GenSplat: Bridging the Generalization Gap in 3DGS Language Comprehension

Supplementary Material

This supplementary material is organized as follows. Sec. A provides additional details of the GenSplat pipeline, including the formulation of 3D Gaussian Splatting, Gaussian clustering, geometry-aware visible Gaussian features, instruction tuning, and the architecture of the instance and referring decoders. Sec. B presents additional quantitative results on efficiency comparison, grounding without object names, referring segmentation ScanRefer, dense captioning, and open-vocabulary understanding. Sec. C provides qualitative visualizations across multiple tasks and datasets.

A. Details in Pipeline

3D Gaussian Splatting (3DGS) [13] models a scene as an explicit set of anisotropic Gaussians. Each Gaussian is defined by a center $\mathbf{x} \in \mathbb{R}^3$ and a covariance matrix Σ. Its density is given by:

$$G(\mathbf{x}) = \exp\left(-\tfrac{1}{2}\mathbf{x}^\top \Sigma^{-1}\mathbf{x}\right). \tag{1}$$

To enable stable differentiable optimization, the covariance is factorized into a rotation matrix R and a scaling matrix S:

$$\Sigma = RSS^\top R^\top. \tag{2}$$

For rendering, Gaussians are projected onto the camera plane using a viewing transformation W and the Jacobian J of the local affine approximation of the projection. The covariance in camera space becomes:

$$\Sigma' = JW\Sigma W^\top J^\top. \tag{3}$$

Each Gaussian point carries learnable attributes, including position $\mathbf{x}$, spherical harmonics color coefficients $\mathbf{c} \in \mathbb{R}^k$, opacity α, rotation quaternion $q \in \mathbb{R}^4$, and scale $\mathbf{s} \in \mathbb{R}^3$. For each pixel, colors and opacities of all overlapping Gaussians are computed according to their projected densities and are composited in a front-to-back order via alpha blending.

Instance and Referring Decoders. Both the instance decoder and the referring decoder follow a similar architecture, each implemented as a lightweight Transformer decoder designed to process a small set of query embeddings. The decoder consists of 3 layers, each containing a multi-head self-attention module, a cross-attention module, and a feed-forward subnetwork. We use 200 learnable queries, which interact with Gaussian features through cross-attention. Each layer adopts a 256-dimensional hidden size, 8 attention heads, and a 1024-dimensional MLP with GELU activation.

Table A1. Dataset statistics for joint instruction tuning.

Dataset	Task	Size
ScanRefer	referring segmentation	37K
Nr3D	referring segmentation	29K
Sr3D	referring segmentation	66K
Multi3DRefer	referring segmentation	44K
ScanQA	visual question answering	30K
SQA3D	visual question answering	89K
Scan2Cap	dense captioning	37K
Nr3D-Captioning*	dense captioning	29K
Total	-	364K

During inference, these queries progressively aggregate semantic cues from the clustered Gaussian features, enabling the decoder to reason about spatial relationships and produce instance-level or referring-level predictions. Because the structure is shared across both modules, the referring decoder inherits the same architectural properties, differing only in the conditioning signals it receives (e.g., language-conditioned features in referring tasks).

Details of Instruction Tunning. In Table A1, we summarize all datasets used in our instruction tuning stage. We include four referring-segmentation datasets (ScanRefer [4], Nr3D [1], Sr3D [1], and Multi3DRefer [26]), two visual question-answering datasets (ScanQA [2] and SQA3D [18]), and two dense-captioning datasets (Scan2Cap [6] and Nr3D-Captioning* [10]). Together, these datasets cover a diverse range of supervisory signals—object grounding, language-conditioned reasoning, and dense object-level descriptions—allowing our model to learn comprehensive 3D language understanding capabilities. The Nr3D-Captioning* is constructed following Chat-Scene [10]. Each Nr3D sample, originally designed for referring expression grounding, is paired with an automatically generated dense caption describing the target object. This transforms Nr3D into an additional dense-captioning resource aligned with its original annotations.

For the third training stage, we update only a small set of parameters while keeping the rest of the model frozen. Specifically, we train the LoRA of the MLLM, the referring decoder, and its associated projector layer. The projector is introduced to map the segmentation-token features produced by the MLLM into a feature space compatible with the referring decoder, ensuring effective interaction between the two components.

