PanoGS: Gaussian-based Panoptic Segmentation for 3D Open Vocabulary Scene Understanding
— Supplementary Material —

Hongjia Zhai[1] Hai Li[2] Zhenzhe Li[1] Xiaokun Pan[1] Yijia He[2] Guofeng Zhang[1†]
[1]State Key Lab of CAD & CG, Zhejiang University [2]RayNeo

In this supplementary document, we first provide the implementation details in Sec. A. Next, we supply additional analysis and more visualization results of our methods in Sec. B. The limitations of our approach are introduced in Sec. C.

A. Implementation Details

A.1. Datasets

Following the setting in [12], we use the selected 10 scenes from ScanNetV2 [2] for evaluation, and we train our model with every 20 frames from the given video images. Besides, following the same evaluation setting, we also use the provided point clouds for initialization, freeze the coordinates of the point clouds, and disable the original densification process in 3DGS [3]. The selected 10 scenes in [12] are *scene0000*, *scene0062*, *scene0070*, *scene0097*, *scene0140*, *scene0200*, *scene0347*, *scene0400*, *scene0590*, and *scene0645*. All selected scenes are evaluated on the *00* trajectory. The 19 classes used for text queries and evaluation are *wall*, *floor*, *cabinet*, *bed*, *sofa*, *table*, *door*, *window*, *bookshelf*, *picture*, *counter*, *desk*, *curtain*, *refrigerator*, *shower curtain*, *toilet*, *sink*, and *bathtub*.

For Replica dataset [9], we follow the setting in [6], the commonly-used 8 scenes {*room0-2*, *office0-4*} are used for evaluation, and we train our model with every 10 frames from the given video images. Since the Replica dataset [9] contains more attributes, two additional labels (*other furniture* and *ceiling*) are used for evaluation. The whole label set and its *Thing* and *Stuff* definitions are shown in Tab. A.

Class ID	Class Name	Type
0	wall	Stuff
1	floor	Stuff
2	cabinet	Thing
3	bed	Thing
4	chair	Thing
5	sofa	Thing
6	table	Thing
7	door	Stuff
8	window	Stuff
9	bookshelf	Stuff
10	picture	Stuff
11	counter	Stuff
12	desk	Thing
13	curtain	Stuff
14	refrigerator	Stuff
15	shower curtain	Stuff
16	toilet	Thing
17	sink	Thing
18	bathtub	Stuff
19	other furniture	Stuff
20	ceiling	Stuff

Table A. Classes and their type (*stuff* or *thing*) used in our experiments for ScanNetV2 [2] and Replica [9] datasets.

A.2. Details of Baselines

OpenGaussian†. When evaluating the performance of 3D point-level semantic segmentation, OpenGaussian [12] removes 3D Gaussian primitives with opacity less than a threshold and only conducts evaluation on the primitives with opacity greater than the threshold. In order to perform a fair and complete 3D panoptic segmentation evaluation, we do not filter primitives with small opacity. We mark this difference as OpenGaussian†.

LangSplat*. The results of LangSplat [7] shown in Table 2 of OpenGuassian [12] are too worse. Because LangSplat only works well in small scenes with obvious foreground objects and large overlaps. It does not perform well in indoor scene datasets, like ScanNetV2 [2] and Replica [9]. Therefore, to enhance its performance on indoor scenes, we use LSeg [5] to extract multi-view visual-language features and lift to 3D space to formulate a 3D feature cloud for compression and distillation. Following the original setting in [7], we use their auto-encoder to compress the fused 3D feature cloud and perform the same learning optimization

process in their paper.

A.3. Details of Our Approach

For the latent pyramid tri-planes, we use coarse feature planes with a resolution 20 cm and fine feature planes with 6 cm. All feature planes have 10 channels, and we employ sum operation on the features from all resolution channels, which results in 10-dimensional features. Our 3D language feature decoder is two-layer MLPs with $\{128, 512\}$ channels in the hidden layers, respectively. We train each scene on a single NVIDIA RTX 4090 card. During the reconstruction of 3D Gaussian primitives, we perform 20 optimization iterations of rendered appearance and depth for each frame. We use the Adam optimizer to update the attributes of 3D Gaussian primitives, the learning rates for rotation, opacity, and scaling are set to 0.001, 0.05, and 0.001, respectively. To learn the 3D language feature, we also use the Adam optimizer to update MLPs and latent parametric features, with a learning rate of 0.001. The betas and weight decay for the Adam optimizer are set to (0.9, 0.99) and 1e-6.

A.4. Evaluation Metrics

To evaluate the 3D segmentation performance, we use the 3D semantic segmentation and panoptic segmentation metrics with the following definition.

