JUNE 18-22, 2023
CVPR
VANCOUVER, CANADA

四川大學
SICHUAN UNIVERSITY

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

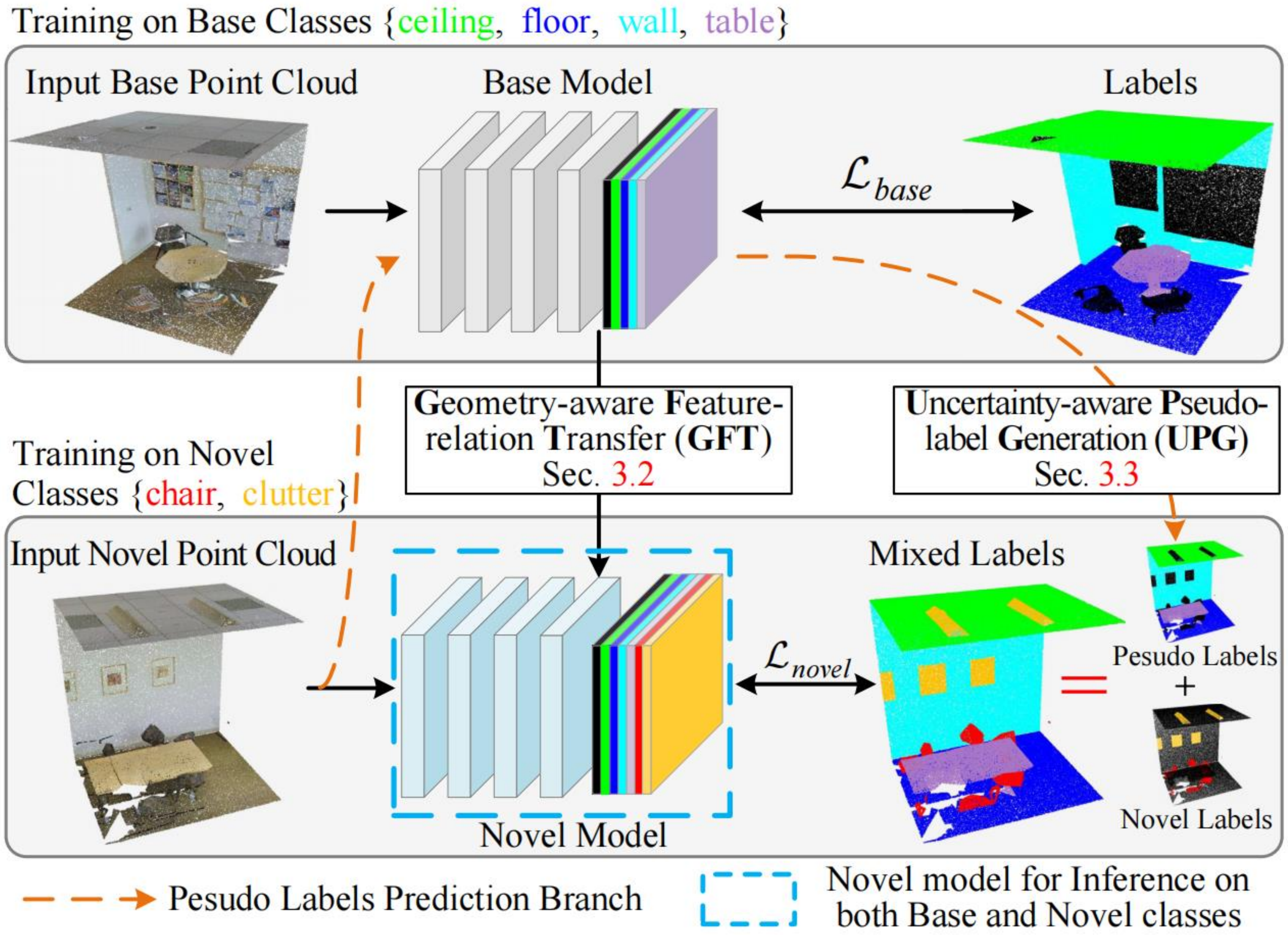
Yuwei Yang[1] Munawar Hayat[2] Zhao Jin[1] Chao Ren[1] Yinjie Lei[1,✉]

