SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

November 2024, Vol. 67, Iss. 11, 212104:1–212104:15
https://doi.org/10.1007/s11432-023-4085-9

PointSmile: point self-supervised learning via curriculum mutual information

Xin LI[1,2], Mingqiang WEI[1,2*] & Songcan CHEN[1]

[1]*School of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing* 210016*, China;*
[2]*Shenzhen Research Institute, Nanjing University of Aeronautics and Astronautics, Shenzhen* 518063*, China*

Received 17 October 2023/Revised 9 January 2024/Accepted 23 February 2024/Published online 22 October 2024

Abstract Self-supervised learning is attracting significant attention from researchers in the point cloud processing field. However, due to the natural sparsity and irregularity of point clouds, effectively extracting discriminative and transferable features for efficient training on downstream tasks remains an unsolved challenge. Consequently, we propose PointSmile, a reconstruction-free self-supervised learning paradigm by maximizing curriculum mutual information (CMI) across the replicas of point cloud objects. From the perspective of how-and-what-to-learn, PointSmile is designed to imitate human curriculum learning, i.e., starting with easier topics in a curriculum and gradually progressing to learning more complex topics in the curriculum. To solve "how-to-learn", we introduce curriculum data augmentation (CDA) of point clouds. CDA encourages PointSmile to follow a learning path that starts from learning easy data samples and progresses to learning hard data samples, such that the latent space can be dynamically affected to create better embeddings. To solve "what-to-learn", we propose maximizing both feature- and class-wise CMI to better extract discriminative features of point clouds. Unlike most existing methods, PointSmile does not require a pretext task or cross-modal data to yield rich latent representations; additionally, it can be easily transferred to various backbones. We demonstrate the effectiveness and robustness of PointSmile in downstream tasks such as object classification and segmentation. The study results show that PointSmile outperforms existing self-supervised methods and compares favorably with popular fully supervised methods on various standard architectures. The code is available at https://github.com/theaalee/PointSmile.

Keywords PointSmile, self-supervised learning, curriculum mutual information, point cloud, representation learning

1 Introduction

There is an increasing demand to capture the real world by 3D sensing techniques for applications such as Metaverse and digital twins [1]. The captured scenes are often represented in a simple and flexible form, i.e., point cloud [2]. In recent years, researchers have made considerable efforts to employ deep learning to understand point clouds [3,4]. The first step toward understanding point clouds is extracting discriminative geometric features [5]; this is referred to as geometric representation learning (GRL). Ideally, when fed with sufficient annotated data, a GRL model to the extent that it can be combined with various neural networks, e.g., PointNet [6], PointNet++ [7], and DGCNN [8], to facilitate downstream tasks such as classification and segmentation [9]. However, real-world scenarios often lack labeled 3D scans, and human annotations of these scans are labor-intensive due to their irregular structures [10,11]. Although training on synthetic scans is promising to alleviate the shortage of labeled real-world data, such trained GRL models will inevitably suffer from domain shifts.

Self-supervised learning, as an unsupervised learning paradigm, can overcome the shortcomings of supervised models and is validated in 2D fields [12–14]. This explains the recent surge of interest in extracting powerful features using self-supervised learning [15–17] for 3D point clouds [18–21]. Most existing self-supervised learning methods follow the widely used encoder-decoder architecture, wherein the update of their encoders' parameters depends on the reconstruction of point cloud objects in the decoder. However, (i) reconstructing the 3D objects is not always effective due to the discrete nature of

* Corresponding author (email: mingqiang.wei@gmail.com)

point clouds; (ii) the unimodal losses, such as mean squared error and cross-entropy, are not feasible to recover various geometric details in the original data; (iii) these models are computationally intensive to formulate the complex relationships in the data; further, it is difficult to optimize these models.

Consider how teachers teach complex knowledge to freshmen. They may make a plan of curriculum learning, wherein the easy and intuitive pertinent knowledge is first presented, followed by difficult and abstract knowledge [22]. This curriculum design enables students to leverage previously learned knowledge to easily learn contents of increasing complexity as the class progresses, thereby reducing the perceived abstraction of new knowledge.

In this study, we attempt to use this strategy of curriculum learning for extracting discriminative features of unlabeled point clouds that are transferable to downstream tasks. To enable GRL to imitate the curriculum learning strategy designed for humans, we need to know "how to learn" and "what to learn". For how-to-learn, we introduce curriculum data augmentation (CDA) to construct two types of replicas for each unlabeled 3D object, namely, easy samples and hard samples. Thus, GRL can follow a learning path, starting from easy samples to gradually progressing to hard samples, such that the latent space is dynamically affected to create better embeddings. For what-to-learn, we propose to implement the maximization of curriculum mutual information (CMI) across the replicas of an unlabeled 3D object. This encourages better extraction of discriminative features of point clouds. Herein, CMI means the mutual information [23] in the curriculum. It can be maximized by CDA in both feature- and class-wise CMI. First, we maximize feature-wise CMI to enhance the similarity and denseness of features belonging to the same class in the feature space. Second, maximizing only feature-wise CMI may lead to different classes becoming further and further apart in the feature space. Thus, we formulate class-wise CMI to emulate the human perception that two similar objects but with different labels can have closer semantic features (e.g., a single sofa and a chair with a backrest). Note that the term class means the representation cluster divided in an unsupervised manner; that is, this division does not yield a real classification label. By maximizing feature- and class-wise CMI jointly, the inner- and intraclass representations can be distributed uniformly in the feature space.

Thus, we propose a reconstruction-free, self-supervised representation learning paradigm for point clouds via maximizing curriculum mutual information, dubbed as PointSmile. Instead of using extracted representations to reconstruct 3D objects, we enlarge the correlation between the representations obtained from different replicas of the same object. PointSmile is conceptually simple and easy to implement, and it learns useful geometric representations. Compared to existing methods, it does not require a complex pretext task or cross-modal data for functioning. With even just one linear layer, it provides classification results comparable to that of the supervised methods. Additionally, after fine-tuning, PointSmile yields even better performance than random initialization. We evaluate PointSmile on multiple downstream tasks. First, we perform shape classification in ModelNet40 [24], a synthetic object dataset. Second, we perform shape classification in ScanObjectNN [25], a real-world object dataset, to assess PointSmile's transferability. Third, we perform part segmentation and semantic segmentation to verify the ability of PointSmile to capture essential fine-grained features. In our experiments, we employ two widely used point cloud networks as feature extractors to assess the generalization of PointSmile. PointSmile outperforms most self-supervised methods for all downstream tasks in different datasets and backbone networks. It even outperforms some supervised methods. For both single-layer classification and fine-tuning segmentation, PointSmile is effective in helping different backbones to better learn point cloud representations.

The main contributions of our work are threefold:

• We propose PointSmile, a novel self-supervised learning paradigm via CMI. PointSmile possesses higher abstraction and maintains the invariance of geometric transformations. It is decoder-free; in other words, it avoids the complicated and unstable reconstruction of 3D objects and can be flexibly combined with mainstream neural networks, such as PointNet/PointNet++ and DGCNN.

• We propose a "how-and-what-to-learn" strategy to (i) increase the degree of difficulty in data augmentation step by step (this process is called CDA) and (ii) maximize the CMI. Implementing this strategy ensures that PointSmile effectively learns the discriminative features of point clouds without any annotation.

• We demonstrate PointSmile's efficacy through extensive evaluations in several downstream tasks, i.e., object classification and part segmentation. It not only achieves a better performance than its competitors but also demonstrates a better generalization capability. Moreover, we demonstrate PointSmile's superiority by comparing it to existing self-supervised learning methods.

2 Related work

Representation learning (RL), which aims to automatically discover the feature patterns in the data, is a significant aspect of deep learning. Although RL has achieved significant success in image processing, RL is still not well explored for processing point clouds, which is referred to as GRL. In the subsequent sections, we review current supervised and self-supervised GRL methods, followed by mutual information maximization.

2.1 Supervised representation learning

Many supervised GRL methods have emerged in recent years; we classify them into conversion-based, point-based, and graph-based methods.

