CLR-GAM: Contrastive Point Cloud Learning with Guided Augmentation and Feature Mapping

Anonymous authors
Paper under double-blind review

Abstract

Point cloud data plays an essential role in robotics and self-driving applications. Yet, it is time-consuming and nontrivial to annotate point cloud data while they enable learning discriminative 3D representations that empower downstream tasks, such as classification and segmentation. Recently, contrastive learning based frameworks show promising results for learning 3D representations in a self-supervised manner. However, existing contrastive learning methods cannot encode and associate structural features precisely and search the higher dimensional augmentation space efficiently. In this paper, we present CLR-GAM, a novel contrastive learning based framework with Guided Augmentation (GA) for efficient dynamic exploration strategy and Guided Feature Mapping (GFM) for similar structural feature association between augmented point clouds. We empirically demonstrate that the proposed approach achieves state-of-the-art performance on both simulated and real-world 3D point cloud datasets for three different downstream tasks, i.e., 3D point cloud classification, few-shot learning, and object part segmentation. The code and pretrained models are made available in the supplementary material.

1 Introduction

Scene understanding is of key importance in a wide range of applications including healthcare, medicine, entertainment, robotics, and human-machine interaction. Identifying surrounding objects in the scene and their interrelations are the core research problems for any scene understanding framework. Several 3D vision research problems (e.g., 3D point cloud classification (Qi et al., 2017a;b; Wang et al., 2019), detection (Misra et al., 2021), and segmentation (Qi et al., 2017b; Thomas et al., 2019; Wang et al., 2019)) have drawn much attention recently. However, obtaining 3D point cloud representations from the raw point clouds is challenging and often requires supervision, which causes high annotation costs. As a result, self-supervised learning for 3D point cloud representations has witnessed much progress and has the potential to improve sample efficiency and generalization for these scene understanding tasks. Existing works are mainly based on generative models (Achlioptas et al., 2018; Han et al., 2019a; Wu et al., 2016), reconstruction (Eckart et al., 2021; Han et al., 2019b; Li et al., 2018a; Yang et al., 2018; Zhao et al., 2019), pretext task (Wang et al., 2021; Poursaeed et al., 2020; Sauder & Sievers, 2019; Hassani & Haley, 2019; Sun et al., 2021; Yang et al., 2021; Rao et al., 2020), and contrastive learning (Zhang & Zhu, 2019; Sanghi, 2020; Xie et al., 2020; Huang et al., 2021; Liu et al., 2021; Zhang et al., 2021; Du et al., 2021). Much progress has been made in recent contrastive learning based methods. However, we observe the following two limitations.

Issue 1: With augmentations like cropping and nonrigid body transformation, the shape of an augmented object is entirely different from the original object, leading to ambiguity for contrastive learning. For instance, if we remove the back part of a "Chair" point cloud, the resulting point cloud could be similar in shape to a sample of the "Table" class, as shown in Figure 1.a. It poses a challenge for contrastive learning based methods because they do not access class labels for training.

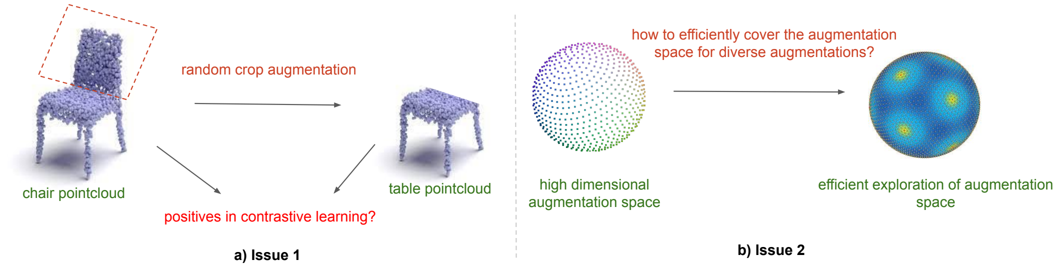


Figure 1: Motivation for CLR-GAM: a) motivation for Guided Feature Mapping, for better association b) motivation for Guided Augmentations, for better exploration of augmentation space

Issue 2: contrastive learning requires a variety of augmentations to learn discriminative 3D point cloud representations. However, searching over these high-dimensional augmentations is time-consuming and does not guarantee proper coverage with a dynamic limited number of samples.

In this work, we introduce two novel modules, i.e., guided feature mapping (GFM) and guided augmentation (GA), to overcome the above limitations. We introduce the GFM module to associate features of the same structure between two augmented samples for effective feature association under heavy shape deformation. The GA module is present to efficiently explore higher-dimensional augmentation spaces with dynamically limited samples for diverse coverage of the augmentation space. We conduct extensive experiments to validate the effectiveness of the proposed contrastive learning framework. Specifically, we benchmark three downstream tasks, i.e., classification, few-shot learning, and object part semantic segmentation. We obtain state-of-the-art performance on the three tasks, and extensive ablative studies are conducted to justify the designed choice.

Our main contributions: i) We propose Guided Augmentation (GA) and Feature Mapping (GFM) for learning discriminative 3D point cloud representations. ii) Our proposed approach achieves state-of-the-art performance on three downstream tasks, i.e., object classification, few-shot learning, and part segmentation. iii) Extensive ablatives studies are presented to justify our design choices.

