Hyperbolic Uncertainty-Aware Few-Shot Incremental Point Cloud Segmentation

Supplementary Material

1. Overview

In this supplementary document, we discuss in detail the architecture with an illustrative diagram (in Fig. 1) and its hyperparameter details (in Sec. 2).

Following that, we discuss the preliminary concepts related to operations in Hyperbolic space with their associated formulae (in Sec. 3). We detail the evaluation metrics used in our work (in Sec. 4). We discuss the pseudocode for HIPO following the same, to better understand the working of our method, which we believe will assist the reader to easily understand the working of HIPO (in Sec. 5). After briefly justifying the use of Hyperbolic space over Euclidean space and the need for Riemannian optimization for hierarchical data (in Sec. 6), we detail the dataset setup (in Sec. 7) and finally provide some additional experimental results(in Sec. 8). We conclude with a qualitative comparative visualization of the baseline methods versus our proposed method (in Fig. 2), as well as additional experimental results, including the ablation of the backbone $\mathcal{E}(\cdot)$ architecture (Table 2) and a comprehensive result for the 4 - 3T setting on the S3DIS dataset, highlighting how performance evolves across incremental sessions (Table 3).

2. Architecture and hyperparameter details

We have adopted PointTransformer [16] as the backbone architecture $\mathcal{E}(.)$ due to its recent state-of-the-art performance in point cloud recognition tasks. The architecture comprises several sequentially placed blocks, each containing two components. For our proposed HIPO framework, we implement a Hyperbolic Classifier (Möbius + softmax) head $\mathcal{H}(.)$ as shown in Fig. 1. We also use a learning rate scheduler whereby, after 50 epochs, the learning rate decays by 0.5. We empirically choose the nearest neighbors $K = 12$. We have incorporated the following preprocessing techniques during training: point clouds are augmented by Gaussian jitter and random rotation around the z-axis.

3. Preliminary definitions and formulae

All the important variables are detailed in Table 1.

Poincaré Ball Model: An n-dimensional Poincaré Ball model with curvature $-c$ is defined as hyperbolic space $\mathbb{B}_c^n = \{x \in \mathbb{R}^n : c\|\boldsymbol{x}\| < 1\}$ and $\mathrm{g}_x^{\mathbb{B}_c} = (\lambda_x^c)^2 \mathbb{I}_n$ is the corresponding Riemannian metric tensor at the point x. λ_x^c is the conformal factor defined as $\lambda_{\boldsymbol{x}}^c = \frac{2}{\left(1-c\|\boldsymbol{x}\|^2\right)}$. Here, $\mathbb{I}_n$ is the Euclidean metric tensor.

Möbius Addition: For $\forall\, \boldsymbol{z_1} \in \mathbb{B}_c^n$ and $\forall\, \boldsymbol{z_2} \in \mathbb{B}_c^n$, the Möbius addition is defined as follows:

$$\boldsymbol{z_1} \oplus_c \boldsymbol{z_2} = \frac{(1 + 2c\langle \boldsymbol{z_1}, \boldsymbol{z_2}\rangle + c\left\|\boldsymbol{z_2}\right\|^2)\boldsymbol{x} + (1 - c\left\|\boldsymbol{z_1}\right\|^2)\boldsymbol{z_2}}{1 + 2c\langle \boldsymbol{z_1}, \boldsymbol{z_2}\rangle + c^2\left\|\boldsymbol{z_1}\right\|^2\left\|\boldsymbol{z_2}\right\|^2}. \tag{1}$$

Exponential Map: For $\forall \boldsymbol{z} \in \mathbb{B}_c^n$, and $\forall \boldsymbol{v} \in \mathcal{T}_{\boldsymbol{z}}\mathbb{B}_c^n$ (i.e., $\mathcal{T}_z\mathbb{B}_c^n$ can be seen as Euclidean space), the exponential map $\exp_z^c(v) : \mathcal{T}_z\mathbb{B}_c^n \to \mathbb{B}_c^n$ is described as follows:

$$\exp_{\boldsymbol{z}}^c(\boldsymbol{v}) = \boldsymbol{z} \oplus_c \frac{1}{\sqrt{c}} \tanh\left(\frac{\sqrt{c}\lambda_{\boldsymbol{z}}^c\left\|\boldsymbol{v}\right\|}{2}\right)[\boldsymbol{v}]. \tag{2}$$

Here $\boldsymbol{z}$ is the anchor.

