Edge-Aware 3D Instance Segmentation Network with Intelligent Semantic Prior

Supplementary Material

Overview

In this supplementary material, we supply further explanations and visualizations of our main paper "Edge-Aware 3D Instance Segmentation Network with Intelligent Semantic Prior". We first explain more details about the implementation and large-scale datasets [1, 2, 6, 12] (Sec. 1). Also, we provide more qualitative analysis for additional experimental results in diverse scenarios (Sec. 2).

1. Experimental Setup

1.1. Datasets

We train and evaluate the overall performance using four landmark datasets for 3D instance segmentation: ScanNetV2 [6], ScanNet200 [12], S3DIS [1], and STPLS3D [2].

ScanNetV2. The ScanNetV2 [6] dataset consists of high-quality, large-scale 3D point data with 1613 scenes from various room types, including hotels, libraries, and offices. It comprises 1201 scenes for the training, 312 for the validation, and 100 unseen in the training for the test. Each scene is captured with RGB-D cameras and categorized with 20 classes of semantic and instance segmentation labels.

ScanNet200. To cover diverse real-world environments, ScanNet200 [12] expands the original ScanNet [6] dataset with fine-grained 200 categories. ScanNet200 enables a more practical evaluation of how well methods can handle less common instances (*e.g*. *coat rack* or *candle*) and challenging, long-tail distribution scenes. In our experiments, we evaluate using 18 classes for ScanNetV2 and 198 classes for ScanNet200, excluding *wall* and *floor* categories.

S3DIS. The S3DIS [1] dataset is another extensive benchmark, comprising 271 scenes from 6 areas within three different buildings. It is annotated with 13 semantic categories, and we employ all these classes for evaluation. Following the standard protocol [1, 8, 13], we report segmentation performance on Area 5 (the scenes in Area 5 for validation and the others for training) and 6-fold cross-validation.

STPLS3D. The STPLS3D [2] dataset is a large-scale aerial photogrammetry dataset with real and synthetic 3D point clouds. It includes 25 urban scenes covering 6 km², categorized into 14 classes. Following [3, 17], we use scenes 5, 10, 15, 20, and 25 for evaluation and the rest for training.

1.2. Implementation Details

Using PyTorch deep learning framework, we implement our experimental setup with the following settings. For ScanNet [6], we adopt a 5-layer U-Net [11] with five hierarchical resolutions as the backbone network, following [3, 7, 9, 14, 17]. The transformer decoder comprises 6 layers, each with 8 heads, and we employ Fourier absolute position encoding [15] with the temperature set to 10,000. We train our model for 512 epochs with a batch size of 4, using a single RTX3090 GPU. For S3DIS [1] and STPLS3D [2], we utilize Res16UNet34C of MinkowskiEngine [5] as the feature backbone. Here, we utilize a modified transformer decoder inspired by Mask2Former [4], where cross and self-attention are swapped. This decoder layer leverages 8-headed attention and a feedforward network with 1024 dimensional features. We train our model for 600 epochs with a batch size of 4, utilizing a single RTX A6000 GPU. We apply the AdamW [10] optimizer with a learning rate of 2×10^{-4} for all four datasets. We also utilize a polynomial scheduler for ScanNet and a one-cycle scheduler for others. During training, voxels are randomly sampled at fixed numbers, whereas all voxels are used for evaluation. This sampling strategy is not only memory-efficient but can also serve as a dropout. We select the top 100 instances for evaluation according to the highest scores. Further, the nearest points k and the threshold value τ are set to 20 and 5, respectively, for dynamically extracting pseudo edge labels.

2. Additional Experimental Studies

Parameters for Pseudo Edge Label Calculation In this section, we analyze the variations of dynamically generated pseudo edge labels based on key parameters (k and τ). To supervise the Edge Prediction Module, we first calculate pseudo edge labels for all input 3D point cloud scenes. As described in our main paper (Section 3.3), we compare the instance labels of the central point with those of its k neighbor points. Then, if the count of distinct label points exceeds a predefined threshold value τ, we set the central point as the edge label. Here, it is noteworthy that these parameters (k and τ) significantly influence the clarity and thickness of the pseudo edges. Therefore, we perform qualitative examinations using various parameter values to generate the robust pseudo edge labels. As shown in Fig. 1 (row 1), we first fix the τ at 5 and experiment with different k values (10, 20, 30): a lower k (*i.e*., 10) results in incomplete noisy edges, while a higher k (*i.e*., 30) thickens the edges without fine details. Also, in the second row, we keep the k at 20 and vary the τ values (2, 5, 10): if the τ is too large (*i.e*., 10), the pseudo edge labels becomes insufficient, lacking in details. Through these studies, we ultimately generate useful pseudo edge labels with optimal parameters ($k = 20$ and $\tau = 5$) for Edge Prediction Module optimization, encouraging the network to utilize edge-advanced features.