2 Related Works

Contrastive Learning on Point Clouds. Following the recent success of contrastive self-supervised learning for images, recent works (Du et al., 2021; Huang et al., 2021; Liu et al., 2021; Sanghi, 2020; Xie et al., 2020; Zhang & Zhu, 2019; Zhang et al., 2021) explore contrastive learning for point cloud. PointContrast (Xie et al., 2020) applies contrastive loss for point-wise features generated from the neural network for a point cloud transformed using two random augmentions, to learn invariant features. Zhu et al. (2021) uses feature memory bank (He et al., 2020) for storing negatives and positives for hard sample mining. Huang et al. (2021) propose STRL that applies spatial augmentation for temporally correlated frames in a sequence point cloud dataset, and performs contrastive learning. Recently, Afham et al. (2022) propose CrossPoint to learn cross-modal (image and point cloud) representations via contrastive learning. All these methods rely on contrastive learning of encoded global features of point clouds, ignoring the structural deformations that lead to intra-class confusion. Recently, the authors of PointDisc (Liu et al., 2021) apply a point discrimination loss within an object for enforcing similarity in features for points within a local vicinity. PointDisc makes the geometric assumption of a fixed radius for obtaining positives from the encoded features of the same point cloud. In this work, we introduce the GFM to identify structurally similar features between two different augmentations of the same point cloud without any geometric assumptions. We empirically demonstrate the effectiveness of the proposed GFM for learning discriminative 3D representations for three different downstream tasks.

Guided Augmentation. Several guided augmentation approaches for image modality (Charalambous & Bharath, 2016; Hauberg et al., 2016; Rogez & Schmid, 2016; Peng et al., 2015;

Dixit et al., 2017) have shown to synthesize variable realistic samples for training. It is an important problem to generalize an algorithm to cover the unseen samples in the test data, which is expected to have wide variations of augmentation. In the context of human posture, Charalambous & Bharath (2016) generates synthetic videos for gait recognition and Rogez & Schmid (2016) augments images with 2D poses using 3D MoCAP data for pose estimation. For improving image detection, Peng et al. (2015); Su et al. (2015) renders 3D CAD models with variable texture, background, and pose for generating synthetic images. Hauberg et al. (2016) learn class specific transformations (diffeomorphism) from an external data, whereas another work (Miller et al., 2000) synthesizes new images using an iterative process. Since the existing works are for task specific and designed for supervised learning of image modality, they require class labels during training. AGA (Dixit et al., 2017) extends to the feature space to be class agnostic, but it requires a huge corpus of annotated datasets with class labels to pretrain. We cannot directly adapt those approaches to self-supervised point cloud learning approaches, so we find exploration strategies in reinforcement learning are relevant for unsupervised guided augmentation.

Exploration of High Dimensional Spaces. Efficient exploration in high dimensional space is a fundamental problem in reinforcement learning. Different strategies such as selecting new state including epsilon-greedy, selecting random states with epsilon probability (Mnih et al., 2015), upper confidence bounds (Auer, 2002), boltzmann exploration (Watkins, 1989; Sutton, 1990) using softmax over the utility of actions and thomson sampling (Agrawal & Goyal, 2012). The motivation or curiosity to explore new states is coined as intrinsic motivation (Oudeyer & Kaplan, 2008), which is adapted into Bellemare et al. (2016); Haber et al. (2018); Houthooft et al. (2016); Oh et al. (2015); Ostrovski et al. (2017); Pathak et al. (2017); Stadie et al. (2015) as intrinsic reward to quantify how different the new state is from already explored states. Some existing methods (Haber et al., 2018; Houthooft et al., 2016; Oh et al., 2015; Pathak et al., 2017; Stadie et al., 2015) use error in prediction as an intrinsic reward, while others use count-based techniques (Ostrovski et al., 2017; Bellemare et al., 2016). However, the computation of intrinsic reward using function approximation is slow to catch up and is not efficient enough for contrastive learning. In this work, we introduce a guided augmentation mechanism for efficient exploration of new states using a memory-based module motivated by Badia et al. (2020). Badia et al. construct an episodic memory-based intrinsic reward using k-nearest neighbors over the explored states to train the directed exploratory policies.

3 Methodology

3.1 Preliminaries and Notation

We denote a point cloud as P_i, which consists of unordered set of points $\mathbf{x}_{j=1:n}$ and $\mathbf{x}_j \in \mathbb{R}^3$, where the parameter n is number of points, and a point $\mathbf{x}_j$ is in 3D coordinate space. A point cloud P_i can be augmented by changing scale $\mathbf{a}_k^S \in \mathbb{R}^3$, translation $\mathbf{a}_k^T \in \mathbb{R}^3$, rotation $\mathbf{a}_k^R \in \mathbb{R}^3$, cropping $\mathbf{a}_k^C$, and jittering $\mathbf{a}_k^J$. The combined set of the above operations is denoted as $\mathbf{a}_k$, where $\mathbf{a}_k = [\mathbf{a}_k^C, \mathbf{a}_k^S, \mathbf{a}_k^R, \mathbf{a}_k^T, \mathbf{a}_k^J]$. Given a point cloud P_i, we apply the order defined in $\mathbf{a}_k$ to obtain an augmented point cloud P_i^k. In the remaining of this paper, we use i, j, k as the index of a point cloud $P_i \in \mathbb{R}^{n\times 3}$ and the corresponding encoded features $F_i \in \mathbb{R}^{n\times d}$, a point in point cloud $x_j = P_i(j) \in \mathbb{R}^{1\times 3}$ and a row of the encoded features $F_i(j) \in \mathbb{R}^{1\times d}$, and an augmentation operation $\mathbf{a}_k$, respectively. Note that the parameter n is the number of points in a point cloud.

