GroundingMate: Aiding Object Grounding for Goal-Oriented Vision-and-Language Navigation
Supplementary Materials

Qianyi Liu[1,2] Siqi Zhang[3] Yanyuan Qiao[4] Junyou Zhu[1,2] Xiang Li[1,2]
Longteng Guo[1] Qunbo Wang[1] Xingjian He[1] Qi Wu[4] Jing Liu[1,2*]
[1]Institute of Automation, Chinese Academy of Sciences
[2]School of Artificial Intelligence, University of Chinese Academy of Sciences
[3]Department of Computer Science and Technology, Tongji University
[4]Australian Institute for Machine Learning, The University of Adelaide
liuqianyi2022@ia.ac.cn, jliu@nlpr.ia.ac.cn

1. More Experiments

1.1. Ablation of MLLM

Considering the inherent difficulty in distinguishing among identically named objects from multiple perspectives and the limited capability of existing MLLMs to understand multiple images, we seek to find the optimal number of homonymous candidate objects to provide to MLLM during Stage 2. On the one hand, providing too few candidates may fail to include the target object; on the other hand, an excessive number of candidates can overwhelm the MLLM, complicating its ability to make accurate judgments amid complex visual contexts. As shown in Table 1, with a fixed confusion threshold of 0.10, the best performance is achieved when providing two candidate objects.

Table 1. Ablation study on the number of candidates for Stage 2. Candidates are the top objects with the highest probabilities predicted by the AutoVLN model based on Stage 1 predictions.

Methods	Number of Candidates	RGS↑	RGSPL↑
Baseline	-	36.58	26.76
Qwen-VL	2	**41.12**	**30.14**
	3	40.53	29.61
	4	40.56	29.62

1.2. Does the Confusion Mechanism Reflect Perplexity?

To validate the hypothesis that the confusion-based assistance mechanism actually reflects the difficulties the agent faces while locating target objects, we compared it with a probability-based mechanism. In the probability-based modal, the agent seeks assistance based on a probability p, specifically by sampling from $Bernoulli\ (p)$. If the sample yields a 1, the agent requests help; otherwise, it relies on its initial judgment. As demonstrated in Table 2, the performance of probability-based assistance is inferior to that of the confusion-based mechanism. This comparison substantiates that the degree of perplexity is a reliable indicator of the complexity associated with object grounding tasks.

[*]Corresponding author.

Table 2. The ablation study of comparing the confusion-based assistance approach with the probability-based approach.

Methods	Probability	RGS↑	RGSPL↑
Basline	-	36.58	26.76
Probability-Based	$0.3^{\dagger}$	39.39	28.72
	$0.3^{\ddagger}$	39.22	28.61
	$0.4^{\dagger}$	38.68	28.20
	$0.4^{\ddagger}$	38.74	28.35
	$0.5^{\dagger}$	38.71	28.29
	$0.5^{\ddagger}$	38.74	28.40
	$0.6^{\dagger}$	38.03	27.92
	$0.6^{\ddagger}$	37.92	27.88
	$0.7^{\dagger}$	37.80	27.66
	$0.7^{\ddagger}$	37.83	27.68
Confusion-Based (Ours)	-	**41.47**	**30.30**

1.3. Confusion Statistics

As shown in Table 3, we have compiled statistics on the confusion in object identification using the AutoVLN [1] method on the REVERIE [3] dataset. Consistent with the setting of this study, we only included instances where the agent successfully navigated to the correct location, as defined by the REVERIE dataset where Remote Grounding Success is achieved when the target object is within a 3-meter radius at the endpoint **and** the correct object is se-

Table 3. Statistical analysis of confusion in object selection at the navigation endpoint by agents using the AutoVLN [1] model on the REVERIE [3] dataset, restricted to cases of successful navigation.

Splits	Result	Number	Entropy	Top-2 Probability Gap
Val Train Seen	Success	121	0.1103	0.9106
	Failure	2	1.2188	0.3398
	Total	123	0.1284	0.9013
Val Seen	Success	689	0.1427	0.8927
	Failure	236	0.8362	0.4159
	Total	925	0.3197	0.7710
Val Unseen	Success	1288	0.2313	0.8024
	Failure	680	0.6825	0.4981
	Total	1968	0.3872	0.7090

