Appendix (Supplementary Materials)

A. Introduction.

In this supplementary material, we provide more details regarding baseline architecture (Appendix B), the boundary problem Appendix C, visualization results (Appendix D), the training setup (Appendix E), the effect of temperature (Appendix F), the effect of design regarding sub-scene annotation (Appendix G), and experiment results (Appendix H).

Especially, CBL achieves a new stat-of-the-art on S3DIS with the newly released transformer model (Tab. 7).

B. Architecture of ConvNet Baseline

We show the specific architecture of our ConvNet baseline in Fig. 1. With a consistent notation, $\mathcal{X}^n$ is the point cloud in sub-sampling stage n, f_i is the feature of point x_i, and $N^n = |\mathcal{X}^n|$ with $N = N^0$. We use the multi-scale head on all baselines when adapting the CBL.

C. Further Analysis on Boundary Problem

We further account for the type of areas and class-specific analysis for better exploring the boundary problem. Specifically, we provide per-class IoU score that is separately calculated on boundary area $\mathcal{B}_l$ and inner area $\mathcal{X} - \mathcal{B}_l$.

As shown in Tab. 2, we evaluate for all three baselines with and without the proposed CBL. We notice that, large improvements are made on small objects, *e.g.* column, which aligns with the observation in **??** in main paper. We would like to add that, despite that CBL focuses only on boundaries, improvements are also made on inner area. We hypothesize the reason might be that the false boundary in model predicted segmentation is restrained, as features in inner area implicitly becomes more similar when the features across boundaries are optimized to be more distinctive by the CBL.

Moreover, for all three baselines, the improvement on boundary area is much more than that made on inner area, which is summarized in Tab. 1.

Therefore, with metrics separately calculated on boundary and inner area, we clearly see that the improvement brought by CBL is mainly from the boundary areas. Such observation further emphasizes the importance of clear scene boundaries in point cloud segmentation task.

baselines (+ CBL)	mIoU		OA		mACC	
	boundary	inner	boundary	inner	boundary	inner
RandLA-Net [10]	+3.3	+1.4	+4.1	-0.3	+3.4	+2.4
CloserLook3D [14]	+0.6	+0.2	+0.1	+0.2	+0.7	+0.4
ConvNet	+2.5	+2.0	+1.0	+0.7	+3.2	+2.8

Table 1. The improvement brought by CBL on different baselines and types of area (boundary / inner area).

D. More Visualizations

We provide more qualitative results as a support for the improvement made by CBL on boundaries. The visualization results include various scenes, including rooms (Fig. 3), cluttered space (Fig. 4), hallways (Fig. 5), and offices (Fig. 6). For each scene, we further attempt to visualize the features discrimination between center points and their corresponding neighbors and the results are presented in the every second row. Specifically, we calculate the normalized feature distance between the point feature f_i and features of its neighboring points $\{f_j \mid x_j \in \mathcal{N}_i\}$. We then take the mean distance for visualization.

According to the presented figures, it shows that the CBL significantly enhances the feature distances around the scene boundaries and improves the baseline to obtain a more detailed and cleaner boundary in prediction for different type of scenes. The visualization is done on S3DIS testset Area 5.

E. Training Setup in Details

For the RandLA-Net [10] and CloserLook3D [14] baselines, we follow their instructions of released code for training and evaluation, which are here (RandLA-Net) and here (CloserLook3D), respectively. Especially, in CloserLook3D [14], there are two non-parametric module, we use the one with sin/cos spatial embedding.

For the ConvNet baseline, we use the SGD optimizer to train for 600 epoch, with a weight decay of 0.001. We set the initial learning rate to 0.01 and use a momentum of 0.98 with a decay rate of $0.1^{1/200}$. It roughly takes 24 hours to train on 4 Nividia v100 GPUs, and we does not observe obvious increase in training time after applying the CBL.