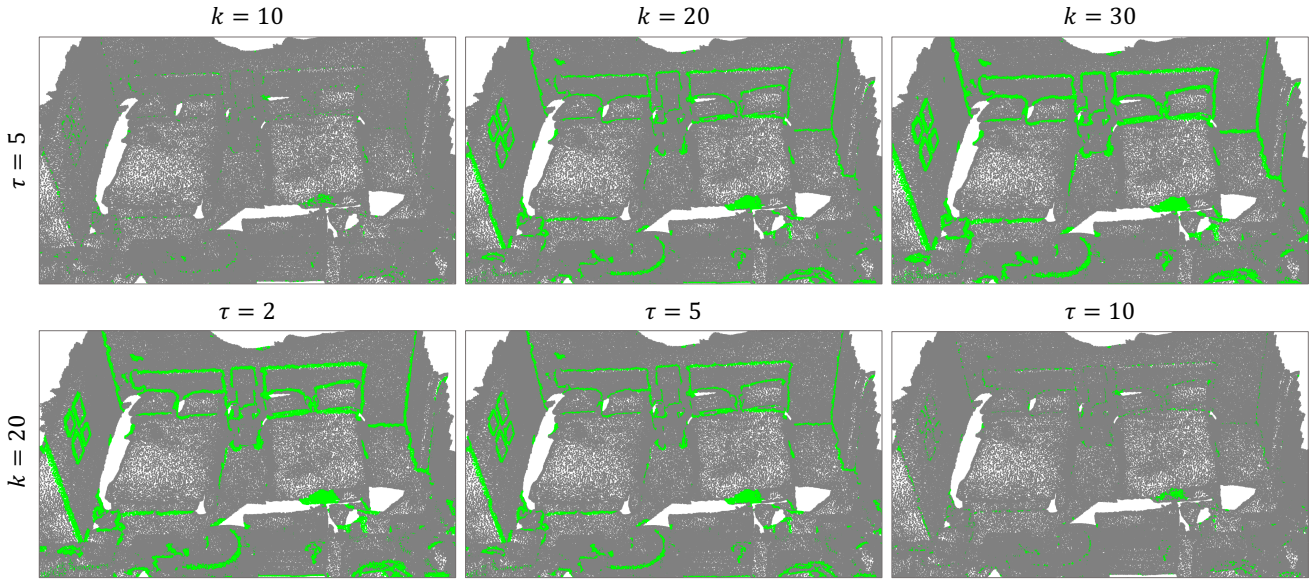

Figure 1. Qualitative visualizations of pseudo edge labels with diverse parameters. In the first row, we visualize the pseudo edges (green) with various neighboring points k at the fixed threshold value τ. The second row represents the effect of the threshold value τ at the fixed neighboring points k. Here, we empirically observe that each parameter significantly impacts the quality of the pseudo edges. Note that we ultimately calculate practical pseudo edge labels using optimal parameters ($k = 20$ and $\tau = 5$) to supervise the Edge Prediction Module, guiding the network to reduce misclassifications near edges with edge-aware features. Best viewed in color.

