# INS-Conv: Incremental Sparse Convolution for Online 3D Segmentation (Supplementary)

Leyao Liu*[1], Tian Zheng*[1], Yun-Jou Lin[2], Kai Ni[3], and Lu Fang[1✉]

[1]Department of Electronic Engineering, Tsinghua University
[2]OPPO US Research Center
[3]HoloMatic Technology (Beijing) Co., Ltd.

## A. Details of INS-Conv Network

In this section, we explain the detailed network architecture of our INS-Conv network in A.1, followed by the training details in A.2.

### A.1. Network Architecture

We follow the similar UNet-like architecture as in [5]. The network architecture is shown in Fig. 1. We describe the layer configuration in Tab. 1. For each resolution level $i$ ($i$ from 0 to $L$, $L = 6$), the configuration of the extract, downscale and upscale modules are given respectively, where M controls the channel width of network. We set M $= 32$ for the m32 model, and M $= 64$ for the m64 model.

Table 1. Details of layer configuration, where M controls the channel width of network.

| Module | Layer | K | S | $C_{in}$ | $C_{out}$ |
|---|---|---|---|---|---|
| $f_i^{extract}$ | bn+ssc | 3 | 1 | M*(i+1) | M*(i+1) |
| | bn+ssc | 3 | 1 | M*(i+1) | M*(i+1) |
| | resnet addition | | | | |
| $f_i^{downscale}$ | bn+conv | 2 | 2 | M*(i+1) | M*(i+2) |
| $f_i^{upscale}$ | addition | | | | |
| | bn+deconv | 2 | 2 | M*(i+1) | M*i |
| | channel linear | | | M*i | M*i |
| | bn+ssc | 3 | 1 | M*i | M*i |
| | bn+ssc | 3 | 1 | M*i | M*i |
| | resnet addition | | | | |

### A.2. Training Details

We generate three different completeness for each scene: 33%, 66%, and 100%, which is measured by the number of points. We trim the RGBD sequence to generate these partial scenes. Each scene and its partial scenes are put into the same batch. Our network outputs semantic probability, instance embedding and uncertainty for each voxel. An offset vector pointing to the instance centroid is also predicted, which we refer to [5]. We use cross-entropy loss to train the semantic probability, and the discriminative loss [2] and temporal consistency loss to train the instance embedding. The weighted BCE loss is used to train the uncertainty term. The neighbor propagation module is then added to each INS-SSC layer, on top of the pretrained model. It is trained to minimize the MSE between the last layer features of INS-Conv and 'full' propagation.

* Equal contribution.

✉ Corresponding author. Mail: fanglu@tsinghua.edu.cn.

Acknowledgement: This work is supported in part by Natural Science Foundation of China (NSFC) under contract No. 62125106, 61860206003 and 62088102, in part by National Key R&D Program of China (2021ZD0109901), in part by Beijing National Research Center for Information Science and Technology (BNRist) under Grant No. BNR2020RC01002.

## B. Implementation Details

In this section, we provide some extra implementation details. In INS-Conv, for layers that have bias terms, they are not linear maps. We make a modification to these layers that we only add the bias value to sites that are previously inactive. In the instance clustering stage, we perform mean-shift clustering on the updated points, and the distance metric is the Euclidean distance between the predicted embeddings. For semantic segmentation, we fuse current predicted semantic labels to global. We maintain a label and a label weight for each point in the global model. If the currently predicted label of a point is different from the global saved label, we decrease the corresponding label weight, and vice versa. If the label weight is smaller than zero, we change the global saved label to the currently predicted label. For the details, please refer to [9]. We set voxel size to 0.02m in all experiments.