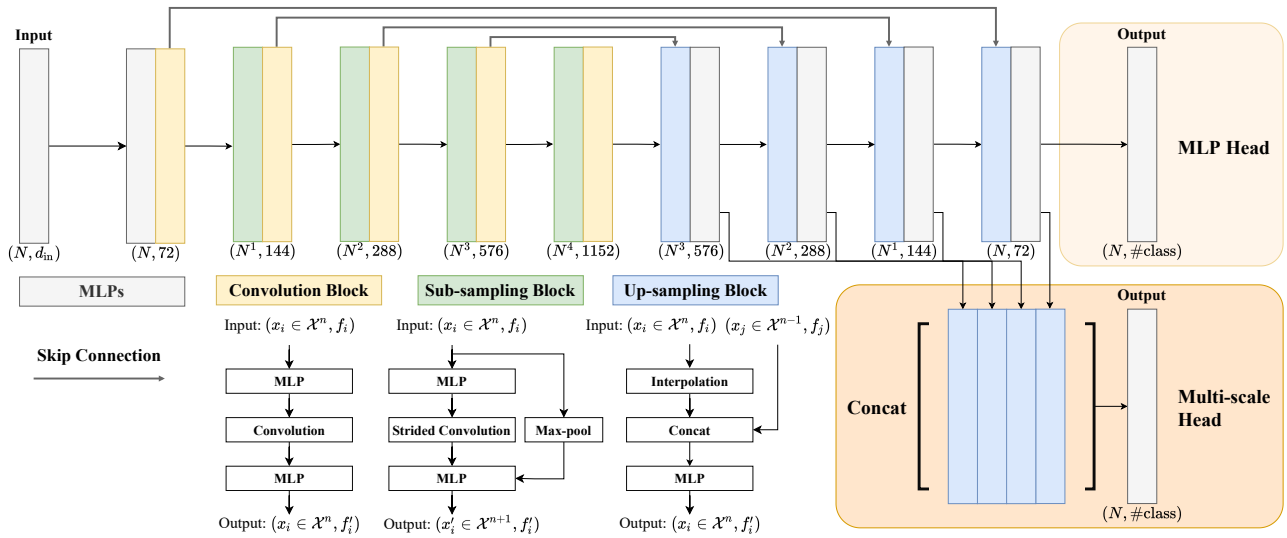


Figure 1. The detail architecture of ConvNet baseline.

methods	mIoU	OA	mACC	ceiling	floor	wall	beam	column	window	door	table	chair	sofa	bookcase	board	clutter
RandLA-Net [10]	44.1	67.1	59.1	65.5	69.4	52.2	0.0	21.4	28.6	55.0	55.0	56.0	41.1	41.2	45.8	42.1
+ CBL	47.4	71.2	62.5	78.2	85.9	56.0	0.0	30.3	25.7	42.6	58.4	60.9	50.0	42.5	52.2	44.2
CloserLook3D [14]	50.0	76.6	58.5	80.7	88.6	63.9	0.0	21.1	15.6	57.5	73.3	64.7	52.2	43.1	37.2	52.6
+ CBL	50.6	76.7	59.2	80.9	88.6	64.6	0.0	26.5	15.6	55.9	73.0	65.0	50.4	47.6	38.4	51.2
ConvNet	50.1	76.5	58.3	80.4	88.3	63.5	0.0	26.5	15.2	58.3	72.1	63.4	52.3	40.8	38.7	52.2
+ CBL	52.6	77.5	61.5	80.5	88.8	65.7	0.0	32.5	20.9	61.8	71.7	62.4	52.5	46.7	47.4	52.5

(a) The full metrics calculated on boundary points from ground truth (*i.e.*, $\mathcal{B}_l$) only.

methods	mIoU	OA	mACC	ceiling	floor	wall	beam	column	window	door	table	chair	sofa	bookcase	board	clutter
RandLA-Net [10]	65.8	89.6	73.0	93.3	98.6	84.6	0.0	25.9	65.7	46.5	81.1	88.9	65.4	75.5	71.9	58.2
+ CBL	67.2	89.3	75.4	93.0	99.1	84.6	0.0	37.3	64.1	39.4	82.7	91.5	79.3	75.9	73.9	56.0
CloserLook3D [14]	70.7	92.2	75.2	96.4	99.9	86.5	0.0	25.9	55.1	76.5	95.9	87.1	81.9	75.1	72.5	66.2
+ CBL	70.9	92.4	75.6	96.5	99.9	86.9	0.0	27.0	59.3	78.1	95.7	87.7	80.8	75.4	69.4	65.6
ConvNet	71.2	92.1	75.5	95.0	99.8	85.9	0.0	34.6	56.0	82.7	95.4	87.4	81.3	73.8	68.4	65.7
+ CBL	73.2	92.8	78.3	95.3	99.9	88.0	0.0	38.4	62.2	76.4	95.9	87.5	82.7	81.2	75.2	68.6

(b) The full metrics calculated on inner points from ground truth (*i.e.*, $\mathcal{X} - \mathcal{B}_p$) only.

Table 2. The improvement CBL brought on baselines, separately calculated in boundary area (a) and inner area (b). The red denotes improvement is made on baseline.

temperature	mIoU	OA	mACC
0.3	70.67	89.16	77.91
0.5	70.98	89.31	78.27
1	71.33	89.40	78.69
2	70.73	89.10	77.98
10	70.03	88.97	77.58

Table 3. The effect of temperature on CBL.

F. Effect of Temperature in CBL

We conduct empirical study on ScanNet [5] validation set to analyze the effect of temperature τ in the CBL (**??**). We use the ConvNet baseline and train for 600 epoch on training set. As shown in Tab. 3, we find that the proper temperature for CBL is within $(0.5, 2)$, and we set the temperature to $\tau = 1$ by default.

