Towards Robust Few-shot Point Cloud Semantic Segmentation (Supplementary Material)

Yating Xu[1]
xu.yating@u.nus.edu
Na Zhao[2]
na_zhao@sutd.edu.sg
Gim Hee Lee[1]
gimhee.lee@nus.edu.sg

[1] Department of Computer Science
National University of Singapore
Singapore

[2] Singapore University of Technology and Design
Singapore

1 Ablation Study

Analysis of different R values. Tab. 1 shows the ablation study of different number of components for each shot in the component-level clean noise separation. 'R=1' is the shot-level representation. It can be seen that the performance of 'R=1' is generally worse than that of the component-level contrastive learning, which verifies that the feature is sub-optimized with a single holistic aggregation. By dividing into local components, we can get more fine-grained and diverse positive and negative samples with 'R=4' having the best performance.

model	0%	In-episode Noise		Out-episode Noise	
		20%	40%	40%	60%
R=1	67.62	65.83	55.94	65.17	57.46
R=2	67.93	**66.28**	57.41	65.08	58.57
R=4	**68.21**	66.02	**58.01**	**66.09**	**58.84**
R=8	67.40	65.58	56.66	65.52	57.94

Table 1: Effects of different number of components in CCNS.

Analysis of different noise ratios in CCNS. We analyze different combination of noise ratio in the episodic training since our component-level clean noise separation is conducted among the clean and noisy shots. '$\{0.2, 0.4\}$' has large performance drop when comparing with '$\{0, 0.2, 0.4\}$', which suggests that it is very necessary to include noise-free episodes during training. By further adding noise ratio of 0.6 (with the restriction that any number of noisy class should not outnumber the clean shots), there is again a significant drop in performance. We can conclude that only a mix of a proper portion of noisy and clean episodes during training can bring decent improvement in the noisy test.

Analysis of different scales in MDNS. Tab. 3 presents the analysis of different scales in the multi-scale degree-based noise suppression. Due to space limitation, we only provide the comparison of selected scales from the many possibilities of combinations. We first

Training Noise	0%	In-episode Noise		Out-episode Noise	
		20%	40%	40%	60%
{0.2, 0.4}	63.66	60.91	49.51	59.95	50.55
{0, 0.2, 0.4}	**68.21**	**66.02**	**58.01**	**66.09**	**58.84**
{0, 0.2, 0.4, 0.6}	66.40	65.39	55.00	63.39	56.11

Table 2: Effects of different simulated noise combinations.

analyze what constitutes a good scale. It is almost guaranteed that the holistic scale $\{1/1/1\}$ gives decent performance since the mean representation covers the general information. The performance varies a lot when the foreground objects are divided into fine-grained scales. By comparing $\{2/2/1\}$, $\{1/2/2\}$ and $\{2/1/2\}$, we can see that a cut on the z-axis causes a significant drop in performance on the heavy noise setting. By comparing $\{3/3/1\}$ with $\{2/2/1\}$, we can see that the cuts that are too fine-grained cause a performance drop due to the severe lack of the global information in the sub-shots. Overall, $\{1/1/1\}$ and $\{2/2/1\}$ are the good scales and their combination achieves the best performance.

$\{n_x/n_y/n_z\}$	In-episode Noise		Out-episode Noise	
	20%	40%	40%	60%
{1/1/1}	65.94	57.27	65.90	58.70
{2/2/1}	**66.47**	56.56	65.96	57.99
{1/2/2}	66.37	55.29	65.96	54.56
{2/1/2}	66.31	54.81	65.44	54.96
{3/3/1}	65.99	54.69	65.24	57.69
{1/1/1} & {2/2/1}	66.02	**58.01**	**66.09**	**58.84**
{1/1/1} & {2/2/1} & {3/3/1}	65.81	**58.01**	66.00	57.51

Table 3: Effects of different scale choices in MDNS.

2 t-SNE Visualization

Fig. 1 presents visualization of the feature distribution of testing classes on the S3DIS via t-SNE [5]. Each color represents a different class. By learning with our proposed component-level clean noise separation, the intra-class is more compact and inter-class is more separable.

Fig. 2 presents visualization of the feature distribution of training classes on the S3DIS via t-SNE [5]. Different colors represent different classes. It is obvious that the class-wise feature distribution of our method is more distinctive and compact than AttMPTI. Together with Fig. 1, we can conclude that our feature representation learning is able to not only make feature embedding of seen classes discriminative but also generalize to unseen classes.

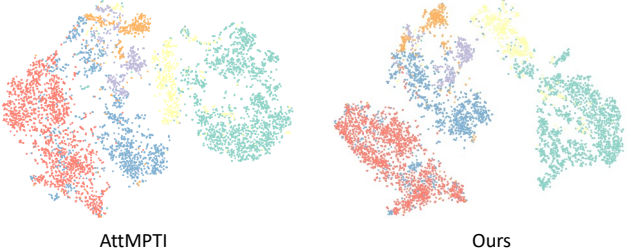

Figure 1: t-SNE [5] comparison on the **testing classes** of S3DIS.

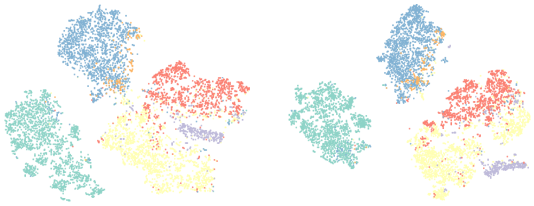

Figure 2: t-SNE [5] comparison on the **training classes** of S3DIS.

