes around 24 hours when training on 100% data.

A.2.2 SR3D

We present SR3D's results in App. B.2. SR3D contains 65844 training samples and 17726 testing samples. Each sample's description is synthesized by simple spatial relations, such as *farthest* and *beside* with simple sentence structures like "*the monitor that is farthest from the printer.*" We conduct the object classification and warm-up process as in NR3D and continue on SR3D data pairs, with around 2.7k and 27k training steps for 1% and 100% data using batch size 24. SR3D 100% data training takes around 48 hours with one NVIDIA Tesla V100 GPU.

A.2.3 ScanRefer

For ScanRefer, following M3DRef-CLIP [20], we use PointGroup [10] to classify object proposals. We do not apply the warm-up due to noisy and imperfect object proposals and labels under ScanRefer's setting. The batch size is 32, and the GPU is a single NVIDIA Tesla V100 GPU. Training on 100% data requires around 60 hours.

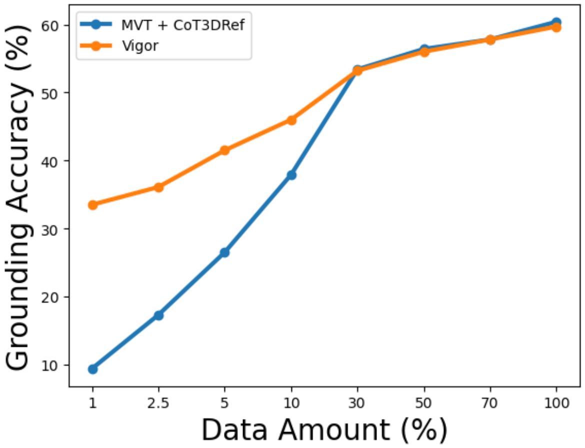


Figure S1. **Quantitative results on NR3D** We can see that when the amount of data is relatively many (above 30%), our Vigor is comparable to MVT+CoT3DRef [2]. However, as the amount of data reduces, our Vigor performs better over MVT-CoT3DRef.

A.2.4 Baselines

For baselines [1–3, 6, 8, 9, 13, 17, 18, 20, 21] on NR3D and ScanRefer in Table 1, 2, and 3, we utilize their official public implementations with different amounts of available training samples and the full testing set to evaluate their low-resource performance. For the full-data (100%) scenario, we acquire their performance either on the official leaderboard of NR3D/ScanRefer or their published papers.

B. Additional Quantitative Results

B.1. More Results on NR3D

In Sec. 4, we have shown comparisons between Vigor and several state-of-the-arts under data efficient scenarios (1% $\sim$ 10% of data). Here, we provide a comparison between Vigor and MVT+CoT3DRef [2] from 1% to 100% of NR3D data in Fig. S1, where Vigor outperforms CoT3DRef with a noticeable margin when the amount of data is limited and is comparable to CoT3DRef when the data amount is over 30%, showing that our Vigor is suitable under different settings.

Table S1 further displays the detailed performance on different official subsets of NR3D. Among the subsets, the **Hard** subset contains samples with more than 2 distractors, where a distractor is an object having the same class name as the target object, and the **Easy** subset is the contrary. **View-dependent** samples contain relations where rotating the point cloud scene will affect the referred ground-truth target object (e.g., left and right), and **View-independent** samples are contrary. Vigor accentuates itself with decent capabilities on different subsets, consistently followed by the two variants of CoT3DRef [2] that also feature the con-

Table S1. **Grounding accuracy on the official NR3D subsets [1].** For implementation and comparison purposes, only the setting of 10% of training data is considered.