G. Effect of Design of Sub-scene annotation

While the sub-scene annotation is a distribution, we only use the simple $\arg\max$ when evaluating the boundary points. Therefore, it raises two particular question: 1) is it necessary to maintain the distribution? 2) is there any better way in utilizing the sub-scene annotation than the $\arg\max$?

In this section, we explore other alternatives and answer to this two questions with a particular focus of how they affect the model performance on boundaries.

Necessities of maintaining distribution. There are two main reasons to leverage the average pooling on labels and

	mIoU (%)	Ground	Building	Pole	Bollard	Trash can	Barrier	Pedestrian	Car	Natural
HDGCN [13]	68.3	99.4	93.0	67.7	75.7	25.7	44.7	37.1	81.9	89.6
ConvPoint [2]	75.9	99.5	95.1	71.6	88.7	46.7	52.9	53.5	89.4	85.4
RandLANet [10]	78.5	99.5	97.0	71.0	86.7	50.5	65.5	49.1	95.3	91.7
KP-Conv [18]	82.0	99.5	94.0	71.3	83.1	78.7	47.7	78.2	94.4	91.4
FKAConv [3]	82.7	99.6	98.1	77.2	91.1	64.7	66.5	58.1	95.6	93.9
PyramidPoint [19]	82.9	99.6	97.1	74.6	84.3	56.0	65.9	79.1	95.1	93.9
ConvNet	76.2	99.5	96.3	68.5	67.4	41.4	41.5	80.6	96.3	94.1
+ CBL	78.6	99.5	96.7	72.1	72.6	46.2	60.4	70.1	97.2	93.2

Table 4. Quantitative results on Paris-Lille-3D of NPM3D [16] benchmark, results obtained from online benchmark site by the time of submission. The red denotes the improvement made on baseline.

maintain the distribution. First, current methods may not preserve the original input points after sub-sampling, *e.g.* grid sub-sampling in KPConv [18]. Therefore, the original label of a sub-sampled point is not presented and the sub-scene annotation is thus demanded. Although we may use the label of the nearest point for approximation, Tab. 5 shows that CBL (nearest) is sub-optimal. Second, despite that we only use the "argmax" result of the sub-scene annotation, maintaining distribution still preserves more information than just maintaining "argmax" result. As "argmax" discards the minor classes during sampling, such elimination of minority may further accumulate through more sub-sampling stages and leads to imprecise boundary, as depicted in Fig. 2. Experimentally, in Tab. 5, though CBL (argmax) improves boundary (B-IoU), it compromises overall performance.

Better treatment than Argmax. While "argmax" is straight forward, it introduces the problem of "label-flipping" when the distribution of sub-scene annotation is close to a uniform distribution, *i.e.*, when the number of points of different classes are roughly the same.

To avoid this, we leverage the KL divergence as a measure of the semantic distance among sub-scene annotations. We then threshold on the KL-distance to determine if two sub-scene annotations belong to the same semantic class or not, which further enables us to determine the boundary points in sub-sampled point cloud. Specifically, we set the threhold to 0.5 and CBL (kl) can be bring a small improvement on overall performance, and a slightly larger boost on boundary performance, as in Tab. 5. Yet, as "thresholding KL distance" introduces extra hyper-parameters and complexity, we opt for "argmax" for simplicity in the main paper.

Summary. Therefore, we summarize the reason for designing the sub-scene annotation as a distribution as it can preserve much more information and can be extended to a more robust boundary determination using KL-distance.

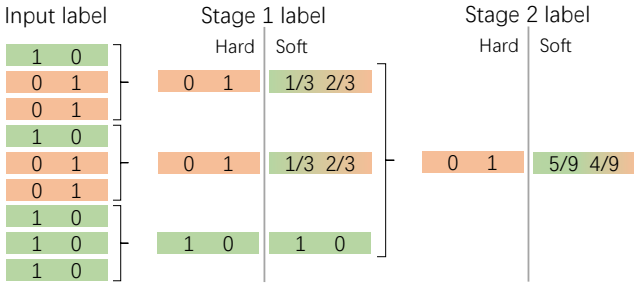


Figure 2. With every 3 points being sub-sampled into 1 in each stage, tracking distribution (soft label) describes original input faithfully, but hard label fails due to accumulated errors.

methods	mIoU			B-IoU
	overall	@boundary	@inner	
ConvNet	67.4	50.1	71.2	59.6
ConvNet + CBL	69.4	52.6	73.1	61.5
ConvNet + CBL (nearest)	68.3	52.1	71.8	60.9
ConvNet + CBL (argmax)	66.8	50.6	70.4	60.6
ConvNet + CBL (kl)	69.5	52.5	73.2	62.0

Table 5. Same setting as in **??** in main paper.