3D Semantic Segmentation. For class c, its mean Intersection over Union (mIoU) and mean Accuracy (mAcc.) metrics can be computed via the following equations:

$$\text{mIoU} = \frac{1}{C}\sum_{c=1}^{C}\frac{\text{TP}_c}{\text{TP}_c + \text{FN}_c} \tag{1}$$

$$\text{mAcc.} = \frac{1}{C}\sum_{c=1}^{C}\frac{\text{TP}_c}{\text{TP}_c + \text{FP}_c + \text{FN}_c} \tag{2}$$

where TP_c, FP_c, and FN_c denote **point-level** true positives, false positives, and false negatives for class c, respectively.

3D Panoptic Segmentation.. Following [1], we use the panoptic reconstruction quality (PRQ) as an average measure across the class categories with the following equations:

$$\text{PRQ}^c = \frac{\sum_{(i,j)\in \text{TP}_c}\text{IoU}(i,j)}{|\text{TP}_c| + 0.5\times|\text{FP}_c| + 0.5\times|\text{FN}_c|} \tag{3}$$

where TP_c, FP_c, and FN_c denote the **instance-level** true positives, false positives, and false negatives for class c, respectively.

For the evaluation of panoptic segmentation, we view the predicted semantic instance with an overlap larger than 25% with GTs as true positive matches. Besides, for the performance of classes in the *Thing* and *Stuff* sets, we denote them as PRQ (T) and PRQ (S), respectively.

B. More Experiment Results

Table B. 3D Instance segmentation results of ScanNetV2 [2] Dataset.

Methods	Input	Type	AP
OpenMask3D	3D	*Sup.*	**47.61**
MaskClustering	2D + 3D	*Z.S.*	33.92
OpenGaussian	2D	*Z.S.*	18.63
Ours	2D	*Z.S.*	<u>38.78</u>

Comparison of 3D Instance Segmentation. Tab. B shows additional instance segmentation results. 2D and 3D indicate using GT images and point clouds as input. *Sup.* indicates the supervised methods trained on the ScanNetV2, and *Z.S.* represents the zero-shot setting. Our approach achieves the best results among *2D* and *Z.S.* settings, only worse than the supervised method, OpenMask3D [10], which uses the supervised approach, Mask3D [8], to extract 3D instances.

Complexity Analysis of Graph Construction. In *scene0000*, for language-guided graph-cut, we construct edges (∼570k) for reconstructed primitives (∼82k) and use breadth-first search to construct super-primitives (∼500), which takes ∼4s. For progressive graph-based clustering, 4 iterations take ∼1.2s to obtain final 3D instances (52).

Table C. 3D panoptic segmentation results of different combination with graph cut (G.C) on ScanNetV2 [2] Dataset.

Ablations	PRQ (T)	PRQ (S)
OpenGaussian (*G.C*)	9.00	9.27
LangSplat (*G.C*)	8.68	14.39
Ours (*G.C*)	<u>19.36</u>	<u>25.15</u>
Ours (*Full*)	**33.84**	**36.22**

Table D. 3D panoptic segmentation results of different edge construction strategy on ScanNetV2 [2] Dataset.

Edges	PRQ (T)	PRQ (S)
Feat.	19.04	21.36
KLD	<u>29.77</u>	<u>32.59</u>
Ours	**33.84**	**36.22**

Performance of others using graph-cut (*G.C*). *G.C* is only

used to generate super-points and is already used in many methods [4, 11, 13, 14], such as MaskClustering [13]. As shown in Tab. B our AP is 4.86% higher than MaskClustering [13]. From Tab. C, we can know that *G.C* is not the key to improving instance segmentation performance. Our algorithm can greatly improve segmentation performance.

Gains of our edge strategy (JSD). As shown in Tab. D, using language features (Feat.) to construct edges is sensitive to the semantic similarity threshold and has poor performance. Compared with Kullback-Leibler divergence (KLD), our JSD is a symmetric function and our performance is better than KLD by a margin.

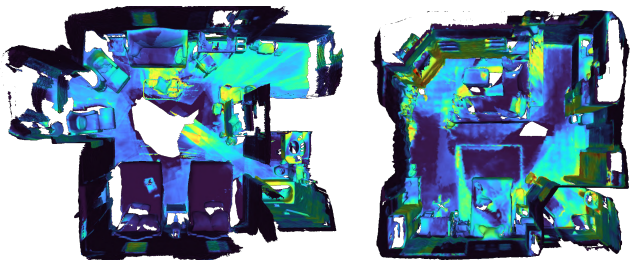


Figure A. Visualization of the uncertainty of fused feature cloud. We show the variance of the feature cloud that fused from multi-view 2D feature maps.

Visualization of 3D Feature Uncertainty. In Fig. A, we show the variance of the 3D feature cloud obtained by projecting the reconstructed 3D Gaussian primitives back to the multi-view 2D feature map and then fusing it. The brighter the color, the greater the uncertainty. The figure shows that in some semantically complex areas, the features extracted from multiple views maybe inconsistent and with low confidence.

