Geometry and Uncertainty-Aware 3D Point Cloud Class-Incremental Semantic Segmentation
– Supplementary Material –

	S^0 Split	S^1 Split
S3DIS [1]	0:ceiling, 1:floor, 2:wall, 3:beam, 4:column, 5:window, 6:door, 7:table, 8:chair, 9:sofa, 10:bookcase, 11:board, 12:clutter	0:beam, 1:board, 2:bookcase, 3:ceiling, 4:chair, 5:clutter, 6:column, 7:door, 8:floor, 9:sofa, 10:table, 11:wall, 12:window
ScanNet [2]	0:wall, 1:floor, 2:chair, 3:table, 4:desk, 5:bed, 6:bookshelf, 7:sofa, 8:sink, 9:bathtub, 10:toilet, 11:curtain, 12:counter, 13:door, 14:window, 15:shower curtain, 16:refrigerator, 17:picture, 18:cabinet, 19:other furniture	0:bathtub, 1:bed, 2:bookshelf, 3:cabinet, 4:chair, 5:counter, 6:curtain, 7:desk, 8:door, 9:floor, 10:other furniture, 11:picture, 12:refrigerator, 13:shower curtain, 14:sink, 15:sofa, 16:table, 17:toilet, 18:wall, 19:window

Table 1. Two split paradigms S^0 and S^1 on S3DIS and ScanNet datasets for 3D point cloud class-incremental semantic segmentation. The number before class names (*e.g.* 0, 1, 2, ...) represents the label. S^0 is organized by the original class label order of datasets. S^1 introduces classes in alphabetical order. We change the number of novel classes to evaluate our approach. For example, when training model on C_{novel}=5 under S3DIS S^0 split, 0:ceiling to 7:table are used as base classes, while 8:chair to 12:clutter are applied as novel classes.

In the supplementary material, we will first show the split of different paradigms S^0 and S^1 on S3DIS and ScanNet datasets in Appendix A, and then we show some extra ablation studies in Appendix B. Finally, we will provide the class-wise IoU results of the C_{novel}=5 case in Appendix C for detailed comparison.

Appendix A. Dataset Split

Tab. 1 shows the dataset split paradigms on S3DIS [1] and ScanNet [2] dataset in our experiments. S^0 split is organized according to the specified order in the original dataset, while S^1 split is arranged according to the alphabet order.

Appendix B. Extra Ablation Studies

Results with various backbones. Tab. 2 shows the experimental results of our method across different point cloud backbones (PointNet++ [5], PointConv [7] and DGCNN [6]). From the table, we can see that our approach has a consistent and superior performance close to the joint training (upper bound).

Table 2. Experiments with various backbones on S3DIS dataset.

Backbone	Methods	C_{novel}=3 / S0		
		0-9	10-12	*all*
PointNet++ [5]	Ours	<u>48.93</u>	<u>42.64</u>	<u>47.48</u>
	Joint Training	51.06	44.91	49.64
PointConv [7]	Ours	<u>49.67</u>	<u>45.53</u>	<u>48.72</u>
	Joint Training	49.82	48.65	49.55
DGCNN [6]	Ours	<u>45.15</u>	<u>45.33</u>	<u>45.19</u>
	Joint Training	48.62	41.44	46.97

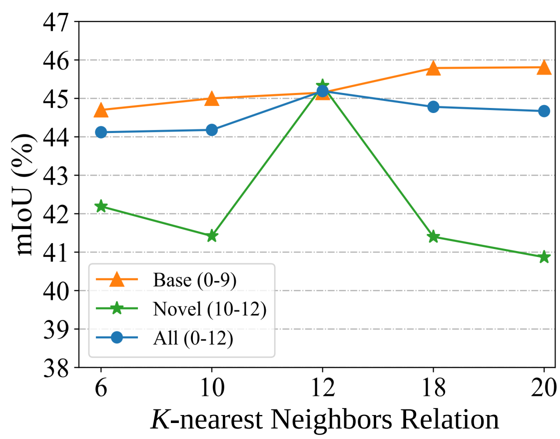

Figure 1. Effects of different K-nearest Neighbors in GFT module on S3DIS datasets (S^0) of C_{novel}=3.