[1]Sichuan University [2]Monash University

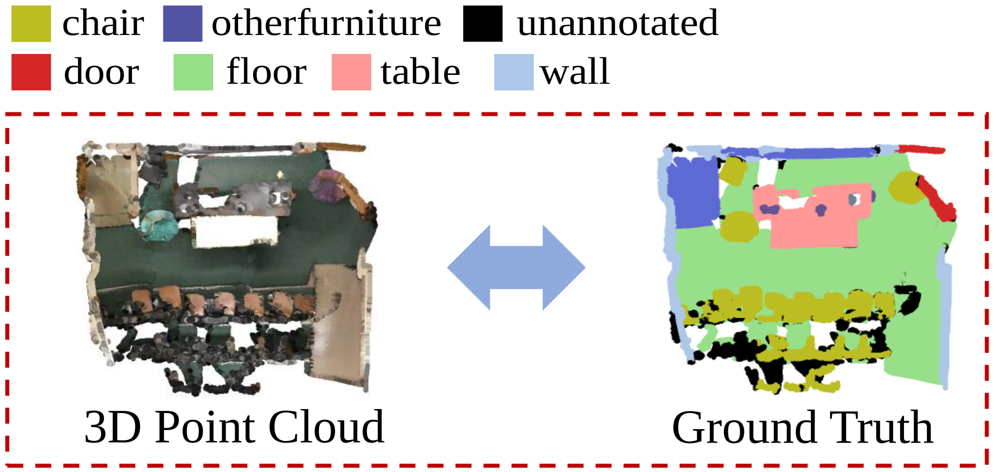
THU-PM-110

00 Preview

- We are the first to propose a **class-incremental learning framework for 3D point cloud semantic segmentation;**
- To **transfer previous knowledge** and **prevent forgetting** caused by unstructured point cloud, we propose a **GFT module;**
- To **tackle the semantic shift** issue where old classes are indiscriminately collapsed into the background, we design an **UPG strategy.**

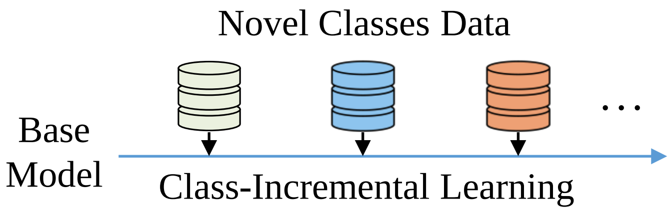


01 Background and Motivation



In the traditional point cloud semantic segmentation setting, all classes are <u>**learned at once**</u> (Joint Training)

New categories are gradually discovered in real-life scenarios, and updating the model to cater for these new categories <u>**requires large memory storage and expensive re-training**</u>



Class-Incremental Learning provides a promising paradigm

Catastrophic Forgetting?

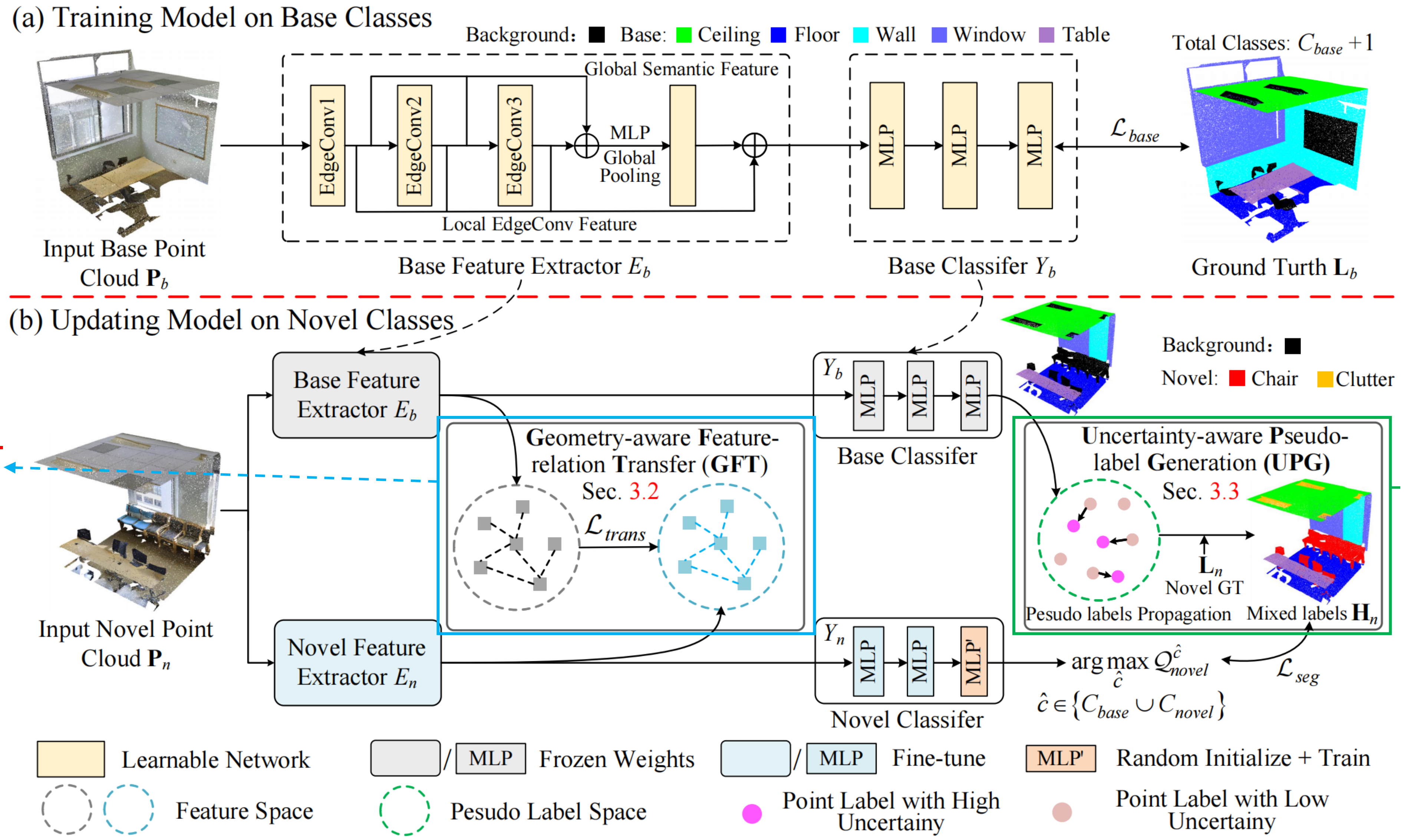
Semantic Shift?