3.2 Framework

The detailed architecture of the CLR-GAM framework, a contrastive learning based approach with the proposed GA and GFM modules, is depicted in Figure 2. We briefly introduce the overall contrastive learning algorithm in this section. First, a point cloud P_i is transformed into P_i^1 and P_i^2 by applying two augmentation operations $\mathbf{a}_1$ and $\mathbf{a}_2$. We utilize a Siamese architecture with shared weights for feature extraction. In this work, we utilize PointNet (a MLP based method) (Qi et al., 2017a) and DGCNN (a graph convolution based method) (Wang et al., 2019) to extract features that are invariant to the input order.

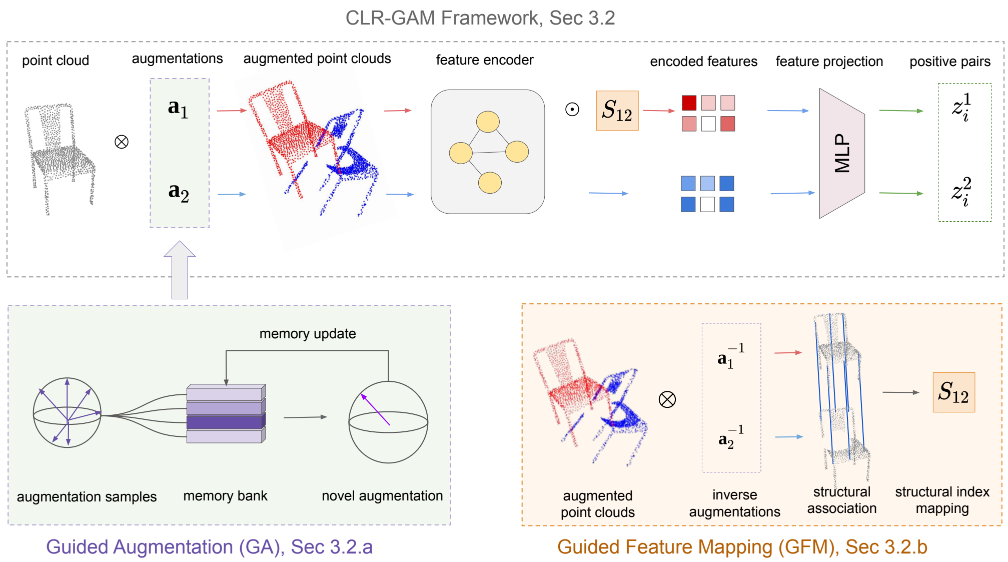


Figure 2: The proposed CLR-GAM framework with guided augmentation (GA) and guided feature mapping (GFM). $\otimes$ is the augmentation operator, $\odot$ is the indexing operator and S_{12} is the structural index mapping.

The augmented point clouds $P_i^1, P_i^2 \in \mathbb{R}^{n\times 3}$ are encoded into latent space $F_i^1, F_i^2 \in \mathbb{R}^{n\times d}$, respectively. The parameter n is the number of points, and d is the feature dimension. The augmented point clouds P_i^1, P_i^2 could contain different structures, while both point clouds originate from the same point cloud P_i. To ensure an effective feature association between F_i^1 and F_i^2, we introduce the Guided Feature Mapping (GFM) module to associate the features that belong to the same structure between two augmented point clouds. The feature F_i^1 is mapped to F_i^{12} to entail similar structural features when F_i^2 is considered. The features F_i^{12} and F_i^2 are pooled and projected into the projected latent space, resulting z_i^1 and z_i^2, respectively. We perform contrastive loss to enforce that the latent representation distance between the same point clouds (positives) features is smaller than the distance between the features from different point clouds (negatives) in a minibatch. In addition, contrastive learning heavily relies on the quality of augmentation. An efficient strategy for exploring the augmentation space is indispensable. We introduce a guided augmentation search to explore various augmentations efficiently, motivated by Badia et al. (2020).

a) Guided Augmentation: Augmentation is the key to the success of self-supervised contrastive learning. We hypothesize that if we can efficiently identify a wide range of informative augmentations, a discriminative representation can be learned. Existing approaches apply random sampling in augmentation spaces, which leads to ineffective augmentation and a high computational burden. Thus, we utilize a dynamic and efficient exploration strategy commonly used in reinforcement learning to mitigate the limitation.

