Supplementary Material for "Learning Superpoint Graph Cut for 3D Instance Segmentation"

Le Hui[†]**, Linghua Tang**[†]**, Yaqi Shen, Jin Xie**[*]**, Jian Yang**[*]
PCA Lab, Nanjing University of Science and Technology, China
{le.hui, tanglinghua, syq, csjxie, csjyang}@njust.edu.cn

A Overview

This supplementary material provides more details on network architecture, visualization, and ablation study of our method. We also analyze the limitation and discuss the impact of our method. Specifically, in Sec. B, we provide specific network architecture and more details about the superpoint feature extraction. In Sec. C, we provide more visualization results, quantitative results, and ablation study of our proposed method. In Sec. D, we discuss the limitations and impacts of our method.

B Network Architecture

Superpoint feature extraction. To extract superpoint features, we first use the submanifold convolution to extract voxel-level features, where the raw 3D points are converted into voxels by performing voxelization. Specifically, we follow [5] and use a 3D U-Net structure constructed by five submanifold convolution blocks. The produced feature dimension of 3D U-Net is 32. Then, we average the voxel features belonging to the same instance to produce the initial superpoint features, where the superpoints are generated by using the method in [7]. After that, we adopt conditioned edge convolution network [6] on the superpoint graph to extract superpoint features, where the long-range context information can be captured through graph convolution. Finally, we can obtain 32-dimensional superpoint features. In addition, we use two classification heads based on voxel features and superpoint features to predict the semantic classes, respectively.

Edge feature learning network. In the edge feature learning network, we learn edge embeddings in both the coordinate and feature space to predict edge scores for proposing instances. For edge embeddings in the coordinate space, we first use a two-layer multi-layer perceptron (MLP) network to predict the 3-dimensional offsets of superpoints. Then, based on the shifted coordinate space, we construct the k-NN graphs for each node pair $(u, v) \in E$, where E is the edge set on the superpoint graph. We adopt a three-layer MLP network to learn channel-wise attention weights in cross-graph attention for extracting edge embeddings. Similarly, for edge embeddings in the feature space, we also construct the k-NN graphs for each node pair $(u, v) \in E$, and adopt a three-layer MLP network to learn channel-wise attention weights in cross-graph attention for extracting edge embeddings. Finally, for each $(u, v) \in E$, we combine the edge embeddings in both the coordinate and feature spaces and the geometric distance (L_2 distance) between u and v in the shifted coordinated space. The two-layer MLP network followed by Sigmoid is used to produce edge scores of the superpoint graph.

Superpoint graph cut network. In the superpoint graph network, we present the bilateral graph attention to extract instance embeddings. Specifically, we first adopt the three-layer MLP network

[†]Equal Contributions, [*]Corresponding authors.

Le Hui, Linghua Tang, Yaqi Shen, Jin Xie, and Jian Yang are with PCA Lab, Key Lab of Intelligent Perception and Systems for High-Dimensional Information of Ministry of Education, and Jiangsu Key Lab of Image and Video Understanding for Social Security, School of Computer Science and Engineering, Nanjing University of Science and Technology, China.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

on the coordinate and feature spaces to capture the geometry differences between the superpoints of the instance and the instance centroid, respectively. Here we can obtain two attention weights from the coordinate and feature spaces. Then, we execute element-wise production of the two attention weights to obtain the bilateral weight. After that, we use the softmax function to normalize the bilateral weight. Finally, we sum the weighed superpoint embeddings within the instance to obtain the instance embedding.

C More Results

C.1 Quantitative Results

ScanNet v2 test set. In Table 1, we also report the 3D instance segmentation results on the ScanNet v2 test set in terms of AP, AP_{50}, AP_{25}. Note that the results in the table are mean average precision over 18 categories. It can be observed that our method achieves the best results in terms of AP and the second-best results in terms of AP_{50} and AP_{25}. For AP_{25}, the results of our method are 14%, 12%, and 16% lower than those of SoftGroup [9] in the categories of the bookshelf, other furniture, and refrigerator, respectively. These three categories bring about 3% performance drop on the mean AP_{25} over all categories. Similarly, these categories also lead to the performance drop in terms of AP_{50}. In addition, the samples of these three categories are much smaller than other categories, such as the chair, table, and cabinet. We visualize the ScanNet v2 test set and observe that the numbers of the bookshelf and refrigerator are about 12 and 13 samples. Due to the small number of samples, the performance on these categories fluctuates greatly. If you predict one more correct instance, the performance will be improved by about 8%. Therefore, about two object prediction errors cause the performance gap in these categories that have a small number of samples. Nonetheless, our method can achieve high-quality object instances with higher IoU scores. Since other methods cannot effectively obtain high-quality instances, $i.e.$, IoU score > 0.5, our method achieves the best results on AP.

