LIRA: Reasoning Reconstruction via Multimodal Large Language Models

Supplementary Material

Contents

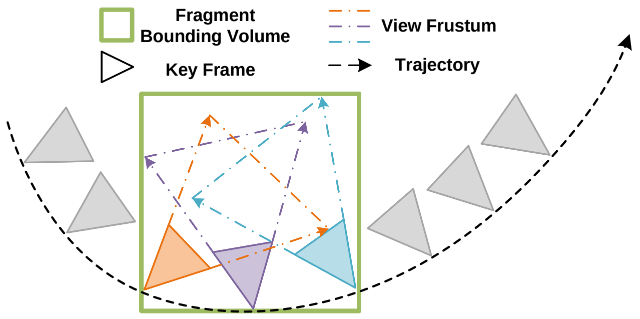

Figure 1. Illustration for the definition of FBV.

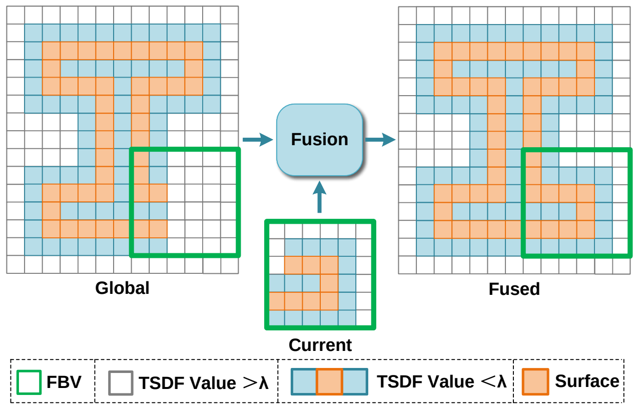

Figure 2. 2D toy examples to illustrate the fusion process between current information and global information during geometric reconstruction and instance fusion.

1. Visualization of FBV

Inspired by [2, 27], we provide the definition of fragment bounding volume (FBV) in Fig. 1. The geometric reconstruction process, 2D reasoning segmentation process, and instance fusion process of LIRA are mainly conducted within each FBV.

In addition, Fig. 2 provides an illustration of the fusion process between current information and global information during geometric reconstruction and instance fusion. During each fusion process, only the global information within the FBV is updated.

2. Queryable Map-Based Methods VS LIRA

Fig. 3 provides a visual comparison between queryable map-based methods and the proposed LIRA. Existing methods [7, 9, 14, 23] mainly rely on explicit instructions and queryable maps to achieve instruction-guided online 3D reconstruction. They first perform geometric reconstruction and extract all instance features and masks. Then, instructions are matched with the pre-defined maps. To enhance comprehension, some approaches [5, 7, 12] employ large language models (LLMs) for pre-processing or post-processing. However, these methods contain much instruction-irrelevant information, and exhibit limited interaction and reasoning between target instance features and instruction features. Particularly for implicit instructions involving complex reasoning, they are more difficult to handle.

Compared with methods using queryable maps, LIRA achieves better reasoning reconstruction performance by leveraging a multimodal LLM (MLLM) to perform comprehensive vision-instruction fusion and reasoning, and reason about instances that are only relevant to implicit instructions.

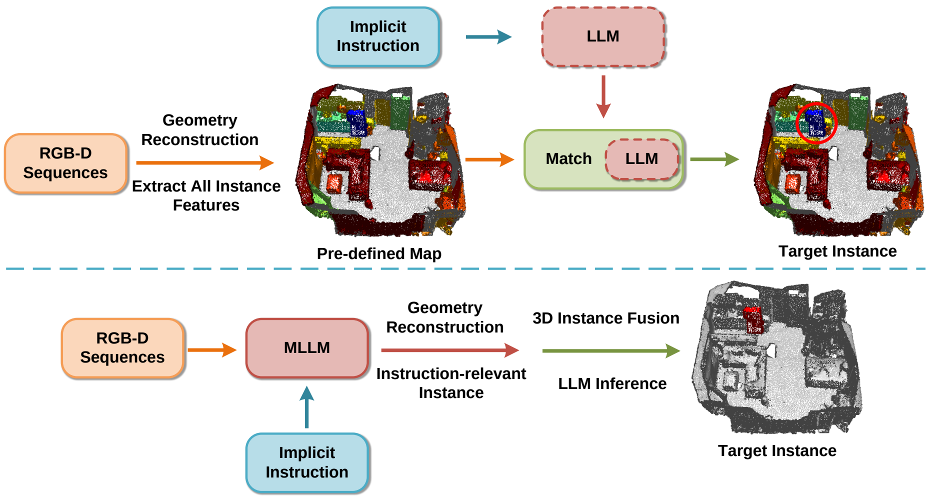

Figure 3. Visual comparison between queryable map-based methods (the part above the blue dashed line) and the proposed LIRA (the part below the blue dashed line). The red dashed boxes are optional modules.

3. Benchmark

3.1. Scene-Instruction Pair Generation

Based on instance attributes in a scene, which include instance segmentation annotations (e.g., class, bounding box, and mask) and attributes inferred by Qwen2-VL [20] (e.g., color), we utilize ChatGPT-4o [16] to design a series of instruction templates and generate scene-instruction pairs. These attributes are the most frequently occurring attributes in human language expressions [1].

We first task ChatGPT-4o with generating a set of base instruction templates based on the above attributes, as illustrated in Fig. 10 (see the end of this paper). The generation of these base instruction templates primarily leverages attributes such as color, class, size, and 3D spatial position. By permuting and combining these attributes, ChatGPT-4o produces a rich and diverse array of instruction templates. Fig. 11 provides some examples of base instruction templates.

Second, ChatGPT-4o generates implicit instruction templates based on the attributes of objects in the scene, as described in Fig. 12. The generation process of these implicit instruction templates mainly relies on the classes of the objects. The generated implicit instructions are required to provide detailed descriptions of the target attributes without explicitly mentioning the attribute names. Furthermore, the instructions should be as diverse as possible. Fig. 13 presents several examples of implicit instruction templates.