Frame-Level Supervision for GAFS. In the training stage of GAFS, we must determine which frames are relevant to the input textual expression. For referring-

Table A2. Efficiency comparison on ScanRefer [4]. Training cost is measured in GPU hours on a single H100. Inference reports per-scene time averaged over all text queries. 3DGS reconstruction time is excluded for all methods.

Methods	Training Cost (GPU Hours) ↓				Inference Cost ↓	Params. ↓	Peak Memory ↓		Perf.
	Semantic	Instance	Referring	Total			Train	Inference	mIoU ↑
ReferSplat [9]	-	-	∼331h†+ 16h	∼347h	105s	-	5.6G	3.1G	20.2
3D-LLaVA$_{GS}$ [8]	-	-	45h^{1} + 22h	67h	25s	6.8G	19.5G	14.7G	29.8
Ours	32h	54h	12h*+ 32h	130h	3s*+ 46s	7.2G	56.3G	20.6G	**43.6**

†: offline (Grounded-SAM) pre-processing time. ^{1}: 2D-3D lifting time of GT masks using camera poses and depth. *: our frame selection time.

Table A3. Grounding without object names on ScanRefer [4]. Target object names in the referring expressions are replaced with the generic token "object" following [24].

Method	mIoU ↑	A@0.25 ↑	A@0.5 ↑
LISA-7B [16]	12.9	20.4	3.7
MLLM-For3D [11]	30.5	33.1	31.2
Ours	**35.4**	**47.5**	**35.5**

segmentation datasets such as ScanRefer, Nr3D, Sr3D, and Multi3DRefer, this relevance can be obtained naturally: each frame comes with a binary ground-truth mask, and a frame is considered text-relevant if its rendered pixels contain the ground-truth object specified by the expression.

For the ScanQA dataset, although the annotations are in the form of questions and answers, each sample still corresponds to a specific target object. Therefore, we can identify the relevant frames by checking whether the frame contains the object instance referenced by the object ID.

For SQA3D, however, frame-level supervision is not explicitly provided. To construct this supervision, we follow a language-driven approach by using GPT-5 to decide frame relevance. For each frame and its associated question, we feed both into GPT-5 and ask it to judge whether the frame contains sufficient visual evidence related to the query. This gives us a reliable frame-level label that can be used during GAFS training.

Fair Comparison with Point-Cloud-Based MLLMs. Following LiftGS [3], during the Gaussian reconstruction stage, we assume that the number of Gaussians matches the number of point-cloud, by fixing Gaussian positions and disabling densification process [25]. This design choice allows our method to be directly comparable to point-cloud–based approaches, ensuring a fair evaluation protocol across different 3D representations.

B. Additional Quantitative Results

B.1. Efficiency Comparison

We provide a comprehensive efficiency analysis comparing GenSplat with existing methods. Table A2 reports training cost, inference cost, model parameters, and peak GPU memory on a single H100 GPU. ReferSplat [9] requires per-scene optimization, whose cost scales linearly with the number of scenes (*e.g.*, ∼347 GPU hours including ∼331h for Grounded-SAM [22] pre-processing and 16h for optimization on the ScanRefer validation set). In contrast, our method requires only a one-time training cost (∼130 GPU hours) and is generalizable across scenes, delivering significantly better referring segmentation performance (43.6 vs. 20.2 mIoU). We further adapt 3D-LLaVA [8] for 3DGS inputs (denoted as 3D-LLaVA$_{GS}$): although easier to train (67h), it achieves a comparable model size (6.8G vs. 7.2G parameters) but substantially lower performance (29.8 vs. 43.6 mIoU), confirming that the gains of GenSplat stem from our structured learning process rather than increased compute. Moreover, our inference speed (49s per scene) is 2× faster than ReferSplat (105s) without requiring any test-time per-scene optimization, demonstrating a favorable efficiency–performance trade-off.

Table A4. Performance comparison on **Scan2Cap (val)** dataset.