Distance: For $\forall \boldsymbol{z_1}, \boldsymbol{z_2} \in \mathbb{B}_c^n$, the distance $d_c : \mathbb{B}_c^n \times \mathbb{B}_c^n \to \mathbb{R}$ is given by the following:

$$d_c(\boldsymbol{z_1},\ \boldsymbol{z_2}) = \frac{2}{\sqrt{c}} \tanh^{-1}\left(\sqrt{c}\left\|-\boldsymbol{z_1} \oplus_c \boldsymbol{z_2}\right\|\right). \tag{3}$$

Geodesic Distance: The geodesic distance between two points $\boldsymbol{z_1}, \boldsymbol{z_2} \in \mathbb{B}_c^n$ is given by:

$$d_{\mathbb{B}}(z_1, z_2) = \operatorname{arcosh}\left(1 + 2\frac{\|z_1 - z_2\|^2}{(1 - \|z_1\|^2)(1 - \|z_2\|^2)}\right).$$

4. Evaluation Metrics

AIA and Last: We define average incremental accuracy **AIA** and accuracy after learning the final task **Last** in the main text following [6]. Let the accuracy after learning the task[1] t be:

$$A^{(\leq t)} = \frac{\#\textit{correctly classified samples in } \bigcup_{t'=0}^{t} D_{\textit{test}}^{(t')}}{\sum_{t'=0}^{t} |D_{\textit{test}}^{(t')}|}$$

Then, after training for all T sessions, **AIA** $= \frac{1}{T}\sum_{k=1}^{T} A^{(\leq k)}$ and **Last** $= A^{(\leq T)}$.

Here $\mathcal{D}_{\textit{test}}^{(t)}$ is the test-set for the session $S^{(t)}, t \geq 0$, and $\#$ is "the number of". In other words, $A^{(\leq t)}$ means the average accuracy of all the test data from task 0 to task t.

Average Forgetting Rate for CIL: Apart from the *classification accuracy* (ACC), we report another popular CIL evaluation metric *average forgetting rate*. The popular definition of average forgetting rate is the following:

$$\boldsymbol{\mathcal{F}^{(t)}} = \frac{1}{t-1}\sum_{i=1}^{t-1}(A_i^{(i)} - A_i^{(t)})$$

where $A_i^{(t)}$ is the accuracy of task i's test set on the CL model after task t is learned [7], which is also referred to as *backward transfer* in other literature [8].

[1] We use task and session synonymously

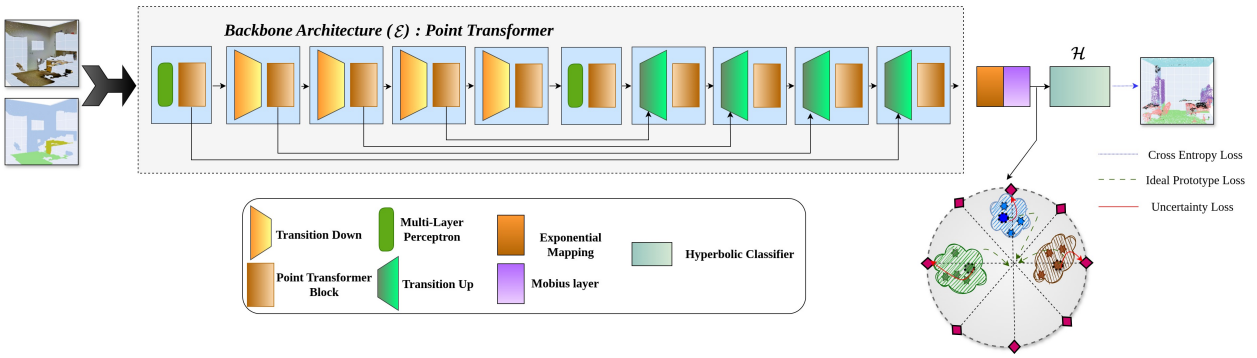

Figure 1. **Architecture diagram:** We use the PointTransformer network as our backbone architecture. The input is passed through $\mathcal{E}$, which consists of a set of Point Transformer blocks and Transition up and Transition down blocks. Then, the feature vector is passed through the exponential mapping and Möbius layer to be projected to the hyperbolic space. The learning is guided by the cross entropy loss, ideal prototype loss, and uncertainty loss, as depicted in the diagram.