ScanNet v2 validation set. In addition to the mean AP, AP_{50}, and AP_{25} in the main paper, we report the detailed performance of each category in Table 2. Note that the results of SoftGroup [9] are obtained by using the pretrained models provided by the official code. Note that the obtained results are slightly lower than those listed in the main paper of SoftGroup. It can be observed that the performance gap between our method and SoftGroup is small on the bookshelf and refrigerator categories. Compared with the test set of 100 scenes, the validation set has 312 scenes. The numbers of samples of the bookshelf and refrigerator are 77 and 57, respectively. Compared with the performance gap in the test set of these two categories, the performance gap in the validation set is reduced. Considering our method performs well in most of the categories, our method achieves the best results in terms of AP, AP_{50}, and AP_{25}.

C.2 Ablation Study

Analysis of area constraint. We perform the ablation study on the ScanNet v2 validation set to demonstrate the effectiveness of the area constraint used for learning compact offset on the superpoint graph. Specifically, we compute the mean absolute error (dubbed "Offset MAE") that indicates the L_1 distance between the shifted superpoint centers and the instance centers. We also compute the standard deviation (dubbed "Offset SD") of the L_2 distance between the shifted superpoint centers and the instance centers. In Table 3, we report the Offset MAE, Offset SD, as well as the AP, AP_{50}, and AP_{25}. It can be observed that our method equipped with the area constraint (dubbed "GraphCut w/ area constraint") can achieve the best results. In Figure 1, we also visualize the shifted superpoint centers, which are computed by adding the original superpoint centers and the predicted superpoint offsets. Note that the nodes of the superpoint graph "GraphCut w/ area constraint" and "GraphCut w/o area constraint" are colored with the same color as the graph-level instance ground truth for a better view.

Analysis of soft threshold θ. To mitigate semantic prediction errors, we use the soft threshold θ to associate the superpoints with multiple classes. Here, we adjust the θ from 0 to 1 to conduct ablation studies on the ScanNet v2 validation set. Table 4 lists the average precision for different values of the θ. It can be observed that the performance is stable between 0.1 to 0.4 for θ. According to the experimental results, we set $\theta = 0.2$ in this paper.

Table 1: Instance segmentation results on the ScanNet v2 hidden test set in terms of mAP, mAP_{50}, and mAP_{25}. Note that the best results are highlighted in **bold** and the second-best results are underlined. The reported results are from the ScanNet benchmark on 19/5/2022.

Method	AP	bathtub	bed	bookshe	cabinet	chair	counter	curtain	desk	door	other	picture	fridge	s.curtain	sink	sofa	table	toilet	window
3D-SIS [4]	16.1	40.7	15.5	6.8	4.3	34.6	0.1	13.4	0.5	8.8	10.6	3.7	13.5	32.1	2.8	33.9	11.6	46.6	9.3
GSPN [13]	15.8	35.6	17.3	11.3	14.0	35.9	1.2	2.3	3.9	13.4	12.3	0.8	8.9	14.9	11.7	22.1	12.8	56.3	9.4
3D-MPA [2]	35.5	45.7	48.4	29.9	27.7	59.1	4.7	33.2	21.2	21.7	27.8	19.3	41.3	41.0	19.5	57.4	35.2	84.9	21.3
PointGroup [5]	40.7	63.9	49.6	41.5	24.3	64.5	2.1	57.0	11.4	21.1	35.9	21.7	42.8	66.0	25.6	56.2	34.1	86.0	29.1
SSTNet [7]	<u>50.6</u>	73.8	54.9	49.7	31.6	69.3	17.8	37.7	19.8	33.0	46.3	57.6	51.5	85.7	49.4	63.7	45.7	94.3	29.0
HAIS [1]	45.7	70.4	56.1	45.7	36.4	67.3	4.6	54.7	19.4	30.8	42.6	28.8	45.4	71.1	26.2	56.3	43.4	88.9	34.4
SoftGroup [9]	50.4	66.7	57.9	37.2	38.1	69.4	7.2	67.7	30.3	38.7	53.1	31.9	58.2	75.4	31.8	64.3	49.2	90.7	38.8
GraphCut (ours)	**55.2**	100	61.1	43.8	39.2	71.4	13.9	59.8	32.7	38.9	51.0	59.8	42.7	75.4	46.3	76.1	58.8	90.3	32.9