Finally, ChatGPT-4o randomly selects from the base instruction template library and the implicit instruction template library to generate a series of language instruction templates for a given specific scene. A subset of objects and their attributes are randomly selected and inserted into the chosen instruction templates to generate specific scene-instruction pairs. The prompt for generating scene-instruction pairs is shown in Fig. 14. The generated instructions are further polished by ChatGPT-4o to make them more consistent with human language expressions. We also enhance the quality of these generated instructions by randomly sampling them for human corrections. Fig. 15 provides several examples of the final generated instructions for specific scenes.

For the reasoning reconstruction task, we integrate corresponding segmentation masks (both 3D and 2D) with scene-instruction pairs to generate scene-instruction-mask data pairs.

3.2. Removal of Erroneous Projected Pixels

In the data collection pipeline of the benchmark, we first extend the attributes of instances in the ScanNetV2 dataset [3]. For an instance in a given scene, its point cloud is projected into an image from a certain viewpoint. We filter out erroneous projected pixels caused by occlusion. Specifically, to eliminate erroneous projected pixels caused by occlusion, we also project other instances in the scene into the same image. If a projected pixel of the target instance is close to other instances in the image and the target instance is farther from the camera center than the other instances, it is considered occluded and should be removed.

3.3. 2D Segmentation Annotations

The ScanNetV2 dataset provides instance segmentation annotations for images. However, the quality of the 2D segmentation annotations is poor, making them unsuitable as supervisory information for model training. Our data production pipeline automatically generates 2D segmentation

	ReasonRecon	**ReasonRecon-Extension**
Number of scene-instruction pairs	12,500	14,800
Number of scenes	125	1,513
Number of implicit instructions	6,155	7,333
Number of targets	14,718	29,819
Number of annotated images	201,336	2,477,378
Number of Multi-class answers	430	317
Number of multi-target answers	1,698	4,184
Number of zero-target answers	776	993
Number of instructions describing spatial relations	4,642	5,422
Number of instructions per scene	100	9.8
Average length of instructions	12.51	12.69

Table 1. Statistics for ReasonRecon and ReasonRecon-Extension.

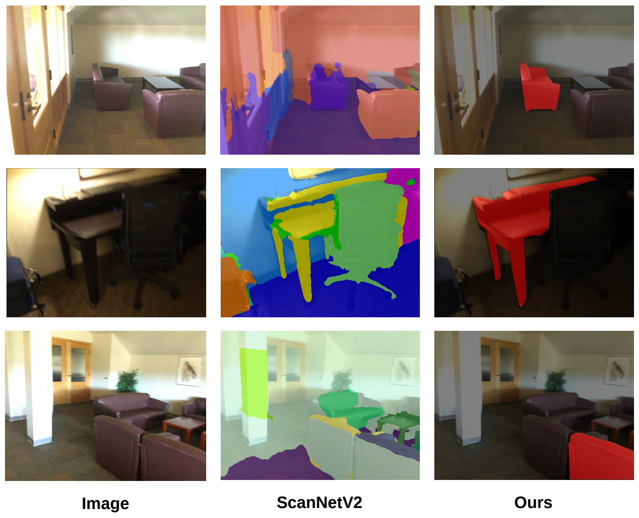

Figure 4. Visual comparison of the image segmentation annotations provided by the ScanNetV2 dataset and those provided by our benchmark. In ReasonRecon, the instruction-relevant instances in a scene-instruction pair are visualized.

annotations, the quality of which is close to that of manually annotated segmentation masks. Fig. 4 shows a comparison of the image segmentation annotations provided by the ScanNetV2 dataset and those provided by our benchmark. In ReasonRecon, the instruction-relevant instances in a scene-instruction pair are visualized.

3.4. Statistics of ReasonRecon

To construct ReasonRecon, we meticulously select a set of frequently occurring scenes from the ScanNetV2 dataset. These scenes are commonly seen in daily life and contain a rich diversity of objects. For each scene, we generate a rich set of instructions, thereby forming scene-instruction data pairs. The detailed data statistics of ReasonRecon are shown in Tab. 1. We select 125 scenes from the ScanNetV2 dataset and generate 100 instructions for each scene, resulting in a total of 12,500 scene-instruction pairs. In addition, ReasonRecon has 6,155 implicit instructions, 201,336 2D annotated images, and 4,642 instructions describing spatial relations (e.g., determining size and distance relationships).

3.5. ReasonRecon-Extension Benchmark

In addition to *ReasonRecon*, we also propose its extended version — *ReasonRecon-Extension*. The data production pipeline of ReasonRecon-Extension is the same as that of ReasonRecon. ReasonRecon-Extension extends ReasonRecon with low-frequency scenes, including all the scenes in the ScanNetV2 dataset. In addition, in consideration of computational and storage resource constraints, relatively few instructions are constructed for each scene. The detailed data statistics of ReasonRecon-Extension are shown in Tab. 1. ReasonRecon-Extension has 1,513 scenes, each with an average of 9.8 instructions, for a total of 14,800 scene-instruction data pairs. In addition, ReasonRecon-Extension has 7,333 implicit instructions, 2,477,378 2D annotated images, and 5,422 instructions describing spatial relations. The training set and test set are also divided into 8: 2.

4. Loss Function

The training of LIRA is divided into two steps: 2D reasoning segmentation and instance fusion. The loss functions are also divided into two parts.