Visualization of Learned Language Feature. In Fig. B, we show the visualization results of the language features learned by different methods. As can be seen from the figure, compared with the previous 3DGS-based methods (LangSplat [7] and OpenGaussian [12]), our method can learn smoother and more consistent language features (without noise primitives, which are affected by alpha-blending and its low opacity). Compared with OpenScene [6], the features we learned can distinguish finer-grained objects, such as carpet and floor, bed and sofa, *etc*.

Effects of Different Latent Code Lengths. In Fig. C, we show the effect of using different lengths of the latent codes g from the latent pyramid tri-plane. As can be seen from the figure, our method can still achieve relatively good results in the case of extremely short features (lengths less than 5).

Effects of Different Feature Cloud Resolutions. Compared to previous 3DGS-based methods, which bind the discrete features to each explicit Gaussian primitive, our approach regresses the language feature from a latent pyramid tri-plane representation. Previous discrete representation lacks the inherent feature smoothness and is easily affected by the resolution of the explicit representation. Our approach can achieve better segmentation results even under the supervision of sparse feature cloud. In Fig. D, we demonstrate the segmentation effect of our method under different spatial sampling rates of the 3D feature cloud. As can be seen from the figure, our method only requires 10% of the data to achieve a relatively stable segmentation performance.

Detailed Segmentation Performance of Each Scene. The detailed 3D semantic and panoptic segmentation performance of our approach and different baselines are shown in Tab. F and Tab. G, respectively.

Memory and Training time. In Tab. E, we show the training time of different methods, as well as the memory requirement for 3D open vocabulary scene understanding. Due to OpenScene [6] was trained on the entire dataset, including many data processing, *e.g.*, feature sampling, it is different from the current 3DGS-based approaches, so we do not list its training time. As shown in the table, OpenScene [6] and OpenGaussian [12] explicitly store the high-dimensional language features of each point or primitives, so they require a lot of memory. Though OpenGaussian [12] uses quantization and clustering to reduce the memory requirements, it is still higher than the approach (LangSplat [7]) of using MLPs to regress language features implicitly. We use a latent pyramid tri-plane to regress language features from a low-dimensional space, which reduces the number of parameters required, speeds up our convergence, and enables us to learn smoother and more accurate features.

Table E. Training time, and memory usage of different methods on scene *scene0000* from ScanNetV2 [2] Dataset.

Method	Training Time	Memory.
LangSplat [7]	52 mins	16.79 MB
OpenGaussian [12]	15 mins	65.8MB
OpenScene [6]	—	158.92 MB
Ours	12 mins	8.33 MB

More Visualization Results. In the main paper, we only show some visualization results of semantic and panoptic maps. So, we show more visualization results here. The open vocabulary query results, 3D semantic segmentation results on Replica [9] dataset, 3D panoptic segmentation results on ScanNetV2 [2] and Replica [9] datasets are shown in Fig. E, Fig. F, Fig. G, and Fig. H, respectively.

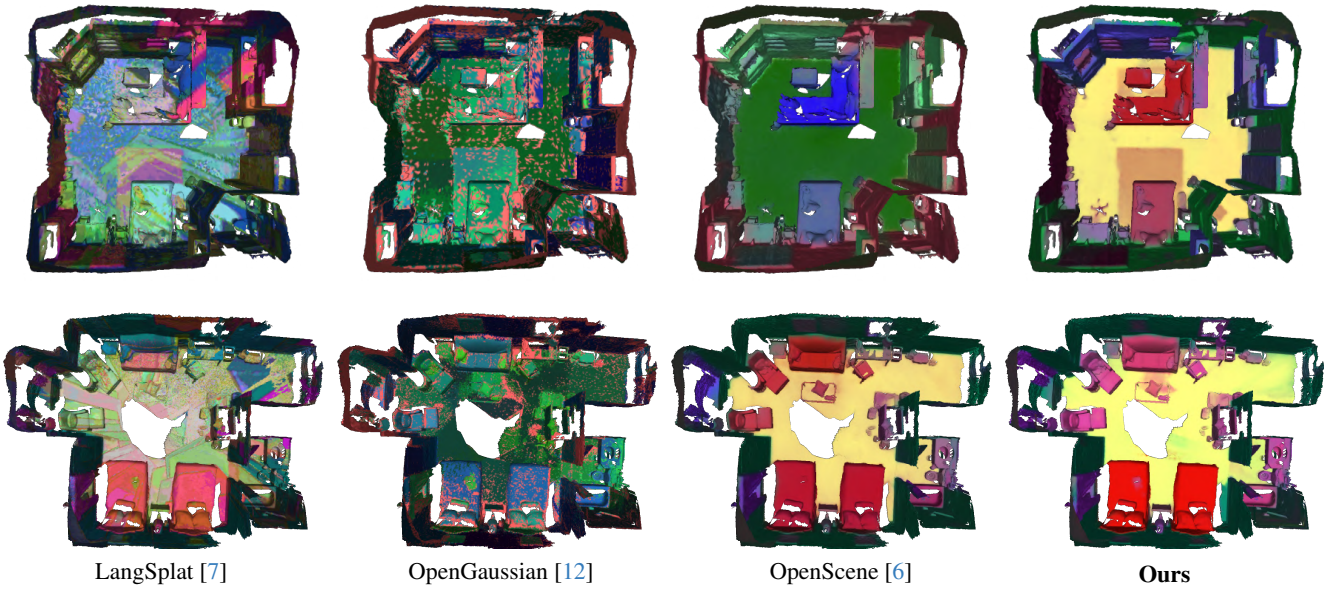


Figure B. Visualization results of the learned 3D language feature of different methods. For better visualization, we perform principal components analysis (PCA) on the learned high-dimensional language features.