Method	AP_{50}	bathtub	bed	bookshe	cabinet	chair	counter	curtain	desk	door	other	picture	fridge	s.curtain	sink	sofa	table	toilet	window
3D-SIS [4]	38.2	100	43.2	24.5	19.0	57.7	1.3	26.3	3.3	32.0	24.0	7.5	42.2	85.7	11.7	69.9	27.1	88.3	23.5
GSPN [13]	30.6	50.0	40.5	31.1	34.8	58.9	5.4	6.8	12.6	28.3	29.0	2.8	21.9	21.4	33.1	39.6	27.5	82.1	24.5
3D-MPA [2]	61.1	100	83.3	76.5	52.6	75.6	13.6	58.8	47.0	43.8	43.2	35.8	65.0	85.7	42.9	76.5	55.7	100	43.0
PointGroup [5]	63.6	100	76.5	62.4	50.5	79.7	11.6	69.6	38.4	44.1	55.9	47.6	59.6	100	66.6	75.6	55.6	99.7	51.3
SSTNet [7]	69.8	100	69.7	88.8	55.6	80.3	38.7	62.6	41.7	55.6	58.5	70.2	60.0	100	82.4	72.0	69.2	100	50.9
HAIS [1]	69.9	100	84.9	82.0	67.5	80.8	27.9	75.7	46.5	51.7	59.6	55.9	60.0	100	65.4	76.7	67.6	99.4	56.0
SoftGroup [9]	**76.1**	100	80.8	84.5	71.6	86.2	24.3	82.4	65.5	62.0	73.4	69.9	79.1	98.1	71.6	84.4	76.9	100	59.4
GraphCut (ours)	<u>73.2</u>	100	78.8	72.4	64.2	85.9	24.8	78.7	61.8	59.6	65.3	72.2	58.3	100	76.6	86.1	82.5	100	50.4

Method	AP_{25}	bathtub	bed	bookshe	cabinet	chair	counter	curtain	desk	door	other	picture	fridge	s.curtain	sink	sofa	table	toilet	window
3D-SIS [4]	55.8	100	77.3	61.4	50.3	69.1	20.0	41.2	49.8	54.6	31.1	10.3	60.0	85.7	38.2	79.9	44.5	93.8	37.1
GSPN [13]	54.4	50.0	65.5	66.1	66.3	76.5	43.2	21.4	61.2	58.4	49.9	20.4	28.6	42.9	65.5	65.0	53.9	95.0	49.9
3D-MPA [2]	73.7	100	93.3	78.5	79.4	83.1	27.9	58.8	69.5	61.6	55.9	55.6	65.0	100	80.9	87.5	69.6	100	60.8
PointGroup [5]	77.8	100	90.0	79.8	71.5	86.3	49.3	70.6	89.5	56.9	70.1	57.6	63.9	100	88.0	85.1	71.9	99.7	70.9
SSTNet [7]	78.9	100	84.0	88.8	71.7	83.5	71.7	68.4	62.7	72.4	65.2	72.7	60.0	100	91.2	82.2	75.7	100	69.1
HAIS [1]	80.3	100	99.4	82.0	75.9	85.5	55.4	88.2	82.7	61.5	67.6	63.8	64.6	100	91.2	79.7	76.7	99.4	72.6
SoftGroup [9]	**86.5**	100	96.9	86.0	86.0	91.3	55.8	89.9	91.1	76.0	82.8	73.6	80.2	98.1	91.9	87.5	87.7	100	82.0
GraphCut (ours)	<u>83.2</u>	100	92.2	72.4	79.8	90.2	70.1	85.6	85.9	71.5	70.6	74.8	64.0	100	93.4	86.2	88.0	100	72.9

Table 2: The detailed instance segmentation results on the ScanNet v2 validation set in terms of mAP, mAP_{50}, and mAP_{25}. Note that the best results are highlighted in **bold**.

Method	AP	bathtub	bed	bookshe	cabinet	chair	counter	curtain	desk	door	other	picture	fridge	s.curtain	sink	sofa	table	toilet	window
SoftGroup [9]	45.7	67.1	45.9	35.4	38.7	72.0	14.5	39.8	28.6	33.7	41.4	34.9	49.2	50.4	38.1	52.2	54.8	93.3	32.7
GraphCut (ours)	**52.2**	65.6	60.1	24.0	41.8	82.1	27.2	43.6	34.1	44.6	50.1	54.0	55.1	56.7	58.4	52.2	58.8	96.9	35.2

Method	AP_{50}	bathtub	bed	bookshe	cabinet	chair	counter	curtain	desk	door	other	picture	fridge	s.curtain	sink	sofa	table	toilet	window
SoftGroup [9]	67.1	87.1	69.9	66.4	62.2	84.8	41.5	58.4	59.1	50.8	60.2	54.4	68.4	68.6	65.5	72.4	78.8	100	58.6
GraphCut (ours)	**69.1**	84.8	81.9	48.8	62.1	93.9	44.1	60.5	62.8	62.0	62.9	60.6	66.2	70.2	77.6	71.2	80.0	100	56.0

Method	AP_{25}	bathtub	bed	bookshe	cabinet	chair	counter	curtain	desk	door	other	picture	fridge	s.curtain	sink	sofa	table	toilet	window
SoftGroup [9]	78.6	87.1	76.6	76.3	76.7	88.6	72.3	76.1	81.2	61.3	71.7	65.4	73.1	75.2	85.6	88.0	85.7	100	73.6
GraphCut (ours)	**79.3**	86.7	84.8	68.2	74.2	96.2	71.7	76.2	80.2	76.1	70.4	66.5	66.8	79.9	90.6	83.8	85.4	100	74.0

Table 3: Ablation study on the ScanNet v2 validation set for area constraint. "Offset MAE" denotes the mean absolute error (*i.e.*, L_1 distance) between the shifted superpoint centers and the instance centers, and "Offset SD" denotes the standard deviation of the L_2 distance between the shifted superpoint centers and the instance centers. The best results are highlighted in **bold**.