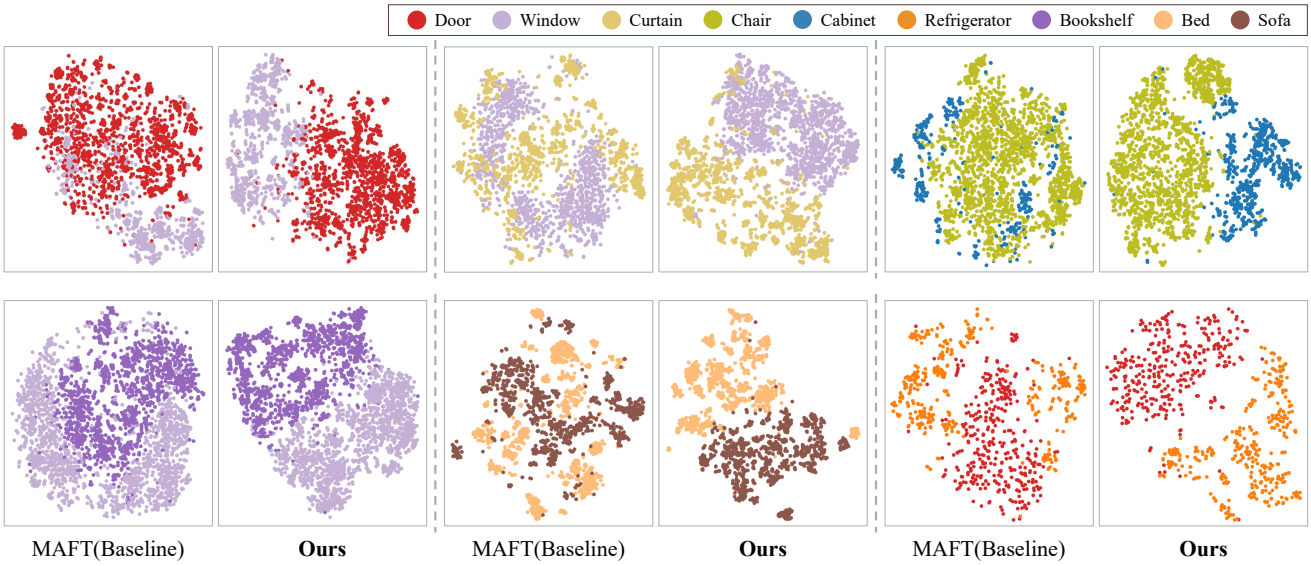

Figure 2. t-SNE [16] visualizations of query features representing each instance. Compared to the existing state-of-the-art model MAFT [8], which produces disorganized clusters for challenging instances with similar appearances, ours creates more distinctive clusters. Here, we confirm that our Semantic Network effectively encourages the network to learn instance-specific semantic knowledge from intelligent text embedding priors. Note that the color map (top right) represents semantic labels. Best viewed in color.

Significance of Semantic Priors. In addition to our main paper (Section 4.4), we validate the efficacy of our suggested Semantic Network. We present more comparative t-SNE [16] visualizations of query features for diverse cases of visually similar instances (*e.g.*, *window* and *curtain*, *bed* and *sofa*) in Fig. 2. The query features of the baseline model (MAFT [8]) are wildly distributed without pattern, causing challenges in identifying instances in feature space. This issue is also evident in the category probability maps of individual queries in Fig. 3. The MAFT queries struggle

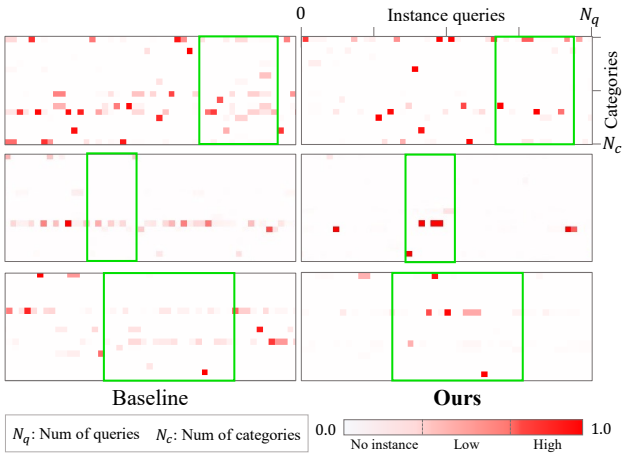

Figure 3. Category probability distribution of instance queries. The x-axis denotes queries that represent corresponding instances and the y-axis denotes categories. The baseline model (MAFT [8]) estimates categories with unclear and fuzzy low (light red) probabilities. However, our model classifies instances with relatively high (dark red) confidence of probabilities. Best viewed in color.

to estimate categories with clear probabilities, uncertainly spread across multiple categories, leading to large duplications along the y-axis. To relieve this confusion, our Semantic Network explicitly instructs the basic queries using text embeddings to learn deep semantic details. Finally, ours forms relatively clear and distinct clusters with highly confident instance queries. These impressive results highlight the semantic recognition capabilities of our model, which proficiently leverages instance-specific semantic knowledge.