Conversion-based methods convert point clouds into regular 2D grids [26] or 3D voxels [24, 27] or develop hand-crafted feature descriptors [28, 29], that facilitate the smooth operation of traditional 2D/3D CNNs. KD-Net [30] uses more efficient data structures and omits the calculation of empty voxels. PointGrid [31] integrates point and mesh representations by sampling a constant number of points in each embedded volumetric grid cell, thereby efficiently extracting geometric details by 3D CNNs. MVCNN [32] and RangeNet++ [33] generate multi-view features by rendering point clouds as 2D images for different downstream tasks. However, these conversion-based techniques are sensitive to noise and outliers, hard to capture fine-grained geometric details, too. Also, they often introduce excessive memory costs.

Graph-based methods regard points as the nodes of a graph and create edges based on their spatial/feature relationships [34]. KCNet [35] defines kernels based on Euclidean distances and geometric affinities of neighboring points. DGCNN [8] collects the nearest neighboring points in the feature space and uses EdgeConv to dynamically identify semantic cues for feature extraction. 3D-GCN [36] develops deformable kernels by extracting local 3D features across scales and focusing on shift and scale-invariant properties in point cloud analysis. AdaptConv [37] exploits adaptive kernels to replace the weight-sharing operation used in standard graph convolution and adaptively establishes diverse connections between different points in the local neighborhood from different semantic parts.

Point-based methods handle the irregularity of point clouds by directly manipulating them rather than introducing various intermediate representations. As the pioneer model, PointNet leverages multilayer perceptrons independently on each point to process points directly. During pooling operations, PointNet and DeepSets [38] abandon considerable local features that are indispensable to describing 3D shapes. PointNet++ [7] addresses this weakness of PointNet by exploiting a hierarchical structure to extract local features and introducing sampling and grouping operations. Similar ideas exist in PointCNN [39] and PointConv [40]. These methods first establish the topological relationship between points to extract local semantic features, subsequently aggregating the features by concatenating the features or improving the representation capability with recurrent neural network (RNN). PointMLP [41] does not consider a sophisticated local geometric extractor to be significant for performance and only uses residual feed-forward multilayer perceptrons (MLPs) without any other local feature exploration. Transformers in encoder and/or decoder configurations, have been successfully applied in natural language processing (NLP) and computer vision (CV). Numerous past studies also use Transformers for point cloud processing, such as Point Transformer [42] and PCT [43].

For supervised learning methods, there is an urgent need to solve the problem of how to efficiently obtain accurate labels.

2.2 Self-supervised representation learning

Generative methods in the field of point cloud representation learning aim to learn features through the process of encoding the point cloud into a feature or distribution and subsequently decoding it back to the original point cloud. These methods, such as those proposed in [19, 21, 44, 45], employ self-reconstruction as a means of feature learning. Recently, a wide variety of Transformer-based self-supervised methods have been proposed. For example, Point-Bert [46] predicts discrete tokens and Point-MAE [47] randomly masks patches of the input point clouds and reconstructs the missing points. An alternative to generative methods is using generative adversarial networks for generative modeling [48,49]. However, a disadvantage of these methods is that they require reconstruction or generation of 3D shapes. As mentioned earlier, reconstructing a 3D shape may be expensive or impossible.

Discriminative methods learn point cloud representations based on auxiliary hand-crafted prediction tasks. For this class of methods, it is not optimal to reconstruct the 3D shapes directly from the representation. Jigsaw3D [50] uses a 3D Jigsaw puzzle as a self-supervised learning task and trains an encoder for downstream tasks through contrastive techniques. PointContrast [51] proposes a pretext task, in which the representation of a single point cloud from different views should remain consistent and focus on high-level scene understanding tasks. Based on this task, PointContrast investigates a unified comparative paradigm framework for 3D representation learning. CrossPoint [52] combines information from both 3D and 2D modalities, focusing on powerful features shared between the different modalities. Although it is straightforward and efficient, it demands difficult-to-obtain point cloud rendering outcomes. To make contrastive learning tasks easier, Du et al. [53] used self-similar point cloud patches from a single point cloud as positive or negative examples and actively learned hard negative examples near positive samples for discriminative feature learning. STRL [54] is a direct extension of BYOL [13] to 3D point clouds; it learns representations through the interaction of online and target networks.

Additionally, some methods aim to integrate discriminative approaches with a self-supervised strategy called clustering. For example, Zhang et al. [55] combined contrastive learning with offline clustering to learn representations; their method has shown promising results. However, their approach typically requires training in two stages to achieve the desired performance.

These discriminative approaches mainly focus on the association of positive and negative samples or the design of pretext tasks. Therefore, the quality of these factors also influences the learned features. In contrast to the existing works that leverage generative and discriminative approaches, through PointSmile, we introduce a more straightforward way of using mutual information that does not require a decoder and does not require redundant pretext tasks but yields better representation.

2.3 Mutual information maximization

Information theory has long been applied as a tool for training deep learning networks using 2D images. IMSAT [56] uses data augmentation to impose invariance on discrete representations by maximizing mutual information between data and their representation. More recently, Oord et al. [57] proposed the framework of contrastive predicting coding (CPC), which combines the prediction of future observations with a probabilistic contrastive loss. CPC uses embeddings to capture maximal information about future samples. Deep InfoMAX [58] maximizes the mutual information between input data and learned high-level representations and has the advantage of performing orderless autoregression. However, it computes mutual information over continuous random variables, which requires complex estimators. In contrast, IIC [59] obtains mutual information of discrete variables with simple and exact computations. DGI [60] relies on maximizing mutual information between patch representations and corresponding high-level summaries of graphs. Inspired by the aforementioned methods, we extend mutual information from the 2D to 3D field and consider its maximization from local and global perspectives simultaneously.

3 Method

3.1 Overview

Imagine when we start a plan of curriculum learning, the easy and intuitive knowledge will be first taught, followed by the hard and abstract knowledge. Such wisdom of curriculum learning inspires our GRL paradigm. That is, by proposing a "how-and-what-to-learn" strategy, we design a self-supervised GRL framework, called PointSmile.

At the top level, we show in Figure 1 our PointSmile, a decoder-free model that learns discriminative features from 3D point clouds in a self-supervised manner. PointSmile consists of three main components, i.e., (i) CDA to construct easy and hard replicas of each 3D object, and to increasingly add the portion of hard samples during learning; (ii) a shared encoder E to learn geometric representations, and (iii) two CMI modules (i.e., feature-wise CMI and class-wise CMI) to maximize the mutual information between features extracted from independently-augmented pairs of each point cloud.

3.2 Curriculum augmented pairs from CDA

Motivation. We observe that a GRL model is prone to overfitting by purely given easy samples since these easy samples only involve simple geometric transformations. Therefore, we can mine hard samples

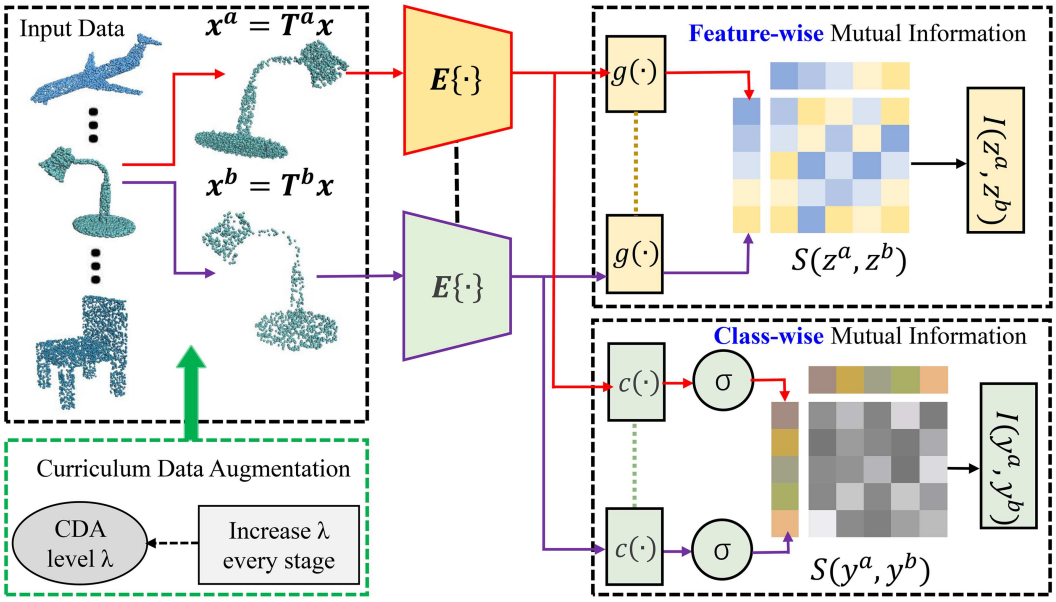

Figure 1 (Color online) Overview of PointSmile. PointSmile imitates the learning path employed by curriculum designers to facilitate students' learning. Consequently, PointSmile comprises three main parts: (i) CDA, which is used to construct easy and hard replicas of each 3D object and to gradually increase the portion of hard replicas during learning; (ii) a shared encoder E, which is used to learn geometric representations; (iii) two CMI modules, which are used to maximize the feature- and class-wise CMI jointly. x denotes an input point cloud batch and x^a, x^b denote two different replicas of x obtained from CDA.