The ranges of each dimension of rotation $\mathbf{a}^R$, translation $\mathbf{a}^T$, and scaling $\mathbf{a}^S$ are $[0, 2\pi)$ radians, $[-1, 1]$ meters, and $[0.5, 1]$, respectively. Since the jittering and cropping operations are point specific, we ignore them in guided augmentation for simplicity. Specifically, motivated by Badia et al. (2020), we utilize a memory bank M to save explored augmentation samples $\mathbf{a}_m$, where m is the index of a slot. The goal is to ensure that the new sample is different from the explored samples. It is worth noting that it is hard to obtain this behavior when just the average of L-norm distance is used to select novel augmentations. To start, we first randomly sample N augmentations $\hat{a}_{k=1:N}$ from the augmentation space $\mathbf{a}_k$. We compute the distance of a new sample $\hat{\mathbf{a}}_k$ from all the explored samples in the memory bank $\mathbf{a}_m$. The design is used to evaluate the novelty of a sample. A novel augmentation $\mathbf{a}_k^*$ is identified by using equation 1.

$$\mathbf{a}_k^* = \arg_{\hat{a}_k} \max \frac{1}{\sqrt{\sum_{m\in M} K(\mathbf{a}_m, \hat{\mathbf{a}}_k)} + c} \tag{1}$$

where $K(\mathbf{a}_m, \mathbf{a}_n) = \frac{\epsilon}{d(\mathbf{a}_m, \mathbf{a}_n)+\epsilon}$. The distance function d between two augmentations is the L_2-norm. The parameters c, ϵ are small values added for numerical stability. The memory bank is updated with the selected novel augmentation $\mathbf{a}_k^*$. The operation is applied twice on each point cloud P_i in an iteration to obtain two novel augmentations $\mathbf{a}_1, \mathbf{a}_2$. The two augmentations are applied to input point cloud P_i, as shown in Figure 2. Note that if the augmentations of rotation angles 2π and 0 are the same in the angular space, we utilize an angular distance measure, i.e., $d_R(\mathbf{a}_m^R, \mathbf{a}_n^R) = \sum(0.5 - \mid |\mathbf{a}_m^R - \mathbf{a}_n^R| - 0.5|)$, instead of using L_2 distance. To be consistent with different scales and ranges of augmentations, we normalize each augmentation to $[0, 1]$ before computing the total distance d as shown in equation 2, where α_R, α_T,and α_S are the weights for the three distances.

$$d(\mathbf{a}_m, \mathbf{a}_n) = \alpha_R d_R(\mathbf{a}_m^R, \mathbf{a}_n^R) + \alpha_T ||\mathbf{a}_m^T - \mathbf{a}_n^T||_2 + \alpha_S ||\mathbf{a}_m^S - \mathbf{a}_n^S||_2 \tag{2}$$

b) Guided Feature Mapping: To learn discriminative point cloud representations, it is crucial to project features with similar structural characteristics for contrastive learning. Existing methods may fail to identify the structural similarity between the two augmented point clouds because certain augmentations (e.g., cropping, scaling) could lead to heavy deformations of an augmented point cloud with a completely different shape from the original class and similar to a different class. Based on our observation, when both the augmentations $\mathbf{a}_1, \mathbf{a}_2$ contains crop operations, this results in very limited structural similarity between the augmented point clouds. So we exclude the crop augmentation $\mathbf{a}_1^C$ from the augmentation $\mathbf{a}_1$. In $\mathbf{a}_2$, it uses all the augmentations, i.e., rotation, translation, scaling, cropping, and jittering. Note that $\mathbf{a}_k^R, \mathbf{a}_k^T, \mathbf{a}_k^S$ are invertible operations as they are applied on the whole point cloud. The operation $\mathbf{a}_k^J$ is a point-specific operation and invertible. On the other hand, the cropping operation $\mathbf{a}_k^C$ is not invertible as the information is lost. An invertible augmentation operation can be written as $P_i = (\mathbf{a}_1)^{-1} \otimes P_i^1$, where P_i^1 is an augmented point cloud, P_i is the original point cloud, and $\otimes$ denotes an augmentation operator. The equation holds because the augmentation $\mathbf{a}_1$ does not contain a cropping operation. Whereas the augmentation inverted point cloud of P_i^2 results in $P_i^C = (\mathbf{a}_2)^{-1} \otimes P_i^2$, a cropped point cloud. The crop operation is ignored in the inverse operation with $\mathbf{a}_2$, as it is not invertible. The order of points and their structures cannot be directly associated between these two augmented point clouds even with the same number of points. The closest point association mapping S_{12} between points of inverted point clouds of P_i^1 and P_i^2 is calculated based on equation 3. The structural index mapping S_{12} retains only the indices of the closest points of P_i^1 to P_i^2, for every point in P_i^2 with index j.