Method	Modality	C@0.5↑	B-4@0.5↑	M@0.5↑	R@0.5↑
Expert Models:					
Scan2Cap [6]	PC	39.1	23.3	22.0	44.8
MORE [12]	PC	40.9	22.9	21.7	44.4
3D-VisTA [28]	PC	61.6	34.1	26.8	55.0
Vote2Cap [5]	PC	61.8	34.5	26.2	54.4
3D MLLMs:					
Chat-Scene [10]	PC+I	77.2	36.4	**28.0**	**58.1**
3D-LLaVA [8]	PC	78.8	36.9	27.1	57.7
Ours	GS+I	**83.4**	**37.2**	27.7	57.8

B.2. Grounding without Object Names

Following the challenging setting proposed by EDA [24], where the target object name in the referring expression is replaced with a generic token "object", we evaluate GenSplat's spatial reasoning capability beyond reliance on semantic shortcuts. As shown in Table A3, we compare our method with LISA-7B [16] (a 2D reasoning segmentation MLLM) and MLLM-For3D [11] (a concurrent work that lifts 2D MLLM predictions onto 3D representations) on the ScanRefer [4] dataset under this setting. The results

Table A5. Performance comparison on **Ref-LERF** [9] dataset. mIoU is reported.

Method	ramen	figurines	teatime	kitchen	average
Grounded SAM [22]	14.1	16.0	16.9	16.2	15.8
LangSplat [19]	12.0	17.9	7.6	17.9	13.9
GOI [20]	27.1	16.5	22.9	15.7	20.5
ReferSplat [9]	35.2	25.7	31.3	24.4	29.2
Ours	**38.8**	**33.9**	**33.1**	**25.6**	**32.9**

demonstrate that: (1) naively applying a 2D MLLM (*i.e.*, LISA-7B) and lifting its per-frame predictions to 3D suffers from significant multi-view inconsistency (A@0.5 = 3.7); (2) MLLM-For3D [11] improves over LISA-7B by adopting a spatial consistency strategy, yet still falls short compared to our method; and (3) our method naturally generalizes to this setting and achieves the best performance across all metrics (+4.9 mIoU and +14.4 A@0.25 over MLLM-For3D), confirming that GenSplat leverages genuine spatial reasoning rather than semantic shortcuts from object names.

B.3. 3D Dense Captioning

3D Dense Captioning requires the model to generate object-level descriptions that capture both the instance itself and its spatial relations to nearby objects. In our setup, we adopt the widely used strategy of employing Mask3D [23] to obtain mask proposals as visual prompts. Since Mask3D outputs are unavailable during training, we utilize a visual sampler [8] to translate the predefined prompt representation into the semantic space of visual features. As shown in Table A4, GenSplat attains the strongest performance, demonstrating its capability in producing accurate and informative instance-level captions.

B.4. 3D Referring Segmentation on Ref-LERF

The Ref-LERF dataset extends the original LERF data. While LERF provides only simple semantic-level text labels, Ref-LERF enriches each object with fine-grained language expressions. Building on LangSplat's mask annotations (LERF-OVS), Ref-LERF further introduces detailed referring expressions (typically around five per object), emphasizing both appearance and spatial relationships. In total, the dataset offers 295 descriptions for 59 objects (236 for training and 59 for testing), enabling more expressive grounding and supporting evaluation of referring 3D Gaussian Splatting tasks.

In Table A5, we compare our method with several representative baselines on the Ref-LERF benchmark, including the zero-shot Grounded-SAM as well as two per-scene optimization approaches, LangSplat and ReferSplat. Across all four scenes, our method consistently achieves the best performance, surpassing per-scene optimization pipelines and improving the average score by 12.7%.

B.5. Detailed Comparison on ScanRefer

We conduct a detailed comparison on the five scenes used in ReferSplat, as reported in Table A6. The results show that our method consistently outperforms all competing approaches across these diverse scenes.

B.6. 3D Open-Vocabulary Understanding

Our Gaussian Encoder is trained on the ScanNet [7] training split by distilling 2D foundation model features extracted from SAM [15] and CLIP [21]. This optimization strategy enables the encoder to learn geometry-aware and language-aligned semantic representations without requiring explicit 3D annotations.

Evaluation on ScanNet. To assess the open-vocabulary capability, we follow a semantic-level evaluation protocol on the ScanNet validation set. Specifically, each official ScanNet category name is fed into CLIP to obtain its textual embedding, which is then compared with the semantic features of Gaussians using cosine similarity to generate category predictions. We evaluate on 12 randomly selected validation scenes, including `scene0050_02`, `scene0144_01`, `scene0221_01`, `scene0300_01`, `scene0354_00`, `scene0389_00`, `scene0423_02`, `scene0427_00`, `scene0494_00`, `scene0616_00`, `scene0645_02`, and `scene0693_00`. The category set includes common indoor object classes such as wall, floor, cabinet, bed, chair, sofa, table, door, window, bookshelf, picture, counter, desk, curtain, refrigerator, shower curtain, toilet, sink, and bathtub. Quantitative results are shown in Table 4 of the main paper and Table A7.