Method	Offset MAE	Offset SD	AP	AP_{50}	AP_{25}
GraphCut w/o area constraint	0.373	0.182	51.3	68.0	78.6
GraphCut w/ area constraint	**0.319**	**0.115**	**52.2**	**69.1**	**79.3**

Table 4: Ablation study on the ScanNet v2 validation set for different values of the hyper-parameter θ. The best results are highlighted in **bold**.

Metrics	$\theta = 0.01$	$\theta = 0.1$	$\theta = 0.2$	$\theta = 0.3$	$\theta = 0.4$	$\theta = 0.5$	$\theta = 0.6$	$\theta = 0.7$	$\theta = 0.8$	$\theta = 0.9$
AP	46.4	52.1	**52.2**	52.0	51.5	50.6	49.3	46.3	42.5	34.3
AP_{50}	60.5	69.0	69.1	**69.2**	68.9	67.5	66.1	63.1	58.9	49.4
AP_{25}	68.4	68.9	**79.3**	79.1	79.0	78.3	77.3	75.1	72.0	62.6

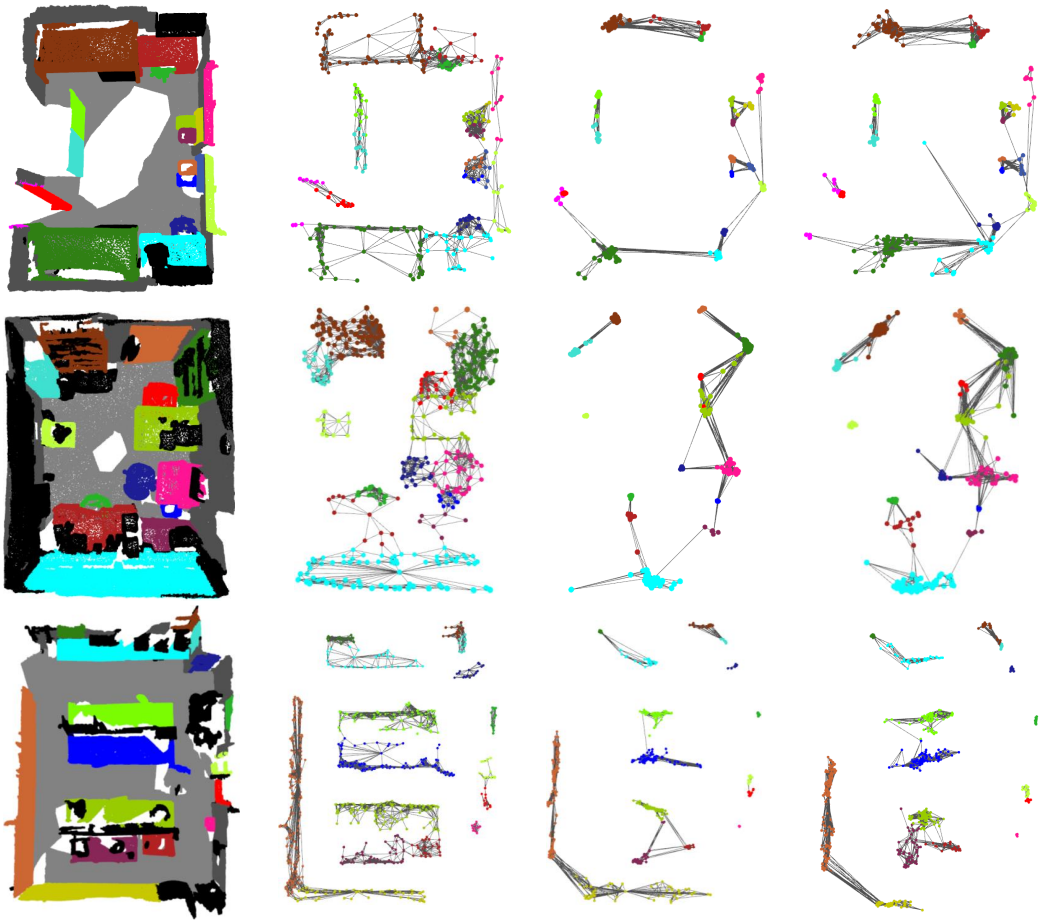

Figure 1: The visualization results of superpoint offsets on the superpoint graph. Note that the nodes of the superpoint graphs "GraphCut w/ area constraint" and "GraphCut w/o area constraint" are colored with the same color as the graph-level instance ground truth for a better view.