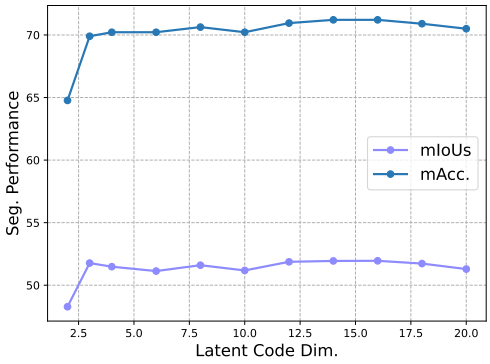


Figure C. 3D semantic segmentation performance with different latent code lengths on ScanNetV2 [2] dataset.

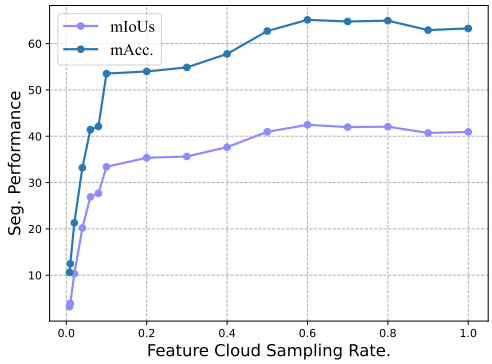


Figure D. 3D semantic segmentation performance with different feature cloud sampling rates on *scene0000* from ScanNetV2 [2] dataset.

C. Limitations

While our approach shows impressive performance on 3D open vocabulary scene understanding, there still remain two limitations. Firstly, our system relies on accurate 2D segmentation masks [4] , which are used to guide the clustering process of 3D Gaussian primitives. Besides, we can not generate multi-level 3D instance masks, which is also limited by the 2D image segmentation models.

References

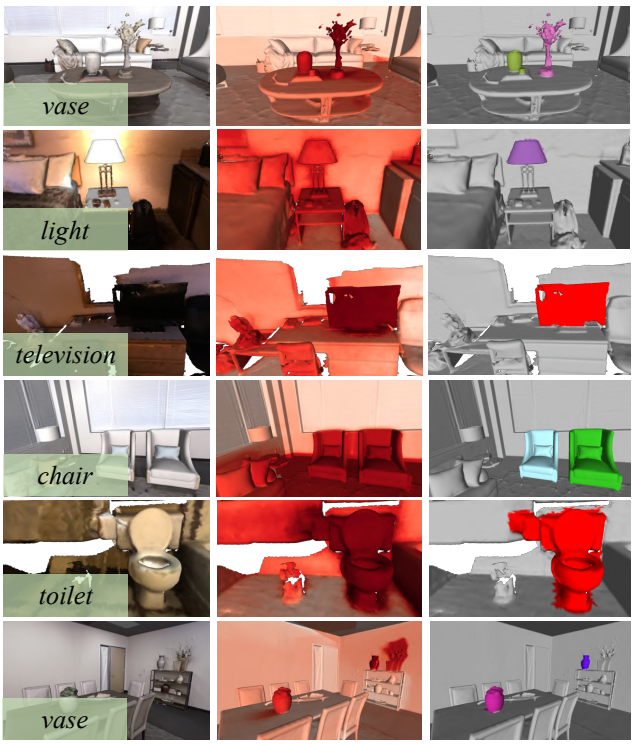
[1] Manuel Dahnert, Ji Hou, Matthias Nießner, and Angela Dai. Panoptic 3d scene reconstruction from a single rgb image. *Advances in Neural Information Processing Systems*, 34: 8282–8293, 2021. 2

[2] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. ScanNet: Richly-annotated 3D reconstructions of indoor scenes. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2432–2443, 2017. 1, 2, 3, 4, 6, 8

[3] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42 (4), 2023. 1

[4] 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 *IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023. 3, 4

[5] Boyi Li, Kilian Q Weinberger, Serge Belongie, Vladlen



RGB & Query OpenScene [6] Ours

Figure E. Qualitative results of open vocabulary query. The query text is in the lower left of the RGB image. For OpenScene, the redder the color, the higher the similarity. We use different colors to distinguish different instances found in the query.