Evaluation on LERF-OVS. For LERF-OVS [14], we follow LangSplat [19] to distill 2D language features into the semantic-level pretrained *Gaussian Encoder* for open-vocabulary segmentation. Results are reported in Table A8.

Overall, our method achieves state-of-the-art performance on ScanNet and yields competitive results on LERF-OVS, demonstrating strong generalization ability across diverse open-vocabulary 3D scene benchmarks.

C. Additional Qualitative Results

We present qualitative visualizations across multiple tasks and datasets to complement the quantitative results above.

Referring Segmentation on ScanRefer. In Fig. A1, we show referring segmentation results on additional scenes from ScanRefer, with frame-wise visualizations from diverse viewpoints. These results reinforce the quantitative findings, showing that our method consistently produces clearer and accurate predictions than competing approaches across diverse viewpoints and scene configurations.

Fine-Grained Instance Disambiguation. In Fig. A2, we visualize multiple queries from the same scene, where different textual expressions are used to refer to different chair

Table A6. Comparison of 3D referring segmentation on five scenes (selected by ReferSplat [9]) from the ScanRefer [4] dataset. Best results are **bolded**. mIoU is reported.

Methods	Training Type	scene0011_00	scene0015_00	scene0019_00	scene0025_00	scene0030_00	average
Grounded-SAM [22]	Zero-shot	11.4	14.5	14.6	10.1	11.3	12.4
LangSplat [19]	Per-scene opt.	8.6	14.3	16.2	7.4	10.2	11.2
ReferSplat [9]	Per-scene opt.	14.4	21.3	**30.9**	3.2	12.5	16.5
Ours	Generalized	**23.0**	**16.9**	28.8	**42.6**	**29.9**	**28.2**

Table A7. Comparison of 3D open-vocabulary understanding on **ScanNet** [7] (12 scenes). Best results are **bolded**.

Methods	mIoU↑	mAcc.↑
LSeg [17]	56.1	74.5
LERF [14]	31.2	61.7
LangSplat [19]	24.7	42.0
Featrure3DGS [27]	59.2	75.1
Ours	**61.2**	**78.9**

Table A8. mIOU comparison of 3D open-vocabulary understanding on **LERF-OVS** [14]. Best results are **bolded**.

Methods	Ramen	Figurines	Teatime	Kitchen	Average
Feature-3DGS [27]	43.7	58.8	40.5	39.6	45.7
LangSplat [19]	51.2	44.7	65.1	44.5	51.4
GOI [20]	52.6	63.7	44.5	41.4	50.6
ReferSplat [9]	**55.1**	**67.5**	50.1	48.9	55.4
Ours	50.4	60.6	**67.4**	**60.2**	**59.7**

instances. Although all targets belong to the same semantic category, our model is able to accurately resolve which specific object each expression refers to. This demonstrates that GenSplat can reason over fine-grained linguistic cues and reliably localize the correct instance. By progressively aligning linguistic concepts with 3D Gaussian primitives, GenSplat effectively bridges the gap between free-form language understanding and 3D spatial reasoning.

Referring Segmentation on Ref-LERF. We provide qualitative results on Ref-LERF in Fig. A3. The visualizations show that our method produces more accurate and better-localized segmentations for both short phrases and complex long expressions, demonstrating strong capability in understanding fine-grained object descriptions and spatial context in cluttered 3D environments.

Open-Vocabulary Segmentation on ScanNet. Visual examples of the open-vocabulary predictions are provided in Fig. A4, illustrating the model's ability to generalize to diverse categories and complex indoor layouts.