Visual Comparison. In this section, we present additional qualitative visualization results of our framework EASE against the existing state-of-the-art models, MAFT [8] and Mask3D [13], in Fig. 4 and Fig. 5. For better comparison, we also visualize yellow and green colored boxes to emphasize the critical differences in semantic (Sem.) and instance (Inst.) results. First, as shown in Fig. 4, our model outperforms existing methods in accurately classifying instances with similar shapes (*e.g.*, *refrigerator* and *cabinet*, *chair* and *sofa*). These results demonstrate that context details from text embeddings can serve as intelligent semantic insights for effectively perceiving complex 3D instances. Furthermore, as shown in Fig. 5, in Scene8, Scene10, and Scene11, our model consistently segments single objects as a whole unit, unlike the baseline model, which tends to fragment them into multiple parts. Also, our model accurately distinguishes between objects situated closely, as illustrated in Scene7, Scene9, and Scene12. These outcomes underscore the effectiveness of our edge module, which enhances the perception of the spatial range of diverse instances.

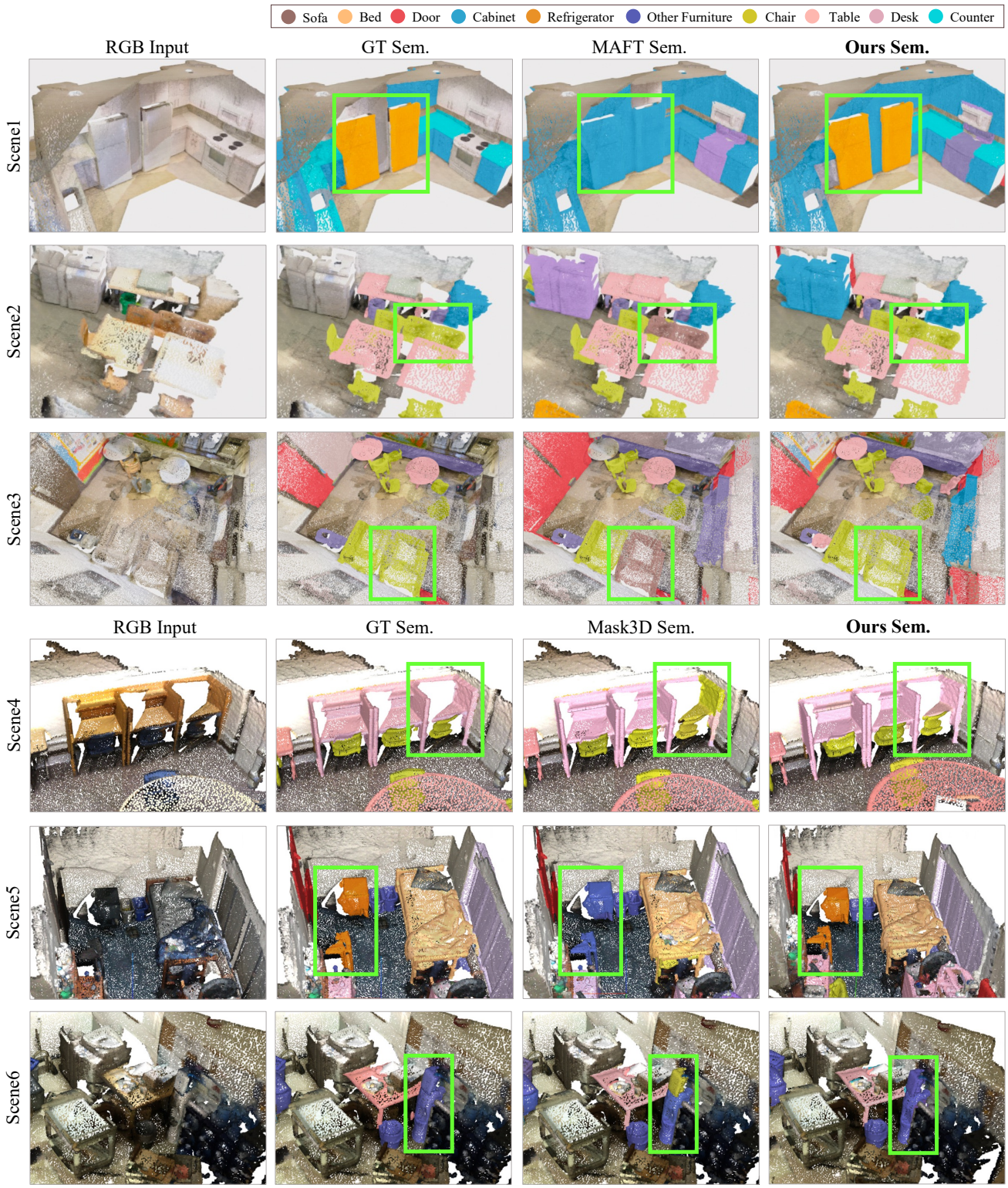

Figure 4. Qualitative comparisons of 3D Instance Segmentation performance on the ScanNetV2 [6] validation set. We visualize semantic (Sem.) masks of the baseline models (MAFT [8], Mask3D [13]) and ours with Ground Truth (GT) masks. The key differences are highlighted using green-colored boxes for better comparison. Note that the color map (top right) represents semantic labels.