H. Further Experiments

Results on ScanNet and NPM3D datasets. We provide the detail results on ScanNet in Tab. 6; and the detail results on NPM3D in Tab. 4.

CBL with Transformer. We use the open-source code base (here) to re-produce the performance of newly released point Transformer [24] on S3DIS [1] Area 5 dataset.

In Tab. 7, the same consistent improvement is made on classes such as column. CBL with better boundaries further boosts the overall performance to 71.0 in mIoU, achieving a new state-of-the-art performance.

Method	mIoU	bathtub	bed	books.	cabinet	chair	counter	curtain	desk	door	floor	other	pic	fridge	shower	sink	sofa	table	toilet	wall	wndw
DCM-Net [17]	65.8	77.8	70.2	80.6	61.9	81.3	46.8	69.3	49.4	52.4	94.1	44.9	29.8	51.0	82.1	67.5	72.7	56.8	82.6	80.3	63.7
VMNet [11]	74.6	87.0	83.8	85.8	72.9	85.0	50.1	87.4	58.7	65.8	95.6	56.4	29.9	76.5	90.0	71.6	81.2	63.1	93.9	85.8	70.9
SparseConvNet [8]	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
MinkowskiNet [4]	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
O-CNN [20]	76.4	75.8	79.6	83.9	74.6	90.7	56.2	85.0	68.0	67.2	97.8	61.0	33.5	77.7	81.9	84.7	83.0	69.1	97.2	88.5	72.7
OccuSeg [9]	76.2	92.4	82.3	84.4	77.0	85.2	57.7	84.7	71.1	64.0	95.8	59.2	21.7	76.2	88.8	75.8	81.3	72.6	93.2	86.8	74.4
Mix3D [15]	78.1	96.4	85.5	84.3	78.1	85.8	57.5	83.1	68.5	71.4	97.9	59.4	31.0	80.1	89.2	84.1	81.9	72.3	94.0	88.7	72.5
BA-GEM [7] *	63.5																				
PointConv [21]	66.6	78.1	75.9	69.9	64.4	82.2	47.5	77.9	56.4	50.4	95.3	42.8	20.3	58.6	75.4	66.1	75.3	58.8	90.2	81.3	64.2
PointASNL [22]	66.6	70.3	78.1	75.1	65.5	83.0	47.1	76.9	47.4	53.7	95.1	47.5	27.9	63.5	69.8	67.5	75.1	55.3	81.6	80.6	70.3
KP-Conv [18]	68.4	84.7	75.8	78.4	64.7	81.4	47.3	77.2	60.5	59.4	93.5	45.0	18.1	58.7	80.5	69.0	78.5	61.4	88.2	81.9	63.2
FusionNet [23]	68.8	70.4	74.1	75.4	65.6	82.9	50.1	74.1	60.9	54.8	95.0	52.2	37.1	63.3	75.6	71.5	77.1	62.3	86.1	81.4	65.8
JSENet [12]	69.9	88.1	76.2	82.1	66.7	80.0	52.2	79.2	61.3	60.7	93.5	49.2	20.5	57.6	85.3	69.1	75.8	65.2	87.2	82.8	64.9
RFCR [6]	70.2	88.9	74.5	81.3	67.2	81.8	49.3	81.5	62.3	61.0	94.7	47.0	24.9	59.4	84.8	70.5	77.9	64.6	89.2	82.3	61.1
ConvNet + CBL	70.5	76.9	77.5	80.9	68.7	82.0	43.9	81.2	66.1	59.1	94.5	51.5	17.1	63.3	85.6	72.0	79.6	66.8	88.9	84.7	68.9

Table 6. Quantitative results on ScanNet [5] benchmark, results obtained from online benchmark site by the time of submission. We group method by the 3D representation type, which is respectively, from top to down, 3D + mesh, 3D voxel and 3D point, and we also use 3D point. The empty line denotes no record of detailed performance found. The method with * also considers boundary.