$$S(j)_{12} = \arg_q \min ||P_i^C(j) - P_i(q)||_2 \tag{3}$$

The operators $P_i(\cdot)$ and $F_i(\cdot)$ denote indexing operation to point cloud and feature set, respectively. The guided mapped feature F_i^{12} is obtained according to $F_i^{12} = F_i^1(S_{12})$. The feature F_i^{12} is projected to z_i^1 using the feature projection module after pooling. Feature projection module is an MLP to reduce the dimensionality of the features. Similarly, F_i^2 is projected to z_i^2. The contrastive loss (Chen et al., 2020) is utilized to compute the similarity between positives (z_i^1, z_i^2) and negatives from the minibatch. We do not store negatives over multiple iterations in a memory bank for comparability with other techniques (Afham et al., 2022), which is commonly done for improving the performance (He et al., 2020). The loss can be found in equation 4. The similarity measure is the cosine distance between two features, $\text{sim}(z_1, z_2) = (z_1^T z_2)/(||z_1||||z_2||)$. Given a minibatch, the final contrastive loss is $L_c = \frac{1}{2B}\sum_{b=1}^{B}(L_{\mathbf{1},\mathbf{2}}^b + L_{\mathbf{2},\mathbf{1}}^b)$. The parameter τ is temperature 0.5, b is the index of the feature in the minibatch of total size B.

$$L_{\mathbf{1},\mathbf{2}}^i = -log \frac{\exp(\text{sim}(z_i^1, z_i^2)/\tau)}{\sum_{b=1, b\neq i}^{B} \exp(\text{sim}(z_i^1, z_b^1)/\tau) + \sum_{b=1}^{B} \exp(\text{sim}(z_i^1, z_b^2)/\tau)} \tag{4}$$

4 Experiments

4.1 Quantitative Results

a) 3D Object Classification: For this task, we utilize the ModelNet-40 (synthetic) and ScanObjectNN (real-world) datasets. The ModelNet-40 dataset consists of a wide range

Approach	Method	ModelNet-40	
point cloud	3D-GAN (Wu et al., 2016)	83.3	
	Latent-GAN (Achlioptas et al., 2018)	85.7	
	SO-Net (Li et al., 2018a)	87.3	
	FoldingNet (Yang et al., 2018)	88.4	
	MRTNet (Gadelha et al., 2018)	86.4	
	3D-PCapsNet (Zhao et al., 2019)	88.9	
	ClusterNet (Zhang & Zhu, 2019)	86.8	
	VIP-GAN (Han et al., 2019a)	90.2	
+ Image Modality	DepthContrast (Zhang et al., 2021)	85.4	
		PNet	DGCNN
point cloud	Multi-Task (Hassani & Haley, 2019)	-	89.1
	self-contrast Du et al. (2021)	-	89.6
	Jigsaw (Sauder & Sievers, 2019)	87.3	90.6
	STRL (Huang et al., 2021)	88.3	90.9
	Rotation (Poursaeed et al., 2020)	88.6	90.8
	OcCo (Wang et al., 2021)	88.7	89.2
	CLR-GAM (ours)	**88.9**	**91.3**
+ Image Modality	CrossPoint (Afham et al., 2022)	<u>89.1</u>	91.2

Table 1: We pretrained using the proposed contrastive self-supervised learning framework on ShapeNet. We evaluate on the test split of ModelNet-40 by fitting a linear SVM classifier. The reported results are the overall accuracy. Upper subtable uses custom backbone and training strategies.

of 3D objects' CAD models. The dataset contains 12,331 objects that are categorized into 40 classes. We use 9,843 for training and 2,468 for testing. The ScanObjectNN dataset is challenging because data is collected in cluttered environments, so objects could be partially observable due to occlusions. It consists of 15 classes totaling 2,880 objects (2,304 for training and 576 for testing).

We follow the same evaluation strategy as in the existing works (Huang et al., 2021; Afham et al., 2022; Wang et al., 2021). Specifically, we freeze the pretrained point cloud feature extractor pretrained on the ShapeNet dataset. We randomly sample 1024 points from each object for testing classification accuracy on ModelNet-40 and ScanObjectNN. We fit a linear SVM (Cortes & Vapnik, 1995) on the extracted features. The results on the testing set of ModelNet-40 and ScanObjectNN can be found in Table 1 and Table 2, respectively. Additionally, we also conduct experiments using two different backbones, i.e., PNet (Qi et al., 2017a) and DGCNN (Wang et al., 2019), on the two datasets. We demonstrate state-of-the-art performance on the ModelNet-40 dataset using both backbone architectures compared to point cloud pretrained approaches in the bottom sub-table, as shown in Table 1. With the DGCNN backbone, the proposed approach performs better than CrossPoint and DepthContrast. It is worth noting that both methods utilize extra image modality for pretraining, while the proposed contrastive self-supervised learning framework only uses point cloud. Compared to previous SOTA on a single modality (OcCo), the accuracy is improved by 2.35% (with DGCNN).

The results conducted on ScanObjectNN further justify the effectiveness of the proposed framework, as shown in Table 2. State-of-the-art performance is present compared to both point cloud and multimodal pretrained approaches using both backbone architectures. Noticeably, compared to previous SOTA on a single modality (OcCo), the accuracy is improved by 4.8% (with DGCNN). In addition to satisfactory results, we empirically demonstrate that the proposed approach has better generalization capability in a real-world setting under severe occlusions than other methods.

b) Few Shot Object Classification: Few Shot Learning (FSL) is a learning paradigm that aims to train a model that generalizes with limited data. In this experiment, we conduct experiments on N-way K-shot learning, which means that a model is trained on N classes and K samples in each class. The test/query set for each of the N classes consists of 20 unseen samples for all these experiments. We use ModelNet-40 and ScanObjectNN for these experiments. The same pretrained model is used for both classification and FSL tasks with respective backbones. Similar to the classification task, we fit a linear SVM classifier for testing the FSL task. A similar protocol is used in earlier works (Afham et al., 2022; Sharma & Kaul, 2020). We report the results in Tables 3, 4. As there is no a standard benchmark