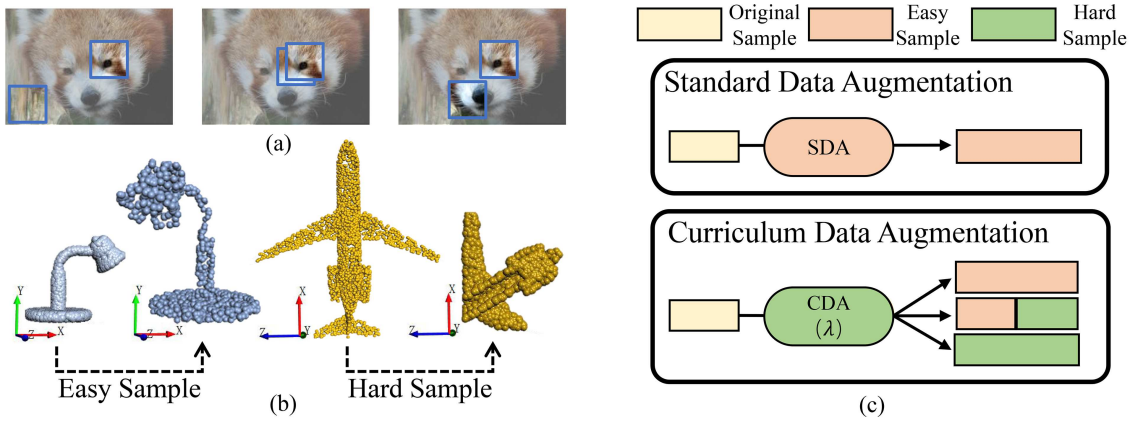

Figure 2 (Color online) Comparison of different ways to obtain augmented samples. (a) Comparison of three different regimes for augmented sample pairs in an image; (b) comparison of easy sample and hard sample; (c) comparison of SDA and CDA.

to enhance the GRL model. However, if feeding the GRL model with the combination of easy and hard samples directly, it might confuse the 'learner' and lead to harm of model training at the initial stage. It is reasonable to learn from easy samples and increasingly add hard samples.

CDA. We leverage CDA to update the difficulty of data augmentation increasingly. The principle of CDA is that the augmented data should resemble the original data. Under this principle, we control the CDA intensity, ensuring the augmented pairs to have lower CMI. This allows the model to gradually accumulate more sophisticated information during the learning process.

We construct two types of replicas for each point cloud. The first type of replicas is called easy samples and are generated by standard data augmentation (SDA) operations including random scale, translation, down-sampling, jitter and rotation. As shown in the left part of Figure 2(b), the easy sample contains geometric transformation that shares much information with the original object. Thus, it is easy to figure out that the two point clouds are from the same object. The second type that shares less information is called hard samples and is generated by using techniques including random X/Y axis flipping, shift, cuboid augmentation, and drop cuboid (certain parts (cubes) of a 3D object are randomly removed). For example, it is hard to identify the replica which remains only the middle-and-rear part by cutting the

original part of an airplane as shown in the right part of Figure 2(b). Easy samples make up easy sample pairs, and so do hard sample pairs. These hard sample pairs have lower CMI than the easy sample pairs and mainly preserve the downstream task-relevant high-level information. To clearly explain why the easy sample pair and the hard sample pair have different CMI, we first explain this phenomenon in 2D, and then change to 3D. We test two different samples (blue frames in Figure 2(a)) as a sample pair of the same image. In the right image of Figure 2(a), the sample pair belongs to different parts of the red panda but contains a few overlapped areas of the foreground. The pair is an optimal mix to provide the most independent information needed by the model. In the left image of Figure 2(a), the sample pair is from the foreground and background, respectively. It shares little information to identify the two regions that belong to the same image, even though they have small mutual information. On the contrary, the sample pair in the middle image shares too much information to learn a useful representation. Unlike 2D images, there is generally no background in a point cloud, and sample pairs in point clouds have larger CMI.

CDA divides the pre-training process into D steps and gradually increases the augmentation intensity λ and the number of hard samples N^c at each step. Suppose that a mini-batch x has N samples, both its λ_k and N_k^c at step k are formulated as

$$\lambda_k = \min\left(\mathrm{ini}\cdot \mathrm{inc}^{\lfloor \frac{k}{\mathrm{step}} \rfloor}, 1\right), \tag{1}$$

$$N_k^c = \lambda_k \cdot N, \tag{2}$$

where ini represents the initial percentage of hard samples, and inc represents an increasing exponential factor used to increase the percentage of the hard samples. step represents the length of iterations in each stage, which can be optionally set fixed. Besides, k denotes the k-th step of CDA and $0 < k \leqslant D$. Compared to the replicas obtained by only geometric transformation operations, our tactics typically have a higher degree of difficulty for an encoder to identify replicas and original data from the same class.

CDA differs from the traditional use of data augmentation: (i) it dynamically influences the latent space to create better embeddings; (ii) it enables the model to learn from the hard samples while keeping the difficulty level within the capability of the model during the whole training process. In general, the more hard samples in the batch, the more difficult that batch is for the model. Even if the difficulty differs due to the random cutting of the excised parts (such as the aircraft with the wings removed, which is harder to recognize but the aircraft with the tail removed is relatively easier), the difficulty can be averaged out because of the presence of multiple samples in the batch. Since CDA is performed in a self-supervised setting, the increased cost of augmentation can be shared across several tasks instead of just one task in a supervised setting.

3.3 CMI maximization

Motivation. We maximize feature-wise and class-wise CMI jointly, rather than a single type of CMI. From a local perspective, the objective of doing joint maximization is to find what is common between two replicas that share redundancy, such as different point clouds of the same object, explicitly encouraging the distillation of the common part while ignoring the rest. From a global perspective, joint CMI maximization helps to distribute features uniformly in the feature space, gaining strong fault tolerance and improving downstream tasks.

Preliminary. Given a set $x = \{x_1, \ldots, x_i, \ldots, x_N\}$ of N random samples, x^a and x^b denote random variables from two different replicas augmented by data augmentation T^a and T^b from CDA. There are multiple ways to compute mutual information [58, 61]. Herein we estimate CMI by maximizing the variational lower bounds of CMI. InfoNCE [57], as the most common lower bound, is formulated as

$$I(x^a, x^b) \geqslant I_{\mathrm{InfoNCE}}(x^a, x^b) \tag{3}$$

$$\stackrel{\mathrm{def}}{=} \mathbb{E}_{p(x_{1:N}^b)p(x_i^a|x_i^b)}\left[\log \frac{\mathrm{e}^{f(x_i^a, x_i^b)}}{\sum_{j=1}^N \mathrm{e}^{f(x_i^a, x_j^b)}}\right] + \log N \tag{4}$$

$$= -\mathscr{L}_{\mathrm{contrast}} + \log N, \tag{5}$$

where $x_{1:N}^b$ are N samples from p_{x^b}. x_i^a is a sample from p_{x^a} associated with x_i^b $(i = 1, \ldots, N)$. p_{x^a} and p_{x^b} are the distribution of x^a and x^b respectively generated by T^a and T^b from x. $p(x_i^a \mid x_i^b)$ means

the distribution of x_i^a for each x_i^b. (x_i^a, x_i^b) has a strong relationship called positive pair, and (x_i^a, x_j^b) $(j = 1, \ldots, N, i \neq j)$ are highly independent and called negative pairs. $f(\cdot, \cdot)$ is a similarity function. $\mathscr{L}_{\text{contrast}}$ is often known as the contrastive loss [12, 14].