Number of nearest neighbors in GFT module. To explore the effects of different K-nearest neighbors in Geometry-aware Feature-relation Transfer (GFT) module, we perform a series of experiments in Fig. 1. As we can see, increasing the number of K improves overall mIoU. However, when K reaches a certain number (*i.e.,* K=12), it starts an adverse impact on the performance. We believe that when euclidean distance is applied for neighbor points sampling on the point cloud with a certain density, larger K will destroy the object geometric structure and thus cause the structured feature relation bias. Besides, increasing K brings more constraints on the old classes, which may interfere with the model's capability to learn novel classes.

Table 3. Class-wise IoU (%) performance comparison of 3D class-incremental segmentation methods on the S3DIS [1] dataset under S^0 split. “BT”, “F&A”, “FT”, “JT” denotes Base Training, Freeze and Add, Fine-Tuning and Joint Training respectively. Asterisk (*) denotes traditional class-incremental methods EWC [3] and LwF [4] in our reproduction for 3D semantic segmentation. For the forgetting-prevention-based method (gray face), the best IoU results for individual class are underlined, and the best mIoU results are in bold.

	S3DIS dataset (S^0), C_{novel}=5															
	base classes								novel classes					mIoU (%)		
	0:ceiling	1:floor	2:wall	3:beam	4:column	5:window	6:door	7:table	8:chair	9:sofa	10:bookcase	11:board	12:clutter	0-7	8-12	*all*
BT	88.74	96.58	73.30	0.00	6.76	40.60	17.61	64.70	-	-	-	-	-	48.54	-	-
F&A	84.11	95.09	65.87	0.00	7.51	40.60	17.26	43.57	18.29	1.90	24.75	4.72	12.01	44.25	12.33	31.98
FT	80.04	89.11	65.13	0.31	3.07	6.29	0.00	35.75	33.83	3.09	43.27	36.56	34.48	34.96	30.25	33.15
EWC*	67.03	84.43	58.38	0.04	10.25	23.14	26.92	44.87	49.68	7.05	42.72	35.87	20.04	39.38	31.07	36.19
LwF*	90.02	96.45	73.70	0.00	3.55	37.61	9.24	45.85	60.01	7.50	43.20	27.82	36.53	44.55	35.01	40.88
Ours	88.10	96.08	73.91	0.04	8.83	41.97	17.20	65.37	55.35	21.05	44.14	41.21	36.05	**48.94**	**39.56**	**45.33**
JT	91.11	96.36	73.52	1.02	10.32	41.67	23.00	64.87	59.26	25.11	44.58	39.34	40.41	50.23	41.74	46.97

Table 4. Class-wise IoU (%) performance comparison of 3D class-incremental segmentation methods on the S3DIS [1] dataset under S^1 split. “BT”, “F&A”, “FT”, “JT” denotes Base Training, Freeze and Add, Fine-Tuning and Joint Training respectively. Asterisk (*) denotes traditional class-incremental methods EWC [3] and LwF [4] in our reproduction for 3D semantic segmentation. For the forgetting-prevention-based method (gray face), the best IoU results for individual class are underlined, and the best mIoU results are in bold.