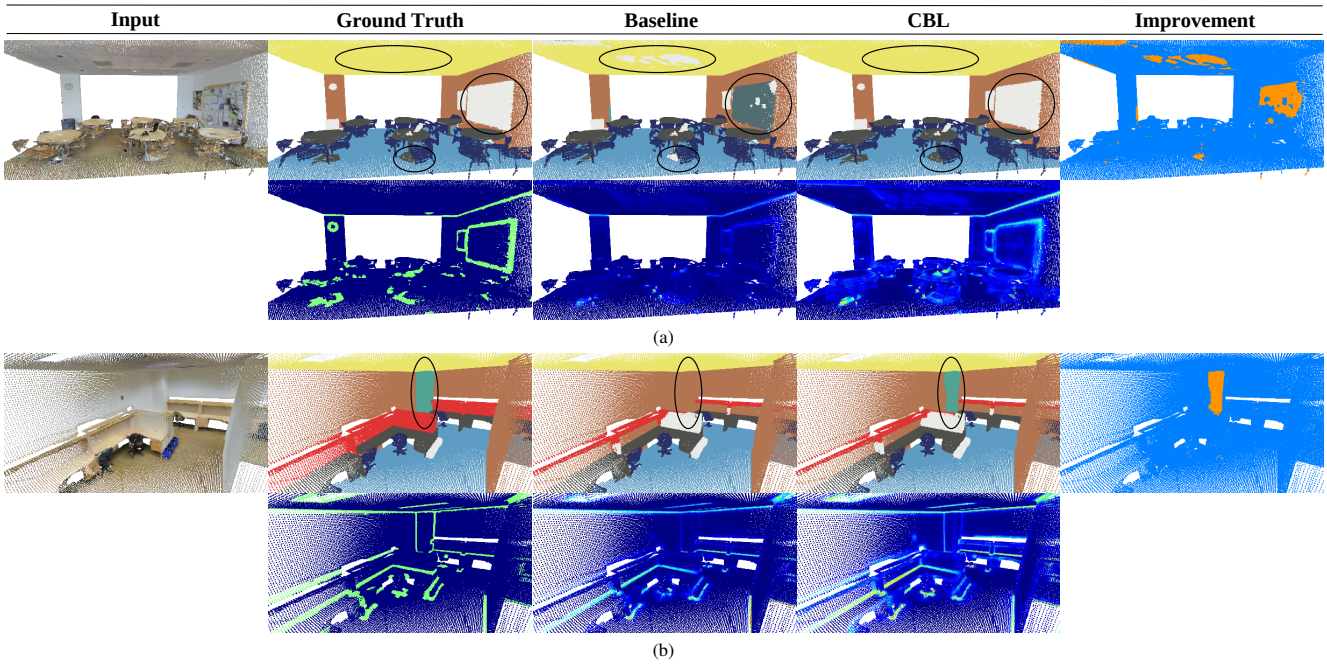


Figure 3. Large rooms. We compare the results of ConvNet baseline with CBL. On the every second row, we visualize the boundary points calculated from the ground truth label, and the feature discrimination among neighboring points for each model. The improvement on the first row and the enhanced feature discrimination on the second row show that CBL improves the features across boundaries to obtain a better segmentation quality on boundary areas. The visualization is done on S3DIS testset Area 5.

methods	mIoU	OA	mACC	ceiling	floor	wall	beam	column	window	door	table	chair	sofa	bookcase	board	clutter
pt trans [24]*	70.4	90.8	76.5	94.0	98.5	86.3	0.0	38.0	63.4	74.3	89.1	82.4	74.3	80.2	76.0	59.3
pt trans [24]	70.0	90.5	76.5	95.2	98.6	85.1	0.0	36.7	62.5	75.9	81.5	91.0	75.1	71.9	76.4	60.2
+ CBL	71.0*	90.9*	77.5*	94.3*	98.3	87.4*	0.0	42.1*	64.0*	78.5*	82.5	88.9*	75.1*	71.1	81.3*	59.6*

Table 7. Quantitative results on S3DIS Area 5 dataset [1], showing the mean IoU (mIoU), overall accuracy (OA), mean accuracy (mACC), and per-class IoU scores. We include both performance reported in original paper (with *, the first row) and the re-produced performance (without *, the second row). We use red to denote improvement over the re-produced point transformer, and * to denote the improvement over the performance reported in original paper.