Since $\log \frac{\mathrm{e}^{f(x_i^a, x_i^b)}}{\sum_{j=1}^N \mathrm{e}^{f(x_i^a, x_j^b)}} \leqslant 0$, the upper bound of $I(x^a, x^b)$ is $\log N$. We maximize $I(x^a, x^b)$ to reach the upper bound. Despite being biased, $I_{\text{InfoNCE}}(x^a, x^b)$ has much lower variance than other unbiased lower bounds of $I(x^a, x^b)$ to allow stable model training.

3.3.1 *Feature-wise CMI*

For feature-wise CMI, our goal is to learn an encoder, i.e., a mapping function E that preserves what is common between different replicas and discards instance-specific details. For example, a round table and a square table should share the same semantic information but they exhibit different instance-specific details. Specifically, each point cloud (i.e., replica) is mapped to a K-dimension assignment feature $f^a = E(x^a)$ and $f^b = E(x^b)$. The feature representations f^a and f^b after E will discard irrelevant information, i.e., data augmentation details. Please note that the encoder E is theoretically independent of any network.

As suggested by [12], we do not apply the instance loss to the feature space directly. Instead, we use a projection head g that consists of two-layer MLPs to map the feature to a subspace via $z^a = g(f^a)$ and $z^b = g(f^b)$. After the projection operation, our goal is changed to maximize $I(z^a, z^b)$. Thus, we utilize $f(z_i^a, z_j^b)$ instead of $f(x_i^a, x_j^b)$ (see (4)). We rewrite the contrastive loss in (4) as

$$\mathscr{L}_{\text{Fea}}^a = \mathbb{E}_{p(z_{1:N}^b) p(z_i^a | z_i^b)} \left[\log \frac{\mathrm{e}^{f(z_i^a, z_i^b)}}{\sum_{j=1}^N \mathrm{e}^{f(z_i^a, z_j^b)}} \right], \tag{6}$$

where Fea stands for the feature.

We specify the similarity function $f(\cdot, \cdot)$ as the scaled cosine similarity S_{ij}^f. The similarity matrix is formulated as

$$S_{ij}^f = \frac{(z_i^a)(z_j^b)^{\mathrm{T}}}{\|z_i^a\| \, \|z_j^b\|} \Big/ \tau_1, \tag{7}$$

where z_i^a and z_j^b are the i-th and j-th rows of z^a and z^b, respectively. And $\tau_1 > 0$ is the feature-wise temperature parameter, $\|\cdot\|$ is the ℓ_2-norm.

Then, we reformulate (6) as

$$\mathscr{L}_{\text{Fea}}^a = \mathbb{E}_{p(z_{1:N}^b) p(z_i^a | z_i^b)} \left[\log \frac{\mathrm{e}^{S_{ii}^f}}{\sum_{j=1}^N \mathrm{e}^{S_{ij}^f}} \right]. \tag{8}$$

Similarly, we can obtain $\mathscr{L}_{\text{Fea}}^b$ for any z_i^b. The feature loss can be formulated as

$$\mathscr{L}_{\text{Fea}} = \frac{1}{2N} \left(\mathscr{L}_{\text{Fea}}^a + \mathscr{L}_{\text{Fea}}^b \right). \tag{9}$$

3.3.2 *Class-wise CMI*

From the perspective of class, the two batches x^a and x^b should have the same sample distribution. Therefore, samples classified as the same class are viewed as positive. Similar to the projection head g, we use another two-layer MLP followed by softmax to form the class head c. Then, we project the feature into an M-dimensional space, where M is equal to or bigger than the number of classes in the pre-trained dataset. We obtain features y^a and y^b via $y^a = c(f^a)$ and $y^b = c(f^b)$. The output $[0, 1]^M$ is interpreted as the distribution of a discrete random variable y over M classes. y_i^a is the i-th column of y^a, i.e., the representation of class i under the first data augmentation. The second augmented representation of class i is y_i^b (the i-th column of y^b). Not only x_i^a and x_i^b should belong to the same class, but also y_i^a and y_i^b should have the same class distribution. For y_i^a, the class loss is

$$\mathscr{L}_{\text{Cls}}^a = \mathbb{E}_{p(y_{1:M}^b) p(y_i^a | y_i^b)} \left[\log \frac{\mathrm{e}^{S_{ii}^c}}{\sum_{j=1}^M \mathrm{e}^{S_{ij}^c}} \right], \tag{10}$$

which has one positive class and $M-1$ negative classes.

Similar to (7), the similarity matrix is given by

$$S_{ij}^{c}=\frac{(y_i^a)(y_j^b)^{\mathrm{T}}}{\|y_i^a\|\,\|y_j^b\|}\Big/\tau_2, \tag{11}$$

where y_i^a and y_i^a are the i-th and j-th columns of z^a and z^b, respectively. τ_2 is the class-wise temperature parameter. After going over every class, the class loss is determined as

$$\mathscr{L}_{\mathrm{Cls}}=\frac{1}{2M}\left(\mathscr{L}_{\mathrm{Cls}}^{a}+\mathscr{L}_{\mathrm{Cls}}^{b}\right). \tag{12}$$

3.3.3 *Total loss*

The overall objective function of PointSmile is defined as

$$\mathscr{L}_{\mathrm{overall}}=\mathscr{L}_{\mathrm{Fea}}+\mathscr{L}_{\mathrm{Cls}}+\beta\mathscr{L}_{\mathrm{CR}}, \tag{13}$$

where $\mathscr{L}_{\mathrm{CR}}$ is a class regularization loss [62] defined as

$$\mathscr{L}_{\mathrm{CR}}=\frac{1}{N}\sum_{M}^{i=0}\left(\sum_{N}^{j=0}y_i^j\right). \tag{14}$$

$\mathscr{L}_{\mathrm{CR}}$ can avoid local optimum solutions or degenerated solutions where all samples fall into the same class (e.g., y is one-hot for all samples). β is a coefficient.

4 Experiments

To learn features transferred effectively to downstream tasks is the primary goal of representation learning. We employ three downstream tasks including classification, part segmentation, and semantic segmentation, to assess the transferability of PointSmile and its competitors. We follow the same procedure as [18], and we train our model on ShapeNet [63]. For a fair comparison, we avoid CDA and CMI operations in downstream tasks. Ablation studies are also presented to analyze the effectiveness of both CDA and joint CMI.

4.1 Pre-training setup

To fairly compare our PointSmile with existing techniques, we adopt PointNet [6] and DGCNN [8] as our feature extractors. For the projection head, we employ a 2-layer MLP to yield a 256-dimensional feature vector projection in the invariant space, and the cluster head is as well. When using PointNet as the backbone, we employ the SGD optimizer [64] with an initial learning rate of 1×10^{-3}, the momentum of 0.9 and the weight decay of 1×10^{-6}. The learning rate scheduler is cosine annealing [65], and the model is trained end-to-end across 200 epochs. The training takes about 33 h on an RTX 3060 GPU. When using DGCNN as the backbone, we use the ADAM optimizer [66] with the weight decay of 1×10^{-4} and the initial learning rate of 1×10^{-3}. We take 124 h to train the model on the RTX 3090 GPU.

4.2 Downstream task setup

Object classification. Given an object represented by a set of points, object classification predicts the class to which the object belongs. We use two benchmarks: ModelNet40 [67] and ScanObjectNN [25]. We perform our synthetic object classification experiments on ModelNet40. ModelNet40 is composed of 12331 meshed models from 40 object categories, split into 9843 training meshes and 2468 testing meshes, on which the points are sampled. ScanObjectNN is a demanding and realistic 3D point cloud classification benchmark dataset made up of occluded objects captured from real indoor scenes which is more challenging. It contains 2880 objects (2304 for training and 576 for testing) from 15 categories. We use the same settings as [6, 8] for fine-tuning.

Specifically, for PointNet, we use the Adam optimizer with an initial learning rate of 1e−3, and the learning rate is decayed by 0.7 every 20 epochs with the minimum value of 1e−5. For DGCNN, we use the

Table 1 Training hype-parameters for pre-training, and downstream fine-tuning for classification and segmentation[a)].