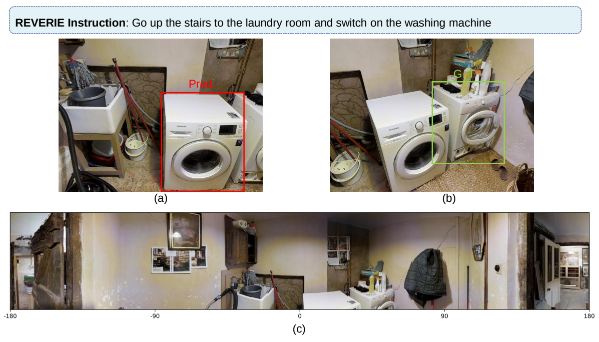
REVERIE Instruction: Go up the stairs to the laundry room and switch on the washing machine
Pred
GT
(a)
(b)
-180
-90
0
90
180
(c)

Figure 1. Schematic representation of the annotation interface. For each erroneous example, the display includes (a) the object predicted by the model, (b) the object annotated as Ground Truth, and (c) the panoramic image information visible to the agent at the navigation endpoint.

lected. We measured the confusion performance in both successful and unsuccessful object identification, as well as overall cases across Val Train Seen, Val Seen, and Val Unseen subsets. We employed two metrics to quantify the model's confusion: the entropy-based confusion, which reflects the uncertainty in object selection, with higher values indicating greater confusion, and the Top-2 Probability Gap, defined as the difference between the probabilities of the two most likely objects predicted by the model [4], where larger values suggest greater certainty in the model's choice. Our observations confirm two phenomena: first, confusion levels are lower in cases of successful object identification, indicating greater certainty when the agent correctly selects an object; second, confusion levels are generally lower in familiar environments (seen during training) compared to unfamiliar ones (not encountered during training), indicating that in novel settings, agents typically exhibit a lack of confidence in their own judgments. Both observations align with our intuitive expectations.

Table 4. Error type statics of the AutoVLN model on the REVERIE dataset Val Unseen split. ① represents attribute errors, ② represents location errors, ③ represents errors due to ambiguous instructions, and ④ represents Ground Truth Labeling Errors.

Error Type	**Class Type Error**	**Discriminating error**			
		①	②	③	④
Number	332	54	258	28	8
		348			

1.4. Errors of the AutoVLN model

We conduct a manual analysis of error types in the AutoVLN model within the REVERIE Val Unseen dataset, specifically focusing on instances where navigation is successful but object identification is incorrect. This analysis aligns with the data recorded in Table 4, encompassing a total of 680 cases. By acquiring panoramic images from the navigational endpoints, and comparing the model-predicted object images against the Ground Truth (G.T.) images, we manually classify all erroneous instances as depicted in Figure 1.

Error types are categorized into two primary groups: class type errors, where the predicted object's name differs from the G.T. object (e.g., a pillow on a sofa misidentified as a flower pot beside it), and discriminating errors, where the predicted and actual objects are of the same type but the agent errs in distinguishing between homonymous objects in the scene. Specifically, discrimination errors are subdivided into four categories: ① Attribute errors, where the chosen object does not match the attributes described in the instructions (such as the color of a pillow or the content of a painting), typically do not involve spatial relationships with other objects. ② Positional errors, often involve relationships with other objects, such as the largest flower pot, the chair closest to the door, or the second chair from the left. The first two categories generally result from the agent's failure to accurately judge the object, while the last two categories are more often related to issues with dataset annotations. ③ Overly vague instructions: when the instructions are too broad, leading to multiple objects fitting the criteria within the scene, yet the G.T. annotation only marks one (e.g., when instructed to retrieve a book from a stack, and multiple books are present, choosing any book should be deemed correct). ④ G.T. annotation errors, such as a closed television screen (reflecting a painting on the opposite wall) mistakenly annotated as a painting on the wall. These error types are systematically cataloged in Table 4. From the table, it is evident that the number of Discrimination Errors slightly exceeds that of Class Type Errors. Within Discrimination Errors, Positional Errors constitute the majority. This indicates that Attribute Errors and Position-Related

Discrimination Errors are the primary reasons preventing the agent from accurately selecting the correct object.

1.5. Comparision with ChatGPT's Performance

We compare our method with the direct utilization of ChatGPT [2] for judgment. We evaluate the performance when only the names of objects are entered as input versus when detailed descriptions of objects are provided. Considering the rate limitations of GPT-4 with image inputs, we limit our testing to the performance of GPT-3.5. We employ the GPT4ROI [5] model to generate detailed textual descriptions for each object image in the REVERIE dataset. This model supports input in the form of bounding box descriptions of objects, producing textual descriptions of these objects, which are then used as supplementary information for input into ChatGPT. It should be noted that this method may result in information loss and the introduction of inaccuracies in the descriptions, making this comparison somewhat challenging to ensure complete fairness. We utilize the AutoVLN model, setting all confusion scores θ to 0.25, as shown in Table 5. Our method outperformed ChatGPT's results in all cases except within the Train Seen dataset.