For 2D reasoning segmentation, this module is trained using a combination of the text generation loss $\mathcal{L}_{txt}$ and the segmentation mask loss $\mathcal{L}_{mask}$. The total loss $\mathcal{L}_{rs}$ of the 2D reasoning segmentation module is the weighted sum of $\mathcal{L}_{txt}$ and $\mathcal{L}_{mask}$. λ_{txt} and λ_{mask} are weight coefficients.

$$\mathcal{L}_{rs} = \lambda_{txt}\mathcal{L}_{txt} + \lambda_{mask}\mathcal{L}_{mask}. \tag{1}$$

$\mathcal{L}_{txt}$ is the auto-regressive cross-entropy (CE) loss for text generation, and $\mathcal{L}_{mask}$ is the mask loss, which encourages the model to produce high-quality segmentation results. To compute $\mathcal{L}_{mask}$, we use a linear combination of per-pixel binary cross-entropy (BCE) loss and DICE loss [15], with corresponding loss weights λ_{bce} and λ_{dice}. Given the supervision targets r_{txt}^{gt} and m_{seg}^{gt}, these losses are formulated as

$$\mathcal{L}_{txt} = \text{CE}(r_{txt}^{gt}, r_{txt}). \tag{2}$$

$$\begin{aligned}\mathcal{L}_{mask} &= \lambda_{bce}\text{BCE}(m_{seg}^{gt}, m_{seg}) \\ &+ \lambda_{dice}\text{DICE}(m_{seg}^{gt}, m_{seg}).\end{aligned} \tag{3}$$

For instance fusion, TIFF is trained using a linear combination of the mask confidence loss $\mathcal{L}_{conf}$ and the similarity loss $\mathcal{L}_{sim}$. The total loss $\mathcal{L}_{if}$ of TIFF is the weighted sum of $\mathcal{L}_{conf}$ and $\mathcal{L}_{sim}$. λ_{if} is the weight coefficient.

$$\mathcal{L}_{if} = \lambda_{if}\mathcal{L}_{conf} + (1-\lambda_{if})\mathcal{L}_{sim}. \tag{4}$$

Specifically, $\mathcal{L}_{conf}$ is the mean squared error (MSE) loss for mask confidence prediction, and $\mathcal{L}_{sim}$ is the per-element BCE loss for similarity matrix generation. Given the supervision targets $\mathcal{S}_{conf}^{gt}$ and $\mathcal{S}^{gt}$, these losses are formulated as

$$\mathcal{L}_{conf} = \text{MSE}(\mathcal{S}_{conf}^{gt}, \mathcal{S}_{conf}). \tag{5}$$

$$\mathcal{L}_{sim} = \text{BCE}(\mathcal{S}^{gt}, \mathcal{S}). \tag{6}$$

5. More Implementation Details

For the training strategy, we first train the 2D reasoning segmentation module for 10 epochs. Then, we freeze the network weights in the 2D reasoning segmentation module, and train TIFF for 20 epochs. The entire network is trained on 8 NVIDIA Tesla A800 GPUs for 15 days. TSDF truncation distance is 12cm, and $d_{\max}$ is set to 3m. $R_{\max}$ and $t_{\max}$ are set to 15° and 0.1m, respectively. The training scripts for the 2D reasoning segmentation module are based on deepspeed [18] engine. The channels of $\mathcal{F}_{\text{proj}}$ are [256, 4096, 4096]. In the instance fusion module, θ_{conf} and θ_{sim} are set to 0.6 and 0.8, respectively.

For the training of the 2D reasoning segmentation module, we use AdamW optimizer with the learning rate and weight decay set to 0.0003 and 0.0, respectively. We also adopt WarmupDecayLR as the learning rate scheduler, where the warmup iterations are set to 100. The weights of the text generation loss λ_{txt} and the mask loss λ_{mask} are set to 1.0 and 1.0, respectively, and those of the bce loss λ_{bce} and the dice loss λ_{dice} are set to 2.0 and 0.5, respectively. Besides, the gradient accumulation step is set to 10. The rank of LoRA [6] is 8.

For the training of TIFF, we use Adam optimizer with the learning rate and weight decay set to 0.001 and 0.0, respectively. We also adopt MultiStepLR as the learning rate scheduler. The learning rate is reduced by half in the 6th, 8th, and 10th epochs. The loss weight λ_{if} of the instance fusion module is 0.5.

Method	**Online**	**AP**	**AP$_{50}$**	**AP$_{25}$**
OpenScene [17] + DBSCAN [4]	✗	2.60	4.42	6.88
OpenScene [17] + Mask3D [19]	✗	3.55	6.35	8.77
OpenIns3D [8] (ODISE [22])	✗	3.26	6.40	8.37
VLMaps-3D [7]	✓	3.92	8.01	9.59
OVIR-3D [14]	✓	4.60	8.93	11.92
MaskClustering [24]	✓	4.87	9.43	13.16
LIRA	✓	**5.13**	**16.69**	**48.68**

Table 2. Quantitative results of reasoning reconstruction on ReasonRecon-Extension.

6. Supplementary Experiments

6.1. Results on ReasonRecon-Extension

We also conduct reasoning reconstruction experiments on ReasonRecon-Extension. As shown in Tab. 2, consistent with the experimental observations on ReasonRecon, LIRA demonstrates superior performance across all accuracy metrics compared to existing online and offline methods. Compared to methods based mainly on explicit instructions or queryable maps, LIRA exhibits superior reasoning and comprehension capabilities for implicit and complex instructions.

6.2. Impact of Dataset Size on Reasoning Reconstruction Performance

We further analyze the impact of dataset size on reasoning reconstruction performance. First, we train two models, using the ReasonRecon training set and the union of the ReasonRecon and ReasonRecon-Extension training sets, respectively. Both models are then evaluated on the ReasonRecon test set. We filter the training data to ensure that the scenes in the test data do not appear in the training data. As shown in Tab. 3, increasing the size of training data further improves the reasoning reconstruction performance.

Similarly, we train two additional models. Their training data consists of the ReasonRecon-Extension training set and the union of the ReasonRecon and ReasonRecon-Extension training sets, respectively. These models are then evaluated on the ReasonRecon-Extension test set. Increasing the training data size also results in further improvements in reasoning reconstruction performance. The above experiments show that the reasoning reconstruction performance of the proposed LIRA is enhanced as the dataset size increases, demonstrating the potential scaling capability of LIRA in terms of dataset size.

Moreover, we test the generalization capability of LIRA. Since LIRA performs reasoning reconstruction by reasoning over vision information and implicit instructions based on the MLLM, we evaluate its generalization capability in unseen scenes. LIRA is trained on scene-instruction pairs from ReasonRecon, where the scenes are selected from fre-

Training Set	Test Set	AP	AP$_{50}$	AP$_{25}$
ReasonRecon	ReasonRecon	11.57	34.39	66.24
ReasonRecon + ReasonRecon-Extension	ReasonRecon	12.81(+1.24)	34.52(+0.13)	67.30(+1.06)
ReasonRecon-Extension	ReasonRecon-Extension	5.13	16.69	48.68
ReasonRecon + ReasonRecon-Extension	ReasonRecon-Extension	5.87(+0.74)	17.36(+0.67)	49.17(+0.49)

Table 3. Impact of dataset size on reasoning reconstruction performance.

