Supplementary Materials for Data Augmentation-free Unsupervised Learning for 3D Point Cloud Understanding

A Algorithm

Here, we provide a pseudo-code for SoftClu training loop in Algorithm 1.

Algorithm 1 Soft clustering (pseudocode).

Input: $\{\mathcal{P}\}$ a set of 3D point clouds and each point cloud has N points; K number of optimization steps.

Output: the backbone f_φ pretrained by using our algorithm.

for i in range(0, K) **do**
  $\mathcal{L}_{tot} = 0$
  for $\mathcal{P} \in \{\mathcal{P}\}$ **do**
    # compute class scores
    $\boldsymbol{S} = \text{softmax}\left(\phi_\alpha\left(f_\varphi\left(\mathcal{P}\right)\right)\right)$
    # compute prototypes
    $\boldsymbol{C}^E = \left\{\frac{1}{\sum_{i=1}^N s_{ij}} \sum_{i=1}^N s_{ij}\boldsymbol{p}_i\right\}_{j=1}^N$
    $\boldsymbol{C}^F = \left\{\frac{1}{\sum_{i=1}^N s_{ij}} \sum_{i=1}^N s_{ij}\boldsymbol{f}_i\right\}_{j=1}^N$
    # compute $\boldsymbol{D}$
    $\boldsymbol{D} = \left\{\lambda\|\boldsymbol{p}_i - \boldsymbol{c}_j^E\|_2^2 + (1-\lambda)\|\boldsymbol{f}_i - \boldsymbol{c}_j^F\|_2^2\right\}_{i,j}^{N,J}$
    # compute γ
    $\boldsymbol{\Gamma} = \text{SINKHORN}\left(\text{stopgrad}\left(\boldsymbol{D}\right), 1e-3, 20\right)$
    $\boldsymbol{\gamma} = N \cdot \boldsymbol{\Gamma}$
    # compute loss
    $\mathcal{L}_{tot}$ += $\mathcal{L}_{soft} + \eta\mathcal{L}_{orth}$
  end for
  # update backbone and segmentation head
  $f_\varphi, \phi_\alpha \leftarrow \text{optimize}\left(\frac{\mathcal{L}_{tot}}{N}\right)$
end for
return f_φ

For the Sinkhorn-Knopp algorithm, we provide a detailed pseudo-code in Algorithm 2.

B Downstream Tasks Setups

Classification. We use ModelNet40 [40] and ModelNet10 [40] benchmark classification datasets. ModelNet40 is composed of 12331 meshed models from 40 object categories, split into 9843 training meshes and 2468 testing meshes, where the points are sampled from CAD models. ModelNet10 dataset contains 4899 pre-aligned shapes from 10 categories with 3991 (80%) shapes for training and 908 (20%) shapes for testing. For SVM training, we randomly sample 1024 points for each shape as in [39].

Part segmentation. We follow [46] and use the ShapeNetPart [52] benchmark dataset that contains 16881 objects of 2048 points from 16 categories with 50 parts in total. We train the linear fully connected layer for 100 epochs by using the AdamW [27] optimizer with batch

Algorithm 2 Sinkhorn-Knopp algorithm (pseudocode).

Input: D distance matrix, $\varepsilon = 1e-3$ and *niters* iterations.

function SINKHORN($\boldsymbol{D}$, ε, *niters*)
  $\boldsymbol{\Gamma} = \exp(\boldsymbol{D}/\varepsilon)$
  $\boldsymbol{\Gamma}/ = \text{sum}(\boldsymbol{\Gamma})$
  $N, J = \boldsymbol{\Gamma}.\text{shape}$
  $\boldsymbol{u}, \boldsymbol{\mu}, \boldsymbol{\nu} = \text{zeros}(N), \text{ones}(N)/N, \text{ones}(J)/J$
  for _ in range(0, *niters*) **do**
    $\boldsymbol{u} = \text{sum}(\boldsymbol{\Gamma}, \text{dim=1})$
    $\boldsymbol{\Gamma} * = (\boldsymbol{\mu}/\boldsymbol{u}).\text{unsqueeze}(1)$
    $\boldsymbol{\Gamma} * = (\boldsymbol{\nu}/\text{sum}(\boldsymbol{\Gamma}, \text{dim=0})).\text{unsqueeze}(0)$
  end for
return $\boldsymbol{\Gamma}$
end function

size of 24, initial learning rate of 0.001, learning rate decay of 0.5 every 20 epoch. We report the overall accuracy (OA) and the mean class intersection over union (mIoU) to evaluate segmentation quality as in [46].