Method	Hard	Easy	View-Dep.	View-Indep.	Overall
Referit3D [1]	19.5	27.3	21.2	24.2	23.3
TransRefer3D [6]	21.6	29.9	22.9	27.0	25.7
SAT [18]	22.5	27.6	21.7	26.6	25.0
BUTD-DETR [9]	25.9	41.9	29.1	34.8	33.3
MVT [8]	22.9	30.3	25.4	27.1	26.5
MVT + CoT3DRef [2]	32.7	43.2	34.0	39.8	37.9
ViL3DRel + CoT3DRef [2]	32.4	44.7	33.4	40.9	38.4
Vigor (Ours)	**39.1**	**53.3**	**45.3**	**46.4**	**46.0**

Table S2. **Data Efficient Grounding accuracy (%) on SR3D.** We show the results trained with 1% and 100% of training data.

Method	Labeled Training Data	
	1%	100%
BUTD-DETR [9]	36.5	67.0
MVT [8]	22.2	64.5
NS3D [7]	**52.4**	62.7
MVT + CoT3DRef [2]	26.9	**73.2**
Vigor (Ours)	51.3	67.1

cept of referential order for 3D visual grounding.

B.2. More Results on SR3D

Table S2 shows the quantitative comparisons on the SR3D dataset against BUTD-DETR, MVT, CoT3DRef and NS3D [7], with the settings of using 1% and 100% of training examples, respectively. From this table, we can see that although NS3D achieves the best performance when using only 1% of training data due to its structured decomposition of the input description into nested logical expressions, its performance saturates when using more data for training. As for CoT3DRef, its design of applying rule-based matching of anchor/target objects as additional guidance for the model is effective on SR3D when using a large amount of data, where the relations are much simpler than NR3D or ScanRefer, but its performance on 1% of data is suboptimal. On the contrary, our Vigor achieves comparable results in both settings, showing that our design is suitable for various amounts of training pairs.

C. Ablation Studies on Training Objectives

This section performs the ablation study on several training objectives used for Vigor mentioned in Sec. 3. In particular, we investigate the influence of $\mathcal{L}_{mask}$, $\mathcal{L}_{crd}$, and $\mathcal{L}_{text}$ for low-resource (1% training data) and full-resource (100% training data) scenarios, as shown in Table S3. Note that we only conduct target object classification on the text feature T when deactivating $\mathcal{L}_{text}$, and apply classification on both anchor and target objects when adopting $\mathcal{L}_{text}$, as mentioned in Sec. 3.3. Also, for all experiments in Table S3, the pre-training with synthetic data in Sec. 3.4 is applied for fair comparison. In summary, $\mathcal{L}_{mask}$, $\mathcal{L}_{crd}$, and $\mathcal{L}_{text}$ all impose positive effects on models' performances under the two settings, verifying Vigor's design in Sec. 3.

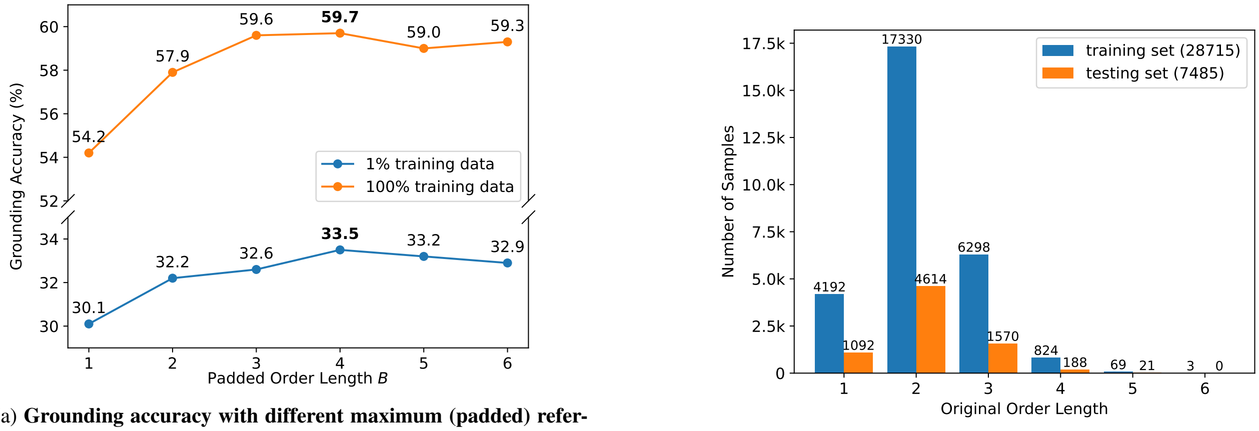
D. Model Performance Under Different Order Length