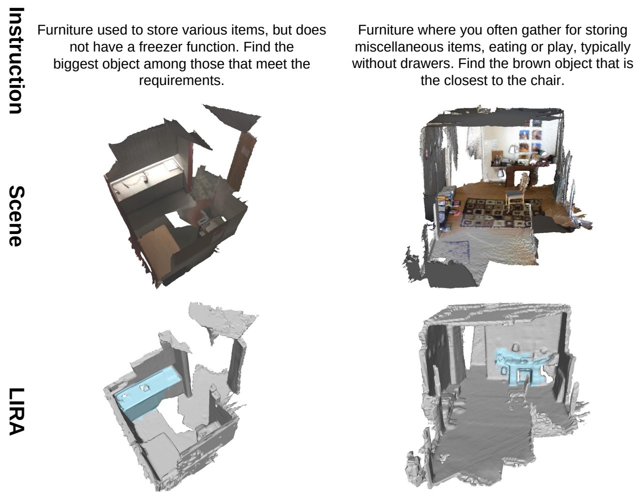
Instruction
Scene
LIRA
Furniture used to store various items, but does not have a freezer function. Find the biggest object among those that meet the requirements.
Furniture where you often gather for storing miscellaneous items, eating or play, typically without drawers. Find the brown object that is the closest to the chair.

Figure 5. Test of generalization capability of LIRA.

quently occurring scenes in the ScanNetV2 dataset. Therefore, we directly test the model trained on ReasonRecon in unseen scenes which are low-frequency scenes from the ScanNetV2 dataset. As shown in Fig. 5, LIRA demonstrates a certain level of generalization capability in some unseen scenes.

6.3. 2D Reasoning Segmentation Results

2D reasoning segmentation is crucial as LIRA primarily relies on this module to reason and comprehend implicit and complex instructions. We also evaluate the performance of our method compared to other segmentation approaches in 2D reasoning segmentation task in Tab. 4. The 2D reasoning segmentation task takes as input an image and a language instruction involving complex reasoning, and outputs the corresponding text response and segmentation mask for each instance. This task supports multiple types of output formats, including zero, one, or multiple instances. The data for the 2D reasoning segmentation task is sourced from the ReasonRecon benchmark, which is designed for the reasoning reconstruction task.

Unless otherwise specified, all methods utilizing LLMs default to using a model of the 7B scale. Inspired by [11, 21], we adopt gIoU and cIoU metrics to represent the segmentation accuracy of masks. gIoU averages the Intersection over Union (IoU) for each mask, whereas cIoU computes the cumulative intersection area over the cumulative union area across the whole dataset. Following the implementation in [21], we compute the No-target-accuracy (N-acc.) for the empty target, which is the ratio of the correctly classified empty-target expressions over all the empty-target expressions in the dataset. In addition, the BLEU (Bilingual Evaluation Understudy) score is used to evaluate the accuracy of text outputs.

Method	gIoU	cIoU	N-acc.	BLEU
SEEM [28]	16.77	14.08	34.01	-
Grounded-SAM [13]	15.56	11.13	30.82	-
LISA [11] (ft)	24.39	26.68	-	-
LISA++ [25] (ft)	60.85	60.33	95.37	88.24
GSVA [21] (ft)	59.07	58.24	95.05	87.89
LLaVA-Grounding [26] (ft)	61.27	61.90	96.54	89.74
Ours	61.53	61.61	96.78	90.35
Ours-13B	**62.71**	**63.31**	**97.08**	**90.79**

Table 4. Quantitative results of 2D reasoning segmentation. "ft" denotes the model is finetuned on the training set.

Experimental results demonstrate that our method achieves superior performance compared to existing methods in terms of mask generation, instance presence detection, and text response. Compared with LISA [11], LISA++ [25] and GSVA [21], which have a similar architecture to our method, our method achieves better 2D reasoning segmentation accuracy. This improvement is primarily attributed to our more effective handling of multi-target and zero-target outputs.

6.4. MDS Visualization of Instance Fusion

The proposed TIFF aggregates the text features, voxel features, and 3D bounding box features of instances to generate features $\mathcal{Q}^{sim}$ for similarity matrix computation between instances. We visualize $\mathcal{Q}^{sim}$ using the multidimensional scaling (MDS) method. The visualization results in Fig. 6 validate that the proposed TIFF successfully learns discriminative feature representation for object matching.

6.5. Impact of Mask Confidence

In TIFF, the mask confidence branch predicts a confidence score $\mathcal{S}_{conf}$ for each candidate instance. If the confidence score is lower than a threshold θ_{conf}, the corresponding

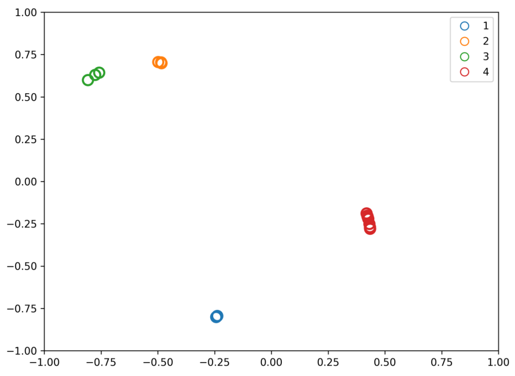

Figure 6. MDS visualization of the instance-specific representation for similarity calculation. Different colors indicate different instances and different points indicate the instance features at different keyframes.