Koltun, and Rene Ranftl. Language-driven semantic segmentation. In *International Conference on Learning Representations*, 2022. 1

[6] Songyou Peng, Kyle Genova, Chiyu "Max" Jiang, Andrea Tagliasacchi, Marc Pollefeys, and Thomas Funkhouser. Openscene: 3d scene understanding with open vocabularies. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 1, 3, 4, 5, 6, 7

[7] Minghan Qin, Wanhua Li, Jiawei Zhou, Haoqian Wang, and Hanspeter Pfister. Langsplat: 3d language gaussian splatting. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20051–20060, 2024. 1, 3, 4, 6

[8] Jonas Schult, Francis Engelmann, Alexander Hermans, Or Litany, Siyu Tang, and Bastian Leibe. Mask3D: Mask Transformer for 3D Semantic Instance Segmentation. 2023. 2

[9] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J. Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, Anton Clarkson, Mingfei Yan, Brian Budge, Yajie Yan, Xiaqing Pan, June Yon, Yuyang Zou, Kimberly Leon, Nigel Carter, Jesus Briales, Tyler Gillingham, Elias Mueggler, Luis Pesqueira, Manolis Savva, Dhruv Batra, Hauke M. Strasdat, Renzo De Nardi, Michael Goesele, Steven Lovegrove, and Richard Newcombe. The Replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019. 1, 3, 6, 7, 9

[10] Ayça Takmaz, Elisabetta Fedele, Robert W. Sumner, Marc Pollefeys, Federico Tombari, and Francis Engelmann. OpenMask3D: Open-Vocabulary 3D Instance Segmentation. In *Advances in Neural Information Processing Systems*, 2023. 2

[11] Thang Vu, Kookhoi Kim, Tung M Luu, Thanh Nguyen, and Chang D Yoo. Softgroup for 3d instance segmentation on point clouds. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2708–2717, 2022. 3, 6, 8, 9

[12] Yanmin Wu, Jiarui Meng, Haijie Li, Chenming Wu, Yahao Shi, Xinhua Cheng, Chen Zhao, Haocheng Feng, Errui Ding, Jingdong Wang, and Jian Zhang. Opengaussian: Towards point-level 3d gaussian-based open vocabulary understanding. 2024. 1, 3, 4, 6

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

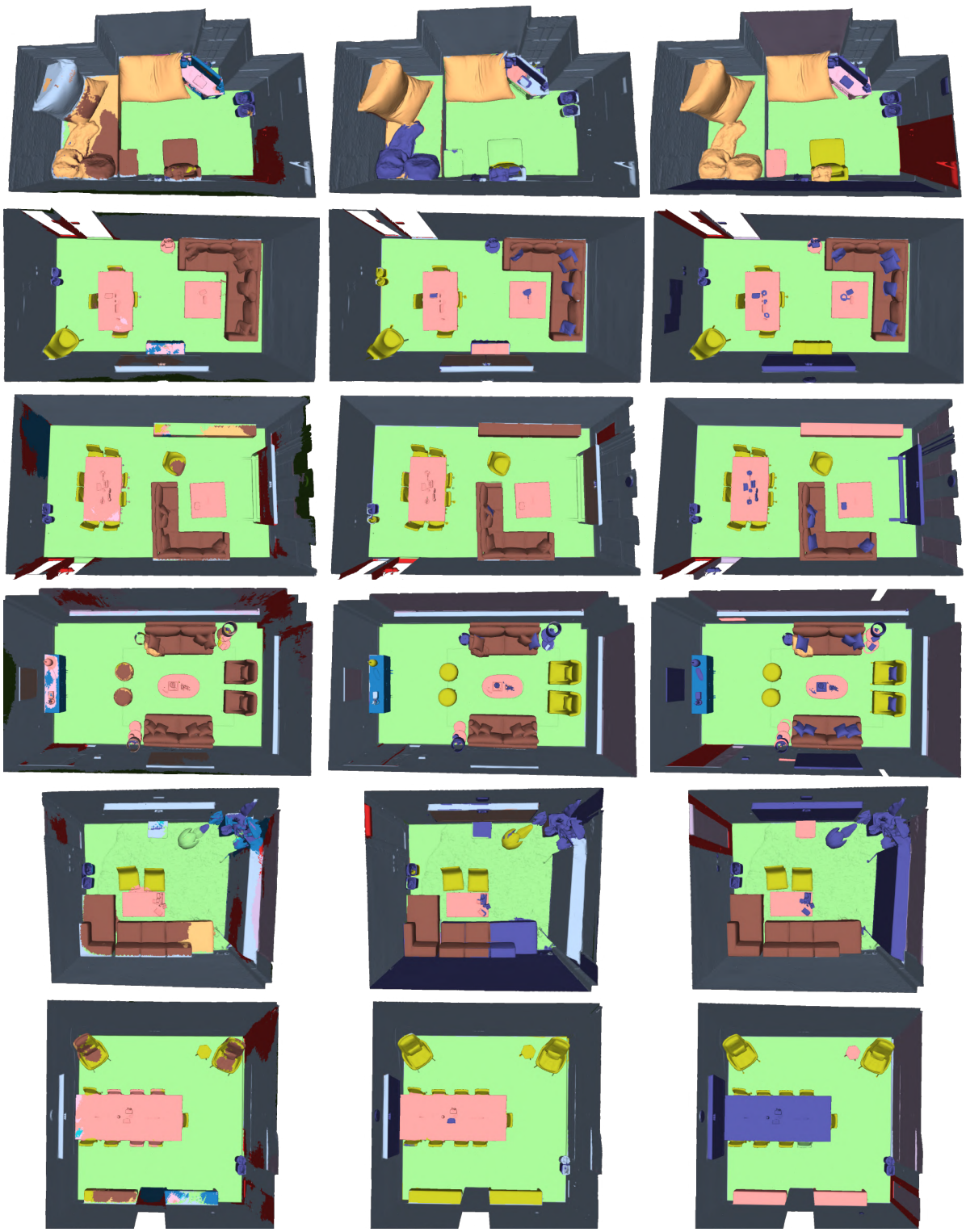
[14] Yingda Yin, Yuzheng Liu, Yang Xiao, Daniel Cohen-Or, Jingwei Huang, and Baoquan Chen. Sai3d: Segment any instance in 3d scenes. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3292–3302, 2024. 3