Semantic segmentation. We evaluate SoftClu features on semantic segmentation by using the S3DIS [3] benchmark dataset. S3DIS consists of 3D scans collected with the Matterport scanner in six indoor areas, featuring 271 rooms and 13 semantic classes. Following the pre-processing, post-processing and training settings as in [46], we split each room into $1m \times 1m$ blocks and use 4,096 points as the model input. For PointNet and DGCNN, we finetune the pre-trained model on areas 1,2,3,4,6 and test them on area 5. As in part segmentation, we report OA and mIoU to quantify the segmentation quality.

For SR-UNet backbone, we finetune the pre-trained model on areas 1-4 and 6 and test them on area 5. The fine-tuning experiments are trained with a batch size of 48 for a total of 10K iterations. The initial learning rate is 0.1, with polynomial decay with a power of 0.9. We set voxel size as 0.05 (5cm) and weight decay as 0.0001. We report mIoU and mAcc to evaluate segmentation quality as in [49].

C More Results

Part segmentation visualizations. Figure 4 shows examples of qualitative part segmentation results obtained with SoftClu after the fine-tuning on the downstream task compared to ground-truth annotations (GT). We can observe that our method provides consistent predictions throughout shapes, also in the case of complex shapes (chair and motorcycle).

Few-shot learning. Few-shot learning (FSL) aims to train a model that generalizes with limited data. We conduct FSL (N-way K-shot learning) for the classification task on ModelNet40 [40] and ModelNet10 [40] benchmark datasets, where the model is evaluated on N classes, and each class contains N samples. We use the same setting and train/test split as OcCo [46] and CrossPoint [2] and report the mean and standard deviation across 10 runs. Table 7 shows the FSL results on ModelNet40, where SoftClu outperforms prior works in all the FSL settings in the DGCNN backbone. Our method with PointNet backbone performs slightly poorly in 5-way 10-short and 10-way 20-short settings compared to results of CrossPoint with PointNet.

Table 7: Few-shot object classification results on ModelNet40. We report mean and standard error over 10 runs. Top results of each backbone is bold.

Encoder	Method	5-way		10-way	
		10-shot	20-shot	10-shot	20-shot
PointNet	Rand	52.0 ± 3.8	57.8 ± 4.9	46.6 ± 4.3	35.2 ± 4.8
	Jigsaw [39]	66.5 ± 2.5	69.2 ± 2.4	56.9 ± 2.5	66.5 ± 1.4
	cTree [11]	63.2 ± 3.4	68.9 ± 3.0	49.2 ± 1.9	50.1 ± 1.6
	OcCo [46]	89.7 ± 1.9	92.4 ± 1.6	83.9 ± 1.8	89.7 ± 1.5
	CrossPoint [2]	$\mathbf{90.9 \pm 4.8}$	93.5 ± 4.4	84.6 ± 4.7	$\mathbf{90.2 \pm 2.2}$
	SoftClu	90.6 ± 4.0	$\mathbf{93.8 \pm 3.2}$	$\mathbf{84.7 \pm 3.6}$	90.1 ± 4.5
DGCNN	Rand	31.6 ± 2.8	40.8 ± 4.6	19.9 ± 2.1	16.9 ± 1.5
	Jigsaw [39]	34.3 ± 1.3	42.2 ± 3.5	26.0 ± 2.4	29.9 ± 2.6
	cTree [11]	68.4 ± 3.4	71.6 ± 2.9	42.4 ± 2.7	43.0 ± 3.0
	OcCo [46]	90.6 ± 2.8	92.5 ± 1.9	82.9 ± 1.3	86.5 ± 2.2
	CrossPoint [2]	92.5 ± 3.0	94.9 ± 2.1	83.6 ± 5.3	87.9 ± 4.2
	SoftClu	$\mathbf{93.6 \pm 3.3}$	$\mathbf{97.3 \pm 2.0}$	$\mathbf{89.1 \pm 1.4}$	$\mathbf{93.2 \pm 3.4}$