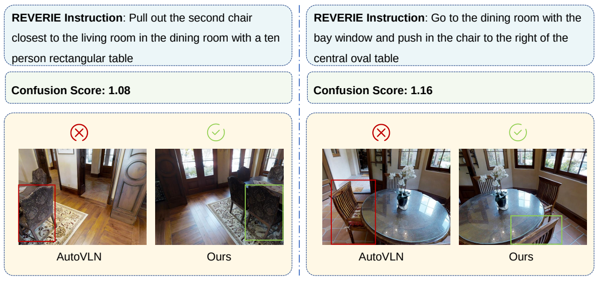
REVERIE Instruction: Pull out the second chair closest to the living room in the dining room with a ten person rectangular table
Confusion Score: 1.08
AutoVLN
Ours
REVERIE Instruction: Go to the dining room with the bay window and push in the chair to the right of the central oval table
Confusion Score: 1.16
AutoVLN
Ours

Figure 2. Comparison between our method and AutoVLN. Our method effectively assists the agent in locating the target object specified in the instructions when confusion arises during object grounding.

Table 5. Comparison of the performance of our method with ChatGPT approach on the REVERIE Train Seen, Val Seen, and Val Unseen datasets. "ChatGPT" represents the scenario where only the names of objects are input, while"ChatGPT ♠" indicates that both the names and descriptive information of the objects are provided.

Method	REVERIE Train Seen		REVERIE Val Seen		REVERIE Val Unseen	
	RGS	RGSPL	RGS	RGSPL	RGS	RGSPL
Baseline	**98.37**	**92.97**	48.42	41.67	36.58	26.76
ChatGPT	98.37	92.97	49.61	42.65	39.62	28.93
ChatGPT♠	95.12	90.19	48.21	41.37	39.88	29.03
Ours	97.56	92.15	**49.75**	**42.81**	**40.59**	**29.60**

1.6. Comparison of Different Confusion Mechanisms

As shown in Table 6, we tested the performance and the number of help requests of the confusion mechanism based on the Top-2 probability gap under different thresholds. Compared to the confusion mechanism based on cross-entropy, both mechanisms perform similarly when the number of help requests is equivalent, indicating that the two confusion mechanisms are similarly effective in reflecting the agent's confusion when searching for the target object mentioned in the instructions.

Table 6. Ablation of different confusion thresholds θ using the confuison mechanism based on Top-2 Probability Gap.

Stages	θ	RGS↑	RGSPL↑	Amount of Assistance
Baseline	-	36.58	26.76	0
Stage 1	0.50	38.54	28.17	577
	0.60	38.82	28.03	656
	0.70	39.14	28.59	741
	0.80	39.65	28.91	818
	0.90	40.22	29.34	920
Stage 2	0.50	39.08	28.59	397
	0.60	39.59	29.03	443
	0.70	39.76	29.11	495
	0.80	40.33	29.41	536
	0.90	**40.90**	**29.89**	577

1.7. Computation Time

As shown in Table 7, we conducted experiments on the REVERIE Val Unseen dataset using a single A6000 GPU. We listed the computation times of different methods at various stages with different confusion thresholds for MLLM, which range from 0.2 to 0.6 seconds. Given that agents will only seek assistance from the MLLM when they experience confusion, the design of the confusion metric can effectively balance inference speed and system performance.

Table 7. The average computation time on REVERIE.

Methods	θ	Stage	Average Time	Methods	θ	Stage	Average Time
AutoVLN+Ours	0.25	1	304 ms	GridMM+Ours	0.25	1	220 ms
		2	550 ms			2	554 ms
	0.20	1	333 ms		0.20	1	279 ms
		2	551 ms			2	542 ms

2. Qualitative Results

We present some examples of failures in object identification by the AutoVLN method on the REVERIE Val Unseen datasets. In the examples shown in Figure 2, the issues relate to distinguishing objects with the same name, which requires the integration of multiple images from different perspectives and is quite challenging. The agent becomes confused when trying to locate the chair described in the in-