	S3DIS dataset (S^1), C_{novel}=5															
	base classes								novel classes					mIoU (%)		
	0:beam	1:board	2:bookcase	3:ceiling	4:chair	5:clutter	6:column	7:door	8:floor	9:sofa	10:table	11:wall	12:window	0-7	8-12	*all*
BT	1.98	30.35	42.37	90.71	60.78	36.38	3.28	32.10	-	-	-	-	-	37.24	-	-
F&A	1.28	35.40	41.19	90.12	45.60	32.43	9.95	42.25	93.75	0.16	14.53	70.85	35.16	37.71	42.89	39.44
FT	0.00	0.36	7.33	55.46	5.98	9.14	0.66	12.61	91.15	10.91	47.02	67.18	37.08	10.99	50.67	26.53
EWC*	0.00	29.97	5.55	85.18	35.37	24.74	1.89	2.80	95.54	6.06	62.43	68.39	41.77	23.19	54.84	35.36
LwF*	0.00	28.42	33.61	90.29	58.78	32.34	5.55	13.66	96.04	6.18	62.72	73.29	37.71	32.83	55.19	41.43
Ours	0.08	32.07	50.16	89.92	62.98	39.19	3.20	27.86	95.43	6.14	63.32	73.80	37.32	**38.17**	**55.20**	**44.72**
JT	0.97	29.47	45.24	89.64	61.02	37.71	11.83	31.16	96.67	22.57	64.72	74.06	42.55	38.38	60.11	46.74

Table 5. Class-wise IoU (%) performance comparison of 3D class-incremental segmentation methods on the ScanNet [2] dataset under S^0 split. “BT”, “F&A”, “FT”, “JT” denotes Base Training, Freeze and Add, Fine-Tuning and Joint Training respectively. Asterisk (*) denotes traditional class-incremental methods EWC [3] and LwF [4] in our reproduction for 3D semantic segmentation. For the forgetting-prevention-based method (gray face), the best IoU results for individual class are underlined, and the best mIoU results are in bold.

	ScanNet dataset (S^0), C_{novel}=5																						
	base classes															novel classes					mIoU (%)		
	0:wall	1:floor	2:chair	3:table	4:desk	5:bed	6:bookshelf	7:sofa	8:sink	9:bathub	10:toilet	11:curtain	12:counter	13:door	14:window	15:shower curtain	16:refrigerator	17:picture	18:cabinet	19:otherfurniture	0-14	15-19	*all*
BT	55.77	91.58	52.90	42.60	27.15	32.58	30.17	33.76	21.88	39.79	35.57	19.05	26.23	25.53	31.42	-	-	-	-	-	37.73	-	-
F&A	53.27	90.89	51.88	40.77	25.76	29.35	28.70	32.74	21.50	39.05	34.68	17.77	18.31	25.81	30.35	0.28	0.00	0.09	4.33	4.15	36.06	1.77	27.48
FT	24.16	25.98	7.35	4.61	8.75	4.35	5.90	9.32	5.05	0.98	14.01	1.12	5.08	13.28	10.88	14.33	17.35	8.17	17.66	10.73	9.39	13.65	10.45
EWC*	42.75	85.48	29.32	9.69	1.55	14.24	14.78	6.78	7.85	4.72	12.34	2.28	9.09	19.16	6.29	16.17	11.13	10.36	15.81	12.61	17.75	13.22	16.62
LwF*	47.95	91.88	38.66	35.41	16.96	32.24	33.14	23.84	19.16	27.64	27.48	10.14	23.06	18.22	9.95	11.37	11.10	10.77	19.50	14.10	30.38	13.37	26.13
Ours	51.59	91.14	45.32	39.15	24.57	29.99	26.21	26.02	19.22	39.89	33.50	11.22	22.27	22.95	29.30	14.44	11.70	11.09	15.27	14.67	**34.16**	**13.43**	**28.98**
JT	53.96	92.17	52.18	42.82	27.12	35.84	35.87	33.75	23.83	36.98	35.63	17.50	26.11	25.73	32.48	19.54	11.30	10.08	24.15	18.08	38.13	16.63	32.76

Table 6. Class-wise IoU (%) performance comparison of 3D class-incremental segmentation methods on the ScanNet [2] dataset under S^1 split. “BT”, “F&A”, “FT”, “JT” denotes Base Training, Freeze and Add, Fine-Tuning and Joint Training respectively. Asterisk (*) denotes traditional class-incremental methods EWC [3] and LwF [4] in our reproduction for 3D semantic segmentation. For the forgetting-prevention-based method (gray face), the best IoU results for individual class are underlined, and the best mIoU results are in bold.