3D point cloud are disordered and unstructured, making it difficult to preserve previous knowledge

The points belonging to old classes are indiscriminately **collapsed into background** during the current learning step

exacerbating forgetting

02 Method Overview

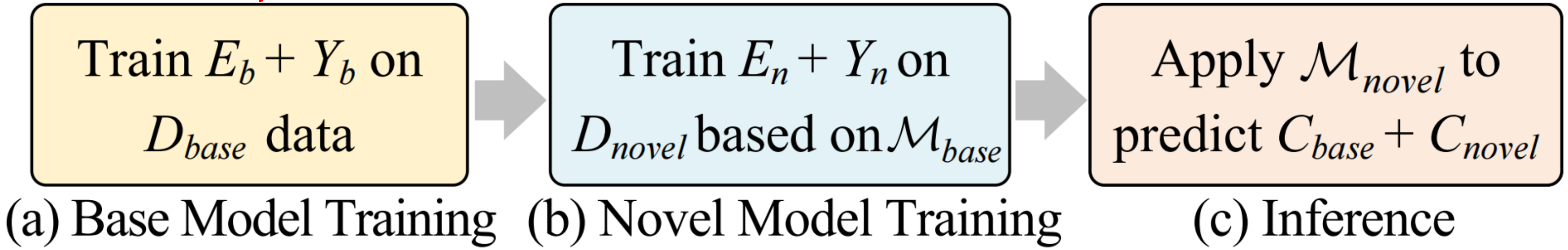


To **distill** previous **point-wise feature relations**

To **tackle the semantic shift**

03 Overall Pipeline → Class-Incremental Segmentation on 3D Point Cloud

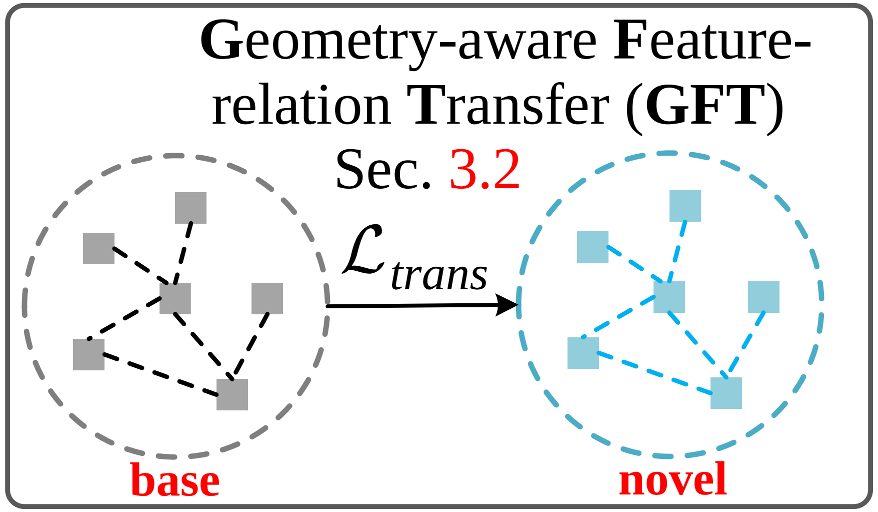
$$\mathcal{L}_{base} = -\sum_{i}\sum_{c}\mathbf{L}_b^i log(Y_b^c(E_b(\mathbf{P}_b^i)))$$



$$\mathcal{L}_{novel} = \mathcal{L}_{seg} + \mathcal{L}_{trans}$$

- **Train the base model** (feature extractor E_b combined with the classifier Y_b, denoted as base/old model M_{base}) on D_{base};
- Use the pre-trained base model to **initialize a new model and randomly initialize the last layer of new classifier Y_n** (denoted as novel model M_{novel}), and train on D_{novel} data;
- Apply novel model M_{novel} to **segment point clouds of all $C_{base}+C_{novel}$ classes** in the evaluation phase.