Method	PNet	DGCNN
Jigsaw (Sauder & Sievers, 2019)	55.2	59.5
OcCo (Wang et al., 2021)	69.5	78.3
STRL (Huang et al., 2021)	74.2	77.9
CLR-GAM (ours)	**75.7**	**82.1**
CrossPoint (Afham et al., 2022)	75.6	81.7

Table 2: 3D Object classification on ScanObjectNN. We pretrained using the proposed contrastive self-supervised learning framework on ShapeNet. We evaluate on test split of ScanObjectNN by fitting a linear SVM classifier. The reported results are the overall accuracy on the test split.

test set, we follow the setting used in Afham et al. (2022); Sharma & Kaul (2020); Wang et al. (2021). Specifically, we report mean and standard deviation over 10 runs.

As shown in Table 3, we observe that the CLR-GAM with DGCNN achieves SOTA compared to all other approaches in the challenging 5-way setting. In the 10-way setting, CLR-GAM performs on-par with CrossPoint (multimodal pretrained) and Occo (single modal pretrained). The results show the same trend as in 1. The few-shot object classification results

	5-way				10-way			
Method	10-shot		20-shot		10-shot		20-shot	
FoldingNet (Yang et al., 2018)	33.4±4.1		35.8±5.8		18.6±1.8		15.4±2.2	
Latent GAN (Achlioptas et al., 2018)	41.6±5.3		46.2±6.2		32.9±2.9		25.5±3.2	
3D-PointCapsNet (Zhao et al., 2019)	42.3±5.5		53.0±5.9		38.0±4.5		27.2±4.7	
PointNet++ (Qi et al., 2017b)	38.5±4.4		42.4±4.5		23.1±2.2		18.8±1.7	
PointCNN (Li et al., 2018b)	65.4±2.8		68.6±2.2		46.6±1.5		50.0±2.3	
RSCNN (Liu et al., 2019)	65.4±8.9		68.6±7.0		46.6±4.8		50.0±7.2	
	PNet	DGCNN	PNet	DGCNN	PNet	DGCNN	PNet	DGCNN
Rand	52.0±3.8	31.6±2.8	57.8±4.9	40.8±4.6	46.6±4.3	19.9±2.1	35.2±4.8	16.9±1.5
Jigsaw (Sauder & Sievers, 2019)	66.5±2.5	34.3±1.3	69.2±2.4	42.2±3.5	56.9±2.5	26.0±2.4	66.5±1.4	29.9±2.6
cTree (Sharma & Kaul, 2020)	63.2±3.4	60.0±2.8	68.9±3.0	65.7±2.6	49.2±1.9	48.5±1.8	50.1±1.6	53.0±1.3
OcCo (Wang et al., 2021)	89.7±1.9	90.6±2.8	92.4±1.6	92.5±1.9	83.9±1.8	82.9±1.3	**89.7±1.5**	86.5±2.2
CLR-GAM (ours)	**91.8±2.6**	**93.7±1.2**	**94.8±2.4**	**96.0±2.6**	**84.6±2.2**	**87.9±2.7**	89.1±2.0	**91.1±1.9**
CrossPoint (Afham et al., 2022)	90.9±4.8	92.5±3.0	93.5±4.4	94.9±2.1	84.6±4.7	83.6±5.3	90.2±2.2	87.9±4.2

Table 3: Few shot object classification on ModelNet-40. A linear SVM is fit on the training set of ModelNet-40 using the pretrained model learned from ShapeNet. Compared with existing methods, the proposed CLR-GAM achieves state-of-the-art performance under different few shot settings. The results are the overall accuracy.

on ScanObjectNN is reported in Table 4. CLR-GAM with DGCNN and PointNet performs SOTA compared to both point cloud and multimodal pretrained approaches. Specifically, on ScanNet we show a large margin improvement (more than 11%) using DGCNN on all sets, and more than 8% improvement with PNET (5 way-20 shot, 10 way-10 shot, 10 way-20 shot). There is a 24% improvement with both DGCNN and PNET backbones in 10 way-20shot. The results further testify that CLR-GAM learns discriminative 3D point cloud representations, and the representations can generalize to challenging real-world setting.