In this section, we analyze the effectiveness of padded order length B of $O_{1:B}$ (and also the number of Referring Blocks) in Fig. S2, which illustrates the performances with different B using 1% and 100% training data in Fig. S2a. We also show the statistics of the original order length generated by our two-stage referential ordering in Fig. S2b. We observe that although performances gradually increase with larger B (i.e., considering more potential anchor objects appeared in D achieves better accuracy), the performance gain saturates at $B = 4$. This can be explained by looking at Fig. S2b, which shows that only a very small amount of data exceeds an order length of 4 in both training and testing pairs. This suggests that our selection of $B = 4$ is reasonable, optimally balancing computational efficiency with prediction accuracy.

We additionally explore Vigor's performance gain for descriptions with different order lengths compared with MVT [8] to show the effectiveness of our design of progressive location to the target object. We split the original NR3D testing set into subsets that possess order lengths of **1**, **2&3**, and **4&5**. An order length of 1 (1092 samples in the testing set) means that only the target object is mentioned in the description. Samples with an order length of 2 or 3 (6184 samples in the testing set) have 1 or 2 anchor objects mentioned other than the target object. Similarly, samples with an order length of 4 or 5 (209 samples in the testing set) have 3 or 4 anchor objects mentioned other than the target object. A longer referential order may generally denote a longer and more com-

Table S3. **Ablation studies on training objectives.** Note that for all ablation settings, the proposed order-aware pre-training in Sec. 3.4 is applied for fair comparison.

$\mathcal{L}_{mask}$	$\mathcal{L}_{crd}$	$\mathcal{L}_{text}$	$\mathcal{L}_{ref}$	1%	100%
			✓	31.1	54.2
		✓	✓	31.6	54.8
	✓	✓	✓	32.4	56.0
✓	✓	✓	✓	33.5	59.7



(a) **Grounding accuracy with different maximum (padded) referential order length** B**.** Note that B is equal to and denoted as the number of Object-Referring blocks.

(b) **Statistics of the referential order length in NR3D.** A longer referential order denotes a more complex referring process for grounding.

Figure S2. **(a) Grounding accuracies with varying Object-Referring block numbers** B **and (b) statistics of referential order length of NR3D.** It can be seen that the model performance saturates at B=4, matching length statistics of NR3D.

plicated description. As shown in Table S4, though only a 2.2% performance gain is obtained for target-only samples, Vigor is more advantageous when dealing with lengthy descriptions, with 5.3% performance gain achieved for descriptions with order length of 4 or 5.

E. Visualization of Responses in Each Referring Block

To show that our Vigor indeed progressively locates the target object following the derived referential order, we visualize the feature response of $F_{1:(B+1)}$ ($B = 4$) in Fig. S3. The blue bounding box indicates the ground-truth target object, and we color the object proposal according to the response of their corresponding features in $F_{1:(B+1)}$, where a brighter color represents a higher response. We can see that the responses to object proposals are originally cluttered. As our referential blocks are applied, the response of anchor/target objects becomes larger and finally locates the ideal target object in the last feature F_5.

F. Details and Prefix Prompt Examples of Two-Stage ICL for Deriving Referential Order

To have Object-Referring blocks $\{R_1, \cdots, R_B\}$ in Sec. 3 to locate the target object properly, it is desirable to extract a proper referential order $O_{1:B}$ from the input description D. With such a referring path constructed, visual features of the associated objects can be updated for grounding purposes. This is inspired by the idea of Chain-of-Thoughts [12,16] in LLM, as noted in CoT3DRef [2]. To achieve such a parsing task, we apply GPT-3.5-Turbo [19] as the description parser using in-context learning (ICL) [5], as depicted in Fig. S4.