	ScanNet dataset (S^1), C_{novel}=5																						
	base classes															novel classes					mIoU (%)		
	0:bathtub	1:bed	2:bookshelf	3:cabinet	4:chair	5:counter	6:curtain	7:desk	8:door	9:floor	10:otherfurniture	11:picture	12:refrigerator	13:shower curtain	14:sink	15:sofa	16:table	17:toilet	18:wall	19:window	0-14	15-19	*all*
BT	35.72	33.32	31.89	20.07	51.95	25.67	17.32	24.74	23.72	91.95	20.06	9.91	12.66	16.60	23.97	-	-	-	-	-	29.30	-	-
F&A	30.25	27.52	27.04	17.12	45.92	20.25	12.17	18.29	20.57	89.11	16.73	6.93	11.57	11.75	23.59	11.73	15.51	14.79	33.83	17.76	25.25	18.72	23.62
FT	1.63	0.03	0.11	2.71	17.61	2.04	0.00	0.39	1.12	49.78	2.87	0.19	0.07	0.00	8.84	28.61	23.91	35.83	52.14	29.65	5.83	34.03	12.88
EWC*	16.33	10.96	11.57	9.24	29.43	4.70	1.75	8.04	18.39	76.79	8.42	4.85	2.23	8.16	13.08	28.23	24.90	30.49	52.86	30.01	14.93	33.30	19.52
LwF*	26.61	31.54	25.48	16.28	49.99	18.25	12.55	21.60	17.15	88.75	15.65	7.93	9.98	17.56	1.27	28.53	35.05	36.88	58.17	30.76	24.04	**37.88**	27.50
Ours	42.90	25.64	26.21	20.52	45.01	25.45	3.82	16.91	20.33	88.99	17.38	7.02	10.84	15.40	24.15	28.96	30.09	35.18	54.83	28.50	**26.04**	35.51	**28.41**
JT	42.49	32.19	35.28	23.62	53.32	24.46	19.34	27.59	26.87	92.14	18.65	10.98	15.47	17.52	22.25	31.61	44.21	33.22	54.15	30.76	30.81	38.79	32.81

Appendix C. Class-wise IoU Performance

Tab. 3 $\sim$ Tab. 6 details the class-wise IoU performance of our method compared with baselines on various datasets split under C_{novel}=5. We implement the joint training as the upper bound by training on both the base and novel classes at once. We notice that the results of our method are significantly better than other approaches on base classes. In particular, for Tab. 3 and Tab. 4, our novel model performs even better on 0-7 (base) classes than training only on base classes (base model). We believe it is due to that our Uncertainty-aware Pseudo-label Generation (UPG) strategy generates more accurate labels of base classes for the new data by eliminating the prediction uncertainties. These pseudo labels will provide better guidance for the novel model training. Moreover, the IoU result of novel class *"sofa"* in Tab. 3 obtain superior performance than it in Tab. 4. This may be due to the confusion between the novel class *"sofa"* and the similar base class *"chair"* in the incremental process of S^1 split, which also shows that different order will have a impact on the results. Moreover, under the experiment of S^1 on ScanNet dataset, our method performs slightly lower than LwF [4] on the novel 15-19 classes. We argue that the novel class samples under S^1 split are more common and numerous (*e.g. "wall"* and *"window"*), which is more likely to cause forgetting of base classes. In this case, our model focuses more (introduces more constraints) on maintaining base classes performance.

References

[1] Iro Armeni, Ozan Sener, Amir R Zamir, Helen Jiang, Ioannis Brilakis, Martin Fischer, and Silvio Savarese. 3d semantic parsing of large-scale indoor spaces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1534–1543, 2016. 1, 3

[2] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5828–5839, 2017. 1, 3

[3] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences (NAS)*, 114(13):3521–3526, 2017. 3

[4] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 40(12):2935–2947, 2017. 2, 3

[5] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 5099–5108, 2017. 1

[6] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics (TOG)*, 38(5):1–12, 2019. 1

[7] Wenxuan Wu, Zhongang Qi, and Li Fuxin. Pointconv: Deep convolutional networks on 3d point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9621–9630, 2019. 1