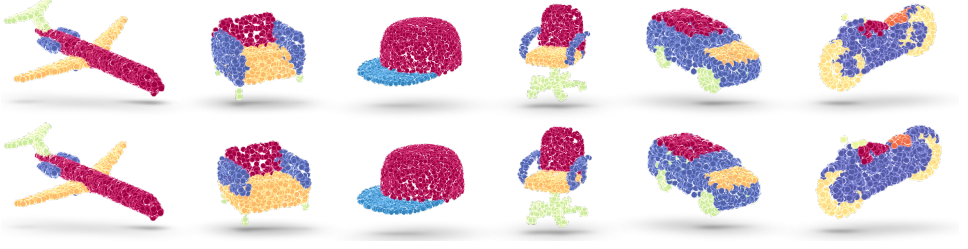

Figure 4: Part segmentation results on ShapeNetPart [52] of SoftClu using the DGCNN encoder (top row) compared to the ground-truth annotations (bottom row).

Batch size. Contrastive methods mine negative examples from the mini-batch can suffer from performance drops when their batch size is not sufficiently large [17]. Because SoftClu does not rely on negative examples, we expect it to be more robust to smaller batch sizes when compared to the contrastive approaches. We empirically show the effect of different batch sizes by comparing the performance of SoftClu with SimCLR [10]. We variate batch sizes from 8 to 48 during pre-training. Table 8 shows that SimCLR experiences degradation of performance when the batch size is 8, likely due to the small number of available negative samples. By contrast, SoftClu maintains a fairly stable performance throughout different batch size configurations.

Computation of soft-labels. We assess our strategy for soft-label assignment based on optimal transport (OT) by comparing it with a typical L2 distance-based approach on ModelNet40 and ModelNet10. Therefore, we assess SoftClu by using Γ computed with Eq. (7) and by using the L2 approach in [6]. Table 9 shows that OT achieves the best performance on all the datasets with both PointNet and DGCNN encoders. This is due to the equal partition constraint in Alg. 2 which prevents solutions from being assigned to the same cluster and affecting the performance.

Table 8: Ablation study results of SoftClu by using DGCNN on ModelNet10 with different batch sizes during pre-training.

Encoder	Method	8	16	24	32	40	48
PointNet	SimCLR	87.5	88.0	88.2	88.1	88.5	88.4
	SoftClu	89.9	89.8	90.2	90.3	90.1	89.9
DGCNN	SimCLR	88.6	89.3	89.4	89.7	89.7	90.1
	SoftClu	91.6	91.8	91.7	91.9	91.9	91.8

Table 9: Ablation study of SoftClu on ModelNet40 and ModelNet10 with soft-labels computed with our approach (OT) and with a typical distance-based assignment (L2).

Dataset	Encoder	Accuracy	
		L2	OT
ModelNet10	PointNet	91.5	**93.4**
	DGCNN	94.1	**94.8**
ModelNet40	PointNet	86.5	**90.3**
	DGCNN	90.4	**91.9**

SoftClu with Transformer backbone. Following [25] setups, we also provide the results with the recent Transformer backbone provided by [25]. As shown in Tab. 10, SoftClu also achieves competitive result.

Table 10: Classification results with a Transformer backbone on ModelNet40.

Encoder	SoftClu	PointViT-OcCo [46]	Point-BERT [54]	MaskPoint [25]
SoftClu	**93.8**	92.1	93.2	**93.8**