Variable	Description
T	Total number of incremental sessions.
$S^{(t)}$	t^{th} incremental session.
$D^{(t)}$	t^{th} incremental session dataset.
$P_k^{(t)}$	k^{th} point cloud sample in the t^{th} session.
$M_k^{(t)}$	Associated labels of the k^{th} point cloud in the t^{th} session.
$C^{(t)}$	Label space of point cloud classes in the t^{th} session.
$z(c_i^{(t)})$	Hyperbolic prototype for the i^{th} class in the t^{th} session.
$\mathbb{B}_c^n$	Poincaré ball of n-dimensional with curvature c.
$\mathcal{O}_{\mathbb{B}}$	Origin of Poincaré ball $\mathbb{B}$.
u_i	Ideal prototype for class i.
$\mathcal{I}^n$	Set describing the boundary of the n-dimensional Poincaré ball formed by the n class prototypes.
$\exp_0(\cdot)$	Exponential map projecting points from Euclidean space to Hyperbolic space.
$\mathcal{H}(\cdot)$	Hyperbolic Classifier head.
$\mathcal{M}$	Möbius layer (also called Hyperbolic Feedforward layer).
$\mathcal{E}(\cdot)$	Feature extractor, the main backbone architecture.
$\mathcal{B}$	Busemann function.
ϕ	Regularization constant controlling strength of regularization effect for the uncertainty loss.

Table 1. **Variables used and associated meaning:** We describe the important variables used to formulate our proposed methodology HIPO.

$$\mathcal{F}_{\boldsymbol{Last}}^{(\boldsymbol{T})} = \frac{1}{T-1}\sum_{i=1}^{T-1}(A_i^{(i)} - A_i^{(T)}),$$

5. Pseudocode for HIPO

We discuss the working of our HIPO framework in the following pseudocode (Algorithm 1). Our method works identically for all the sessions (both base and subsequent incremental sessions). We first position the ideal prototypes corresponding to every individual class on the boundary of the Poincare ball. Following that, as data arrives for each session, we compute their Hyperbolic embedding onto the Poincare ball and try to align its representations with the corresponding ideal prototype. To avoid the vanishing gradient problem, we try to regularize the alignment with an uncertainty measure.

6. Justification of Hyperbolic space and the need for Riemannian optimization

In a recent study, Moreira *et al*. [9] demonstrate that in high-dimensional embeddings, the volume of a hyperbolic ball, similar to its Euclidean counterpart, is concentrated near the boundary. This formed a key argument in their work, where they asserted that a fixed-radius encoder utilizing the Euclidean metric can achieve better performance, irrespective of the embedding dimension. But, in formulating this hy-

Algorithm 1 HIPO: Hyperbolic Ideal Prototype Optimization

Require: $D = \{D_1, \ldots, D_T\}$, $C = \{C_1, \ldots, C_n\}$, Poincaré ball ($\mathbb{B}_c^n$) with curvature ($-c$), max_epochs

$\mathcal{I}^n \leftarrow$ **Optimal_positioning**(C) {*Optimize the ideal prototypes within the unit Poincaré ball to maximize the distance between semantically similar classes*}
for $t = 1$ to T **do**
  $P_k^{(t)}, M_k^{(t)} \leftarrow D^{(t)}$ {*Data and Labels*}
  for epoch $= 1$ to max_epochs **do**
    $v^t \leftarrow$ **mean**$(\mathcal{E}(P_k^{(t)}))$ {*Compute the mean feature vector for each class in Euclidean space*}
    $z^t \leftarrow \mathcal{M}(\exp_0(v^t))$ {*Project to Hyperbolic space to obtain Hyperbolic mean class prototype*}
    $\mathcal{L}_{\text{Alignment}} \leftarrow \mathcal{B}(z^t, \mathcal{I}^n)$ {*Align class prototype to ideal prototype*}
    $\mathcal{L}_{\text{Uncertainty}} \leftarrow$ **Uncertainty_regularizer**(z^t) {*Regularize alignment to avoid close proximity to ideal prototype*}
    loss $\leftarrow \mathcal{L}_{\text{Alignment}} + \mathcal{L}_{\text{Uncertainty}} +$ **CE**$(z^t, M_k^{(t)})$ {*Compute total loss*}
    Perform *loss* backpropagation {*Model parameter update*}
  end for
end for