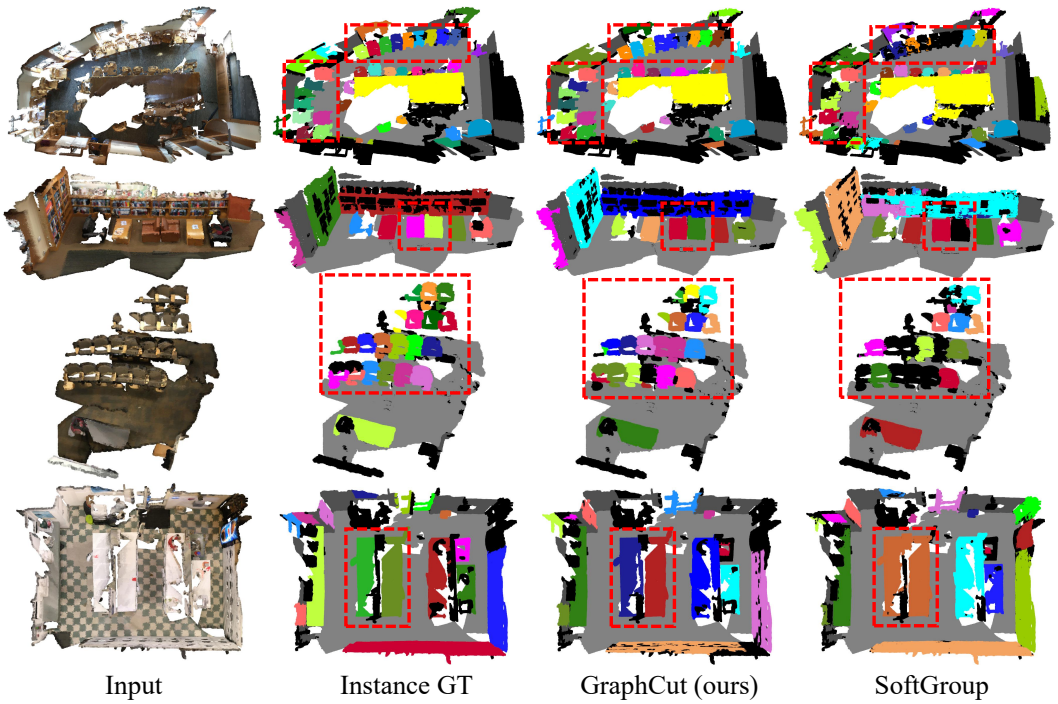

Figure 2: The visualization results of our method and previous state-of-the-art method SoftGroup [9] on the ScanNet v2 validation set. Red rectangles show the differences between the two methods of 3D instance segmentation.

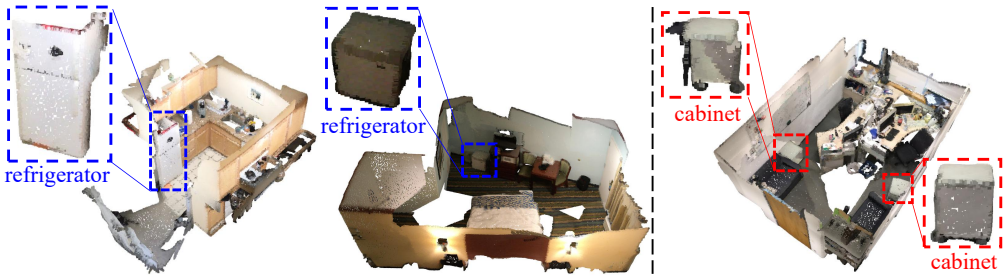

Figure 3: The visualization samples of the refrigerators and cabinets on the ScanNet v2 validation set.

C.3 Visualization Results

Visualization results. In Figure 2, we show more visualization results of 3D instance segmentation on the ScanNet v2 validation set. Compared with SoftGroup [9], our method can effectively segment clustered objects, such as chairs.

Visual process of graph cutting. In order to show the detailed 3D instance segmentation process of our method, we provide the visualization results of each step of our method in Figures 4 and 5. Specifically, given a raw point cloud, we first oversegment it into superpoints and then construct superpoint graph. After that, we perform superpoint graph cutting on the superpoint graph, where the red edges are cut and the blue edges are remained. In this way, we can obtain graph-level instances of the point cloud. Finally, we convert graph-level instances into point-level instances. Note that the instances are randomly colored.