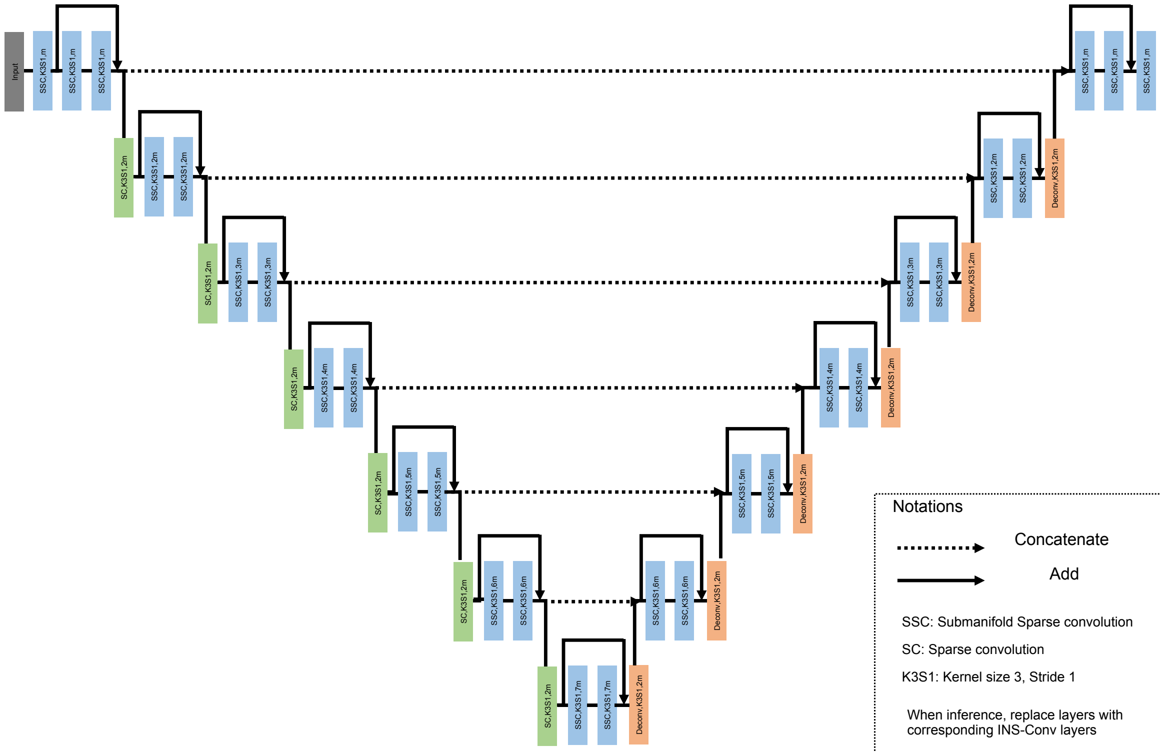


Figure 1. Network architecture. We use a UNet-like submanifold sparse convolutional network as backbone, and replace the layers with corresbonding INS-Conv layers in inference stage.

## C. More Results

More qualitaive results of online semantic and instance results on the validation set of ScanNetv2 are provided in Fig. 2 and Fig. 3. Additionally, per-class semantic mIoU, per-class instance mAP@50 on ScanNetv2 validation set and test set are provided in Tab. 2 to 5. The per-class instance mAP@50 results on sceneNN are shown in Tab. 6. We also provide an online demo of INS-Conv in the attached video, where we integrate the CPU version of INS-Conv into a CPU-based SLAM system [4]. Only using the CPU computing power of a portable device, we achieve online inference speed.

| Method | **mIoU** | bath | bed | bkshf | cab | chair | cntr | curt | desk | door | floor | ofurn | pic | fridg | showr | sink | sofa | tabl | toil | wall | wind |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Ours-m32 | 71.5 | 84.3 | 79.3 | 78.7 | 64.4 | 90.1 | 63.6 | 72.5 | 63.5 | 60.4 | 95.2 | 55.8 | 33.8 | 52.3 | 74.5 | 63.3 | 83.2 | 74.5 | 92.8 | 84.0 | 63.3 |
| Ours-m64 | 72.4 | 87.4 | 81.2 | 79.6 | 67.7 | 91.0 | 64.5 | 74.9 | 60.8 | 62.1 | 95.1 | 57.8 | 36.0 | 52.0 | 72.2 | 67.0 | 83.3 | 72.3 | 93.3 | 85.1 | 65.2 |