However, it is not trivial to have LLM output a referential order from D due to lengthy and noisy descriptions. For example, for an input description "*Look at the king-size bed in the room next to a green chair, Find the pillow on the bed. Not the pillow on the sofa beside the chair.*", one would expect first to find the green *chair* then the king-size *bed* next to the chair, and finally, the *pillow* on the bed. Therefore, the ideal referential order is {*"chair", "bed", "pillow"*}. In the above example, the sentence "*Not the pillow on the sofa which is also beside the chair.*" is redundant since one

Table S4. **Grounding accuracy on NR3D subsets regarding different parsed referential order lengths.** Note that referential order length $= 1$ means only the target object is mentioned in the description. The improvement of Vigor over MVT [8] grows as the parsed order length increases.

Method	Order Length			overall
	1	2&3	4&5	
MVT [8]	59.4	55.0	46.7	55.1
Vigor (Ours)	61.6 (+2.2)	59.6 (+4.6)	52.0 (+5.3)	59.7 (+4.6)

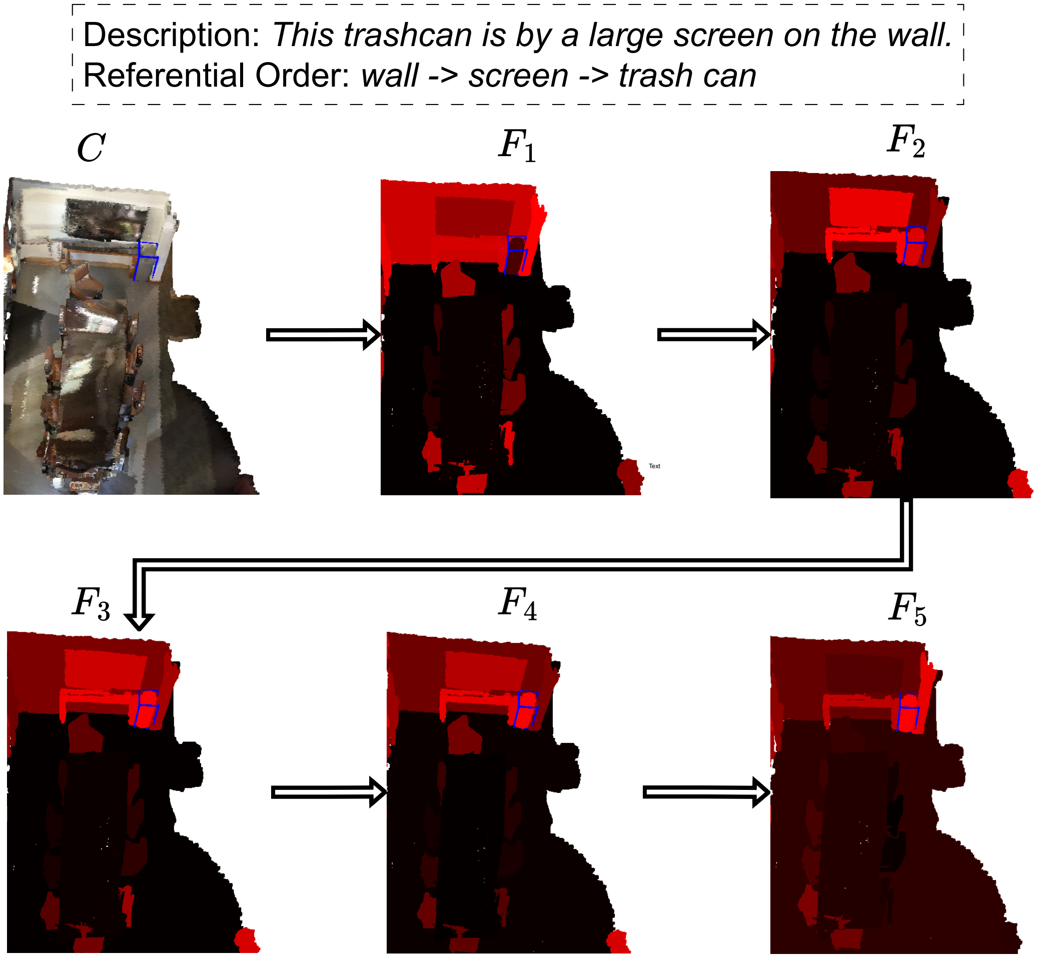

Description: *This trashcan is by a large screen on the wall.*
Referential Order: *wall -> screen -> trash can*