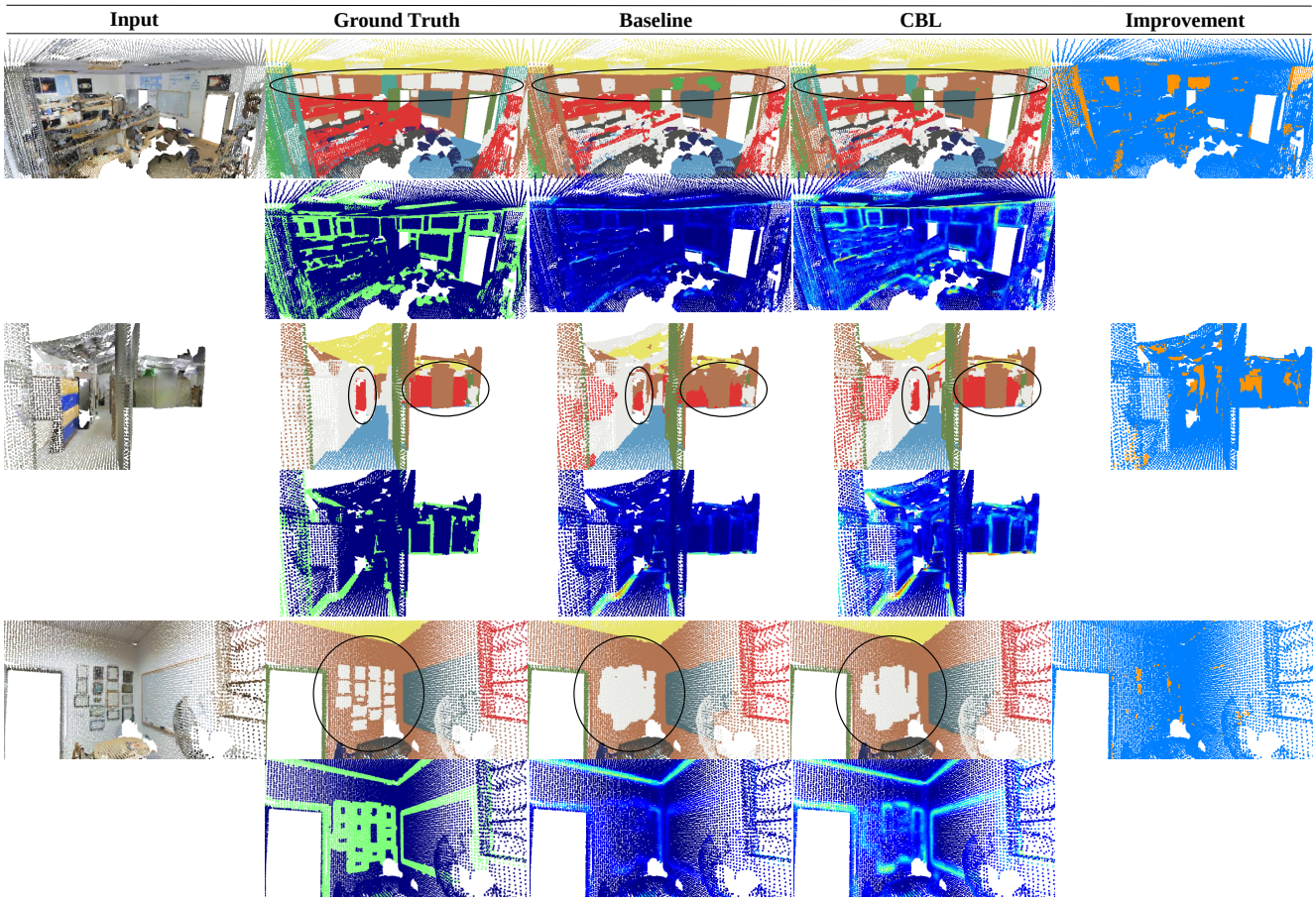
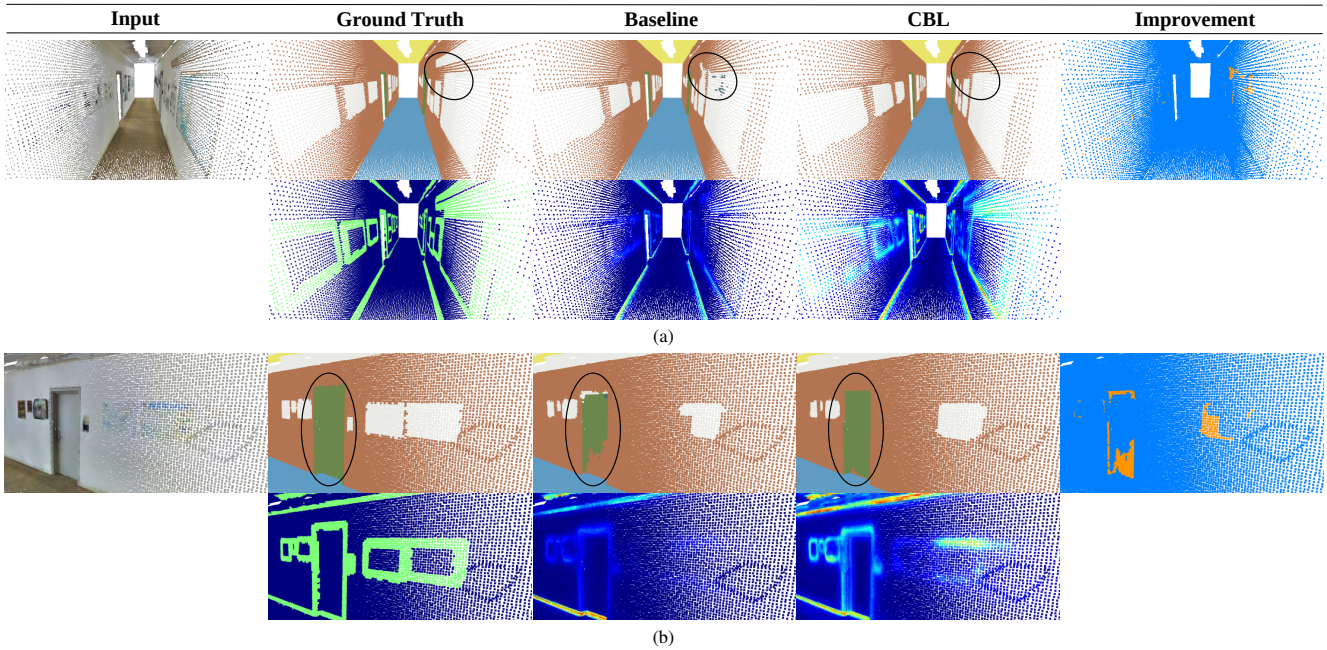