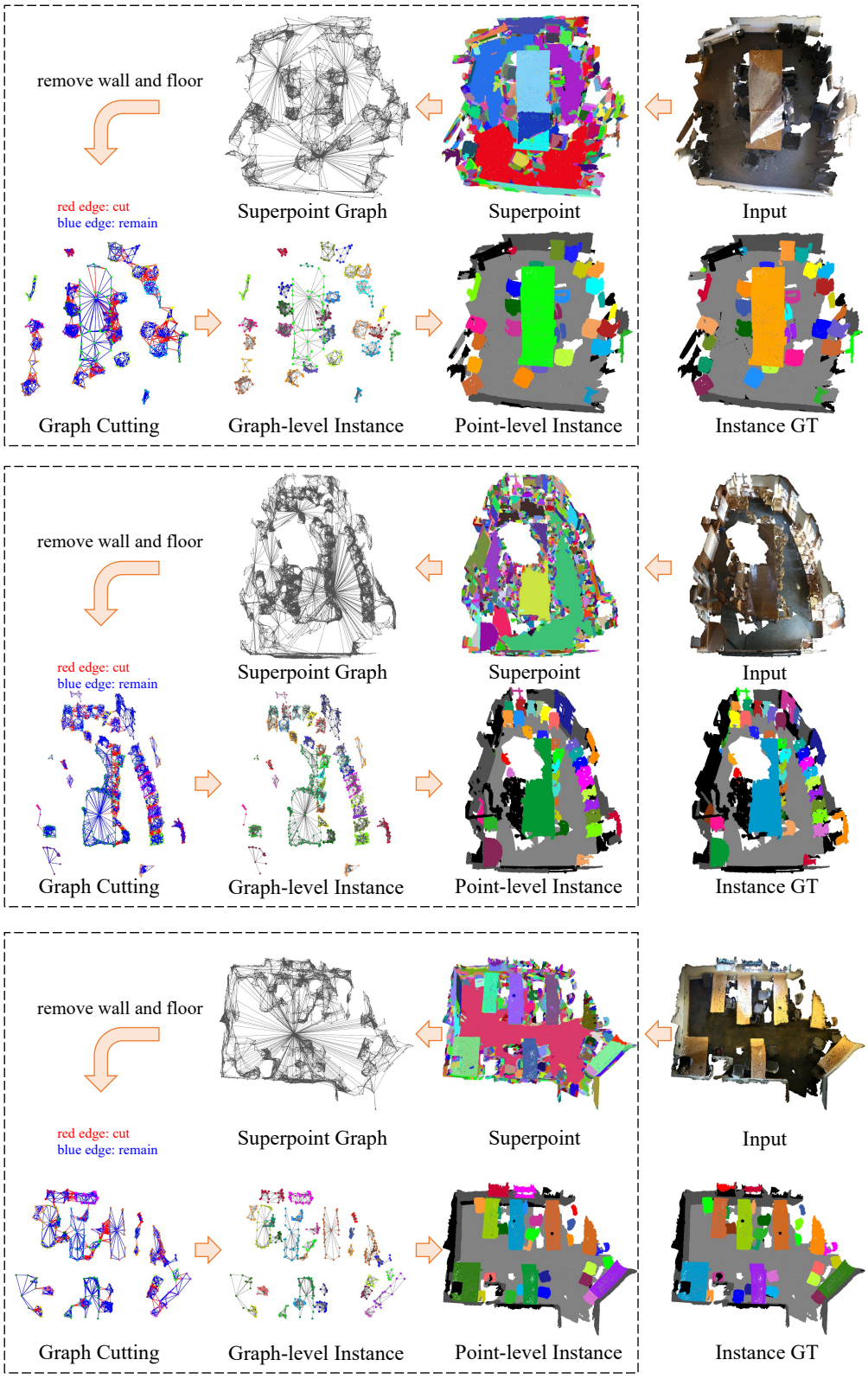

Figure 4: The 3D instance segmentation process of our method on the ScanNet v2 validation set. Note that instances are randomly colored.