Table 2. Per-class semantic segmentation results of our method on the ScanNetV2 [1] validation set.

| Method | **mAP@50** | bath | bed | bkshf | cab | chair | cntr | curt | desk | door | ofurn | pic | fridg | showr | sink | sofa | tabl | toil | wind |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Ours-m32 | 57.4 | 77.4 | 69.7 | 53.4 | 50.3 | 74.6 | 30.9 | 48.5 | 47.0 | 45.1 | 52.3 | 42.0 | 44.8 | 69.7 | 55.9 | 69.7 | 62.5 | 94.8 | 45.2 |
| Ours-m64 | 61.4 | 80.6 | 71.4 | 53.8 | 52.9 | 92.3 | 30.9 | 55.2 | 51.7 | 50.3 | 61.6 | 44.4 | 47.5 | 74.0 | 52.6 | 71.0 | 66.4 | 100.0 | 48.3 |

Table 3. Per-class instance segmentation results of our method on the ScanNetV2 [1] validation set.

| Method | **mIoU** | bath | bed | bkshf | cab | chair | cntr | curt | desk | door | floor | ofurn | pic | fridg | showr | sink | sofa | tabl | toil | wall | wind |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| FA [11] (Online) | 63.0 | 60.4 | 74.1 | 76.6 | 59.0 | 74.7 | 50.1 | 73.4 | 50.3 | 52.7 | 91.9 | 45.4 | 42.3 | 55.0 | 42.0 | 67.8 | 68.8 | 54.4 | 89.6 | 79.5 | 62.7 |
| SV [7] (Online) | 63.5 | 65.6 | 71.1 | 71.9 | 61.3 | 75.7 | 44.4 | 76.5 | 53.4 | 56.6 | 92.8 | 47.8 | 27.2 | 63.6 | 53.1 | 66.4 | 64.5 | 50.8 | 86.4 | 79.2 | 61.1 |
| SC [3] (Offline) | 72.5 | 64.7 | 82.1 | 84.6 | 72.1 | 86.9 | 53.3 | 75.4 | 60.3 | 61.4 | 95.5 | 57.2 | 32.5 | 71.0 | 87.0 | 72.4 | 82.3 | 62.8 | 93.4 | 86.5 | 68.3 |
| MK [5] (Offline) | 73.6 | 85.9 | 81.8 | 83.2 | 70.9 | 84.0 | 52.1 | 85.3 | 66.0 | 64.3 | 95.1 | 54.4 | 28.6 | 73.1 | 89.3 | 67.5 | 77.2 | 68.3 | 87.4 | 85.2 | 72.7 |
| Ours (Online) | 71.7 | 75.1 | 75.9 | 81.2 | 70.4 | 86.8 | 53.7 | 84.2 | 60.9 | 60.8 | 95.3 | 53.4 | 29.3 | 61.6 | 86.4 | 71.9 | 79.3 | 64.0 | 93.3 | 84.5 | 66.3 |

Table 4. Semantic segmentation results on the ScanNetV2 [1] test set in terms of mIoU score on 20 classes, using the m64 model.

| Method | **mAP@50** | bath | bed | bkshf | cab | chair | cntr | curt | desk | door | ofurn | pic | fridg | showr | sink | sofa | tabl | toil | wind |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PF [9] (Online) | 47.8 | 66.7 | 71.2 | 59.5 | 25.9 | 55.0 | 0.0 | 61.3 | 17.5 | 25.0 | 43.4 | 43.7 | 41.1 | 85.7 | 48.5 | 59.1 | 26.7 | 94.4 | 35.9 |
| PG [8] (Offline) | 63.6 | 100.0 | 76.5 | 62.4 | 50.5 | 79.7 | 11.6 | 69.6 | 38.4 | 44.1 | 55.9 | 47.6 | 59.6 | 100.0 | 66.6 | 75.6 | 55.6 | 99.7 | 51.3 |
| OS [5] (Offline) | 67.2 | 100.0 | 75.8 | 68.2 | 57.6 | 84.2 | 47.7 | 50.4 | 52.4 | 56.7 | 58.5 | 45.1 | 55.7 | 100.0 | 75.1 | 79.7 | 56.3 | 100.0 | 46.7 |
| Ours (Online) | 65.7 | 100.0 | 76.0 | 66.7 | 58.1 | 86.3 | 32.3 | 65.5 | 47.7 | 47.3 | 54.9 | 43.2 | 65.0 | 100.0 | 65.5 | 73.8 | 58.5 | 94.4 | 47.2 |