	Pretraining	Classification	Part segmentation	Semantic segmentation
Config	ShapeNet	ModelNet/ScanObjectNN	ShapeNetPart	S3DIS
Optimizer	SGD/AdawW	AdamW/SGD	AdamW/SGD	AdamW/SGD
Learning rate	1e−3/1e−4	1e−3/1e−1	1e−3/1e−1	1e−3/1e−3
Weight decay	1e−6/1e−3	1e−5/1e−3	1e−4/1e−3	1e−4/1e−4
Learning rate scheduler	cosine/cosine	cosine/cosine	cosine/cosine	cosine/cosine
Training epochs	200/250	200/250	200/200	32/100
Batch size	16/16	32/32	24/16	16/16
Number of points	2048/2048	1024/1024	2048/2048	4096/4096

a) Each hyper-parameter has two parts: the former part (i.e., before '/') from PointNet, and the latter is from DGCNN.

SGD optimizer [64] with a momentum of 0.9 and a weight decay of 1e−4. The learning rate starts from 0.1 and then decays using cosine annealing [65] with the minimum value of 1e−3. We use dropout [68] in the fully connected layers before the softmax output layer. The dropout rate is set to 0.7 for PointNet and is set to 0.5 for DGCNN. For all the models, we train them for 200 epochs with a batch size of 32.

Part segmentation. Part segmentation is a challenging fine-grained 3D recognition task. The mission is to predict the part category label (e.g., car wheel, bag handle) of each point for a given object. For 3D object part segmentation, we choose ShapeNetPart [69] that contains 16881 objects of 2048 points from 16 categories with 50 parts in total. Following PointNet [6], we sample 2,048 points from each model. For PointNet, we use the Adam optimizer with an initial learning rate of 1e−3, and the learning rate is decayed by 0.5 every 20 epochs with the minimum value of 1e−5. For DGCNN, we use an SGD optimizer with a momentum of 0.9 and a weight decay of 1e−4. The learning rate starts from 0.1 and then decays using cosine annealing with the minimum value of 1e−3. We train the models for 250 epochs with a batch size of 16.

Semantic segmentation. Semantic segmentation predicts the semantic object category of each point. We evaluate PointSmile on semantic segmentation on Stanford Large Scale 3D Indoor Spaces (S3DIS) [70]. S3DIS consists of 3D scans collected by Matterport scanners from 6 indoor areas, containing 271 rooms and 13 semantic classes. We train the model for 32 and 100 epochs with a batch size of 16 for PointNet and DGCNN. For further details on the training configurations, please refer to Table 1.

4.3 Evaluation on downstream tasks

4.3.1 *Synthetic object classification*

After equally sampling each object with 1024 points, the coordinates (x, y, z) of sampled points are used as input for the classification task. Following each of the methods in the standard experimental process outline [50, 71], we train a simple linear SVM (support vector machine) classifier [72] using the extracted 3D point cloud features, while disabling the pre-trained point cloud feature extractor, to assess the utility of the feature representations in classification.

As shown in Table 2 [6–8, 15, 42, 46, 50, 52, 54, 71, 73–75], our method outperforms the state-of-the-art methods on ModelNet40, no matter whether PointNet or DGCNN is employed. Please note that CrossPoint [52] necessitates multi-modal data support, while our PointSmile is single-modal based. Still, PointSmile outperforms CrossPoint by a margin of 0.9% and 0.6% by using the backbone of PointNet and DGCNN, respectively. Furthermore, our PointSmile using a simple backbone like PointNet, prevails over many self-supervised methods with complex architectures, and it also surprisingly demonstrates superior performance than the original PointNet supervised learning benchmark. It is also shown that initializing our model with pre-trained weights also helps in achieving high accuracy in a fine-tuned manner. Even without a Transformer framework, it can achieve a nearly flat accuracy with Point-Bert. As illustrated in Table 3 [6, 8, 21, 42], our method requires fewer parameters than methods utilizing a decoder [21], and significantly fewer than methods employing a transformer structure.

4.3.2 *Real-world object classification*

To validate the effectiveness of our method on real-world point clouds, we perform classification experiments on ScanObjectNN [25]. We use a simple linear SVM for classification. Table 4 [50, 52, 54, 71] reports the linear evaluation results on ScanObjectNN. Compared with the state-of-the-art self-supervised meth-

Table 2 Comparison of the linear SVM classification on ModelNet40 [24][a)].

Self-sup.	Decoder-free.	PointNet Acc.	DGCNN Acc.	Sup.	Acc.
DeepCluster [73]	✓	86.3	90.4	PointNet [6]	89.2
Jigsaw3D [50]	×	87.3	90.6	PointNet++ [7]	90.7
Rotation3D [74]	✓	88.9	90.8	DGCNN [8]	92.9
Info3D [75]	✓	89.8	91.6	PointTransformer [42]	93.7
OcCo [71]	✓	88.7	90.2	Fine.	Acc.
STRL [54]	✓	88.3	90.0	Transformer-OcCo [71]	92.1
ParAE [15]	×	**90.3**	91.6	Point-Bert [46]	**93.2**
CrossPoint [52]	✓	89.1	91.2	PointSmile (PointNet)	90.7
PointSmile (ours)	✓	90.0	**91.8**	PointSmile (DGCNN)	93.0

a) 'Self-sup.', 'Decoder-free.', 'Sup.', and 'Fine.' denote the models pre-trained in a self-supervised, reconstruction-free, supervised, and fine-tuned manners, respectively. The best results are marked in bold.

Table 3 Pre-trained model parameters.

Encoder	Method	#P
PointNet	Sup. [6]	3.5M
	FoldingNet [21]	4.6M
	PointSmile (ours)	3.8M
DGCNN	Sup. [8]	1.8M
	PointSmile (ours)	2.1M
Transformer	Sup. [42]	22.1M
	Point-BERT	22.1M
	Point-MAE	22.1M

Table 4 Comparison of classification on ScanObjectNN[a)].

Encoder	Method	Acc.
PointNet	OcCo [71]	69.5
	Jigsaw [50]	55.2
	STRL [54]	74.2
	CrossPoint [52]	75.6
	PointSmile (ours)	**75.8**
DGCNN	OcCo [71]	69.5
	Jigsaw [50]	59.5
	STRL [54]	77.9
	CrossPoint [52]	81.7
	PointSmile (ours)	**82.8**

a) The best results are marked in bold.

ods, the accuracy for the DGCNN backbone is significantly improved by 1.1%, indicating that the feature representation learned by our PointSmile can span from synthetic data to realistic real-world settings.

4.3.3 *Part segmentation*

The part classifier is built by using the same architecture as those used in PointNet and DGCNN for part segmentation. Our self-supervised model is initially trained on ShapeNet and then fixed as a feature extractor. The evaluation metrics are OA (overall accuracy) and mIoU (mean intersection over union).

Table 5 [50, 52, 71] summarizes the evaluation results and the comparison of our PointSmile against several alternative methods on ShapeNetPart. Our method improves the part segmentation performance and exceeds the state-of-the-art baselines in terms of OA and mIoU. Furthermore, compared to other self-supervised approaches, our model performs better on the backbone of PointNet and DGCNN, demonstrating that more discriminative features can be learned by maximizing feature- and class-wise CMI.

Moreover, we observe from the results that our model can capture essential fine-grained features. Figure 3 demonstrates some of the qualitative part segmentation results showing that our method can achieve an excellent performance for part segmentation. We find that our method is able to better segment fine details than other methods in various categories. For example, the handle of a bag, the tail

Table 5 Comparison of the OA and mIoU of part segmentation results with self-supervised methods on ShapeNetPart [69][a)].

Encoder	Method	Metrics	
		OA (%)	mIoU (%)
PointNet	Jigsaw3D [50]	93.1	82.2
	OcCo [71]	93.4	83.4
	CrossPoint [52]	93.2	82.7
	PointSmile (ours)	**93.6**	**83.5**
DGCNN	Jigsaw3D [50]	92.7	84.3
	OcCo [71]	94.4	85.0
	CrossPoint [52]	94.4	85.3
	PointSmile (ours)	**94.4**	**85.4**

a) The best results are marked in bold.