	5-way				10-way			
Method	10-shot		20-shot		10-shot		20-shot	
	PNet	DGCNN	PNet	DGCNN	PNet	DGCNN	PNet	DGCNN
Rand	57.6±2.5	62.0±5.6	61.4±2.4	67.8±5.1	41.3±1.3	37.8±4.3	43.8±1.9	41.8±2.4
Jigsaw (Sauder & Sievers, 2019)	58.6±1.9	65.2±3.8	67.6±2.1	72.2±2.7	53.6±1.7	45.6±3.1	48.1±1.9	48.2±2.8
cTree (Sharma & Kaul, 2020)	59.6±2.3	68.4±3.4	61.4±1.4	71.6±2.9	53.0±1.9	42.4±2.7	50.9±2.1	43.0±3.0
OcCo (Wang et al., 2021)	70.4±3.3	72.4±1.4	72.2±3.0	77.2±1.4	54.8±1.3	57.0±1.3	61.8±1.2	61.6±1.2
CLR-GAM (ours)	**71.8±2.8**	**80.6±1.9**	**78.4±3.2**	**86.3±2.3**	**63.8±2.6**	**67.2±1.5**	**69.4±2.8**	**76.4±2.7**
CrossPoint (Afham et al., 2022)	68.2±1.8	74.8±1.5	73.3±2.9	79.0±1.2	58.7±1.8	62.9±1.7	64.6±1.2	73.9±2.2

Table 4: Few shot object classification on ScanObjectNN. A linear SVM is fit on the training set of ModelNet-40 using the pretrained model learned from ShapeNet. Compared with existing methods, the proposed CLR-GAM outperforms state-of-the-art method Wang et al. (2021) with a large margin. Reported results are the overall accuracy.

c) 3D Object Part Segmentation: ShapeNet-part dataset (Yi et al., 2016), which contains 50 different parts from 16 distinct object categories with a total of 16,881 3D objects, is used for 3D part object segmentation. We use the same pretrained model for both classification and FSL tasks with respective backbones. To be consistent with the evaluation for part segmentation, we finetune the pretrained model using 2048 points sampled

Category	Method	Mean IOU
Supervised	PointNet (Qi et al., 2017a)	83.7
	PointNet++ (Qi et al., 2017b)	85.1
	DGCNN (Wang et al., 2019)	85.1
Self-Supervised	Self-Contrast Du et al. (2021)	82.3
	Jigsaw (Sauder & Sievers, 2019)	85.3
	OcCo (Wang et al., 2021)	85.0
	PointContrast (Xie et al., 2020)	85.1
	PointDisc (Liu et al., 2021)	85.3
	CLR-GAM (ours)	**85.5**
+ Image Modality	CrossPoint (Afham et al., 2022)	85.5

Table 5: We report the mean IOU results for 3D object part segmentation on the ShapeNet-part dataset. Supervised methods are trained with randomly initialized weights, whereas self-supervised methods are initialized with pretrained weights learned from ShapeNet.

from point clouds. We observe that the performance of CLR-GAM is better than the other point cloud contrastive learning-based approaches and on-par with CrossPoint (multimodal pretrained). The reported results in Table 5 are average of intersection over union (IOU) computed for each part.

4.2 Qualitative Results

We visualize feature representations (learned from the proposed CLR-GAM) of each point/node in an unseen object's point cloud selected from test sets of ShapeNet and ModelNet-40 in Figure 3. We compute the cosine distance between the feature of a randomly selected point (colored in red) to other points' features in the same point cloud. The color scale is Yellow-Green-Blue. The closest feature in the feature space is yellow, and the farthest is blue.

Our approach learns similar representations for the whole planar region for simple planar structures such as stool (a) and table (b). Moreover, in the case of a chair (f), a complicated planar structure, the proposed model can learn similar features for the back part of a seat. For monitor (k), the plane is assigned with closer/similar features, whereas the features at the corners (structural irregularities) are dissimilar to the center. Similar observation can be found in the case of a knife (e), i.e., the handle and sharp edge have different features. For a curved object like a bathtub (g), the whole tub has similar features except for the legs. Similarly, for the cone (h), the whole curved region has similar features except for the edges. In the case of lamp (i), the curved stand has similar features, separating the stem. For irregular-shaped objects, e.g., flowerpot (c), all leaves have similar features, and different features are learned for pot and stem. For airplane (d), all turbines have similar features since it is relatively small and curved, and the other sharply curved front and back regions of the airplane have similar features.

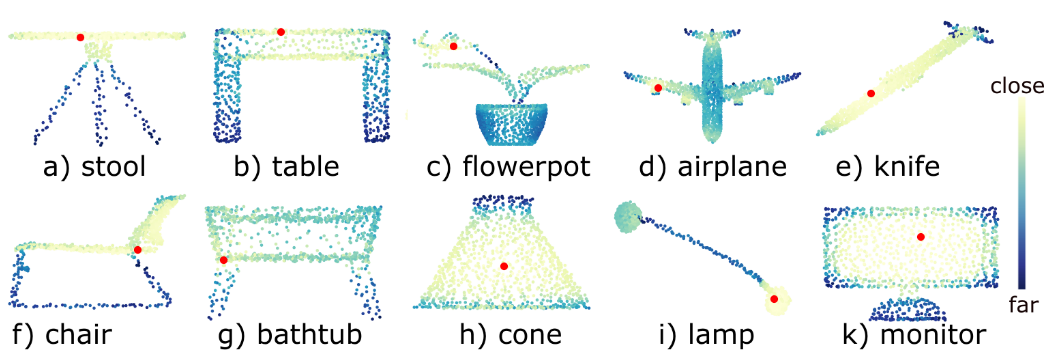


Figure 3: Feature visualization of unseen objects selected from the test sets of ShapeNet and ModelNet-40. For more qualitative results please check the Appendix.