03 Geometry-aware Feature-relation Transfer (GFT)



We argue that the geometry-aware feature relation is discriminative for various semantic categories of 3D point cloud, and can be exploited to migrate knowledge while learning continually.

(1) Apply the Farthest Point Sampling (FPS) to find anchor points;

(2) Use *xyz* coordinates to calculate the distance **between other points and anchor points** to sample *k*-nearest neighbors to form areas reflecting the local geometric structures;

(3) Model the **point-wise relative relationships** within the geometric neighbors:

$$\mathcal{R}^a = \frac{1}{K}\sum_{k\in\mathcal{N}(a)} \overbrace{\boxed{(p_n^{a,k} - p_n^a)}}^{\text{relative } xyz \text{ coordinates}} \oplus \underbrace{\boxed{(\mathbf{F}_n^{a,k} - \mathbf{F}_n^a)}}_{\text{relative features}}$$

(4) Perform **base-to-novel feature relation distillation** via MSE loss:

$$\mathcal{L}_{trans} = \frac{1}{Z}\sum_{a=1}^{Z} ||\mathcal{R}^a_{novel} - \mathcal{R}^a_{base}||^2$$

03 Uncertainty-aware Pseudo-label Generation (UPG)

Different from the traditional Monte Carlo Dropout (MC-Dropout) method, which performs multiple predictions to estimate uncertainty, we apply **neighborhood spatial aggregation method combined with MC-dropout** [27] to complete the estimation of the point distribution uncertainty **at once**.

Conditional probability

Bernoulli distribution over the weights of the neighboring points distribution

$$\mathcal{U}_n^i = -\sum_c \Big[\frac{1}{T}\sum_t q(y_n^i = c|\mathbf{P}_n^i, \hat{\omega}_t)\Big]\log\Big[\frac{1}{T}\sum_t q(y_n^i = c|\mathbf{P}_n^i, \hat{\omega}_t)\Big] + \frac{1}{T}\sum_{c,t} q(y_n^i = c|\mathbf{P}_n^i, \hat{\omega}_t)\log q(y_n^i = c|\mathbf{P}_n^i, \hat{\omega}_t),$$

Predicted Labels

Point Cloud

Bayesian Active Learning by Disagreement (BALD) [15] as our spatial sampling uncertainty estimation function