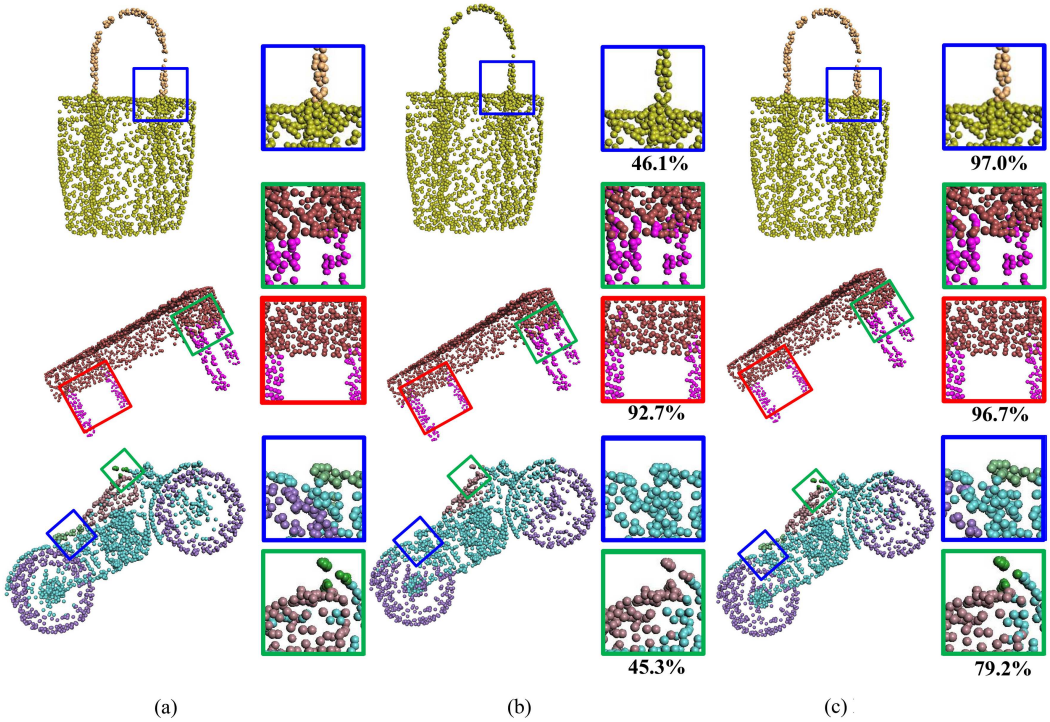

Figure 3 (Color online) Segmentation results on ShapePart of CrossPoint [52] and PointSmile (DGCNN as the encoder). Different colors represent different parts. (a) GT; (b) CrossPoint; (c) PointSmile (ours).

of an airplane, and the wheel of a motorcycle can be all segmented clearly from other parts.

4.3.4 *Semantic segmentation*

Semantic segmentation is a technique for associating points or voxels with semantic object labels, and it is also a fundamental research challenge in point cloud processing. As shown in Table 6 [50, 52, 71], with the PointNet as an encoder, PointSmile achieves 82.4% OA and 55.0% mIoU, outperforming the excellent baselines in terms of OA and mIoU. With the DGCNN as an encoder, PointSmile achieves 86.9% OA and 58.9% mIoU, outperforming CrossPoint, OcCo, and Jigsaw3D. Figure 4 shows the visualization results, where our results are very close to the ground-truths in the different scenes. Besides, PointSmile performs well when segmenting small objects in the scene, such as the clutter on the ceiling.

4.4 Ablations and analysis

4.4.1 *CDA*

Existing research in image processing has demonstrated that a powerful data augmentation technique is crucial for downstream tasks [12]. We also discover that a more sophisticated backbone requires more challenging beginning samples. To verify the significance of CDA, we test our model on ModelNet40 by

Table 6 Comparison of self-supervised methods by the OA and mIoU on the task of semantic segmentation on S3DIS [69][a)].

Encoder	Method	Metrics	
		OA (%)	mIoU (%)
PointNet	Jigsaw3D [50]	80.1	52.6
	OcCo [71]	82.0	54.9
	CrossPoint [52]	81.8	54.5
	PointSmile (ours)	**82.4**	**55.0**
DGCNN	Jigsaw3D [50]	84.1	55.6
	OcCo [71]	84.6	58.0
	CrossPoint [52]	**87.4**	58.4
	PointSmile (ours)	86.9	**58.9**

a) The best results are marked in bold.

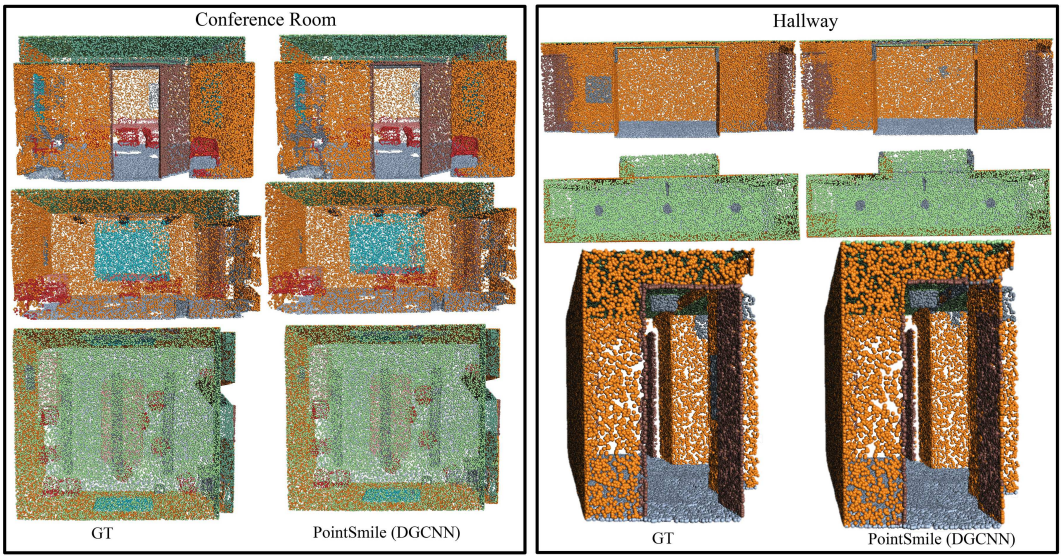

Figure 4 (Color online) Semantic segmentation results (DGCNN as the backbone) on S3DIS. Different colors indicate different objects. We show two different indoor scenes in two boxes. For each scene, we show three views (front, side, and back).

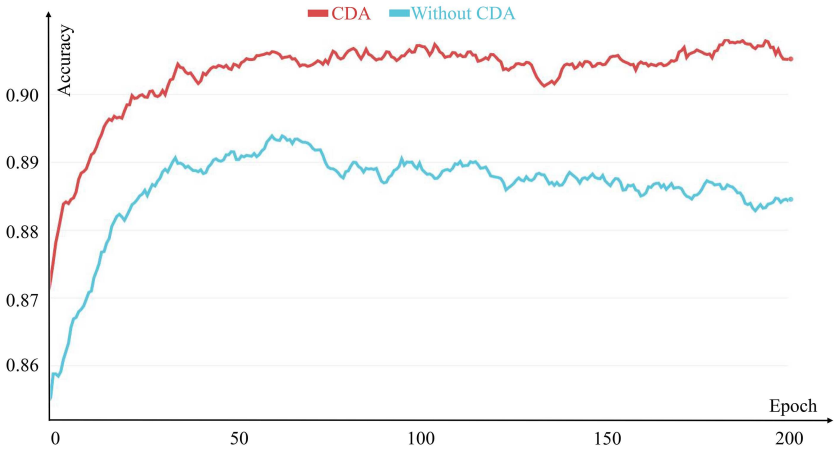

Figure 5 (Color online) Effect of CDA on the classification of the models in ModelNet40 [67] using PointNet [6] as the backbone.

omitting CDA using the backbone of PointNet [6]. Figure 5 shows that CDA enhances the performance of PoineSmile. In addition, with CDA, the accuracy can remain stable and consistently rise without decreasing at a later stage, according to the resulting curve.

Table 7 Accuracy of linear SVM classification using retrained embedding on ModelNet40 [67] for PointSmile[a)].

Encoder	L_{Fea}	L_{Cls}	Acc.
PointNet	✓		88.2
		✓	87.6
	✓	✓	**90.0**
DGCNN	✓		91.1
		✓	90.7
	✓	✓	**91.8**

a) The best results are marked in bold.