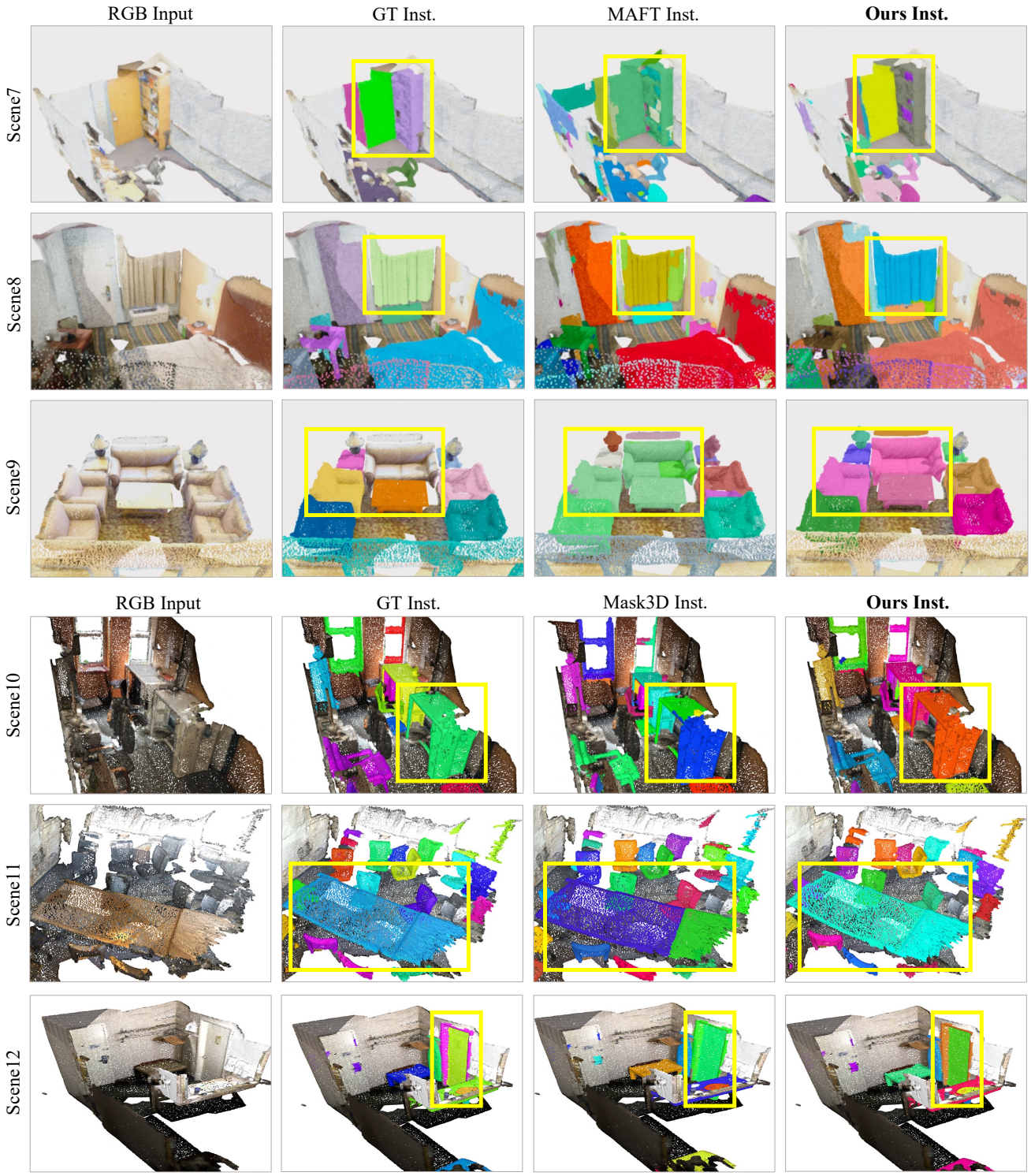

Figure 5. Qualitative comparisons of 3D Instance Segmentation performance on the ScanNetV2 [6] validation set. We visualize instance (Inst.) masks of the baseline models (MAFT [8], Mask3D [13]) and ours with Ground Truth (GT) masks. The key differences are highlighted using yellow-colored boxes. Here, the color does not represent semantic classes but is used to distinguish different instances.

References

[1] Iro Armeni, Ozan Sener, Amir R Zamir, Helen Jiang, Ioannis Brilakis, Martin Fischer, and Silvio Savarese. 3d semantic parsing of large-scale indoor spaces. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1534–1543, 2016. 1

[2] Meida Chen, Qingyong Hu, Zifan Yu, Hugues Thomas, Andrew Feng, Yu Hou, Kyle McCullough, Fengbo Ren, and Lucio Soibelman. Stpls3d: A large-scale synthetic and real aerial photogrammetry 3d point cloud dataset. *arXiv preprint arXiv:2203.09065*, 2022. 1

[3] Shaoyu Chen, Jiemin Fang, Qian Zhang, Wenyu Liu, and Xinggang Wang. Hierarchical aggregation for 3d instance segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15467–15476, 2021. 1

[4] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1290–1299, 2022. 1

[5] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3075–3084, 2019. 1

[6] 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 conference on computer vision and pattern recognition*, pages 5828–5839, 2017. 1, 4, 5

[7] Li Jiang, Hengshuang Zhao, Shaoshuai Shi, Shu Liu, Chi-Wing Fu, and Jiaya Jia. Pointgroup: Dual-set point grouping for 3d instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and Pattern recognition*, pages 4867–4876, 2020. 1