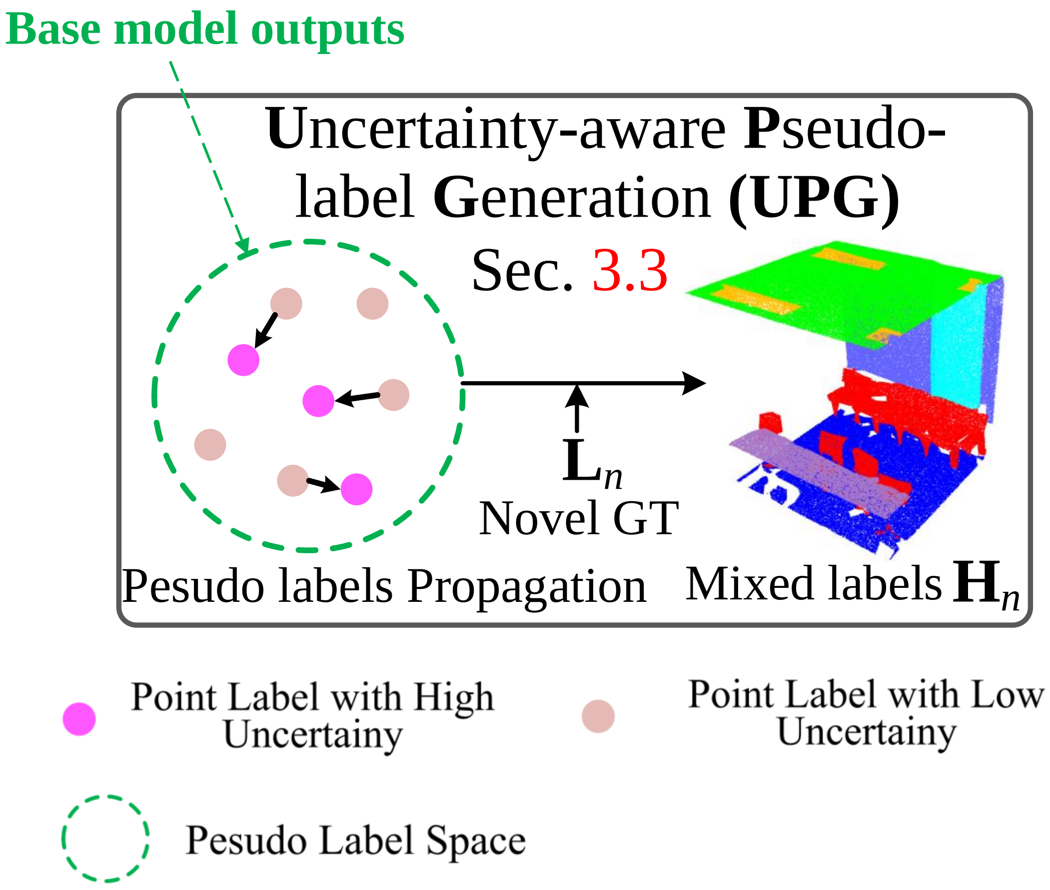
***T*-Neigbors**

Base Model output probability

$$\mathcal{U}_n^i = -\sum_c \Big[\frac{1}{T}\sum_t \mathcal{Q}_{base}^{i,c} * \sigma_n^t\Big]\log\Big[\frac{1}{T}\sum_t \mathcal{Q}_{base}^{i,c} * \sigma_n^t\Big] + \frac{1}{T}\sum_{c,t}(\mathcal{Q}_{base}^{i,c} * \sigma_n^t)\log(\mathcal{Q}_{base}^{i,c} * \sigma_n^t).$$

Calculate the **normalized cosine similarity** between neighbors and point *i* to implement $\hat{\omega}_t$

03 Uncertainty-aware Pseudo-label Generation (UPG)



We set τ as the threshold to determine the points with the high or low uncertainty. For a point with high uncertainty, our strategy is to **replace its prediction with the label of its nearest neighbor *t*** having the low uncertainty:

We combined the pseudo label with the current novel class labels to **form the mixed labels** using:

$$\mathbf{H}_n^i = \begin{cases} \underset{c}{\operatorname{argmax}}\, \mathcal{Q}_{base}^{i,c} & \mathbf{L}_n^i = c'_{bg}, \underset{c}{\operatorname{argmax}}\, \mathcal{Q}_{base}^{c} \neq c_{bg} \text{ and } \mathcal{U}_n^i \leq \tau, \\ \underset{c}{\operatorname{argmax}}\, \mathcal{Q}_{base}^{t,c} & \mathbf{L}_n^i = c'_{bg}, \underset{c}{\operatorname{argmax}}\, \mathcal{Q}_{base}^{c} = c_{bg} \text{ or } \mathcal{U}_n^i > \tau, \\ \mathbf{L}_n^i & \mathbf{L}_n^i \neq c'_{bg}, \\ \text{ignore} & \text{otherwise}, \end{cases}$$

Finally, the **cross-entropy segmentation loss** is constructed for novel model training:

$$\mathcal{L}_{seg} = -\sum_i \sum_{\hat{c}} \mathbf{H}_n^i log(\mathcal{Q}_{novel}^{i,\hat{c}})$$

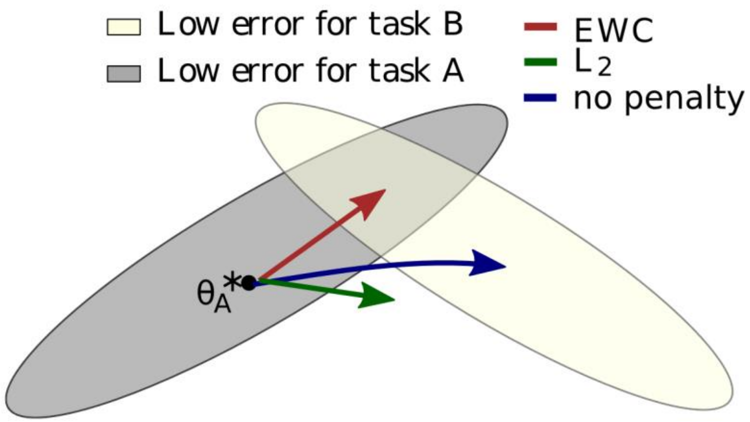
04 Experimental Results

To compare our approach, we design 4 baselines in 2 directions:

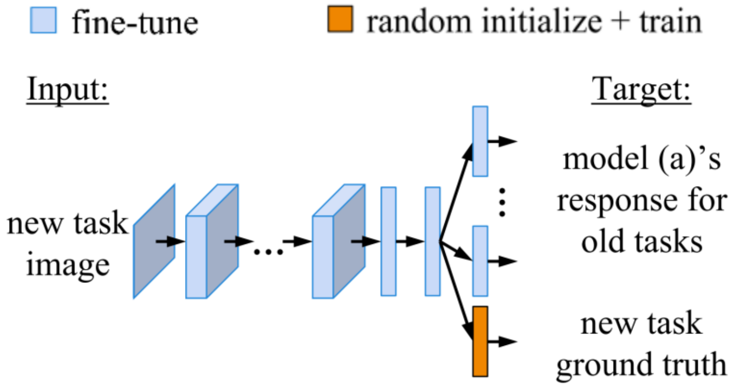
1) ***Direct adaptation methods.*** “Freeze and Add (**F&A**)” : Freeze the base model and adds a novel classifier output layer when training on the D_{novel}. “Fine-Tuning (**FT**)”: Randomly initialize the new classifier last layer and joins the base model for fine-tuning.