Table F. 3D semantic and panoptic segmentation results on Replica dataset [9]. The results of OpenScene [6] are obtained from their pre-trained model. $*$ indicates our better implementation of LangSplat [7]. Compared approaches use SoftGroup [11] to D generate instance mask.

Method	Metric	Room 0	Room 1	Room 2	Office 0	Office 1	Office 2	Office 3	Office 4
LangSplat* [7]	mIoU	7.56	4.26	2.58	3.78	3.21	2.7	7.64	5.51
	mACC.	11.68	6.48	5.69	6.4	5.77	6.05	12.88	9.19
	PRQ (T)	11.46	9.68	0	9.29	0	0	12.56	9.58
	PRQ (S)	2.98	2.69	0	2.28	0	0	3.24	4.82
OpenScene(*Dis.*) [6]	mIoU	49.43	61.89	49.7	37.44	36.4	48.19	44.06	37.63
	mACC.	68.74	79.24	70.95	54.55	45.60	63.97	65.47	55.2
	PRQ (T)	30.97	10.57	18.13	49.83	35.62	34.43	27.52	20.14
	PRQ (S)	7.22	8.69	8.15	4.84	9.93	13.18	6.66	9.87
OpenScene(*Ens.*) [6]	mIoU	45.27	63.39	38.36	59.05	34.98	57.55	53.65	40.02
	mACC.	61.57	74.77	53.47	70.89	45.35	70.04	72.05	50.18
	PRQ (T)	25.39	13.93	23.75	44.29	10.49	36.94	38.57	17.38
	PRQ (S)	5.87	12.70	5.83	12.13	11.78	17.33	8.65	11.86
Ours	mIoU	73.81	72.88	73.65	46.71	24.89	57.46	56.91	42.33
	mACC.	82.06	84.66	89.80	57.16	35.56	66.79	71.72	52.06
	PRQ (T)	53.15	17.03	56.62	54.21	23.58	44.81	39.73	47.00
	PRQ (S)	10.60	31.40	41.60	17.28	29.97	56.87	20.99	37.03