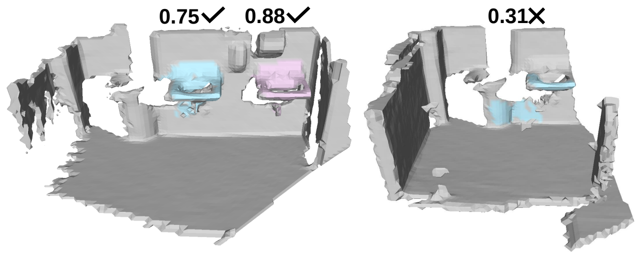

Figure 7. A visual example of the mask confidence branch prediction results. When θ_{conf} is set to 0.6, masks with scores exceeding θ_{conf} are retained (✓), while those with scores below θ_{conf} are discarded (✗).

candidate instance is discarded. Fig. 7 provides a visual example of the mask confidence branch prediction results. When θ_{conf} is set to 0.6, masks with scores exceeding θ_{conf} are retained, while those with scores below θ_{conf} are discarded. The mask confidence branch is necessary since it eliminates many low-quality candidate instances.

6.6. Number of Keyframes

We experiment with a different number of keyframes in each FBV rather than the default 9 keyframes in LIRA. As shown in Tab. 5, when 9 keyframes are used as a FBV, the reasoning reconstruction accuracy is relatively the highest.

6.7. Accuracy Analysis of Candidate Instances

We also provide the accuracy of 3D candidate instances after the instance fusion stage in Tab. 6 for reference. After final target instances are inferred through LLM reasoning in stage III, there is a slight decrease in accuracy metrics.

6.8. Geometric Reconstruction of the Entire Scene

Given a sequence of RGB-D observations, LIRA applies the widely used TSDF fusion for geometric reconstruction. We also provide the results of geometric reconstruction of the entire scene (including target instances and background environment) in Tab. 7 for reference. The implementation of the geometric reconstruction evaluation metrics follows [2, 10]. Experimental results indicate that using standard TSDF fusion yields high-quality geometric reconstruction performance. The most important F-score reaches 0.928, demonstrating both high-quality reconstruction accuracy and completeness.

Method	AP	AP$_{50}$	AP$_{25}$
7 keyframes	11.08	33.72	65.10
9 keyframes	**11.57**	**34.39**	**66.24**
11 keyframes	10.79	33.16	64.55

Table 5. Quantitative results of number of keyframes in the FBV.

Target	AP	AP$_{50}$	AP$_{25}$
3D Candidate Instances	11.98	35.64	70.63
Final Target Instances	11.57	34.39	66.24

Table 6. Accuracy of 3D candidate instances after instance fusion.

Method	Acc	Comp	Precision	Recall	F-score
LIRA	0.062	0.020	0.895	0.969	0.928

Table 7. Results of geometry reconstruction of the entire scene.

6.9. Visualization of 2D/3D Candidate Instances

Fig. 8 presents the visualization results of 2D/3D candidate instances during the reasoning reconstruction process of LIRA, as well as the final target instances.

6.10. More Visualization Results of Reasoning Reconstruction of LIRA

More visualization results of reasoning reconstruction of LIRA are presented in Fig. 9.

6.11. Supplementary Video

We also provide real-time online reasoning reconstruction visualization to further demonstrate the practicability of LIRA in our supplementary video demo. LIRA inputs RGB-D sequences and reconstructs instances that conform to implicit language instructions involving complex reasoning and background environment. In the video demo, we do not visualize the candidate instances generated by LIRA during the intermediate process. The final reasoning reconstruction results at different time steps are only visualized. Specifically, we visualize double-layered mesh. As the process is online, LIRA can halt at any time step.

References

[1] 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, 2020. 2

[2] Xi Chen, Jiaming Sun, Yiming Xie, Hujun Bao, and Xiaowei Zhou. Neuralrecon: Real-time coherent 3d scene reconstruction from monocular video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):7542–7555, 2024. 1, 6

[3] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Niessner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2

[4] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Knowledge Discovery and Data Mining*, pages 226–231, 1996. 4

[5] Qiao Gu, Ali Kuwajerwala, Sacha Morin, Krishna Murthy Jatavallabhula, Bipasha Sen, Aditya Agarwal, Corban Rivera, William Paul, Kirsty Ellis, Rama Chellappa, Chuang Gan, Celso Miguel de Melo, Joshua B. Tenenbaum, Antonio Torralba, Florian Shkurti, and Liam Paull. Conceptgraphs: Open-vocabulary 3d scene graphs for perception and planning. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5021–5028, 2024. 1

[6] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *arXiv preprint arXiv:2106.09685*, 2021. 4

[7] Chenguang Huang, Oier Mees, Andy Zeng, and Wolfram Burgard. Visual language maps for robot navigation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 10608–10615, 2023. 1, 4

[8] Zhening Huang, Xiaoyang Wu, Xi Chen, Hengshuang Zhao, Lei Zhu, and Joan Lasenby. Openins3d: Snap and lookup for 3d open-vocabulary instance segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 169–185, 2025. 4

[9] Krishna Murthy Jatavallabhula, Alihusein Kuwajerwala, Qiao Gu, Mohd Omama, Tao Chen, Shuang Li, Ganesh Iyer, Soroush Saryazdi, Nikhil Keetha, Ayush Tewari, Joshua B. Tenenbaum, Celso Miguel de Melo, Madhava Krishna, Liam Paull, Florian Shkurti, and Antonio Torralba. Conceptfusion: Open-set multimodal 3d mapping. *Robotics: Science and Systems (RSS)*, 2023. 1

[10] Jihong Ju, Ching Wei Tseng, Oleksandr Bailo, Georgi Dikov, and Mohsen Ghafoorian. Dg-recon: Depth-guided neural 3d scene reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 18184–18194, 2023. 6

[11] Xin Lai, Zhuotao Tian, Yukang Chen, Yanwei Li, Yuhui Yuan, Shu Liu, and Jiaya Jia. Lisa: Reasoning segmentation via large language model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9579–9589, 2024. 5

[12] Sergey Linok, Tatiana Zemskova, Svetlana Ladanova, Roman Titkov, Dmitry Yudin, Maxim Monastyrny, and Aleksei Valenkov. Beyond bare queries: Open-vocabulary object grounding with 3d scene graph. In *arXiv preprint arXiv:2406.07113*, 2024. 1

[13] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, and Lei Zhang. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 38–55, 2025. 5