3 Experiment Results on ScanNet

Effectiveness of CCNS and MDNS. We analyze the effectiveness of our proposed component-level clean noise suppression (CCNS) and multi-scale degree-based noise suppression (MDNS) on ScanNet in Tab. 4. Both CCNS and MDNS are effective, and the combination of them achieves best overall performance. It is worth highlighting that the robustness of AttMPTI is improved by simply adding our feature representation learning, *i.e.* CCNS. It verifies our claim that AttMPTI has the potential to be noise robust (by FPS based multi-prototype generation and label propagation), yet is subject to how discriminative the feature embedding is.

model	0%	In-episode Noise		Out-episode Noise	
		20%	40%	40%	60%
AttMPTI	54.16	46.63	31.57	43.31	36.45
AttMPTI+CCNS	**54.79**	47.80	32.92	45.54	**38.57**
Ours	53.50	**49.78**	**38.70**	**47.90**	38.42

Table 4: Effectiveness of CCNS and MDNS on the ScanNet on 2-way 5-shot. 'Ours' consists of both CCNS and MDNS.

Qualitative Results. Fig. 3 presents the qualitative comparison between our method and AttMPTI under a 2-way 5-shot point cloud segmentation with 40% out-episode noise on ScanNet [1]. With the interference of the noisy shots, AttMPTI [7] either fails to segment the target semantic object (see the result in the first row) or wrongly segment some background points as the target class (see the result in the second row). In contrast, our method is able to give reliable segmentation results with respect to the target classes.

4 Data split

We follow the data split of [7], and adopt the split 0 as the testing classes as shown in Tab. 5.

Dataset	Meta-Testing Classes	Meta-Training Classes
S3DIS	beam, board, bookcase, ceiling, chair, column	door, floor, sofa, table, wall, window
ScanNet	bathtub, bed, bookshelf, cabinet, chair, counter, curtain, desk, door, floor	otherfurniture, picture, refrigerator, shower curtain, sink, sofa, table, toilet, wall, window

Table 5: Data split in S3DIS and ScanNet.

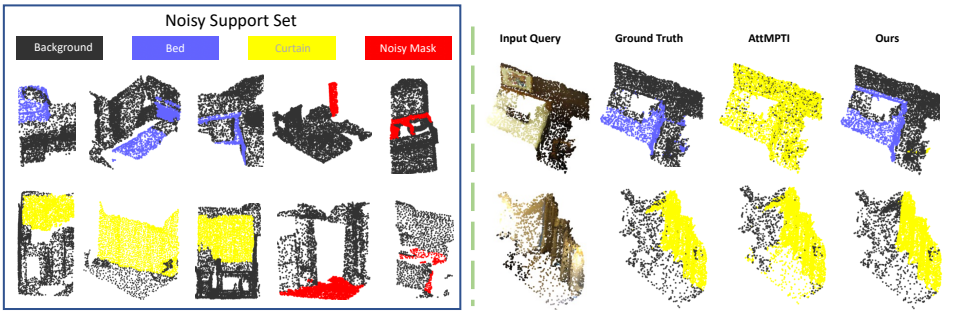

Figure 3: Qualitative comparison between AttMPTI and our method under a 2-way 5-shot point cloud segmentation with 40% out-episode noise on ScanNet. Each row shows the segmentation results of a target class from the corresponding support set in the 2-way setting.

5 Clean Ratio Comparison

Tab. 6 lists the clean ratios of the original support set ('Original') and the filtered support set produced by the MDNS ('Ours') during meta-testing. The clean ratio in each noise setting is given by first computing the percentage of the number of the clean shots in the corresponding set of one episode and then averaging the percentages in all episodes. As can be clearly seen from Tab. 6, our method can significantly improve the clean ratio in all the noise setting.

Model	In-episode Noise		Out-episode Noise	
	20%	40%	40%	60%
Original	0.8000	0.6000	0.6000	0.4000
Ours	**0.9749**	**0.8211**	**0.8476**	**0.5602**

Table 6: Comparison of the clean ratio of the 2-way 5-shot noisy support set in the meta-testing stage on the S3DIS.

6 Baseline Setups

We compare our method with few-shot point cloud semantic segmentation (3DFSSeg) methods AttMPTI [7] and ProtoNet [4], robust few-shot learning (R2DFSL) method Tra-NFS [7] and robust point cloud semantic segmentation (R3DSeg) method PNAL [6]. All methods use the same feature extractor as AttMPTI for fair comparison.

We follow the official code in AttMPTI to train ProtoNet and AttMPTI. For Tra-NFS, we adopt three-layer transformer encoder to generate robust prototype. We also randomly inject noise into the support set by sampling point clouds containing foreground objects from other classes during meta-training. For PNAL, we apply its robust training algorithm on each noisy support set and then test the performance on the corresponding query point cloud in each episode during meta-testing. We do not carry forward the knowledge from one episode to the next as suggested in [3].

References

[1] 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*, pages 5828–5839, 2017.

[2] Kevin J Liang, Samrudhdhi B Rangrej, Vladan Petrovic, and Tal Hassner. Few-shot learning with noisy labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9089–9098, 2022.

[3] Pratik Mazumder, Pravendra Singh, and Vinay P Namboodiri. Rnnp: A robust few-shot learning approach. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2664–2673, 2021.

[4] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30, 2017.

[5] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[6] Shuquan Ye, Dongdong Chen, Songfang Han, and Jing Liao. Learning with noisy labels for robust point cloud segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6443–6452, 2021.

[7] Na Zhao, Tat-Seng Chua, and Gim Hee Lee. Few-shot 3d point cloud semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8873–8882, 2021.