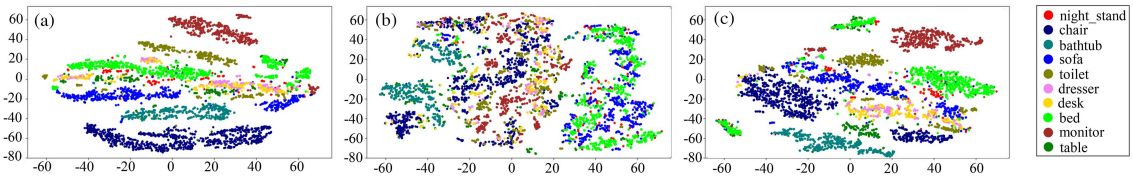

Figure 6 (Color online) t-SNE visualization of the features learned from ModelNet10 after training PointNet as the backbone. The features learned by maximizing joint CMI (c) provide better discrimination of classes than using only feature-wise CMI (a) or class-wise CMI (b).

4.4.2 *Joint CMI*

To examine the effectiveness of our design, we perform the ablation study on ModelNet40 by using (i) only the feature-wise CMI, (ii) only the class-wise CMI, and (iii) joint CMI, to understand their efficacy. The results are reported in Table 7. The feature-wise CMI model gets a classification accuracy of 91.1% for DGCNN and 88.2% for PointNet. Maximizing class-wise CMI can significantly improve the accuracy of the baseline model, which increases nearly by 0.8% and 0.7% for PointNet and DGCNN, respectively. However, utilizing only the class-wise CMI achieves the lowest performance, due to the misallocation of samples that belong to the same category but do not have similar features. Using both types of CMI produces a better outcome than using only one of them. This test reveals that the two types of CMI interact with and complement each other.

The t-SNE plot of the features trained by PointNet as the backbone on ModelNet10 [67] is shown in Figure 6. The left part of Figure 6 is only with feature-wise CMI, showing that points of the same color are almost on the same level, but points of different colors are also farther apart. The middle part is only with class-wise CMI, showing that the points of different colors are more evenly distributed, but there are no clear boundaries. This is detrimental to downstream tasks. The right part is the joint CMI, showing that points of the same color are clustered together and uniformly distributed over the feature space.

It is clear that even without labeled data, both feature- and class-wise settings yield some degrees of class discrimination. Besides, combining the two improves the discrimination boundaries in these classes, resulting in superior discriminative features.

5 Conclusion

We propose PointSmile, a novel self-supervised learning paradigm, for GRL on point clouds. PointSmile is designed to mimic how humans learn new yet difficult knowledge in a "how-and-what-to-learn" manner. To this end, we design a curriculum in which the "learner" follows a learning path, starting from easy samples to gradually progressing to hard samples. PointSmile leverages both fine-grained feature-level and coarse-grained class-level information by maximizing CMI. Thus, PointSmile can extract discriminative features from point clouds more effectively. The efficacy of PointSmile is evidenced through various downstream tasks, including object classification and segmentation, where it consistently outperforms existing methods. While our findings highlight the advantages of PointSmile, it is also crucial to recognize its limitations. Future research should focus on addressing key challenges, such as balancing different CMI types of self-supervised learning. This involves striking a balance between learned high-level semantic information and low-level detailed information. Additionally, we aim to enhance GRL by incorporating features from diverse modalities. Extending the applicability of PointSmile to different 3D domains, such as scenes, and broadening its utility to encompass other downstream tasks, such as detection, are pivotal

avenues for further exploration and refinement of PointSmile.

Acknowledgements This work was supported by National Natural Science Foundation of China (Grant Nos. T2322012, 62172218, 62032011), Shenzhen Science and Technology Program (Grant Nos. JCYJ20220818103401003, JCYJ20220530172403007), and Guangdong Basic and Applied Basic Research Foundation (Grant No. 2022A1515010170).

References

1 Zhu Z, Nan L, Xie H, et al. CSDN: cross-modal shape-transfer dual-refinement network for point cloud completion. IEEE Trans Visual Comput Graphic, 2024, 30: 3545–3563
2 Wei M, Wei Z, Zhou H, et al. AGConv: adaptive graph convolution on 3D point clouds. IEEE Trans Pattern Anal Mach Intell, 2023, 45: 9374–9392
3 Gu L, Yan X, Cui P, et al. PointSee: image enhances point cloud. IEEE Trans Visual Comput Graphic, 2024, 30: 6291–6308
4 Cui Y, Chen R, Chu W, et al. Deep learning for image and point cloud fusion in autonomous driving: a review. IEEE Trans Intell Transp Syst, 2022, 23: 722–739
5 Li X, Li R, Chen G, et al. A rotation-invariant framework for deep point cloud analysis. IEEE Trans Visual Comput Graphic, 2022, 28: 4503–4514
6 Qi C, Su H, Mo K, et al. PointNet: deep learning on point sets for 3D classification and segmentation. In: Proceedings of the Conference IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2017. 77–85
7 Qi C, Yi L, Su H, et al. PointNet++: deep hierarchical feature learning on point sets in a metric space. In: Proceedings of Neural Information Processing Systems, 2017. 5099–5108
8 Wang Y, Sun Y, Liu Z, et al. Dynamic graph CNN for learning on point clouds. ACM Trans Graph, 2019, 38: 1–12
9 Ma L, Li Y, Li J, et al. Multi-scale point-wise convolutional neural networks for 3D object segmentation from LiDAR point clouds in large-scale environments. IEEE Trans Intell Transp Syst, 2021, 22: 821–836
10 Chen H, Wei Z, Xu Y, et al. ImLoveNet: misaligned image-supported registration network for low-overlap point cloud pairs. In: Proceedings of ACM SIGGRAPH, 2022
11 Wang G, Wu J, Tian B, et al. CenterNet3D: an anchor free object detector for point cloud. IEEE Trans Intell Transp Syst, 2022, 23: 12953–12965
12 Chen T, Kornblith S, Norouzi M, et al. A simple framework for contrastive learning of visual representations. In: Proceedings of International Conference on Machine Learning, 2020. 1597–1607
13 Grill J B, Strub F, Altché F, et al. Bootstrap your own latent — a new approach to self-supervised learning. In: Proceedings of Neural Information Processing Systems, 2020. 21271–21284
14 Chen X, Fan H, Girshick R, et al. Improved baselines with momentum contrastive learning. 2020. ArXiv:2003.04297
15 Eckart B, Yuan W, Liu C, et al. Self-supervised learning on 3D point clouds by learning discrete generative models. In: Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021. 8248–8257
16 Zhou J, Wen X, Liu Y, et al. Self-supervised point cloud representation learning with occlusion auto-encoder. 2022. ArXiv:2203.14084
17 Fu K, Gao P, Zhang R, et al. Distillation with contrast is all you need for self-supervised point cloud representation learning. 2022. ArXiv:2202.04241
18 Achlioptas P, Diamanti O, Mitliagkas I, et al. Learning representations and generative models for 3D point clouds. In: Proceedings of the 35th International Conference on Machine Learning, 2018. 40–49
19 Han Z, Wang X, Liu Y S, et al. Multi-angle point cloud-VAE: unsupervised feature learning for 3D point clouds from multiple angles by joint self-reconstruction and half-to-half prediction. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019. 10441–10450
20 Han Z, Shang M, Liu Y, et al. View inter-prediction GAN: unsupervised representation learning for 3D shapes by learning global shape memories to support local view predictions. In: Proceedings of Association for the Advancement of Artificial Intelligence, 2019. 8376–8384
21 Yang Y, Feng C, Shen Y, et al. FoldingNet: point cloud auto-encoder via deep grid deformation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018. 206–215
22 Avrahami J, Kareev Y, Bogot Y, et al. Teaching by examples: implications for the process of category acquisition. Q J Exp Psychol Sect A, 1997, 50: 586–606
23 Kraskov A, Stögbauer H, Grassberger P. Estimating mutual information. Phys Rev E, 2004, 69: 066138
24 Wu Z, Song S, Khosla A, et al. A deep representation for volumetric shapes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015. 1912–1920
25 Uy M A, Pham Q H, Hua B S, et al. Revisiting point cloud classification: a new benchmark dataset and classification model on real-world data. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019. 1588–1597
26 Zhou H, Chen H, Feng Y, et al. Geometry and learning co-supported normal estimation for unstructured point cloud. In: Proceedings of Conference IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020. 13235–13244
27 Maturana D, Scherer S. VoxNet: a 3D convolutional neural network for real-time object recognition. In: Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, 2015. 922–928
28 Chen H, Wei M, Sun Y, et al. Multi-patch collaborative point cloud denoising via low-rank recovery with graph constraint. IEEE Trans Visual Comput Graphic, 2020, 26: 3255–3270
29 Li Z, Zhang Y, Feng Y, et al. NormalF-Net: normal filtering neural network for feature-preserving mesh denoising. Comput-Aided Des, 2020, 127: 102861
30 Klokov R, Lempitsky V. Escape from cells: deep Kd-networks for the recognition of 3D point cloud models. In: Proceedings of IEEE/CVF International Conference on Computer Vision, 2017. 863–872
31 Le T, Duan Y. PointGrid: a DEEP NEtwork for 3D shape understanding. In: Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018. 9204–9214
32 Su H, Maji S, Kalogerakis E, et al. Multi-view convolutional neural networks for 3D shape recognition. In: Proceedings of IEEE/CVF International Conference on Computer Vision, 2015. 945–953
33 Milioto A, Vizzo I, Behley J, et al. RangeNet++: fast and accurate lidar semantic segmentation. In: Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, 2019. 4213–4220
34 Kipf T N, Welling M. Semi-supervised classification with graph convolutional networks. 2016. ArXiv:1609.02907
35 Shen Y, Feng C, Yang Y, et al. Mining point cloud local structures by kernel correlation and graph pooling. In: Proceedings of Conference IEEE Conference on Computer Vision and Pattern Recognition, 2018. 4548–4557