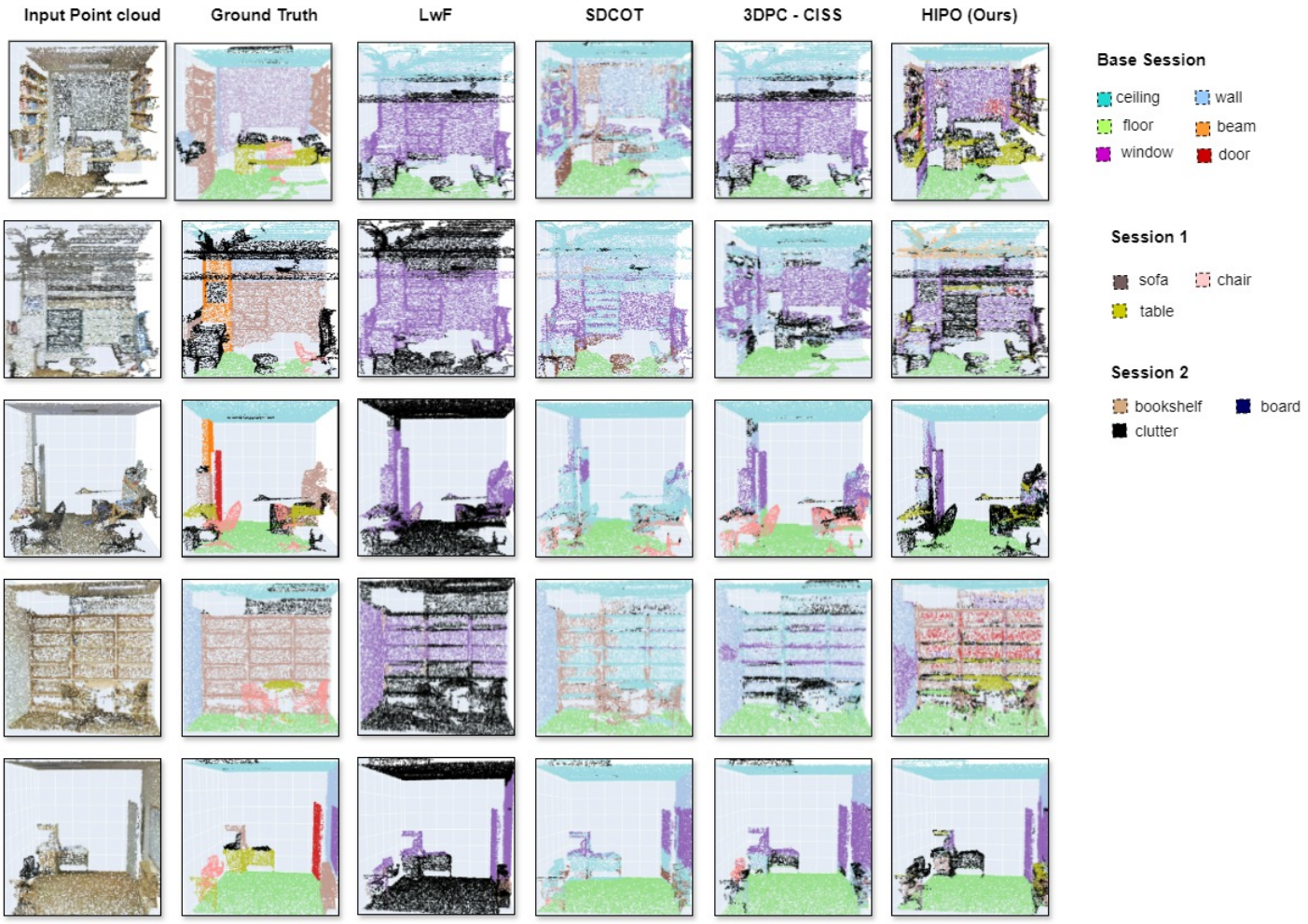

Figure 2. **Qualitative visualization of model performance for 7-3T setting:** Comparing the FSCIL results for 3D point cloud segmentation between our method (HIPO) and baseline approaches, we observe that even in a few-shot setup, our model performs significantly better in segmenting objects such as walls, counters, and others, when compared to the other methods.