2) ***Forgetting-prevention methods.*** Adapt Elastic Weight Consolidation (**EWC**) and Learning without Forgetting (**LwF**) method from **classical incremental learning models** to 3D point cloud incremental segmentation setting.

*Base Training (**BT**): Training model on Base classes.
*Joint Training (**JT**): Joint training on all Base+Novel classes. **(Upper Bound)**



Elastic Weight Consolidation (EWC) [19]



Learning without Forgetting (LwF) [21]

04 Experimental Results

Two different paradigms to develop C_{base} and C_{novel}:

One (S^0) where classes are incrementally introduced as per their **original class label order** in the dataset, and the other (S^1) introduces classes **in an alphabetical order.**

Table 1. Experimental comparisons of 3D class-incremental segmentation methods on S3DIS dataset of S^0 and S^1 split. We apply the mIoU (%) as the evaluation metric. "BT", "F&A", "FT" in the table represents Base Training, Freeze and Add, Fine-Tuning respectively. "JT" denotes Joint Training on all base+novel classes at once. Asterisk (*) denotes traditional class-incremental methods EWC [19] and LwF [21] in our reproduction for 3D semantic segmentation. The joint training is treated as the upper bound, and the best results of incremental learning methods are in bold.

Methods	C_{novel}=5						C_{novel}=3						C_{novel}=1					
	S^0			S^1			S^0			S^1			S^0			S^1		
	0-7	8-12	*all*	0-7	8-12	*all*	0-9	10-12	*all*	0-9	10-12	*all*	0-11	12	*all*	0-11	12	*all*
BT	48.54	-	-	37.24	-	-	46.80	-	-	40.73	-	-	45.00	-	-	45.88	-	-
F&A	44.25	12.33	31.98	37.71	42.89	39.44	44.28	3.34	34.83	41.11	35.64	39.85	44.57	0.05	41.14	45.35	0.05	41.86
FT	34.96	30.25	33.15	10.99	50.67	26.53	28.87	31.56	29.49	17.83	54.69	26.34	29.44	29.52	29.45	23.80	5.74	22.41
EWC*	39.38	31.07	36.19	23.19	54.84	35.36	37.13	37.92	37.31	29.38	55.53	35.41	36.55	19.94	35.27	25.60	9.81	24.39
LwF*	44.55	35.01	40.88	32.83	55.19	41.43	43.07	38.34	41.98	37.69	54.73	41.62	39.94	35.50	39.60	32.16	18.26	31.09
Ours	**48.94**	**39.56**	**45.33**	**38.17**	**55.20**	**44.72**	**45.15**	**45.33**	**45.19**	**39.83**	**57.59**	**43.93**	**44.08**	**35.69**	**43.43**	**40.33**	**19.28**	**38.71**
JT	50.23	41.74	46.97	38.38	60.11	46.74	48.62	41.44	46.97	42.63	60.44	46.74	47.51	40.41	46.97	47.09	42.55	46.74

Base Classes **Novel Classes**

Considering the overall mIoU, our method consistently **achieves the best results.**

04

Experimental Results

Table 2. Experimental comparisons of 3D class-incremental segmentation methods on ScanNet dataset of S^0 and S^1 split. We apply the mIoU (%) as the evaluation metric. "BT", "F&A", "FT" in the table represents Base Training, Freeze and Add, Fine-Tuning respectively. "JT" denotes Joint Training on all base+novel classes at once. Asterisk (*) denotes traditional class-incremental methods EWC [19] and LwF [21] in our reproduction for 3D semantic segmentation. The joint training is treated as the upper bound, and the best results of incremental learning methods are in bold.