Figure 4. Cluttered space. Same as above (Fig. 3).



(a)

(b)

Figure 5. Hallways. Same as above (Fig. 3).

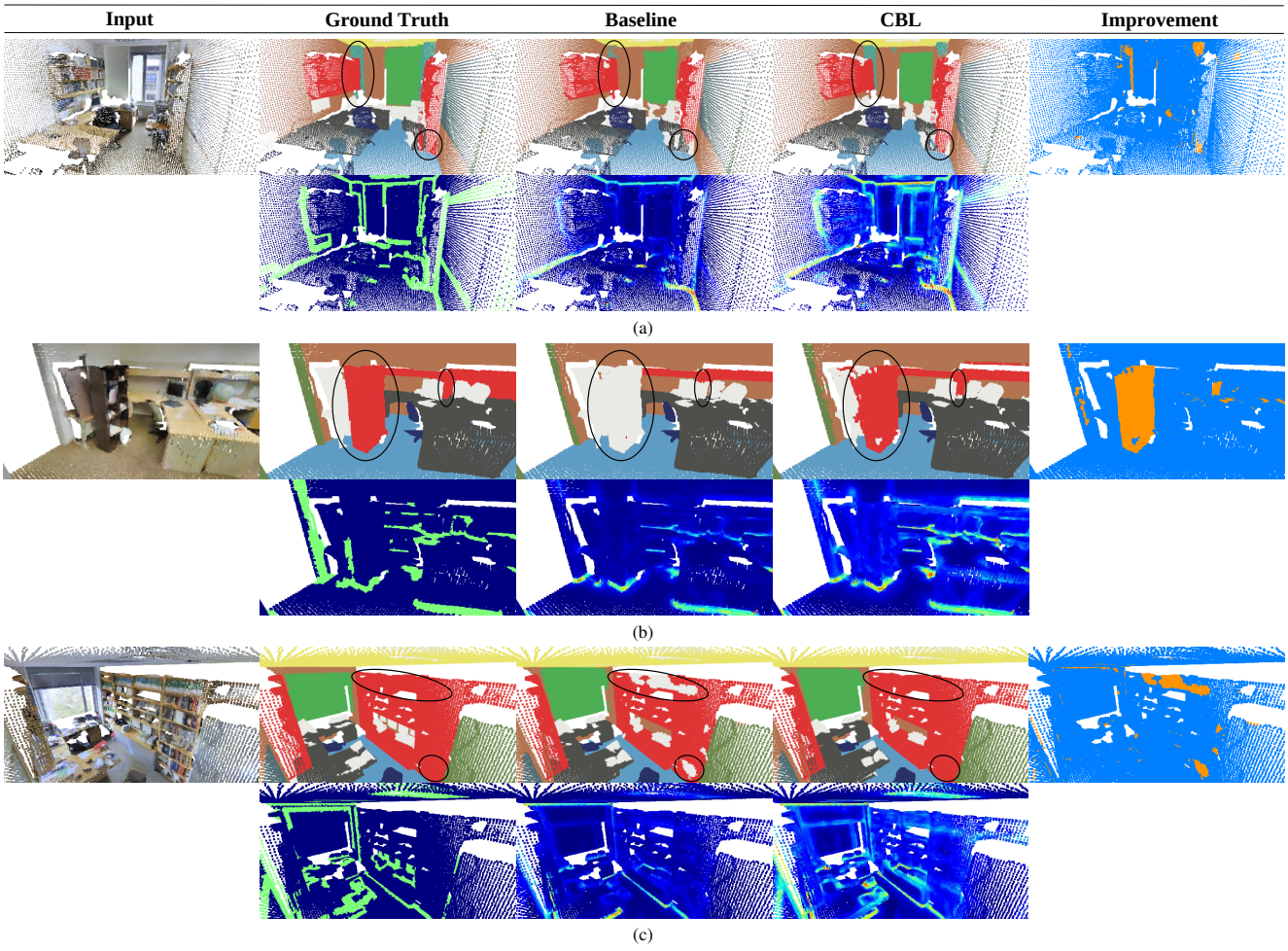


Figure 6. Offices. Same as above (Fig. 3).