Backbone	Methods	S3DIS [1]			ScanNetv2 [4]		
		Last↑	AIA↑	$\mathcal{F}^{(T)}_{Last}$↓	Last↑	AIA↑	$\mathcal{F}^{(T)}_{Last}$↓
DGCNN [12]	LwF [3]	9.70	25.68	23.97	4.14	21.32	26.98
	C-FSCIL [5]	15.37	24.54	28.76	8.95	20.32	23.10
	LGKD [15]	21.60	33.27	17.51	10.20	23.86	21.92
	SDCOT [17]	20.42	32.30	17.82	7.45	18.10	22.65
	3DPC-CISS [14]	21.60	32.65	16.56	13.87	23.52	19.33
	HIPO	**29.30**	**36.62**	**10.98**	**15.87**	**24.52**	**16.33**
PointTransformer [18]	LwF [3]	8.60	21.80	19.80	1.05	17.53	24.72
	C-FSCIL [5]	20.01	31.92	17.86	11.12	24.18	19.59
	LGKD [15]	19.76	30.41	15.97	9.55	25.52	22.45
	SDCOT [17]	12.00	29.46	26.20	0.40	17.83	26.15
	3DPC-CISS [14]	24.20	34.50	15.45	8.10	22.30	21.30
	HIPO	**29.40**	**36.80**	**11.10**	**14.65**	**27.22**	**18.85**

Table 2. Performance comparison in few shot incremental segmentation across the **S3DIS** dataset under a **7-3T** configuration and the **ScanNetV2** dataset under a **10-5T** setup with various backbone architectures. **AIA**(↑) is the *average incremental* mIoU (%). **Last** (↑) is the mIoU after learning the final task. $\mathcal{F}^{(T)}_{Last}$(↓) is the average forgetting rate (%) after learning the final task.

	ceiling	floor	wall	beam	column	window	door	table	chair	sofa	book case	board	clutter
Base Session	91.27	95.58	77.51	0.00	-	-	-	-	-	-	-	-	-
Session 1	90.03	94.21	72.00	0.00	3.94	29.97	8.73	-	-	-	-	-	-
Session 2	88.02	91.87	70.85	0.00	0.96	1.53	2.29	38.51	32.07	32.78	-	-	-
Session 3	87.89	93.34	67.94	0.00	0.00	0.00	0.00	16.68	30.22	25.98	35.22	10.77	22.95

Table 3. **Comparison of class-wise mIoU(%) values:** We compare the performance of HIPO concerning different classes across in the **4-3T** setting of **S3DIS** [1] dataset for 5-shot FSCIL.

pothesis, they relied on a strong first-order approximation to Equation 17 (refer to the main paper), thus assuming a faster convergence to zero. In our view, this approximation and the resulting upper bound leading to Equation 17 appear somewhat restrictive. We acknowledge the findings of [9] but emphasize that the use of Hyperbolic Geometry, rather than Euclidean Geometry, is not excessive. On the contrary, Hyperbolic Geometry offers numerous advantages for addressing the challenging problem of few-shot class-incremental learning. These benefits are evident from the experimental results described in Table [4] and Fig. [2] in the main paper, where Hyperbolic representation enhances class separation. Notably, we achieve improved performance in terms of AIA and $\mathcal{F}^{(T)}_{\text{Last}}$.

7. Dataset Setup

We apply a sliding window [10, 12] to divide the rooms of S3DIS and ScanNet into 7,547 and 36,350 $1m \times 1m$ blocks, respectively, and randomly sample 2048 points in each block as input. For the FSCIL setup, We follow a stricter version of the disjoint setting in [2] where the incremental sessions include only the current classes of the point cloud but not the old or future ones. For FSCIL in 3D point cloud segmentation, we create three scenarios for both in-domain (S3DIS, ScanNetv2) and cross-dataset (ScanNetv2 $\rightarrow$ S3DIS) settings. For **(a)** S3DIS, we have constructed the scenarios 4 - 3T, 7 - 3T and 10 - 3T where scenario M - NT implies M classes in the base session $S^{(0)}$ and N novel classes in each incremental session $S^{(t)}$, $t > 0$ **(b)** ScanNetv2, we have designed 10 - 2T, 10 - 5T and 15 - 5T scenarios **(c)** ScanNetv2 $\rightarrow$ S3DIS, scenarios 10 - 2T, 10 - 4T and 10 - 8T have been constructed likewise. The motivation behind constructing such scenarios stems from the fact that the strength of catastrophic forgetting increases with an increase in the number of incremental sessions as observed in point cloud recognition task [11].