[14] Shiyang Lu, Haonan Chang, Eric Pu Jing, Abdeslam Boularias, and Kostas Bekris. Ovir-3d: Open-vocabulary 3d instance retrieval without training on 3d data. In *Proceedings of The 7th Conference on Robot Learning*, pages 1610–1620, 2023. 1, 4

[15] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *2016 Fourth International Conference on 3D Vision (3DV)*, pages 565–571, 2016. 4

[16] OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, et al. Gpt-4 technical report. In *arXiv preprint arXiv:2303.08774*, 2024. 2

[17] Songyou Peng, Kyle Genova, Chiyu "Max" Jiang, Andrea Tagliasacchi, Marc Pollefeys, and Thomas Funkhouser. Openscene: 3d scene understanding with open vocabularies. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 815–824, 2023. 4

[18] Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, page 3505–3506, 2020. 4

[19] Jonas Schult, Francis Engelmann, Alexander Hermans, Or Litany, Siyu Tang, and Bastian Leibe. Mask3d: Mask transformer for 3d semantic instance segmentation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8216–8223, 2023. 4

[20] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. In *arXiv preprint arXiv:2409.12191*, 2024. 2

[21] Zhuofan Xia, Dongchen Han, Yizeng Han, Xuran Pan, Shiji Song, and Gao Huang. Gsva: Generalized segmentation via multimodal large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3858–3869, 2024. 5

[22] Jiarui Xu, Sifei Liu, Arash Vahdat, Wonmin Byeon, Xiaolong Wang, and Shalini De Mello. Open-vocabulary panop-

tic segmentation with text-to-image diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2955–2966, 2023. 4

[23] Kashu Yamazaki, Taisei Hanyu, Khoa Vo, Thang Pham, Minh Tran, Gianfranco Doretto, Anh Nguyen, and Ngan Le. Open-fusion: Real-time open-vocabulary 3d mapping and queryable scene representation. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9411–9417, 2024. 1

[24] Mi Yan, Jiazhao Zhang, Yan Zhu, and He Wang. Maskclustering: View consensus based mask graph clustering for open-vocabulary 3d instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 28274–28284, 2024. 4

[25] Senqiao Yang, Tianyuan Qu, Xin Lai, Zhuotao Tian, Bohao Peng, Shu Liu, and Jiaya Jia. Lisa++: An improved baseline for reasoning segmentation with large language model. In *arXiv preprint arXiv:2312.17240*, 2024. 5

[26] Hao Zhang, Hongyang Li, Feng Li, Tianhe Ren, Xueyan Zou, Shilong Liu, Shijia Huang, Jianfeng Gao, Leizhang, Chunyuan Li, and Jainwei Yang. Llava-grounding: Grounded visual chat with large multimodal models. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 19–35, 2025. 5

[27] Zhen Zhou, Yunkai Ma, Junfeng Fan, Shaolin Zhang, Fengshui Jing, and Min Tan. Eprecon: An efficient framework for real-time panoptic 3d reconstruction from monocular video. 2024. 1

[28] Xueyan Zou, Jianwei Yang, Hao Zhang, Feng Li, Linjie Li, Jianfeng Wang, Lijuan Wang, Jianfeng Gao, and Yong Jae Lee. Segment everything everywhere all at once. In *Advances in Neural Information Processing Systems*, pages 19769–19782, 2023. 5

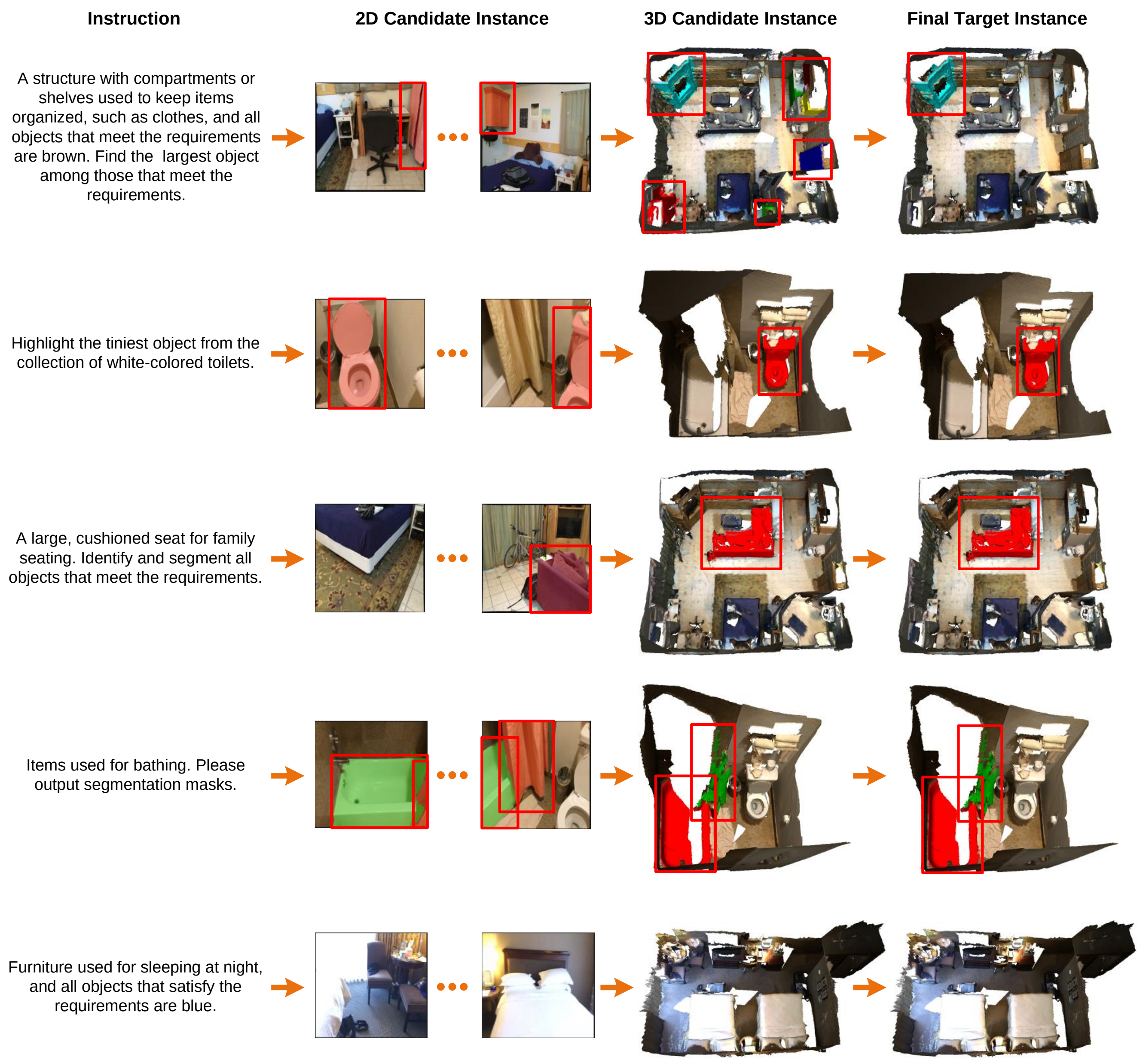
Instruction
2D Candidate Instance
3D Candidate Instance
Final Target Instance
A structure with compartments or shelves used to keep items organized, such as clothes, and all objects that meet the requirements are brown. Find the largest object among those that meet the requirements.
Highlight the tiniest object from the collection of white-colored toilets.
A large, cushioned seat for family seating. Identify and segment all objects that meet the requirements.
Items used for bathing. Please output segmentation masks.
Furniture used for sleeping at night, and all objects that satisfy the requirements are blue.