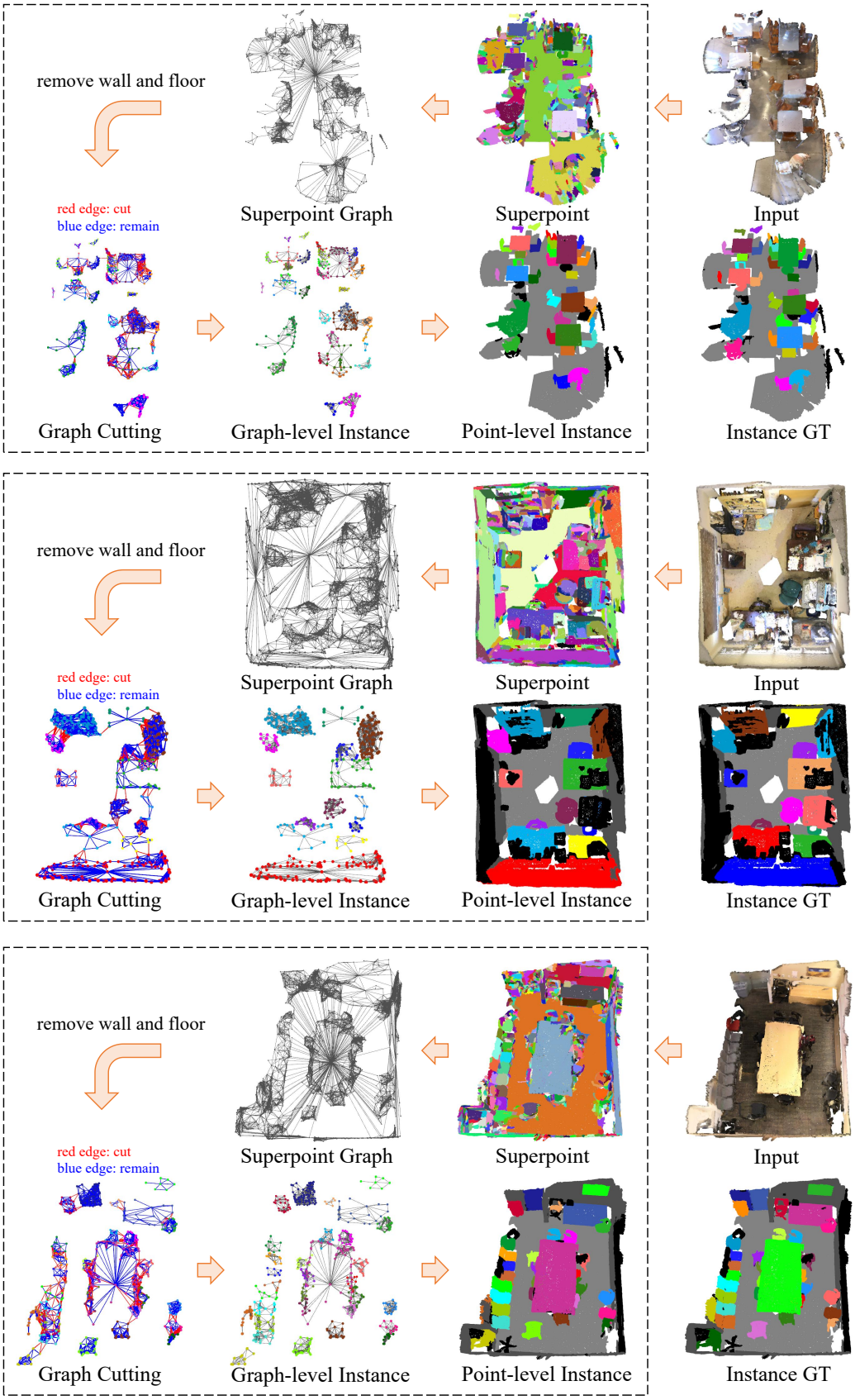

Figure 5: The 3D instance segmentation process of our method on the ScanNet v2 validation set. Note that instances are randomly colored.

Table 5: Inference time of different methods on the ScanNet v2 validation set. For a fair comparison, the runtime is computed on the same TITAN X GPU model.

Method	Superpoint (ms)	Component Time (ms)	Total (ms)
SGPN [10]	-	Backbone (2080), Group merging (149000), Block merging (7119)	158439
ASIS [11]	-	Backbone (2083) Mean shift (172711), Block merging (7119)	181913
GSPN [13]	-	Backbone (2083), Point sampling (9559), Neighbour search (1500)	12702
3D-BoNet [12]	-	Backbone (2083), SCN (667), Block merging (7119)	9202
GICN [8]	-	Backbone (1497), SCN (667), Block merging (7119)	8615
OccuSeg [3]	-	Backbone (189), Supervoxel (1202), Clustering (513)	1904
PointGroup [5]	-	Backbone (128), Clustering (221), ScoreNet (103)	452
SSTNet [7]	195	Backbone (125), Tree Network (229), ScoreNet (74)	623
HAIS [1]	-	Pointwise prediction (154), Hier.aggr. (118), Intra-inst.prediction (67)	339
SoftGroup [9]	-	Pointwise prediction (152), Soft grouping (123), Top-down refinement (70)	345
GraphCut (ours)	195+15	Extract superpoint features (122), Edge score prediction (5), Superpoint graph cut (42)	379

C.4 Inference Time

In Table 5, we report the average runtime of different methods on the ScanNet v2 validation set. Note that except for our method, the rest of the results in this table are derived from SoftGroup [9]. For a fair comparison, we use the same TITAN X GPU to evaluate the runtime of our method. In our GPU environment, we re-evaluated the runtime of SoftGroup and found that the runtime (343ms) is very close to the official time (345ms). Since SSTNet [7] and our GraphCut utilize the same method to generate superpoints, we add the runtime (195ms) of the superpoint generation to the total runtime. Note that our method also needs 15ms to construct the superpoint graph. For our GraphCut, we require 122ms for extracting superpoint features, 5ms for the edge score prediction network, and 42ms for the superpoint graph cut network. The total runtime of our method is 379ms (195+15+122+5+42). It can be observed that the runtime of our method outperforms most methods and is comparable to HAIS [1] and SoftGroup [9].