Figure S3. **Visualization of Responses in Each Referring Block.** An example in NR3D is shown in this figure. We color each object proposal according to their feature response in $F_{1:(B+1)}$, where a larger response with a brighter color. Note that the blue bounding box represents the ground truth target object. We can see that our Vigor progressively locates the target object by considering the referential order by first focusing on the wall then the screen and finally the trash can.

can find the correct target object without this information. If we apply the LLM directly to the original description, the model may be misled by this redundant information and generate a referential order containing the object "*sofa*".

To tackle the above problem, we conduct a two-stage in-context learning (ICL) scheme to remove redundant information in D before producing $O_{1:B}$, as depicted in Fig. S4.

With the given target object O_B, we predict a summarized description $D^{'}$ to remove redundant information in D in the first stage. Then, the entire referential order $O_{1:B}$ given $D^{'}$ and O_B is produced in the second stage. For each stage of our ICL, *10* examples are provided as demonstra-

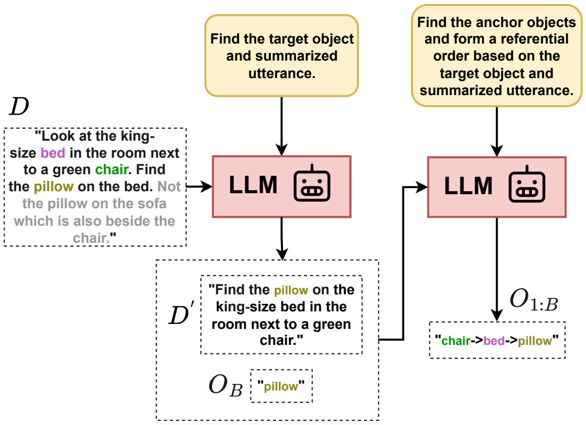

Find the target object and summarized utterance.
Find the anchor objects and form a referential order based on the target object and summarized utterance.
D
"Look at the king-size bed in the room next to a green chair. Find the pillow on the bed. Not the pillow on the sofa which is also beside the chair."
LLM
LLM
D'
"Find the pillow on the king-size bed in the room next to a green chair."
O_B
"pillow"
$O_{1:B}$
"chair->bed->pillow"

Figure S4. **Referential order generation via LLM.** A two-stage ICL is deployed to remove redundant information from D, forming a descriptive order $O_{1:B}$ for locating the target object O_B.

tions of the input prompts for LLM to predict $O_{1:B}$. Due to page limitations, demonstration examples and the complete prefix prompts are presented in the supplementary materials. With $O_{1:B}$ obtained, the order-aware object referring process can be processed accordingly, as we detail next.

We now list the complete prompts of our two-stage ICL using GPT-3.5-Turbo [19] for referential order generation. Also, we exhibit parsing results of 4 samples in the NR3D testing set and compare them with CoT3DRef [2] that also establishes the extraction and usage of referential order.

F.1. Prefix Prompt of First-Stage ICL

The first-stage prompt is used to acquire the summarized description D' and target object O_B of the original description. Our first-stage prompt is as follows:

I have some descriptions, each describing a specific target object in a room. However, they may have some redundant clauses or words. Your task is to summarize them into a shorter description. Also, tell me what the target object.