Figure 8. Visualization of 2D/3D candidate instances and final target instances.

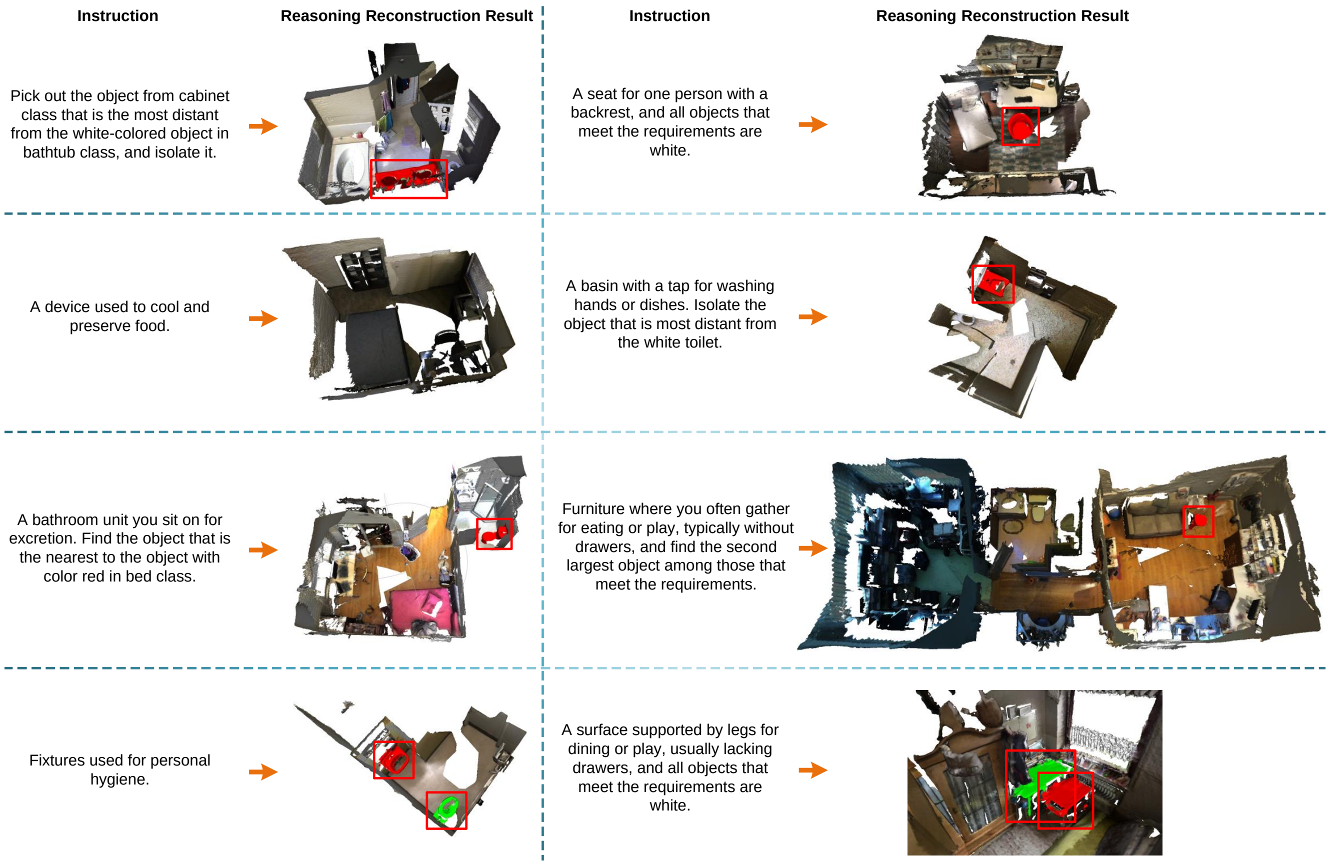
Instruction
Reasoning Reconstruction Result
Instruction
Reasoning Reconstruction Result
Pick out the object from cabinet class that is the most distant from the white-colored object in bathtub class, and isolate it.
A seat for one person with a backrest, and all objects that meet the requirements are white.
A device used to cool and preserve food.
A basin with a tap for washing hands or dishes. Isolate the object that is most distant from the white toilet.
A bathroom unit you sit on for excretion. Find the object that is the nearest to the object with color red in bed class.
Furniture where you often gather for eating or play, typically without drawers, and find the second largest object among those that meet the requirements.
Fixtures used for personal hygiene.
A surface supported by legs for dining or play, usually lacking drawers, and all objects that meet the requirements are white.