36 Lin Z H, Huang S Y, Wang Y C. Convolution in the cloud: learning deformable kernels in 3D graph convolution networks for point cloud analysis. In: Proceedings of Conference IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020. 1800–1809
37 Zhou H, Feng Y, Fang M, et al. Adaptive graph convolution for point cloud analysis. In: Proceedings of IEEE/CVF International Conference on Computer Vision, 2021. 4945–4954
38 Zaheer M, Kottur S, Ravanbakhsh S, et al. Deep sets. In: Proceedings of Neural Information Processing Systems, 2017. 3391–3401
39 Li Y, Bu R, Sun M, et al. PointCNN: convolution on X-transformed points. In: Proceedings of Neural Information Processing Systems, 2018. 828–838
40 Wu W, Qi Z, Li F. PointConv: deep convolutional networks on 3D point clouds. In: Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019. 9621–9630
41 Ma X, Qin C, You H, et al. Rethinking network design and local geometry in point cloud: a simple residual MLP framework. 2022. ArXiv:2202.07123
42 Zhao H, Jiang L, Jia J, et al. Point transformer. In: Proceedings of IEEE/CVF International Conference on Computer Vision, 2021. 16239–16248
43 Guo M H, Cai J X, Liu Z N, et al. PCT: point cloud transformer. Comp Visual Media, 2021, 7: 187–199
44 Zhao Y, Birdal T, Deng H, et al. 3D point capsule networks. In: Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018
45 Hassani K, Haley M. Unsupervised multi-task feature learning on point clouds. In: Proceedings of IEEE/CVF International Conference on Computer Vision, 2019. 8159–8170
46 Yu X, Tang L, Rao Y, et al. Point-BERT: pre-training 3D point cloud transformers with masked point modeling. In: Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022. 19291–19300
47 Pang Y, Wang W, Tay F E, et al. Masked autoencoders for point cloud self-supervised learning, 2022. ArXiv:2203.06604
48 Achlioptas P, Diamanti O, Mitliagkas I, et al. Learning representations and generative models for 3D point clouds. In: Proceedings of International Conference on Machine Learning, 2018. 40–49
49 Han Z, Shang M, Liu Y S, et al. View inter-prediction GAN: unsupervised representation learning for 3D shapes by learning global shape memories to support local view predictions. In: Proceedings of Association for the Advancement of Artificial Intelligence, 2019. 8376–8384
50 Sauder J, Sievers B. Self-supervised deep learning on point clouds by reconstructing space. In: Proceedings of Neural Information Processing Systems, 2019. 12942–12952
51 Xie S, Liu S, Chen Z, et al. Attentional ShapeContextNet for point cloud recognition. In: Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018. 4606–4615
52 Afham M, Dissanayake I, Dissanayake D, et al. Crosspoint: self-supervised cross-modal contrastive learning for 3D point cloud understanding. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022. 9892–9902
53 Du B, Gao X, Hu W, et al. Self-contrastive learning with hard negative sampling for self-supervised point cloud learning. In: Proceedings of ACM Multimedia Conference, 2021. 3133–3142
54 Huang S, Xie Y, Zhu S C, et al. Spatio-temporal self-supervised representation learning for 3D point clouds. In: Proceedings of IEEE/CVF International Conference on Computer Vision, 2021. 6535–6545
55 Zhang L, Zhu Z. Unsupervised feature learning for point cloud understanding by contrasting and clustering using graph convolutional neural networks. In: Proceedings of International Conference on 3D Vision, 2019. 395–404
56 Hu W, Miyato T, Tokui S, et al. Learning discrete representations via information maximizing self-augmented training. In: Proceedings of International Conference on Machine Learning, 2017. 1558–1567
57 van den Oord A, Li Y, Vinyals O. Representation learning with contrastive predictive coding. 2018. ArXiv:1807.03748
58 Hjelm R D, Fedorov A, Lavoie-Marchildon S, et al. Learning deep representations by mutual information estimation and maximization. In: Proceedings of International Conference on Learning Representations, 2019
59 Ji X, Vedaldi A, Henriques J F. Invariant information clustering for unsupervised image classification and segmentation. In: Proceedings of IEEE/CVF International Conference on Computer Vision, 2019. 9864–9873
60 Velickovic P, Fedus W, Hamilton W L, et al. Deep graph infomax. 2019. ArXiv:1809.10341
61 Belghazi M, Baratin A, Rajeshwar S, et al. Mutual information neural estimation. In: Proceedings of International Conference on Machine Learning, 2018. 530–539
62 Meier L, van de Geer S, Bühlmann P. The group Lasso for logistic regression. J R Statist Soc Ser B-Statist Method, 2008, 70: 53–71
63 Chang A X, Funkhouser T A, Guibas L J, et al. ShapeNet: an information-rich 3D model repository. 2015. ArXiv:1512.03012
64 Cherry J. SGD: saccharomyces genome database. Nucleic Acids Res, 1998, 26: 73–79
65 Loshchilov I, Hutter F. SGDR: stochastic gradient descent with warm restarts. 2016. ArXiv:1608.03983
66 Kingma D, Adam J B. A method for stochastic optimization. 2014. ArXiv:1412.6980
67 Sharma A, Grau O, Fritz M. VConv-DAE: deep volumetric shape learning without object labels. In: Proceedings of European Conference on Computer Vision, 2016. 236–250
68 Srivastava N, Hinton G, Krizhevsky A, et al. Dropout: a simple way to prevent neural networks from overfitting. J Mach Learn Res, 2014, 15: 1929–1958
69 Yi L, Kim V G, Ceylan D, et al. A scalable active framework for region annotation in 3D shape collections. ACM Trans Graph, 2016, 35: 1–12
70 Armeni I, Sener O, Zamir A R, et al. 3D semantic parsing of large-scale indoor spaces. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2016. 1534–1543
71 Wang H, Liu Q, Yue X, et al. Unsupervised point cloud pre-training via occlusion completion. In: Proceedings of IEEE/CVF International Conference on Computer Vision, 2021. 9762–9772
72 Cortes C, Vapnik V. Support-vector networks. Machine Learn, 1995, 20: 273–297
73 Caron M, Bojanowski P, Joulin A, et al. Deep clustering for unsupervised learning of visual features. In: Proceedings of European Conference on Computer Vision, 2018. 132–149
74 Poursaeed O, Jiang T, Qiao H, et al. Self-supervised learning of point clouds via orientation estimation. In: Proceedings of International Conference on 3D Vision, 2020. 1018–1028
75 Sanghi A. Info3D: representation learning on 3D objects using mutual information maximization and contrastive learning. In: Proceedings of the European Conference on Computer Vision, 2020. 626–642