Below are 10 examples:

description 1*: Assume you are facing the door in the room. Find the larger cabinet to its left.*
summarized description 1*: When facing the door, the cabinet on the right of it.*
target object 1*: cabinet*

description 2*: The water bottle that is above the easy chair. NOT the smaller water bottle that is above the orange table.*
summarized description 2*: The water bottle that is above the easy chair.*
target object 2*: water bottle*

description 3*: In the bedroom, you will see a sheer curtain. Beside the curtain is the steel window you need to find.*
summarized description 3*: The steel window beside a sheer curtain.*
target object 3*: window*

description 4*: Please find the towel hanging on the wall in the bathroom with the other three towels. You should find the one nearest to the door. Or say it is on the door's right side.*
summarized description 4*: The towel on the wall nearest to the door.*
target object 4*: towel*

description 5*: Between a pencil and a desk lamp on the desk is the backpack you need to find.*
summarized description 5*: The backpack between a pencil and a desk lamp on the desk.*
target object 5*: backpack*

description 6*: In the living room we have three bookshelves. Choose the bookshelf to the right of the clock facing a cabinet.*
summarized description 6*: The bookshelf to the right of the clock faces a cabinet.*
target object 6*: bookshelf*

description 7*: The person wearing a white T-shirt, not the man who is also sitting on the bed but with a jacket.*
summarized description 7*: The person wearing a white T-shirt on a bed.*
target object 7*: person*

description 8*: The purple pillow on the right side of the bed when facing it. Not the one on the left side and the one in the middle of the bed.*
summarized description 8*: The purple pillow on the right side of the bed when facing it.*
target object 8*: pillow*

description 9*: The brown door at the end of the living room, next to the trash cans, which are full of garbage.*
summarized description 9*: The brown door next to the full trash can.*
target object 9*: door*

description 10*: The shoes that are placed in the middle of five shoes near the door in the room.*
summarized description 10*: The middle shoes near the door.*
target object 10*: shoes*
Now for the description [DESCRIPTION], give me the summarized description and the target object. Your answer

must be in the form "summarized description: target object:"

F.2. Prefix Prompt of Second-Stage ICL

The second-stage prompt is used to acquire the referential order $O_{1:B}$ based on the target object O_B and the summarized description D'. The complete prompt is as follows:

I have some descriptions, each describing a specific target object with some supporting anchor objects helping the localization. We can find the specific target object by tracing the referential order of anchor objects step by step. Your task is to provide a correct referential order. Also, tell me what the mentioned anchor objects.
Below are 10 examples:
***description 1**: The water bottle that is above the easy chair.*
***target object 1**: water bottle*
***anchor objects 1**: easy chair*
***referential order 1**: easy chair→water bottle*

***description 2**: The steel window beside a sheer curtain.*
***target object 2**: window*
***anchor objects 2**: curtain*
***referential order 2**: curtain→window*

***description 3**: The trash can that is on the right of the king-size bed.*
***target object 3**: trash can*
***anchor objects 3**: bed*
***referential order 3**: bed→trash can*

***description 4**: The backpack between a pencil and a desk lamp. They are all on a wooden desk.*
***target object 4**: backpack*
***anchor objects 4**: pencil, desk lamp, desk*
***referential order 4**: desk→pencil→desk lamp→backpack*

***description 5**: The cabinet on the right of the door.*
***target object 5**: cabinet*
***anchor objects 5**: door*
***referential order 5**: door→cabinet*

***description 6**: The bookshelf to the right of the clock facing a cabinet.*
***target object 6**: bookshelf*
***anchor objects 6**: clock, cabinet*
***referential order 6**: cabinet→clock→bookshelf*

***description 7**: The person wearing a white T-shirt on a bed.*
***target object 7**: person*
***anchor objects 7**: bed*
***referential order 7**: bed→person*

***description 8**: The purple pillow on the right side of the bed when facing it.*
***target object 8**: pillow*
***anchor objects 8**: bed*
***referential order 8**: bed→pillow*

***description 9**: The brown door next to the full trash can.*
***target object 9**: door*
***anchor objects 9**: trash can*
***referential order 9**: trash can→door*

***description 10**: Please find the towel hanging on the wall in the bathroom with the other three towels. You should find the one nearest to the door. Or say it is on the door's right side.*
***target object 10**: towel*
***anchor objects 10**: wall, door*
***referential order 10**: wall→door→towel*