Figure 9. More visualization results of reasoning reconstruction of LIRA. We augment the reconstructed geometric results with image textures. Single-layered mesh is visualized.

```
messages = [{"role": "system", "content": "You are an AI assistant, doing visual language understanding tasks in the field of 3D scenes. Here are some object attribute information, including color, class, size, and 3D spatial position. Based on these object attribute information, design language instruction templates and introduce questions or descriptions for the above object attributes. The language instruction templates you generate should involve at least one attribute and no more than three attributes. For each selected combination of attributes in an instruction template, provide 10 different forms that convey the same meaning to increase the richness and diversity of the instruction templates. Note that the instruction templates should not be overly complicated or excessively brief.

For example, you can choose the attributes of class and color, and then generate an instruction template: "Locate objects that are of {color} and belong to {class}," and then proceed with rephrasing."}]

for sample in fewshot_samples:
    messages.append({"role": "user", "content": sample["content"]})
    messages.append({"role": "assistant", "content": sample["response"]})
messages.append ({"role": "user", "content": '\n'.join(sample["query"])})
```

Figure 10. The prompt for generating base instruction templates.

"Pick out the object from {class A} that is the {howfar} distant from the {color} object in {class B}, and isolate it."

"Extract the {howsize} item within the group of objects that belong to {class} and have {color}."

"Retrieve instances of {class} that are {color}."

"Every object belonging to {class} should be found."

"Segment out the object in {class B} that is positioned at the {howfar} distance from the object of {color} in {class A}."

"Mark the {howsize} object in the group of {class} that are {color}."

"Find and segment the {howsize} object among the objects that are both {class} and have {color}."

"Locate objects that are of {color} and belong to {class}."

"Gather all objects that are classified as {class}."

"Identify and segment the object in {class B} that is {howfar} from the object in {class A} with {color}."

"Select and extract the object in {class B} with the {howfar} distance to the {color} object in the {class A}."

"Choose the {howsize} item among the objects that are of {class} and are {color}."

Figure 11. Examples of base instruction templates.

```
messages = [{"role": "system", "content": "You are an AI assistant, doing visual language understanding tasks in the
field of 3D scenes. The following object classes are provided: {cabinet, bed, chair, sofa, ...}, and implicit instructions
need to be generated for each class. Implicit instructions require detailed descriptions of these target classes. The
target class name should not appear in any of these implicit instructions. The implicit instructions you generate should
include at least one class, and no more than four classes. The target classes corresponding to the generated implicit
instructions should be strictly accurate and clear. Implicit instructions must not be ambiguous to the point where it is
difficult to determine whether a class belongs to a particular instruction. The descriptions of the implicit instructions
should be as diverse as possible. Note that the instruction templates should neither be overly complex nor excessively
brief.

For example, you could select the classes of bed, chair, and sofa, and then generate an instruction template like:
"Furniture used for sitting or lying down."}]

for sample in fewshot_samples:
    messages.append({"role": "user", "content": sample["content"]})
    messages.append({"role": "assistant", "content": sample["response"]})
messages.append ({"role": "user", "content": '\n'.join(sample["query"])})
```

Figure 12. The prompt for generating implicit instruction templates.

```
'"content": "Items used in the bathroom to aid bathing and maintain hygiene."
"response": "sink, shower curtain, toilet, bathtub."

'"content": "Furniture that people use to organize their clothes and personal belongings."
"response": "cabinet, bookshelf, desk."

"content": "A flat surface often in kitchens, bathrooms, and bars, used for food preparation or holding items."
"response": "counter."

"content": "A waterproof barrier hung to prevent water from splashing out during bathing."
"response": "shower curtain."

"content": "A flat surface on legs generally used for play, or dining and typically has no drawers."
"response": "table."

'"content": "Typically placed in rooms to hold belongings, such as hats and quilts, securely and neatly."
"response": "cabinet."

'"content": "Appliances or furniture used to store food."
"response": "refrigerator, cabinet."

'"content": "A piece of furniture used for writing or computer work."
"response": "desk."
```

Figure 13. Examples of implicit instruction templates.

```
messages = [{"role": "system", "content": "You are an AI assistant, doing visual language understanding tasks in the
field of 3D scenes. First, a language instruction template is randomly selected from base instruction templates. Given a
specific scene and the attributes of all objects in that scene, including color, class, size, and 3D spatial position, a
subset of objects and their attributes is randomly selected and inserted into the chosen instruction template to generate
a specific scene-instruction pair. To increase the likelihood of implicit instructions, for each class, you select an implicit
instruction template with a probability greater than 50% and incorporate it into the generated scene-instruction pair.
Finally, the generated instruction is polished to ensure it aligns with human language expression.

For example, there is an object in a scene with the following attributes: {color: white, class: sink, size: [0.74, 0.552,
0.269], position: [2.692, 5.291, 0.788]}. You can generate an instruction for this scene: "White fixtures used for
personal hygiene. Find the smallest object among those that meet the requirements."}]

for sample in fewshot_samples:
    messages.append({"role": "user", "content": sample["content"]})
    messages.append({"role": "assistant", "content": sample["response"]})
messages.append ({"role": "user", "content": '\n'.join(sample["query"])})
```

Figure 14. The prompt for generating scene-instruction pairs.

"A single-person seat, not for lying down, and find the second smallest object among those that meet the requirements."

"A bathroom unit you sit on for excretion. Find the object that is the nearest to the object with color red in bed class."

"A structure with compartments or shelves used to keep items organized, such as clothes, and all objects that meet the requirements are brown. Find the largest object among those that meet the requirements."

"A large, cushioned seat for family seating. Identify and segment all objects that meet the requirements."

"A basin with a tap for washing hands or dishes, and all objects that meet the requirements are white."

"Furniture where you often gather for eating or play, typically without drawers, and find the second largest object among those that meet the requirements."

"A surface supported by legs for dining or play, usually lacking drawers, and all objects that meet the requirements are white."

"Pick out the object from cabinet class that is the most distant from the white-colored object in bathtub class, and isolate it."

"A bowl with a seat for bodily waste disposal, and find the largest object among those that meet the requirements."

"A basin with a tap for washing hands or dishes. Isolate the object that is most distant from the white toilet."

Figure 15. Examples of scene-instruction pairs.