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 *ECCV*, 2020. 1

[2] Daichi Azuma, Taiki Miyanishi, Shuhei Kurita, and Motoaki Kawanabe. Scanqa: 3d question answering for spatial scene understanding. In *CVPR*, 2022. 1, 7

[3] Ang Cao, Sergio Arnaud, Oleksandr Maksymets, Jianing Yang, Ayush Jain, Ada Martin, Vincent-Pierre Berges, Paul McVay, Ruslan Partsey, Aravind Rajeswaran, et al. From thousands to billions: 3d visual language grounding via render-supervised distillation from 2d vlms. In *ICML*, 2025. 2

[4] Dave Zhenyu Chen, Angel X Chang, and Matthias Nießner. Scanrefer: 3d object localization in rgb-d scans using natural language. In *ECCV*, 2020. 1, 2, 4, 6, 7

[5] Sijin Chen, Hongyuan Zhu, Xin Chen, Yinjie Lei, Gang Yu, and Tao Chen. End-to-end 3d dense captioning with vote2cap-detr. In *CVPR*, 2023. 2

[6] Zhenyu Chen, Ali Gholami, Matthias Nießner, and Angel X Chang. Scan2cap: Context-aware dense captioning in rgb-d scans. In *CVPR*, 2021. 1, 2

[7] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, 2017. 3, 4, 9

[8] Jiajun Deng, Tianyu He, Li Jiang, Tianyu Wang, Feras Dayoub, and Ian Reid. 3d-llava: Towards generalist 3d lmms with omni superpoint transformer. In *CVPR*, 2025. 2, 3

[9] Shuting He, Guangquan Jie, Changshuo Wang, Yun Zhou, Shuming Hu, Guanbin Li, and Henghui Ding. ReferSplat: Referring segmentation in 3d gaussian splatting. In *ICML*, 2025. 2, 3, 4, 8

[10] Haifeng Huang, Yilun Chen, Zehan Wang, Rongjie Huang, Runsen Xu, Tai Wang, Luping Liu, Xize Cheng, Yang Zhao, Jiangmiao Pang, et al. Chat-scene: Bridging 3d scene and large language models with object identifiers. In *NeurIPS*, 2024. 1, 2

[11] Jiaxin Huang, Runnan Chen, Ziwen Li, Zhengqing Gao, Xiao He, Yandong Guo, Mingming Gong, and Tongliang Liu. Mllm-for3d: Adapting multimodal large language model for 3d reasoning segmentation. In *NeurIPS*, 2025. 2, 3

[12] Yang Jiao, Shaoxiang Chen, Zequn Jie, Jingjing Chen, Lin Ma, and Yu-Gang Jiang. More: Multi-order relation mining for dense captioning in 3d scenes. In *ECCV*, 2022. 2

[13] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM TOG*, 2023. 1

[14] Justin Kerr, Chung Min Kim, Ken Goldberg, Angjoo Kanazawa, and Matthew Tancik. Lerf: Language embedded radiance fields. In *ICCV*, 2023. 3, 4

[15] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *ICCV*, 2023. 3

[16] Xin Lai, Zhuotao Tian, Yukang Chen, Yanwei Li, Yuhui Yuan, Shu Liu, and Jiaya Jia. Lisa: Reasoning segmentation via large language model. In *CVPR*, 2024. 2

[17] Boyi Li, Kilian Q Weinberger, Serge Belongie, Vladlen Koltun, and René Ranftl. Language-driven semantic segmentation. *arXiv:2201.03546*, 2022. 4

[18] Xiaojian Ma, Silong Yong, Zilong Zheng, Qing Li, Yitao Liang, Song-Chun Zhu, and Siyuan Huang. Sqa3d: Situated question answering in 3d scenes. In *ICLR*, 2023. 1

[19] Minghan Qin, Wanhua Li, Jiawei Zhou, Haoqian Wang, and Hanspeter Pfister. Langsplat: 3d language gaussian splatting. In *CVPR*, 2024. 3, 4

[20] Yansong Qu, Shaohui Dai, Xinyang Li, Jianghang Lin, Liujuan Cao, Shengchuan Zhang, and Rongrong Ji. Goi: Find 3d gaussians of interest with an optimizable open-vocabulary semantic-space hyperplane. In *ACM MM*, 2024. 3, 4

[21] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 3

[22] Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kunchang Li, He Cao, Jiayu Chen, Xinyu Huang, Yukang Chen, Feng Yan, et al. Grounded sam: Assembling open-world models for diverse visual tasks. *arXiv:2401.14159*, 2024. 2, 3, 4

[23] Jonas Schult, Francis Engelmann, Alexander Hermans, Or Litany, Siyu Tang, and Bastian Leibe. Mask3D: Mask Transformer for 3D Semantic Instance Segmentation. In *ICRA)*, 2023. 3

[24] Yanmin Wu, Xinhua Cheng, Renrui Zhang, Zesen Cheng, and Jian Zhang. Eda: Explicit text-decoupling and dense alignment for 3d visual grounding. In *CVPR*, 2023. 2