Now for the description: [DESCRIPTION], give me the anchor objects and the referential order. Your answer must be in the form "referential order, anchor objects:. "

F.3. Examples of Derived Referential Order

To show that our two-stage ICL produces reasonable referential orders, we provide examples and comparisons between ours and CoT3DRef [2]'s parsing results. In particular, we leverage CoT3DRef's released prompt to query the GPT-3.5-Turbo. We display *4* examples in Table S5, where CoT3DRef misses an anchor object *"bed"* in the third example and includes a redundant object *"shelves"* as an anchor object in the fourth example, while our Vigor produces proper results in both cases. This shows the effectiveness of the two-stage ICL strategy.

G. Limitations and Social Impact

G.1. Limitations

G.1.1 LLM-parsed Referential Order

Since Vigor utilizes LLMs to generate the referential order from the input description, despite of our introduced pretraining strategy to warm the training process, the correctness of the extracted referential order would affect the training and the performance of Vigor. For example, Table S6 shows the accuracy on the SR3D dataset and NR3D dataset that our GPT-3.5-Turbo-parsed referential orders correctly place the target object at the last position. We can see that for the NR3D dataset, where the relations in the descriptions are much more complicated, the accuracy of the iden-

Table S5. **Examples of LLM-parsed referential order from CoT3DRef [2] and Vigor.** Note that the blue text represents the ideal anchor/target objects, and the red text represents the redundant object that should not appear in the referential order. We can see that MVT misses one anchor object in the third example and includes a redundant object in the fourth example, while Vigor predicts proper order for both cases.

Description	CoT3DRef [2]	Vigor (Ours)
The pillow closest to the foot of the bed.	bed → pillow	bed → pillow
Facing the bed, it's the large white pillow on the right. The second one from the headboard.	bed → headboard → pillow	bed → headboard → pillow
The front pillow on the bed with the laptop.	bed → pillow	laptop → bed → pillow
The window near the table, not the one near the shelves.	table → shelves → window	table → window

Table S6. **Our adopted GPT-3.5-Turbo's zero-shot accuracy (%) on identifying the class name of the target object in the referring descriptions.** Note that since the ground-truth labels and orders of the anchor objects are not available, we are only allowed to check if the target object is correctly placed at the last position in the parsed referential order as an indirect verification of the reliability of the orders.

Method	NR3D		SR3D	
	train	test	train	test
GPT-3.5-Turbo	86.9	89.1	96.4	96.1

tification of the target objects is 86.9% and 89.1% for the training and testing sets, respectively. The gap between this accuracy and an absolutely reliable prediction (i.e., 100% accuracy), though small, would still affect the training stability and testing accuracy of our visual grounding pipeline. As a result, better usage of the LLM to produce perfect referential order is one of the future research directions to pursue.

G.1.2 Order Length Decision

As detailed in Sec. A, we set the length B of referential order (as well as the number of referring blocks) as 4 to conduct batch-wise training. Although we have shown that our choice of B is reasonable for balancing computational efficiency and prediction accuracy in Sec. D by conducting proper experiments on NR3D, this choice appears to be dataset-specific. We leave this as a future direction to develop a more flexible architecture to be able to dynamically adjust B according to the LLM-parsed referential order for each sample during training and testing.

G.2. Broader Impact

G.2.1 Applications of 3D visual grounding

Although the experiments in this paper are conducted on indoor datasets only, the task of 3D visual grounding is not restricted to indoor scenes. For the autonomous driving industry, 3D visual grounding could also be an important topic to study along with 3D object detection. As our approach is designed to promote the data efficiency of 3D visual grounding tasks, we look forward to seeing future works that consider the concept of our Vigor and apply it to autonomous driving systems since it is hard to collect enormous amounts of data for autonomous driving.

G.2.2 Potential Negative Impacts

Although the experiments in this paper show that our Vigor outperforms current SOTAs in data-efficient scenarios, one must make sure that Vigor is well-validated before applying it to a new data domain. Without properly transferring Vigor's grounding ability to the corresponding domain, the performance could be non-ideal, introducing potential safety risks, especially in critical applications like autonomous driving.

