Supplementary Material: SAM-guided Graph Cut for 3D Instance Segmentation

Haoyu Guo[1*] He Zhu[2*] Sida Peng[1] Yuang Wang[1]
Yujun Shen[3] Ruizhen Hu[4] Xiaowei Zhou[1]
[1]Zhejiang University [2]Beijing Normal Univeristy [3]Ant Group [4]Shenzhen Univeristy

A. Points sampling in projection mask

To sample $k = 5$ points uniformly and not too far from the boundary of the projection mask of each superpoint in each view, we first use the Euclidean Distance Transform to compute the distance from each pixel within the mask to its boundary, creating a distance map. We then select the point with the maximum value in the distance map to ensure it is near the center. To prevent subsequent sampled points from being too close to this first point, we set the values in the distance map within a certain area around this point to zero. This process is iteratively repeated for sampling the remaining points.

B. Structure of GNN

The Graph Neural Network (GNN) in our method consists of a 5-layer Graph Convolutional Network (GCN) and a 3-layer Multi-Layer Perceptron (MLP). The GCN has an input channel size of 256, which corresponds to the channel size of the SAM features. It has a hidden layer width of 128 and an output channel size of 128. The MLP has an input channel size of 257, which includes the concatenated GCN features of two nodes and one edge weight. Its hidden layer width is 128, and it has an output channel size of 1, corresponding to the affinity score of an edge.

C. Comparison with Panoptic Lifting

We observed that Panoptic Lifting struggles to extract satisfactory geometry, so we render the results of Panoptic Lifting in several views and visualize our method in nearby views for comparison. We show the results in **??**.

D. Analyses of different graph cut method

Based on the graph constructed using SAM, we tested segmenting the graph using normalized cuts, DBSCAN, and the direct graph partition method used in our approach, both with and without using the GNN (without means directly using edge weight). The comparison results are shown in the **??**. From the results, it's evident that the use of GNN generally improves most metrics for normalized cuts, while DBSCAN and the direct graph partition method show comprehensive improvements across all metrics. Furthermore, regardless of the use of GNN, the direct graph partition method consistently outperforms both normalized cuts and DBSCAN. Our analysis suggests that while normalized cuts and DBSCAN are adept at obtaining a rough segmentation for graphs with unreliable edge affinities, they are less capable of achieving finer segmentation results even when edge affinities are highly reliable.

E. Discussions of SAM guidance

As shown in the ablation studies in our paper, both the node features and edge weights calculated based on SAM are effective for our method, with the edge weights being particularly crucial. To further analyse their effectiveness, we attempted to remove both and use PointNet++ to compute node features. Specifically, we utilized PointNet++ to extract features from the point cloud, averaging the features within a superpoint to serve as the node feature. We employed the same loss function as in our method and optimized the network parameters of both PointNet++ and the GNN simultaneously. We found that this approach resulted in very poor performance.

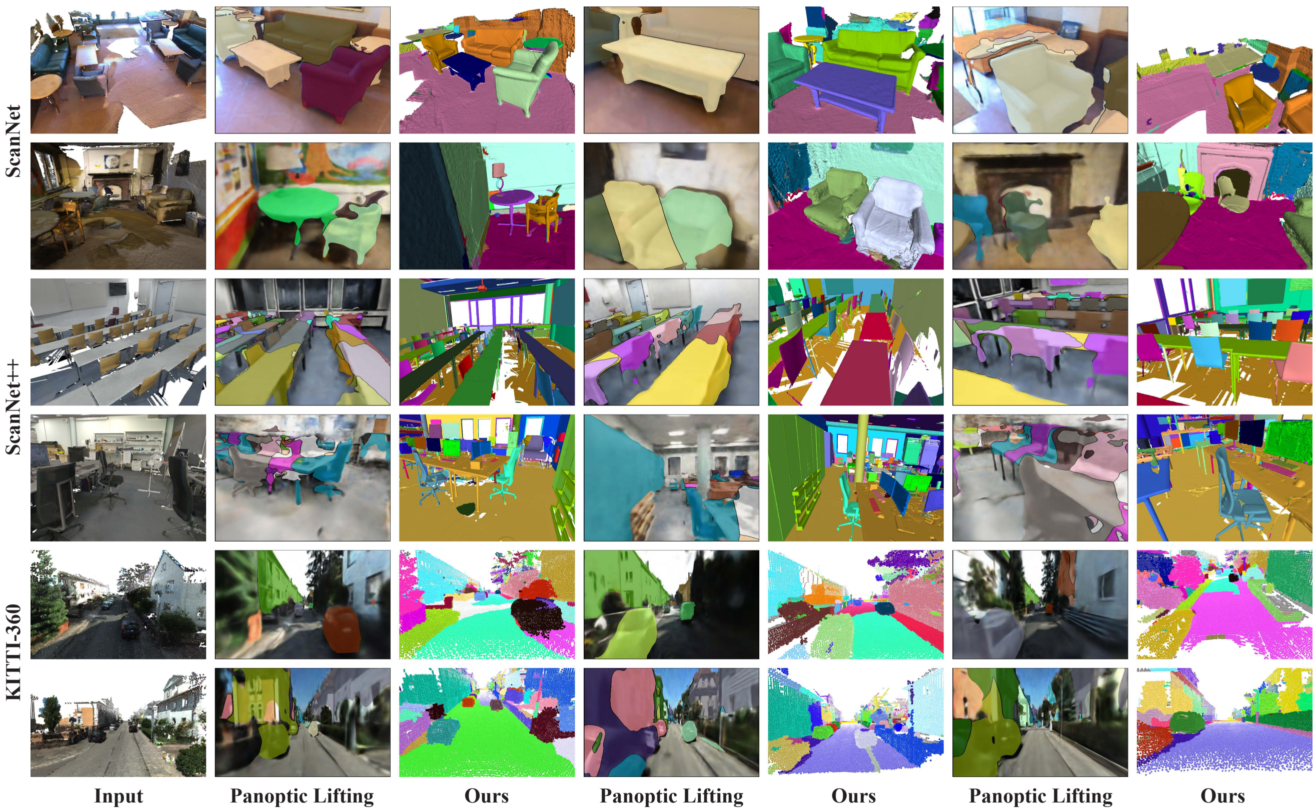

Figure 1. **Comparison with Panoptic Lifting.**

	ScanNet			ScanNet++			KITTI-360		
	mAP	AP_{50}	AP_{25}	mAP	AP_{50}	AP_{25}	mAP	AP_{50}	AP_{25}
SAM-based Graph + NCuts	5.1	13.6	39.5	7.5	16.5	34.2	10.1	18.4	31.0
SAM-based Graph + DBSCAN	6.6	14.1	24.7	8.2	15.0	22.6	12.9	23.5	35.6
SAM-based Graph + Graph partition	12.9	30.0	57.7	12.6	24.5	40.2	12.6	25.5	41.9
SAM-based Graph + GNN + NCuts	8.9	21.5	47.7	8.0	16.9	32.4	8.9	16.4	29.4
SAM-based Graph + GNN + DBSCAN	7.1	15.0	26.4	8.3	15.3	23.4	13.3	23.9	36.0
SAM-based Graph + GNN + Graph partition	**15.1**	**33.3**	**59.1**	**12.9**	**25.3**	**43.6**	**14.7**	**28.0**	**43.2**

Table 1. **Ablation studies of different graph cut methods.**