Methods	C_{novel}=5						C_{novel}=3						C_{novel}=1					
	S^0			S^1			S^0			S^1			S^0			S^1		
	0-14	15-19	*all*	0-14	15-19	*all*	0-16	17-19	*all*	0-16	17-19	*all*	0-18	19	*all*	0-18	19	*all*
BT	37.73	-	-	29.30	-	-	34.03	-	-	30.84	-	-	31.57	-	-	30.78	-	-
F&A	36.06	1.77	27.48	25.25	18.72	23.62	32.58	0.86	27.82	26.95	7.37	24.02	30.99	0.95	29.49	30.41	0.01	28.89
FT	9.39	13.65	10.45	5.83	34.03	12.88	8.43	10.98	8.82	4.88	40.94	10.29	8.02	10.46	8.14	4.76	7.57	4.90
EWC*	17.75	13.22	16.62	14.93	33.30	19.52	15.70	11.74	15.11	8.78	31.74	12.22	15.66	6.76	15.21	12.24	8.84	12.07
LwF*	30.38	13.37	26.13	24.04	**37.88**	27.50	26.22	13.88	24.37	22.76	**42.34**	25.70	22.15	12.56	21.67	20.63	**13.88**	20.29
Ours	**34.16**	**13.43**	**28.98**	**26.04**	35.51	**28.41**	**28.38**	**14.31**	**26.27**	**28.79**	40.31	**30.52**	**25.74**	**12.62**	**25.08**	**24.16**	12.97	**23.60**
JT	38.13	16.63	32.76	30.81	38.79	32.81	35.46	17.44	32.76	31.65	39.38	32.81	33.53	18.08	32.76	32.91	30.76	32.81

Our approach achieves promising results, **closer to the joint training (upper bound)** using all data at once.

04 Experimental Results

Table 3. Experiments with various backbones on S3DIS dataset.

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

Our approach has a **consistent and superior performance** close to the joint training with various backbones.

Multi-step increment (compared to adding novel classes at once) is **more challenging,** since the model need to deal with the semantic shift of both old and the unknown future classes.

Table 4. Multi-step incremental segmentation of overlapped setting on S3DIS datasets in S^0 split. We use mIoU (%) as the evaluation metric. The first 8 classes are base, and the remaining 5 classes are novel. Instead of the incremental procedure in the disjoint setting of C_{novel}=5, We use multi-step increments, each step increments 1 class, total increments 5 times.

	0	1	2	3	4	5	6	7	8	9	10	11	12	*all*	*base*	*novel*
Base Training	88.74	96.58	73.30	0.00	6.76	40.60	17.61	64.70	-	-	-	-	-	-	48.54	-
Step 1	88.09	95.63	73.46	0.00	7.93	39.70	22.76	63.06	34.42	-	-	-	-	47.23	48.83	34.42
Step 2	85.46	95.66	71.57	0.00	0.90	32.01	19.74	50.07	13.69	3.73	-	-	-	37.28	44.43	8.71
Step 3	85.72	95.76	72.07	0.00	0.81	34.79	11.74	51.78	12.32	3.84	44.31	-	-	37.56	44.08	20.16
Step 4	85.91	95.37	65.17	0.00	0.00	31.28	6.82	44.29	0.21	5.04	40.10	8.02	-	31.85	41.11	13.34
Final Step 5	88.04	95.83	65.89	0.00	0.00	34.48	8.01	44.38	0.07	3.56	36.18	10.63	33.50	32.35	42.08	16.79

04 Experimental Results → Incremental classification across datasets

Acc_o^* The base model's accuracy;

Acc_o Accuracy on base classes using the final incremental model;

Acc_n Accuracy on novel classes using the final incremental model;

$$\Delta = \frac{Acc_o^* - Acc_o}{Acc_o^*} \times 100\%$$

The lower Δ represents less forgetting.

The statistics of the cross-dataset incremental classification

Settings	Task	#Classes	#Train	#Test
ModelNet40→ScanObjectNN	Old	26	4999	1496
	Novel	11	1496	475
ModelNet40→ModelNet10	Old	30	5852	1560
	Novel	10	3991	908

Backbone	Methods	ModelNet40 →ScanObjectNN				ModelNet40 →ModelNet10			
		$Acc_o^*\uparrow$	$Acc_o\uparrow$	$Acc_n\uparrow$	$\Delta\downarrow$	$Acc_o^*\uparrow$	$Acc_o\uparrow$	$Acc_n\uparrow$	$\Delta\downarrow$
DGCNN [38]	lwf-3D [6]*	92.91	73.34	79.41	21.06	91.71	87.14	93.32	4.98
	+GFT	92.91	76.31	81.19	17.87	91.71	88.95	93.32	3.01
	+GFT+UPG	**92.91**	**78.19**	**82.82**	**15.84**	**91.71**	**88.99**	**93.86**	**2.97**
PointNet [28]	lwf-3D [6]*	90.14	84.77	76.87	5.96	88.71	81.59	90.41	8.03
	+GFT	90.14	84.09	77.15	3.92	88.71	82.23	90.43	7.30
	+GFT+UPG	**90.14**	**86.84**	**79.12**	**3.66**	**88.71**	**83.18**	**91.27**	**6.20**
PointConv [39]	lwf-3D [6] *	92.69	87.19	79.33	5.93	91.26	83.59	92.13	8.41
	+GFT	92.69	88.32	79.75	4.71	91.26	83.80	92.03	8.18
	+GFT+UPG	**92.69**	**88.79**	**80.08**	**4.21**	**91.26**	**84.65**	**93.32**	**7.24**

Introducing proposed GFT/UPG module **significantly improve all performance** across various backbone architectures.

