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 Superpoint generation

For the mesh of each scene on ScanNet and ScanNet++, we use the segmentator [2] provided by ScanNet, which adopts the algorithm described in [3] based on the mesh's normals. Specifically, for ScanNet, we directly use the over-segmentation provided in their dataset, which is obtained by running the segmentator with `KThresh` set to 0.01 and `segMinVerts` set to 20. For ScanNet++, due to the higher point cloud density of ScanNet++, we adjust the parameters to `KThresh` as 0.2 and `segMinVerts` as 500 to run the segmentator. For KITTI-360, we use the unsupervised point cloud segmentation algorithm proposed in [4], which first computes geometric features for each point from its 3D position and color values and then obtains the partitioned superpoints by minimizing the global energy function, and we set the regularization strength ρ to 0.1.

B 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 implemented by OpenCV [1] 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.

C Multi-scale mask selection

The 2D segmentation model SAM is designed to output masks at three different scales, each with a corresponding confidence score, to enable segmentation at different granularities. Please refer to their paper for detailed information. In our pipeline, we tend to choose masks with larger areas, as they are more likely to correspond to the segmentation of a complete object. However, when the confidence of a larger area segmentation is high, we will consider masks of smaller areas. Specifically, if the confidence of the largest mask is higher than the others, or lower but within a margin of 0.05, then we choose this mask. If this criterion is not met, we select from the remaining two masks: if the medium-sized mask has a higher confidence than the smallest mask, or its confidence is lower but within a margin of 0.05, then we choose the medium-sized mask; otherwise, we choose the smallest mask.

D 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.

E Implementation details of graph cut

When performing segmentation, for every two vertices, if their affinity is below a certain threshold, we consider them to be unconnected. Conversely, if their affinity is above this threshold, to further improve robustness, we identify paths of length 2 between these two vertices. We then record the number of paths where both edges have high affinity scores and the number where one edge is high and the other low. If the ratio of the latter exceeds a predefined threshold, we regard the two vertices to be unconnected; otherwise, they are considered as connected. Once the connection status of each edge is determined using this method, we employ a union-find algorithm [6] to merge all connected superpoints, resulting in the 3D segmentation of the scene.

F Details of evaluation protocol

We evaluate the class-agnostic AP scores of all methods across all datasets, meaning that during the evaluation, we only consider the accuracy of the masks, without taking into account their semantic categories. This approach follows that of [5]. Since our primary focus is on object instance segmentation, and the baseline Mask3D cannot segment the floors and walls, we exclude predictions of floor and wall regions (only the floor for KITTI-360) from evaluation for all methods to ensure a more fair comparison. Additionally, we also exclude segmentations predicted in other unlabeled regions.

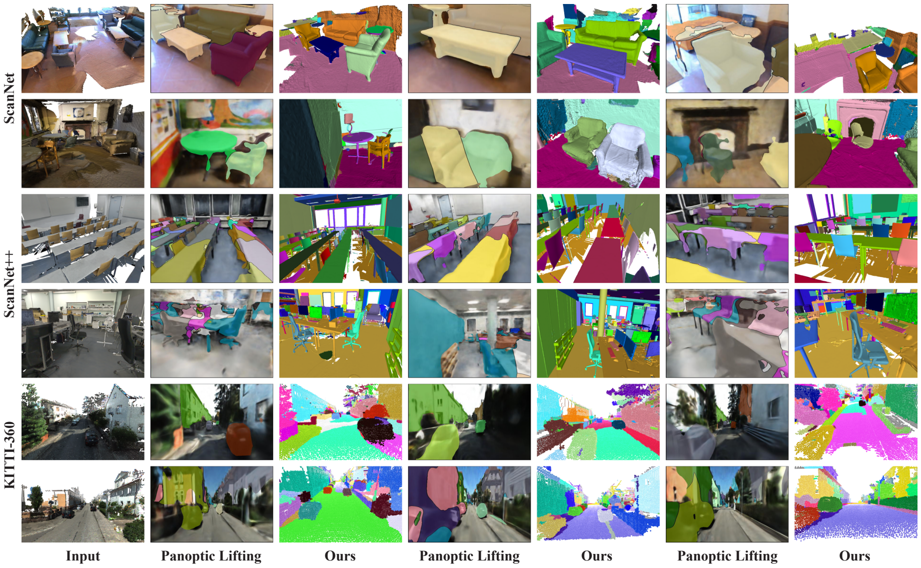

Fig. 1: Comparison with Panoptic Lifting.