8. Additional Experimental results

We present detailed results for the 4 - 3T setting on the S3DIS dataset, highlighting the change in class-wise mIoU values for HIPO in Table 3. As we can observe, there is very minimal forgetting of the classes introduced in the base session. Some classes introduced in subsequent sessions suffer significant forgetting, while others undergo minimal forgetting. This presents an open direction for future research, inspiring further exploration. We also present a visual com-

parative study in Fig. 2, illustrating the performance of key baselines and HIPO in a 7 - 3T setting on the S3DIS dataset. It is evident that HIPO performs much better than all other baselines in accurately classifying the points. For instance, HIPO identifies the "bookshelf" with greater precision compared to the baselines (as seen in the fourth image). Similarly, points belonging to the "clutter" class are correctly identified (in the second image).

8.1. Ablation studies on backbone architecture

We conduct our experiments using two recent state-of-the-art backbone architectures: the DGCNN architecture [12] and the PointTransformer architecture [16].

From Table 2, it is evident that our proposed framework, based on Hyperbolic Ideal Prototype optimization, effectively mitigates catastrophic forgetting for both DGCNN [12] and PointTransformer [18]. This demonstrates that our approach is backbone-agnostic.

9. Limitations

As discussed in section 3.2.1, we position Hyperbolic Ideal Prototypes for n predefined classes at the boundary of a Poincaré Ball. For experiments conducted on S3DIS [1] and ScanNetV2 [4], we assume $n = 13$ and $n = 20$ respectively. While we assume the predefined classes to be equal to the number of categories in the datasets as mentioned earlier, the n predefined object categories can be adapted from the most common indoor object categories in the world [13].

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*, pages 1534–1543, 2016. 4, 5

[2] Fabio Cermelli, Massimiliano Mancini, Samuel Rota Bulo, Elisa Ricci, and Barbara Caputo. Modeling the background for incremental learning in semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9233–9242, 2020. 4

[3] Townim Chowdhury, Mahira Jalisha, Ali Cheraghian, and Shafin Rahman. Learning without forgetting for 3d point cloud objects. In *Advances in Computational Intelligence: 16th International Work-Conference on Artificial Neural Networks, IWANN 2021, Virtual Event, June 16–18, 2021, Proceedings, Part I 16*, pages 484–497. Springer, 2021. 4

[4] 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. 4, 5

[5] Michael Hersche, Geethan Karunaratne, Giovanni Cherubini, Luca Benini, Abu Sebastian, and Abbas Rahimi. Constrained few-shot class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9057–9067, 2022. 4

[6] Gyuhak Kim, Zixuan Ke, and Bing Liu. A multi-head model for continual learning via out-of-distribution replay, 2022. 1

[7] Yaoyao Liu, Yuting Su, An-An Liu, Bernt Schiele, and Qianru Sun. Mnemonics training: Multi-class incremental learning without forgetting. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2020. 1

[8] David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning, 2022. 1

[9] Gabriel Moreira, Manuel Marques, João Paulo Costeira, and Alexander Hauptmann. Hyperbolic vs euclidean embeddings in few-shot learning: Two sides of the same coin. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2082–2090, 2024. 2, 4

[10] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017. 4

[11] Yuwen Tan and Xiang Xiang. Cross-domain few-shot incremental learning for point-cloud recognition. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2307–2316, 2024. 4

[12] 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. 4, 5

[13] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920, 2015. 5

[14] Yuwei Yang, Munawar Hayat, Zhao Jin, Chao Ren, and Yinjie Lei. Geometry and uncertainty-aware 3d point cloud class-incremental semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21759–21768, 2023. 4

[15] Ze Yang, Ruibo Li, Evan Ling, Chi Zhang, Yiming Wang, Dezhao Huang, Keng Teck Ma, Minhoe Hur, and Guosheng Lin. Label-guided knowledge distillation for continual semantic segmentation on 2d images and 3d point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18601–18612, 2023. 4

[16] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 16259–16268, 2021. 1, 5

[17] Na Zhao and Gim Hee Lee. Static-dynamic co-teaching for class-incremental 3d object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 3436–3445, 2022. 4

[18] 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. 4, 5