[8] Xin Lai, Yuhui Yuan, Ruihang Chu, Yukang Chen, Han Hu, and Jiaya Jia. Mask-attention-free transformer for 3d instance segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3693–3703, 2023. 1, 2, 3, 4, 5

[9] Zhihao Liang, Zhihao Li, Songcen Xu, Mingkui Tan, and Kui Jia. Instance segmentation in 3d scenes using semantic superpoint tree networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2783–2792, 2021. 1

[10] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 1

[11] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015. 1

[12] David Rozenberszki, Or Litany, and Angela Dai. Language-grounded indoor 3d semantic segmentation in the wild. In *European Conference on Computer Vision*, pages 125–141. Springer, 2022. 1

[13] Jonas Schult, Francis Engelmann, Alexander Hermans, Or Litany, Siyu Tang, and Bastian Leibe. Mask3d for 3d semantic instance segmentation. *arXiv preprint arXiv:2210.03105*, 2022. 1, 3, 4, 5

[14] Jiahao Sun, Chunmei Qing, Junpeng Tan, and Xiangmin Xu. Superpoint transformer for 3d scene instance segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 2393–2401, 2023. 1

[15] Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in Neural Information Processing Systems*, 33:7537–7547, 2020. 1

[16] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9 (11), 2008. 2

[17] Thang Vu, Kookhoi Kim, Tung M Luu, Thanh Nguyen, and Chang D Yoo. Softgroup for 3d instance segmentation on point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2708–2717, 2022. 1