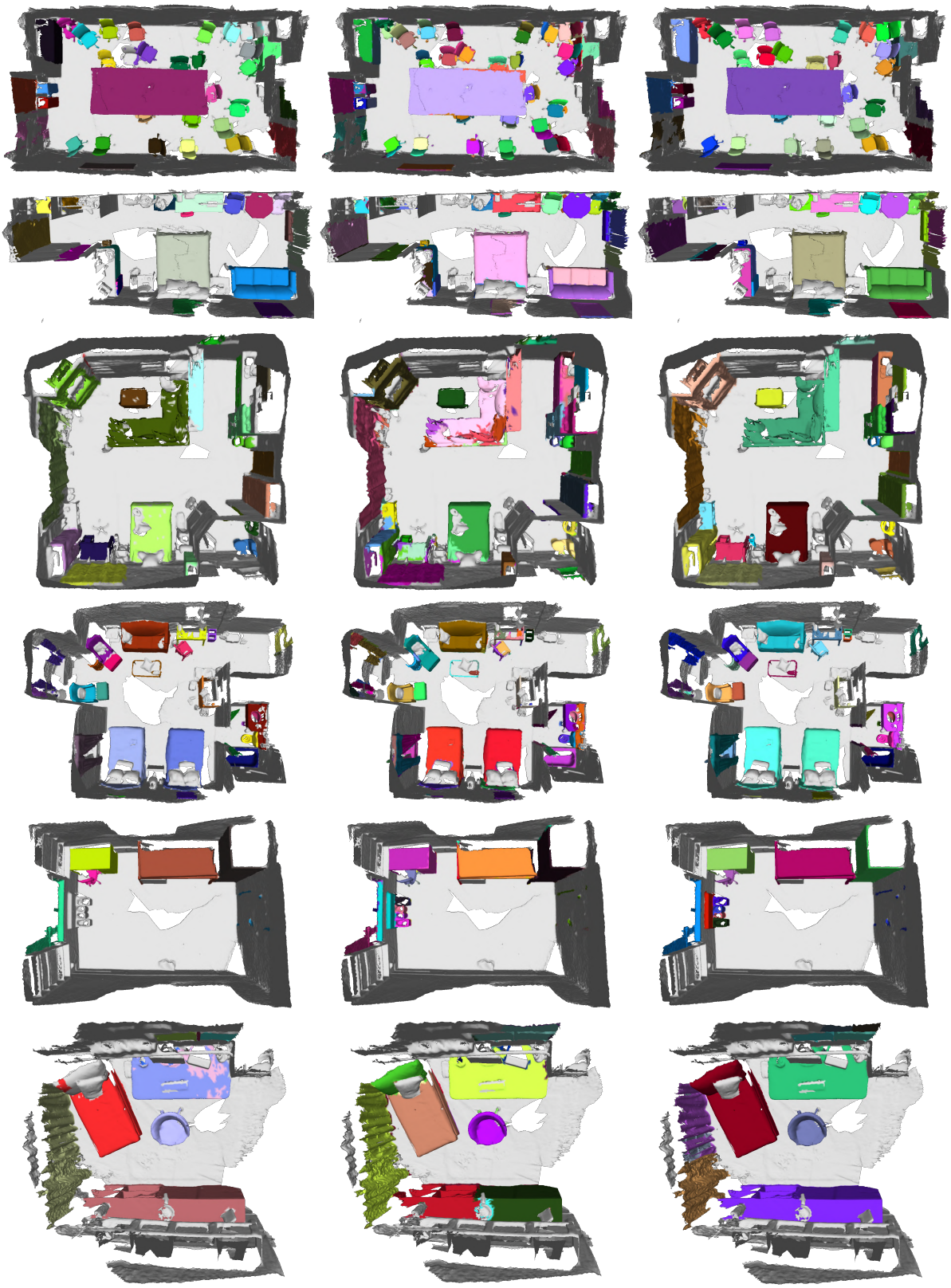
Table G. 3D semantic and panoptic segmentation results on ScanNetV2 [2]. The results of OpenScene [6] are obtained from their pre-trained model. $*$ indicates our better implementation. † indicates no Gaussian filter is used for the evaluation of panoptic segmentation. Compared approaches use SoftGroup [11] to D generate instance mask.

Method	Metrics	0000	0062	0070	0097	0140	0200	0347	0400	0590	0645
LangSplat* [7]	mIoU	16.68	40.42	23.25	43.52	30.25	44.08	31.79	30.01	16.05	18.70
	mACC.	30.50	50.97	36.13	59.57	54.22	62.79	44.97	52.16	30.31	31.33
	PRQ (T)	19.35	31.57	10.96	23.29	46.11	33.71	10.82	18.85	13.49	17.60
	PRQ (S)	12.52	45.99	15.44	42.89	21.47	45.03	34.10	34.14	18.59	14.30
OpenGaussian† [12]	mIoU	23.46	25.53	15.26	28.81	17.36	27.39	24.96	15.72	21.89	21.54
	mACC.	37.73	46.61	27.66	47.08	36.03	41.62	36.53	31.69	31.54	37.05
	PRQ (T)	11.44	27.54	21.99	39.09	29.02	22.21	25.72	0	25.33	26.37
	PRQ (S)	18.32	31.92	15.58	23.01	11.43	26.44	20.88	24.91	12.75	11.94
OpenScene(*Dis.*) [6]	mIoU	37.07	61.19	42.17	64.37	39.23	49.31	63.28	42.08	36.44	28.34
	mACC.	57.15	79.52	65.25	87.05	62.02	73.45	84.90	71.01	59.34	47.71
	PRQ (T)	33.87	51.51	38.95	44.03	63.76	25.42	46.17	49.35	42.03	42.02
	PRQ (S)	24.55	59.01	29.09	47.64	22.35	55.88	55.80	47.00	32.63	27.39
OpenScene(*Ens.*) [6]	mIoU	41.90	62.19	43.17	65.52	40.34	50.62	64.17	43.08	36.33	29.23
	mACC.	64.50	79.52	65.25	87.05	62.09	73.63	84.90	71.01	58.26	47.72
	PRQ (T)	37.06	51.51	38.95	44.03	63.76	24.33	46.17	49.35	40.70	41.70
	PRQ (S)	27.69	59.01	29.09	47.78	22.38	56.02	55.65	47.00	32.05	27.15
Ours	mIoU	41.72	73.25	46.98	59.50	51.76	45.21	58.43	46.55	40.19	43.76
	mACC.	64.49	90.25	65.75	78.11	67.11	65.41	77.92	72.65	55.52	64.84
	PRQ (T)	28.53	36.86	36.45	42.06	55.31	21.04	20.00	25.84	32.94	40.96
	PRQ (S)	37.59	45.53	25.59	43.68	32.47	34.15	41.07	45.26	29.71	26.26