G 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 Fig. 1.

H 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 Tab. 1. 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.

Table 1: Ablation studies of different graph cut methods.

	ScanNet			ScanNet++			KITTI-360		
	mAP	AP$_{50}$	AP$_{25}$	mAP	AP$_{50}$	AP$_{25}$	mAP	AP$_{50}$	AP$_{25}$
NCuts	15.7	31.7	59.0	10.1	18.6	34.7	19.5	30.2	45.0
DBSCAN	10.3	18.6	27.8	10.5	17.2	25.0	20.5	31.4	42.1
Graph partition	19.7	37.7	61.6	13.7	25.2	43.0	22.6	36.2	48.5
GNN + NCuts	18.0	35.0	59.4	11.3	20.1	35.2	18.1	27.7	40.7
GNN + DBSCAN	11.0	19.6	29.2	10.7	17.5	25.9	21.1	32.2	43.0
GNN + Graph partition	**22.1**	**41.7**	**62.8**	**15.3**	**27.2**	**44.3**	**23.8**	**37.2**	**49.1**

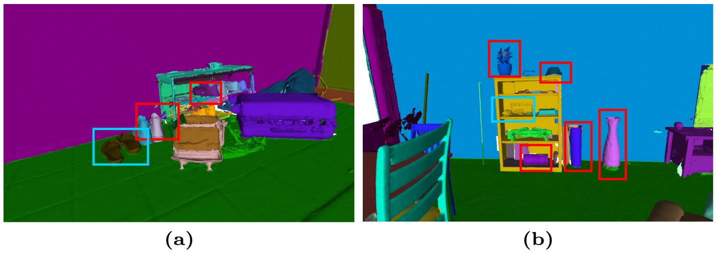
(a) **(b)**

Fig. 2: Segmentation performance on small objects.

I 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 analyze 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.

J Performance on small objects

We show segmentation results on small objects in Fig. 2. Our method successfully segments a considerable number of small objects, such as several shoes in Fig. 2a and some vases and small items on the shelf in Fig. 2b, as framed in red, though some small objects are not separated from each other or their surroundings, as framed in blue.

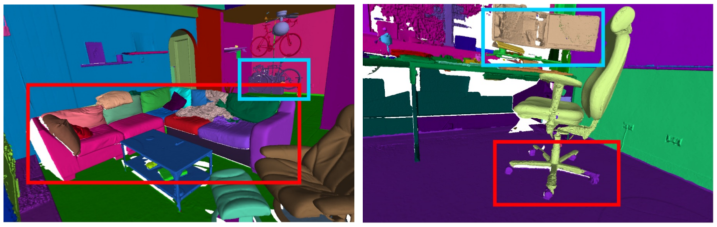

Fig. 3: Failure cases.

K Failure cases

To better understand the performance of our method, we show two typical failure cases (over-segmention and under-segmention) of our method in Fig. 3, framed in red and blue respectively.

References

1. Opencv. https://opencv.org/ 1
2. Segmentator. https://github.com/ScanNet/ScanNet/tree/master/Segmentator 1
3. Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient graph-based image segmentation. IJCV (2004) 1
4. Guinard, S., Landrieu, L., Vallet, B.: Weakly supervised segmentation-aided classification of urban scenes from 3d lidar point clouds. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences (2017) 1
5. Rozenberszki, D., Litany, O., Dai, A.: Unscene3d: Unsupervised 3d instance segmentation for indoor scenes. arXiv preprint arXiv:2303.14541 (2023) 2
6. Tarjan, R.E.: Efficiency of a good but not linear set union algorithm. Journal of the ACM (JACM) (1975) 2