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. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 422–440. Springer, 2020. 2, 3

[2] Eslam Mohamed Bakr, Mohamed Ayman, Mahmoud Ahmed, Habib Slim, and Mohamed Elhoseiny. Cot3dref: Chain-of-thoughts data-efficient 3d visual grounding. *arXiv preprint arXiv:2310.06214*, 2023. 2, 3, 4, 6, 7, 8

[3] Dave Zhenyu Chen, Angel X Chang, and Matthias Nießner. Scanrefer: 3d object localization in rgb-d scans using natural language. In *Proceedings of the European conference on computer vision (ECCV)*, pages 202–221. Springer, 2020. 2

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 4171–4186, 2019. 1

[5] Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, and Zhifang Sui. A survey for in-context learning. *arXiv preprint arXiv:2301.00234*, 2022. 4

[6] Dailan He, Yusheng Zhao, Junyu Luo, Tianrui Hui, Shaofei Huang, Aixi Zhang, and Si Liu. Transrefer3d: Entity-and-relation aware transformer for fine-grained 3d visual grounding. In *Proceedings of the 29th ACM International Conference on Multimedia (MM)*, pages 2344–2352, 2021. 2, 3

[7] Joy Hsu, Jiayuan Mao, and Jiajun Wu. Ns3d: Neuro-symbolic grounding of 3d objects and relations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2614–2623, 2023. 3

[8] Shijia Huang, Yilun Chen, Jiaya Jia, and Liwei Wang. Multi-view transformer for 3d visual grounding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15524–15533, 2022. 2, 3, 5

[9] Ayush Jain, Nikolaos Gkanatsios, Ishita Mediratta, and Katerina Fragkiadaki. Bottom up top down detection transformers for language grounding in images and point clouds. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 417–433. Springer, 2022. 1, 2, 3

[10] Li Jiang, Hengshuang Zhao, Shaoshuai Shi, Shu Liu, Chi-Wing Fu, and Jiaya Jia. Pointgroup: Dual-set point grouping for 3d instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4867–4876, 2020. 2

[11] Diederik P Kingma and Ba Jimmy. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015. 1

[12] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, 2022. 4

[13] Junyu Luo, Jiahui Fu, Xianghao Kong, Chen Gao, Haibing Ren, Hao Shen, Huaxia Xia, and Si Liu. 3d-sps: Single-stage 3d visual grounding via referred point progressive selection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16454–16463, 2022. 2

[14] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, 2019. 1

[15] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Proceedings of Advances in neural information processing systems (NeurIPS)*, volume 30, 2017. 1

[16] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, 2022. 4

[17] Li Yang, Ziqi Zhang, Zhongang Qi, Yan Xu, Wei Liu, Ying Shan, Bing Li, Weiping Yang, Peng Li, Yan Wang, et al. Exploiting contextual objects and relations for 3d visual grounding. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, 2023. 2

[18] Zhengyuan Yang, Songyang Zhang, Liwei Wang, and Jiebo Luo. Sat: 2d semantics assisted training for 3d visual grounding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1856–1866, 2021. 2, 3

[19] Junjie Ye, Xuanting Chen, Nuo Xu, Can Zu, Zekai Shao, Shichun Liu, Yuhan Cui, Zeyang Zhou, Chao Gong, Yang Shen, et al. A comprehensive capability analysis of gpt-3 and gpt-3.5 series models. *arXiv preprint arXiv:2303.10420*, 2023. 4, 6

[20] Yiming Zhang, ZeMing Gong, and Angel X Chang. Multi3drefer: Grounding text description to multiple 3d objects. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (CVPR)*, pages 15225–15236, 2023. 2

[21] Lichen Zhao, Daigang Cai, Lu Sheng, and Dong Xu. 3dvg-transformer: Relation modeling for visual grounding on point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2928–2937, 2021. 2