[25] Yanmin Wu, Jiarui Meng, Haijie Li, Chenming Wu, Yahao Shi, Xinhua Cheng, Chen Zhao, Haocheng Feng, Errui Ding, Jingdong Wang, et al. Opengaussian: Towards point-level 3d gaussian-based open vocabulary understanding. In *NeurIPS*, 2024. 2

[26] Yiming Zhang, ZeMing Gong, and Angel X Chang. Multi3drefer: Grounding text description to multiple 3d objects. In *ICCV*, 2023. 1

[27] Shijie Zhou, Haoran Chang, Sicheng Jiang, Zhiwen Fan, Zehao Zhu, Dejia Xu, Pradyumna Chari, Suya You, Zhangyang Wang, and Achuta Kadambi. Feature 3dgs: Supercharging 3d gaussian splatting to enable distilled feature fields. In *CVPR*, 2024. 4

[28] Ziyu Zhu, Xiaojian Ma, Yixin Chen, Zhidong Deng, Siyuan Huang, and Qing Li. 3d-vista: Pre-trained transformer for 3d vision and text alignment. In *ICCV*, 2023. 2

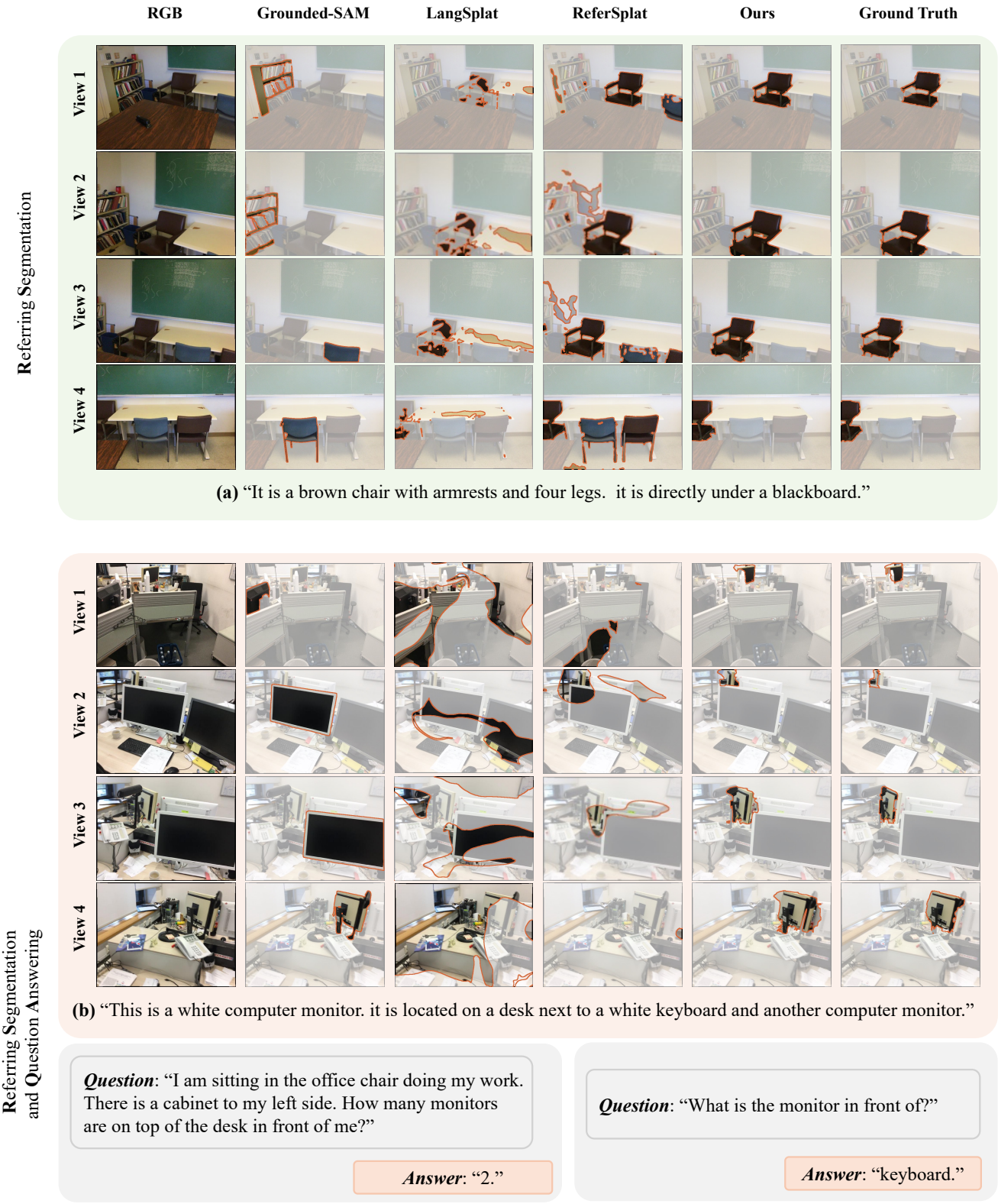

Figure A1. Qualitative visualization on the **ScanRefer** [4] dataset.

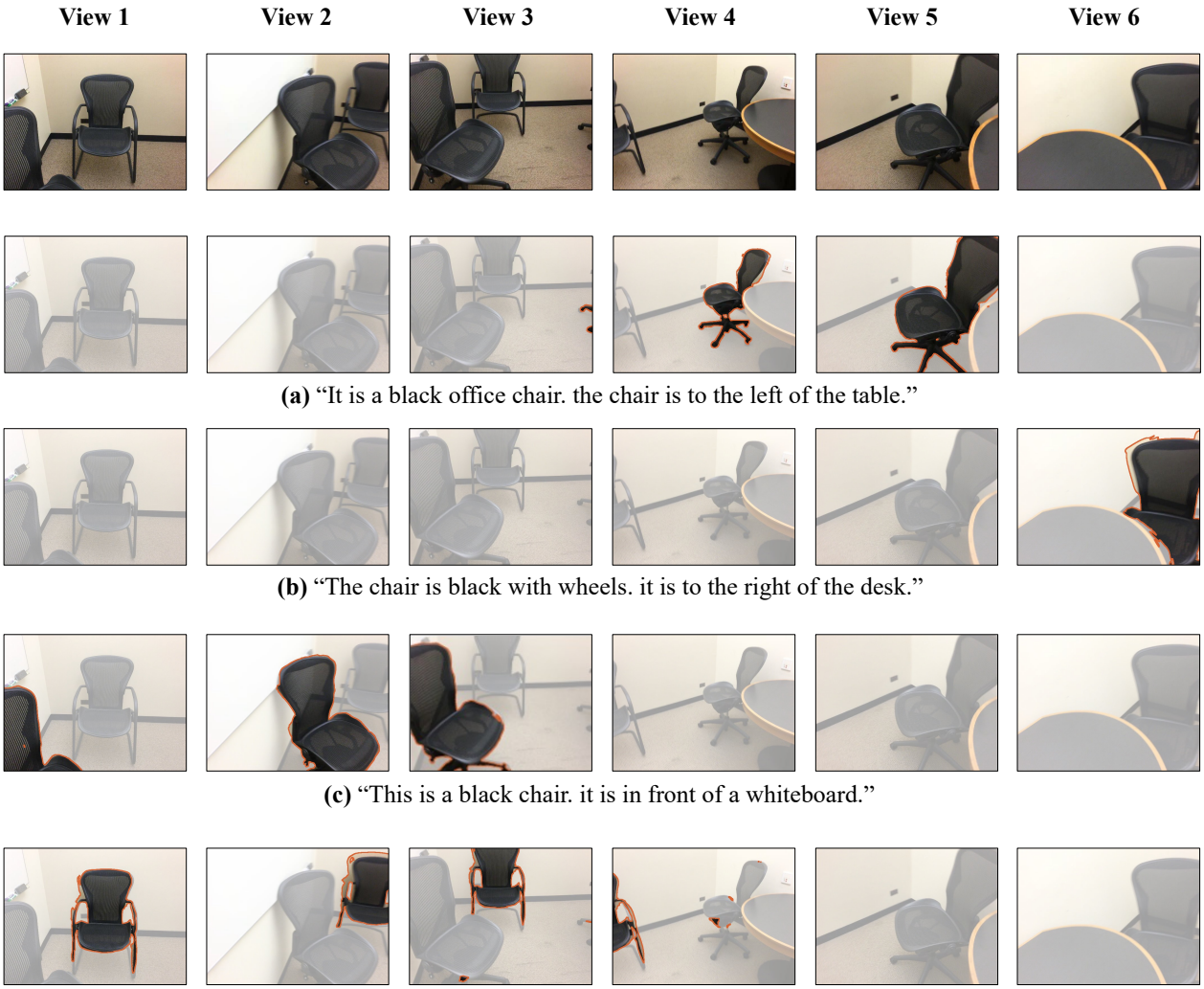

Figure A2. Qualitative results produced by our method on the ScanRefer [4] and ScanQA [2] datasets.