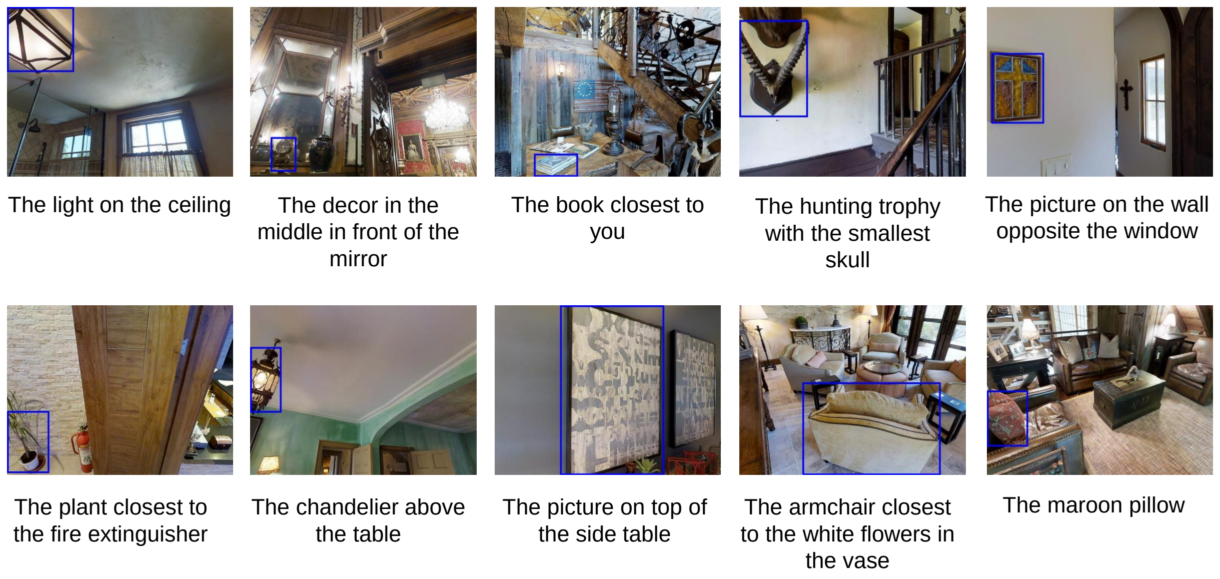
The light on the ceiling
The decor in the middle in front of the mirror
The book closest to you
The hunting trophy with the smallest skull
The picture on the wall opposite the window
The plant closest to the fire extinguisher
The chandelier above the table
The picture on top of the side table
The armchair closest to the white flowers in the vase
The maroon pillow

Figure 3. Examples of REVERIE-OG dataset.

structions, and this confusion significantly exceeds our preset threshold. In our approach, the agent eventually succeeds in finding the correct object, vividly demonstrating the effectiveness of our method.

3. Discussions

3.1. Reproducibility

For ChatGPT, by adding example prompts during invocation, we were able to help it better understand user intent and generate the expected output. In repeated experiments with the default parameter settings, we found that object description information was completely consistent 79.04% of the time, while object name information was fully consistent 92.38% of the time. In most cases where the object description and name were inconsistent, the differences were due to variations in the level of detail in the description rather than differences in key information such as object category. For example, “the picture of flowers closest to the kitchen” versus “the picture of flowers.” Utilizing different extraction results from ChatGPT can cause fluctuations in our method’s RGSPL of ± 0.03. For Qwen-VL, setting a random seed ensures fully consistent outputs in the same environment, and repeated experiments confirmed that this does not impact performance.

3.2. Stability

We utilized in-context learning by providing several examples that conform to the output format when querying MLLM. We used regular expressions to extract other similar expressions, such as ”O (o) bject A.” or ”O (o) ption A.” Experimental statistics revealed that 98.56% of the model outputs adhere to the set format.

3.3. Error Analysis

3.3.1 LLM Errors

For LLM, we manually analyzed 261 instructions across 100 paths and found that in most cases (93.87%), LLMs correctly extracted the required information. However, there were 6.13% cases where errors occurred. These errors can be categorized into two main types: ① extraction of objects describing the destination rather than the target object. For instance, in the instruction “ Go to the toilet room on level 2 that has *knitted figures* located on top of a black table that’s next to the white toilet and observe the quality of the *light*”, ChatGPT misinterpreted the target object, extracting ”knitted figures” as the target instead of ”light”. This type of error accounted for 3.45%, mainly due to the model’s failure to accurately identify the verb-object structure describing the target object; ② extraction of objects subordinate to the target object. For example, in the instruction “Go to the kitchen and clean the counter with the scale”, ChatGPT incorrectly extracted “scale” as the target object, while the correct answer was “counter”. This type of error accounted for 2.69%.