Table 5. Instance segmentation results on the ScanNetV2 [1] test set in terms of mAP@50 score on 18 classes, using the m64 model.

| Method (Offline) | **mAP@0.5** | wall | floor | cabinet | bed | chair | sofa | table | desk | tv | prop |
|---|---|---|---|---|---|---|---|---|---|---|---|
| MLS-CRF [10] | 12.1 | 13.9 | 44.5 | 0.0 | 32.9 | 12.9 | 0.0 | 5.7 | 10.8 | 0.0 | 0.8 |
| OccuSeg | 47.1 | 39.0 | 93.8 | 5.7 | 66.7 | 91.3 | 8.7 | 50.0 | 31.6 | 76.9 | 7.14 |
| Ours (Online) | 57.6 | 21.2 | 88.2 | 39.9 | 75.0 | 89.9 | 64.8 | 40.9 | 43.3 | 90.2 | 22.3 |

Table 6. Instance segmentation results on the SceneNN [6] dataset in terms of mAP@0.5 score of each class, using the m64 model.

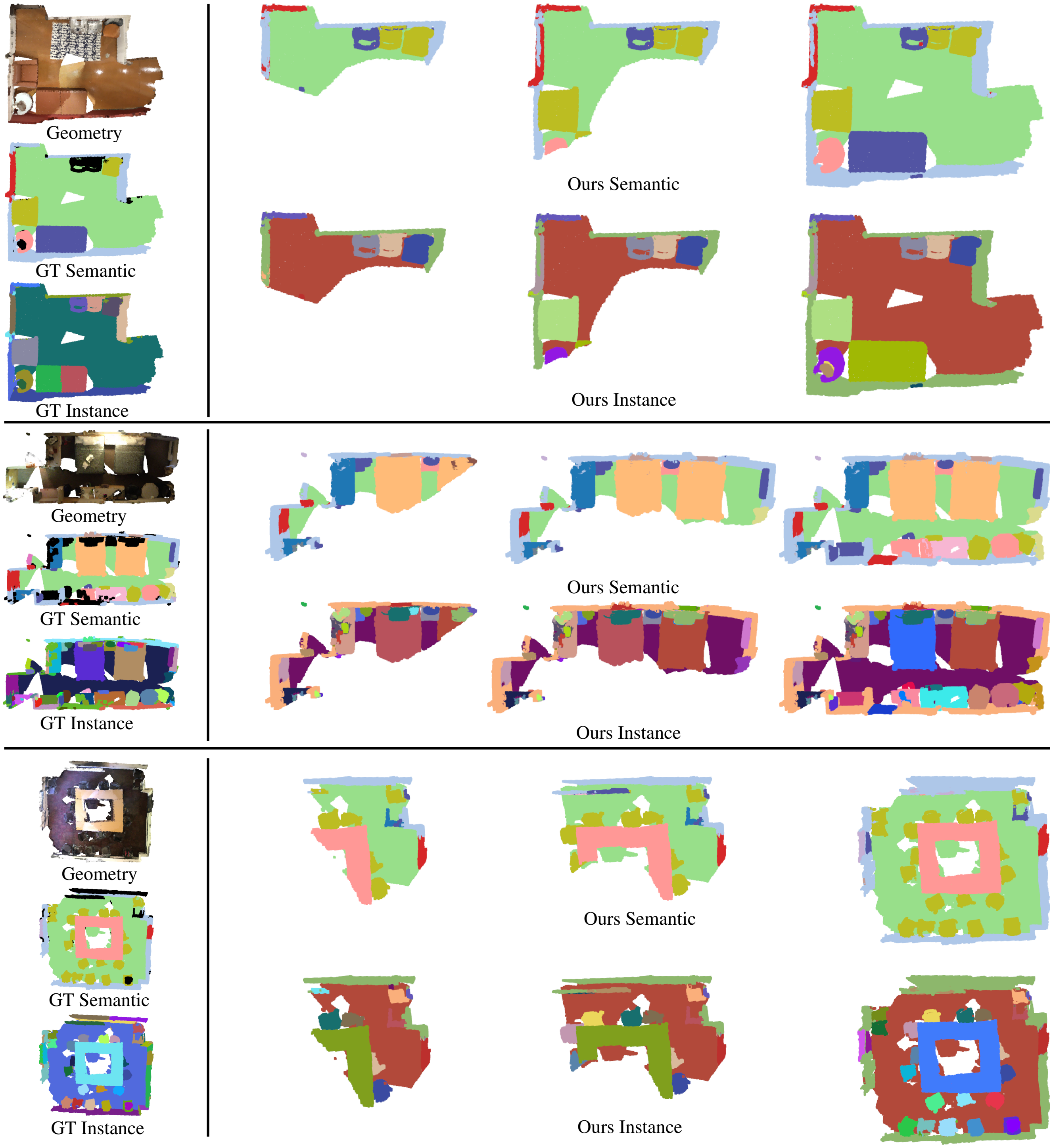


Figure 2. Online semantic and instance results on the validation set of ScanNetv2.