04 Experimental Results

Our approach strikes a balance between keeping the knowledge of base classes and learning the novel classes.

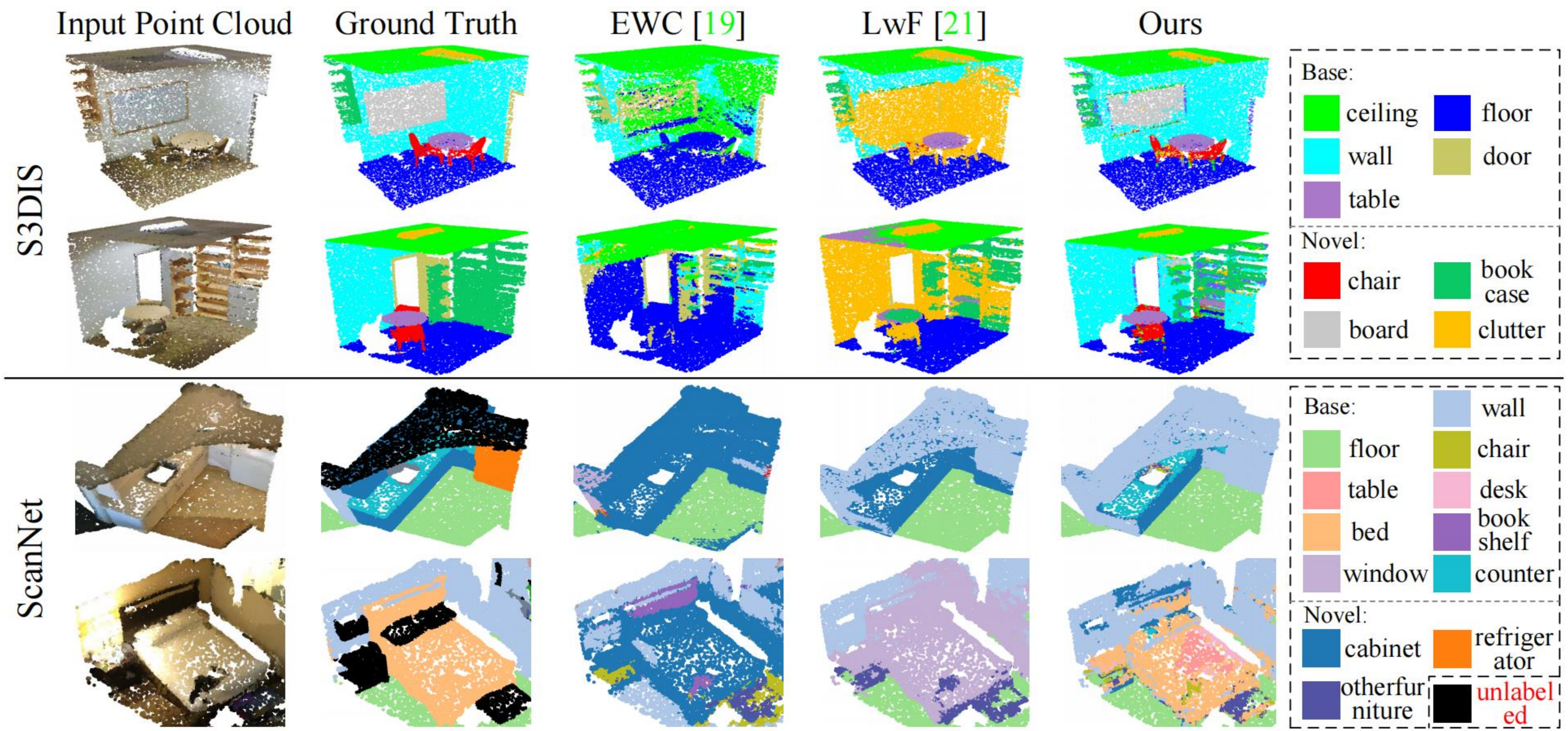


Figure 4. Qualitative comparison with the forgetting-prevention methods EWC [19] and LwF [21] on S3DIS and ScanNet datasets of $C_{novel} = 5$ in S^0 split. Only the base and novel classes included in current point cloud scenario are explained in the legend. Results in black on ScanNet dataset represent unlabeled and do not belong to either the base or novel classes.

JUNE 18-22, 2023 CVPR VANCOUVER, CANADA

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

Project: *https://github.com/leolyj/3DPC-CISS*

THANKS