3.3.2 MLLM Errors

For MLLM, we found that nearly all errors occurred in the disambiguation of objects with similar names. These errors can be categorized into: ① incorrect judgments of spatial relationship between objects, such as "the picture that is further from the crystal door"; ② errors in interpreting object location information, such as "the picture furthest to the left"; ③ incorrect understanding of object attributes, such as the color of a pillow or the content of a painting; ④ errors in understanding comparative relationships between objects, such as "the largest vase"; ⑤ cases where the instruction itself lacked sufficient descriptive information, rather than the MLLM making an error. Upon analysis, these five error types accounted for 48%, 29%, 15%, 4%, and 4% of the total errors, respectively.

3.4. Circumustance of Target Object Absence

When the agent chooses to seek assistance from the MLLM but there is no target object in the final scene, the MLLM may respond in several ways: (1) it might select an object of the same name (or category) from the environment. Through manual analysis of erroneous examples, we found that even if the agent fails to navigate correctly, it often stops in a scene containing an object of the same category. (2) it may reject the query and output that the target object is absent, in which case we maintain the agent's original judgment; (3) it may select objects that frequently co-occur with the target object in daily life, such as "drawer" and "shelf"; or (4) it may choose an incorrect object unrelated to the target.

3.5. Confidence Threshold in Potential Real-world Deployments.

When considering possible future deployments in real-world scenarios, using a perplexity threshold based on an entropy mechanism may be inconvenient. Therefore, we explored a method that uses the probability difference between the two most confident objects predicted by the agent as the threshold, as shown in Table 3 and Table 6. This method is simple to compute, less affected by different environments, and more stable, and its performance is only slightly lower than that of the entropy-based search method. By observing the perplexity of a small number of samples in the real world, this method allows for better calibration of the perplexity threshold and enables users to dynamically adjust it to balance speed and performance. Thus, this perplexity threshold setting method is more practical for real-world applications.

Although existing VLN research has advanced rapidly, there remains a significant gap between its performance and practical deployment. As a result, current studies primarily focus on exploring ways to further enhance performance within simulators, and the Sim2Real transition in VLN tasks still offers substantial room for exploration.

4. Datasets

REVERIE It requires an intelligent agent to localize a remote target object using concise high-level natural language instructions that replicate real-world scenarios. The dataset includes high-level instructions averaging 21 words, emphasizing target locations and objects. With predefined object bounding boxes for each panorama, the agent must accurately select the correct one by the end of the navigation path, which typically consists of 4 to 7 steps.

SOON The instructions averagely comprise 47 words, typically containing attributes such as color and material and spatial relationships with other objects. Unlike REVERIE, which furnishes object bounding boxes for agent selection, SOON asks agents to predict the polar coordinates of the object's bounding box center, incorporating heading and elevation. In alignment with the AutoVLN framework, we utilize the Mask2Former detector to derive the bounding box of objects. Path lengths in SOON vary from 2 to 21 steps, with an average of 9.5 steps.

REVERIE-OG Consistent with REVERIE's dataset partitioning, we divide our datasets into train, val_train_seen, val_seen, and val_unseen, comprising 4065, 49, 515, and 1328 image_text pairs, respectively. As illustrated in Figure 3, we present several examples from the REVERIE-OG dataset, each consisting of natural language descriptions of objects along with annotated bounding boxes, which are highlighted in blue in the images. Consistent with the settings described earlier in this paper, the natural language descriptions of the objects are generated by ChatGPT.

References

[1] Shizhe Chen, Pierre-Louis Guhur, Makarand Tapaswi, Cordelia Schmid, and Ivan Laptev. Learning from unlabeled 3d environments for vision-and-language navigation. In *ECCV*, pages 638–655, 2022. 1, 2

[2] OpenAI. Chatgpt. `https://openai.com/blog/chatgpt/`, 2023. 3

[3] Yuankai Qi, Qi Wu, Peter Anderson, Xin Wang, William Yang Wang, Chunhua Shen, and Anton van den Hengel. REVERIE: remote embodied visual referring expression in real indoor environments. In *CVPR*, pages 9979–9988. IEEE, 2020. 1, 2

[4] Kunal Pratap Singh, Luca Weihs, Alvaro Herrasti, Jonghyun Choi, Aniruddha Kembhavi, and Roozbeh Mottaghi. Ask4help: Learning to leverage an expert for embodied tasks. *Advances in Neural Information Processing Systems*, 35:16221–16232, 2022. 2

[5] Shilong Zhang, Peize Sun, Shoufa Chen, Min Xiao, Wenqi Shao, Wenwei Zhang, Kai Chen, and Ping Luo. Gpt4roi: Instruction tuning large language model on region-of-interest. *arXiv preprint arXiv:2307.03601*, 2023. 3