References

[1] Iro Armeni, Sasha Sax, Amir R Zamir, and Silvio Savarese. Joint 2d-3d-semantic data for indoor scene understanding. *arXiv preprint arXiv:1702.01105*, 2017. 3, 4

[2] Alexandre Boulch. Convpoint: Continuous convolutions for point cloud processing. *Computers & Graphics*, 88:24 – 34, 2020. 3

[3] Alexandre Boulch, Gilles Puy, and Renaud Marlet. Fkaconv: Feature-kernel alignment for point cloud convolution, 2020. 3

[4] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3075–3084, 2019. 4

[5] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017. 2, 4

[6] Jingyu Gong, Jiachen Xu, Xin Tan, Haichuan Song, Yanyun Qu, Yuan Xie, and Lizhuang Ma. Omni-supervised point cloud segmentation via gradual receptive field component reasoning. *CoRR*, abs/2105.10203, 2021. 4

[7] Jingyu Gong, Jiachen Xu, Xin Tan, Jie Zhou, Yanyun Qu, Yuan Xie, and Lizhuang Ma. Boundary-aware geometric encoding for semantic segmentation of point clouds, 2021. 4

[8] 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*, Jun 2018. 4

[9] 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. 4

[10] Qingyong Hu, Bo Yang, Linhai Xie, Stefano Rosa, Yulan Guo, Zhihua Wang, Niki Trigoni, and Andrew Markham. Randla-net: Efficient semantic segmentation of large-scale point clouds. *CoRR*, abs/1911.11236, 2019. 1, 2, 3

[11] Zeyu Hu, Xuyang Bai, Jiaxiang Shang, Runze Zhang, Jiayu Dong, Xin Wang, Guangyuan Sun, Hongbo Fu, and Chiew-Lan Tai. Vmnet: Voxel-mesh network for geodesic-aware 3d semantic segmentation. *CoRR*, abs/2107.13824, 2021. 4

[12] Zeyu Hu, Mingmin Zhen, Xuyang Bai, Hongbo Fu, and Chiew-Lan Tai. Jsenet: Joint semantic segmentation and edge detection network for 3d point clouds. *CoRR*, abs/2007.06888, 2020. 4

[13] Zhidong Liang, Ming Yang, Liuyuan Deng, Chunxiang Wang, and Bing Wang. Hierarchical depthwise graph convolutional neural network for 3d semantic segmentation of point clouds. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8152–8158, 2019. 3

[14] Ze Liu, Han Hu, Yue Cao, Zheng Zhang, and Xin Tong. A closer look at local aggregation operators in point cloud analysis. *ECCV*, 2020. 1, 2

[15] Alexey Nekrasov, Jonas Schult, Or Litany, Bastian Leibe, and Francis Engelmann. Mix3d: Out-of-context data augmentation for 3d scenes, 2021. 4

[16] Xavier Roynard, Jean-Emmanuel Deschaud, and François Goulette. Paris-lille-3d: A large and high-quality ground-truth urban point cloud dataset for automatic segmentation and classification. *The International Journal of Robotics Research*, 37(6):545–557, 2018. 3

[17] Jonas Schult, Francis Engelmann, Theodora Kontogianni, and Bastian Leibe. Dualconvmesh-net: Joint geodesic and euclidean convolutions on 3d meshes. *CoRR*, abs/2004.01002, 2020. 4

[18] Hugues Thomas, Charles R. Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J. Guibas. Kpconv: Flexible and deformable convolution for point clouds. *CoRR*, abs/1904.08889, 2019. 3, 4

[19] Nina Varney, Vijayan K. Asari, and Quinn Graehling. Pyramid point: A multi-level focusing network for revisiting feature layers, 2020. 3

[20] Peng-Shuai Wang, Yang Liu, Yu-Xiao Guo, Chun-Yu Sun, and Xin Tong. O-CNN: Octree-based Convolutional Neural Networks for 3D Shape Analysis. *ACM Transactions on Graphics (SIGGRAPH)*, 36(4), 2017. 4

[21] Wenxuan Wu, Zhongang Qi, and Fuxin Li. Pointconv: Deep convolutional networks on 3d point clouds. *CoRR*, abs/1811.07246, 2018. 4

[22] Xu Yan, Chaoda Zheng, Zhen Li, Sheng Wang, and Shuguang Cui. Pointasnl: Robust point clouds processing using nonlocal neural networks with adaptive sampling. *CoRR*, abs/2003.00492, 2020. 4

[23] Feihu Zhang, Jin Fang, Benjamin W. Wah, and Philip H. S. Torr. Deep fusionnet for point cloud semantic segmentation. In *ECCV*, 2020. 4

[24] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip Torr, and Vladlen Koltun. Point transformer, 2021. 3, 4