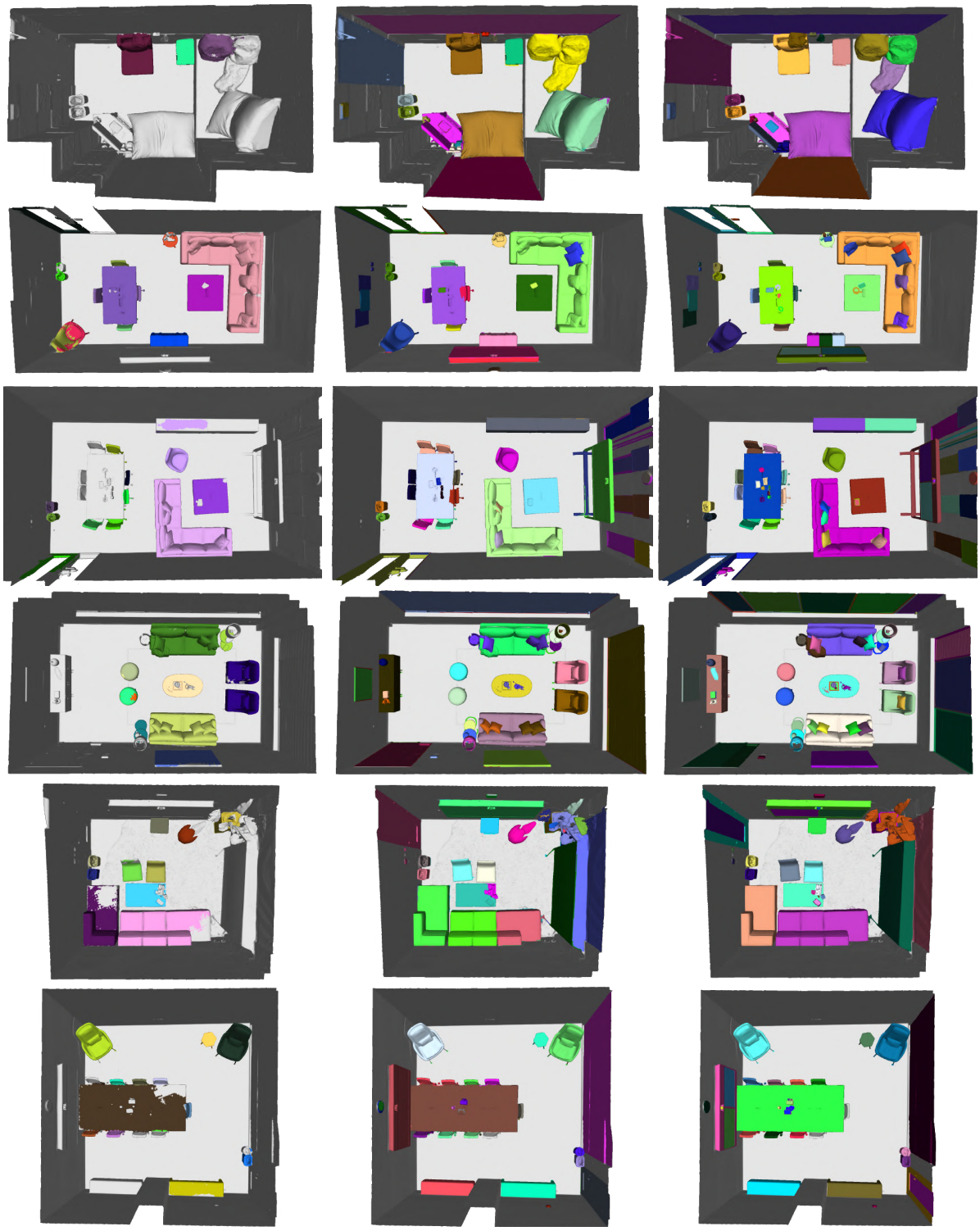
OpenScene [6] **Ours** GT Panoptic

Figure F. Qualitative 3D semantic segmentation comparison. We show some reconstructed semantic maps selected from Replica [9] datasets.



SoftGroup [11] **Ours** GT Panoptic

Figure G. Qualitative 3D panoptic segmentation comparison. We show some reconstructed panoptic maps selected from ScanNetV2 [2].



SoftGroup [11] **Ours** GT Panoptic

Figure H. Qualitative 3D panoptic segmentation comparison. We show some reconstructed panoptic maps selected from Replica [9] dataset.