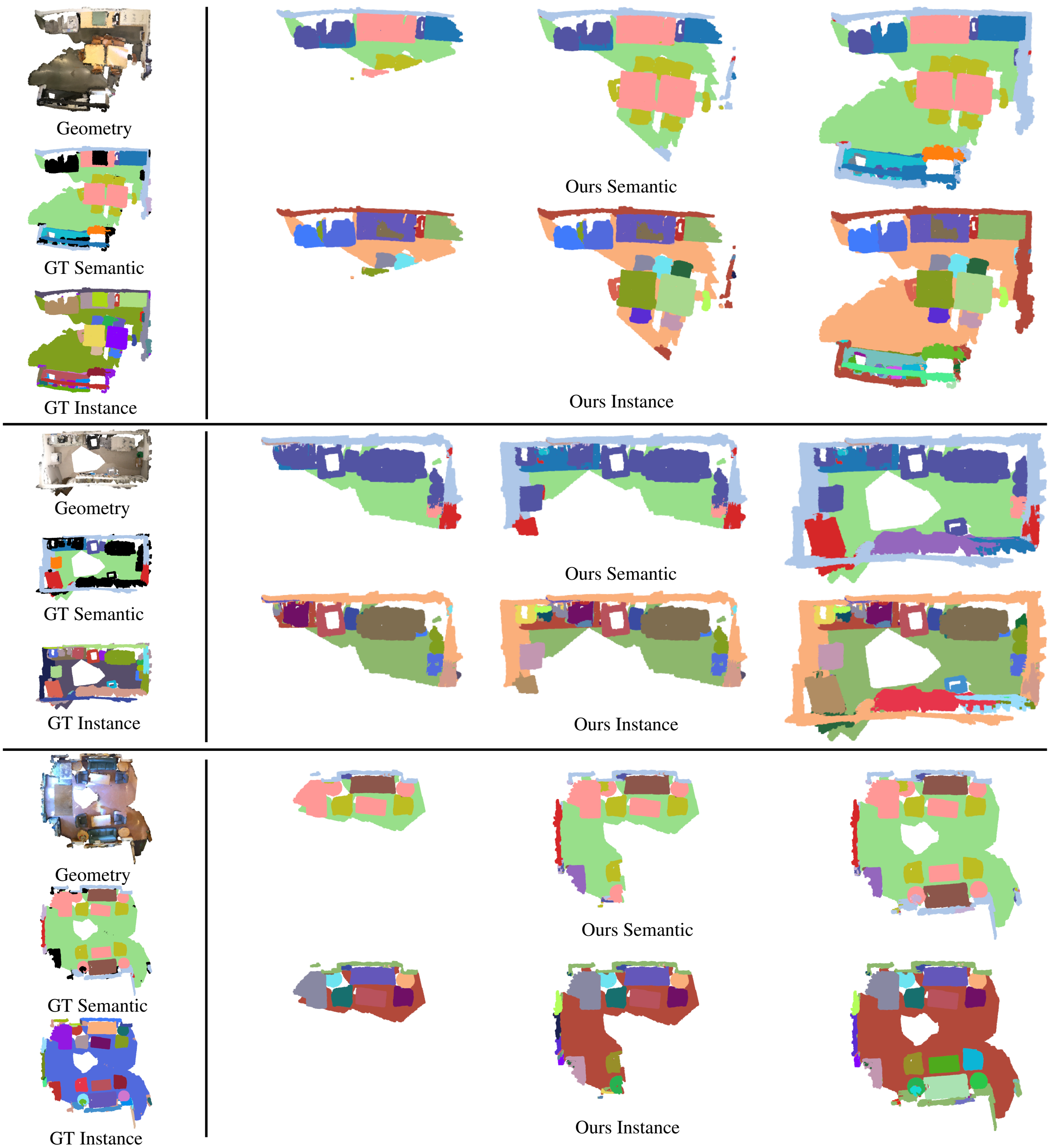


Figure 3. Online semantic and instance results on the validation set of ScanNetv2.

## References


[1] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas A. Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2432–2443, 2017. 3

[2] Bert De Brabandere, Davy Neven, and Luc Van Gool. Semantic instance segmentation with a discriminative loss function. *arXiv preprint arXiv:1708.02551*, 2017. 1

[3] Benjamin Graham, Martin Engelcke, and Laurens van der Maaten. 3d semantic segmentation with submanifold sparse convolutional networks. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9224–9232, 2018. 3

[4] Lei Han and Lu Fang. Flashfusion: Real-time globally consistent dense 3d reconstruction using cpu computing. In *Robotics: Science and Systems*, volume 1, page 7, 2018. 2

[5] Lei Han, Tian Zheng, Lan Xu, and Lu Fang. Occuseg: Occupancy-aware 3d instance segmentation. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2937–2946, 2020. 1, 3

[6] Binh-Son Hua, Quang-Hieu Pham, Duc Thanh Nguyen, Minh-Khoi Tran, Lap-Fai Yu, and Sai-Kit Yeung. Scenenn: A scene meshes dataset with annotations. *2016 Fourth International Conference on 3D Vision (3DV)*, pages 92–101, 2016. 3

[7] Shi-Sheng Huang, Ze-Yu Ma, Tai-Jiang Mu, Hongbo Fu, and Shi-Min Hu. Supervoxel convolution for online 3d semantic segmentation. *ACM Trans. Graph.*, 40:34:1–34:15, 2021. 3

[8] Li Jiang, Hengshuang Zhao, Shaoshuai Shi, Shu Liu, Chi-Wing Fu, and Jiaya Jia. Pointgroup: Dual-set point grouping for 3d instance segmentation. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4866–4875, 2020. 3

[9] Gaku Narita, Takashi Seno, Tomoya Ishikawa, and Yohsuke Kaji. Panopticfusion: Online volumetric semantic mapping at the level of stuff and things. *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4205–4212, 2019. 1, 3

[10] Quang-Hieu Pham, Thanh Nguyen, Binh-Son Hua, Gemma Roig, and Sai-Kit Yeung. Jsis3d: Joint semantic-instance segmentation of 3d point clouds with multi-task pointwise networks and multi-value conditional random fields. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8827–8836, 2019. 3

[11] Jiazhao Zhang, Chenyang Zhu, Lin tao Zheng, and Kai Xu. Fusion-aware point convolution for online semantic 3d scene segmentation. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4533–4542, 2020. 3