D Limitations and Impacts

Limitations. In 3D instance segmentation, it requires to recognize the object instance and predict semantic categories simultaneously. According to the instance segmentation results on the ScanNet v2 validation set, we found that our method performs worse on the refrigerator category in terms of average precision (AP). Although our method can recognize the instances of the refrigerator, the semantic category of the instances is easily predicted to the cabinet, resulting in low AP. In Figure 3, we visualize the samples of refrigerators and cabinets in the ScanNet v2 validation set. It can be observed that mini refrigerators are very similar to cabinets, so mini-refrigerators can be easily classified as cabinets. In addition, we find that the number of samples of the refrigerator is much smaller than in other categories, such as chairs, tables, and desks. Therefore, the above two points lead to the low AP of the refrigerator category. In order to improve the AP of the refrigerator category, we can consider two points: (1) We can mine the context information of the refrigerator

to infer the refrigerator from the surrounding objects. (2) Executing data augmentation for those categories have a small number of samples.

Impacts. The proposed method has a potential impact in autonomous cars and transportation. For autonomous cars, object instances on the road may be incorrectly recognized by the proposed method, which will increase the risk of safe driving. These issues require further research and consideration when building upon this work for 3D instance segmentation in autonomous situation.

Ethical consideration. This work is able to facilitate the development of certain applications. For example, it can help domestic robots avoid potential obstacles in indoor environments. In assisted driving, it can help the driver recognize potential objects that may affect driving in advance. In addition, all datasets used in this paper are publicly available as academic research, and the evaluation metrics used in the experiments are also standard. For negative outcomes, it depends on a specific task and the criteria for assessing positive and negative.

References

[1] Shaoyu Chen, Jiemin Fang, Qian Zhang, Wenyu Liu, and Xinggang Wang. Hierarchical aggregation for 3D instance segmentation. In *ICCV*, 2021.

[2] Francis Engelmann, Martin Bokeloh, Alireza Fathi, Bastian Leibe, and Matthias Nießner. 3D-MPA: Multi proposal aggregation for 3D semantic instance segmentation. In *CVPR*, 2020.

[3] Lei Han, Tian Zheng, Lan Xu, and Lu Fang. OccuSeg: Occupancy-aware 3D instance segmentation. In *CVPR*, 2020.

[4] Ji Hou, Angela Dai, and Matthias Nießner. 3D-SIS: 3D semantic instance segmentation of RGB-D scans. In *CVPR*, 2019.

[5] Li Jiang, Hengshuang Zhao, Shaoshuai Shi, Shu Liu, Chi-Wing Fu, and Jiaya Jia. PointGroup: Dual-set point grouping for 3D instance segmentation. In *CVPR*, 2020.

[6] Loic Landrieu and Martin Simonovsky. Large-scale point cloud semantic segmentation with superpoint graphs. In *CVPR*, 2018.

[7] Zhihao Liang, Zhihao Li, Songcen Xu, Mingkui Tan, and Kui Jia. Instance segmentation in 3D scenes using semantic superpoint tree networks. In *ICCV*, 2021.

[8] Shih-Hung Liu, Shang-Yi Yu, Shao-Chi Wu, Hwann-Tzong Chen, and Tyng-Luh Liu. Learning gaussian instance segmentation in point clouds. *arXiv preprint arXiv:2007.09860*, 2020.

[9] Thang Vu, Kookhoi Kim, Tung M. Luu, Xuan Thanh Nguyen, and Chang D. Yoo. SoftGroup for 3D instance segmentation on 3D point clouds. In *CVPR*, 2022.

[10] Weiyue Wang, Ronald Yu, Qiangui Huang, and Ulrich Neumann. SGPN: Similarity group proposal network for 3D point cloud instance segmentation. In *CVPR*, 2018.

[11] Xinlong Wang, Shu Liu, Xiaoyong Shen, Chunhua Shen, and Jiaya Jia. Associatively segmenting instances and semantics in point clouds. In *CVPR*, 2019.

[12] Bo Yang, Jianan Wang, Ronald Clark, Qingyong Hu, Sen Wang, Andrew Markham, and Niki Trigoni. Learning object bounding boxes for 3D instance segmentation on point clouds. In *NeurIPS*, 2019.

[13] Li Yi, Wang Zhao, He Wang, Minhyuk Sung, and Leonidas J Guibas. GSPN: Generative shape proposal network for 3d instance segmentation in point cloud. In *CVPR*, 2019.