(c) "A small, round container placed on the side of the bear, near the cookie and with warm liquid."

(d) "A spoon-like object, functional and with a long handle, hangs from a perforated wall plate."

(a) "A miniature long-nosed toy, light and fun, standing quietly on the tabletop."

(b) "The long one on the tray."

Figure A3. Qualitative visualization on the **Ref-LERF** [9] dataset.

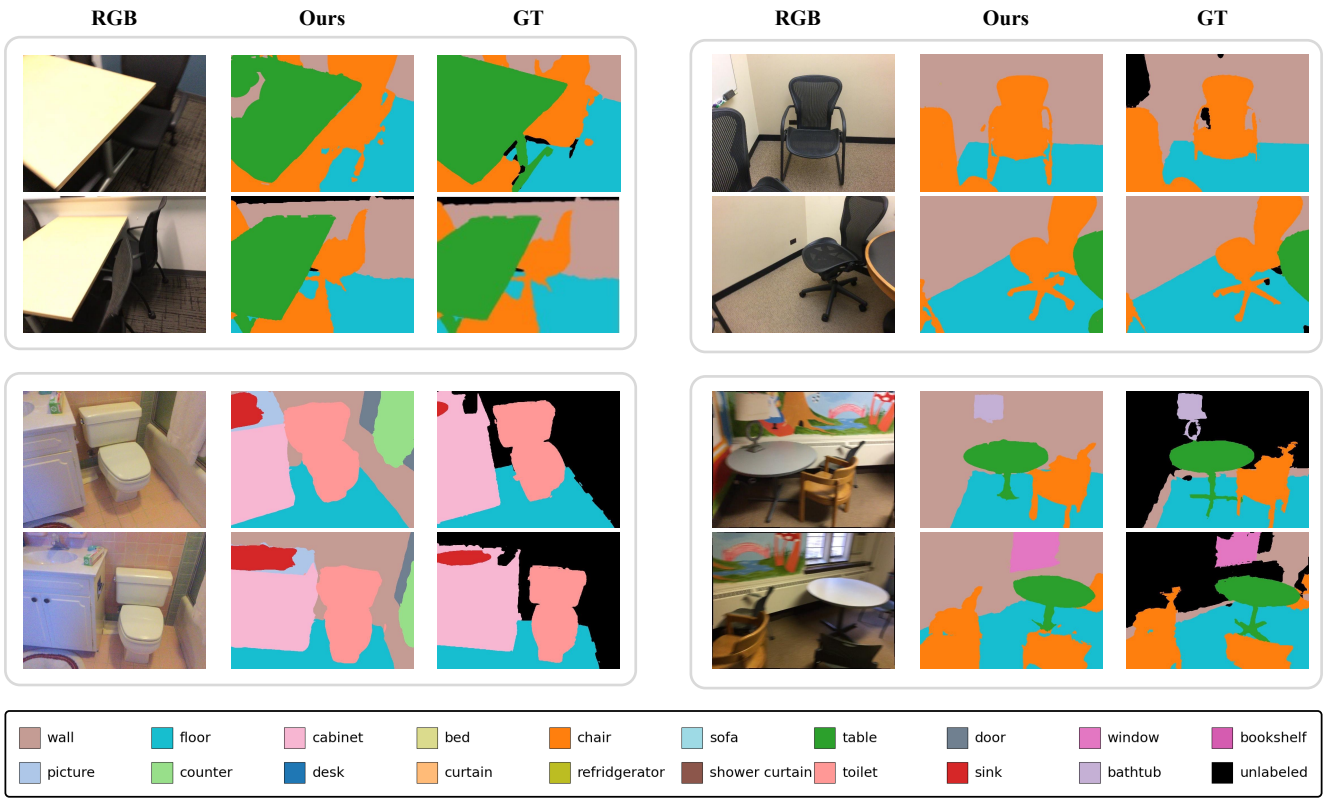

Figure A4. Open-vocabulary segmentation results on the val set of ScanNet [7].