augmentations					novel modules		dataset
jitter	translation	rotation	scaling	crops	GFM	GA	Modelnet-40
✓	✓	✓	✓				84.8
✓	✓	✓	✓	✓			89.7
✓	✓	✓	✓	✓	✓		90.7
✓	✓	✓	✓	✓		✓	90.4
✓	✓	✓	✓	✓	✓	✓	**91.3**

Table 6: Ablation Study of CLR-GAM: Trained on ShapeNet using self-supervised method and evaluated ModelNet-40 using Linear-SVM. Reported results are overall accuracy

4.3 Ablation Study

We conduct an ablation study on ModelNet-40 dataset to understand the contribution of GFM, GA, and augmentation. The results are shown in Table 6. Contrastive learning without cropping achieves around 84.8% in the overall accuracy. With cropping, a large improvement of 4.9% is observed. The result is similar to the performance of CrossPoint (Afham et al., 2022) without multimodal training (i.e., only Intra Modal Instance Discrimination, IMID). We treat the model as the vanilla baseline, i.e., the second row in Table 6. With GFM, we observe a performance improvement by 1.1% compared to the vanilla baseline. A 0.77% improvement is observed when GA is added. When both novel modules are introduced, we observe 1.78% improvement compared to vanilla baseline. The ablative studies demonstrate the effectiveness of the proposed GA and GFM.

We depict all features generated from our CLR-GAM approach on unseen samples of ModelNet-10 test dataset using the DGCNN backbone in Figure 4. To generate t-SNE plots, we use a perplexity of 30. In the vanilla contrastive learning approach, except monitor class, all the other classes have a wider spread making the classes closer. With the proposed GFM, we observe the improvement in nightstand toilet classes, but with a similar overlap of bed bathtub classes as vanilla. With added GA, our proposed approach CLR-GAM, we observe further improvement in toilet class separation from nightstand, and more concentrated class clusters. In all cases, the dresser and night stand had more confusion because of the similarity in shape.

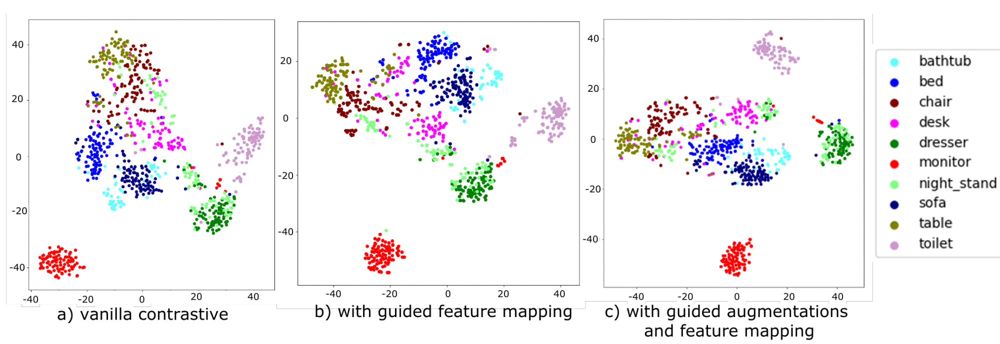


Figure 4: t-SNE plots: visualization of features from three different approaches, generated from unseen samples of ModelNet-10 test dataset.

5 Conclusion

In this paper, we present a contrastive learning framework (CLR-GAM) with guided augmentation (GA) to search augmentation parameters efficiently and guided feature mapping (GFM) to associate structural features precisely. The former is realized by adapting the inverse Dirac delta function with a memory bank, and the latter is fulfilled by associating structural features between two augmented point clouds. Both these processes help boost the contrastive learning of point cloud data. We benchmark on three different downstream tasks and show that our method performs state-of-the-art compared to other methods trained on single modality point cloud data. It also performs similar to or better than a recent multimodal trained approach CrossPoint.

6 Ethics Statement

This paper focuses on contrastive learning for point cloud, a crucial sensory data for a wide range of applications in robotics and intelligent driving systems. Discriminative 3D point cloud representations learned in a self-supervised manner are attractive because it improves sample efficiency (as we present in Section 4.1.b) for training downstream tasks. While promising potentials across various applications are expected, it could potentially have an adverse effect on annotators and annotating companies that rely on annotating point cloud datasets.

7 Reproducibility Statement

The code and pretrained models are made available in the supplementary material. The uncertainty error bars are made available in Appendix and in Table-3,4. For classification and segmentation the reported values are average of 5 runs. For Few shot learning the reported results are average of 10 runs as mentioned in Section 4.1.b.

References

Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas Guibas. Learning representations and generative models for 3d point clouds. In *International conference on machine learning*, pp. 40–49. PMLR, 2018.

Mohamed Afham, Isuru Dissanayake, Dinithi Dissanayake, Amaya Dharmasiri, Kanchana Thilakarathna, and Ranga Rodrigo. Crosspoint: Self-supervised cross-modal contrastive learning for 3d point cloud understanding. *arXiv preprint arXiv:2203.00680*, 2022.

Shipra Agrawal and Navin Goyal. Analysis of thompson sampling for the multi-armed bandit problem. In *Conference on learning theory*, pp. 39–1. JMLR Workshop and Conference Proceedings, 2012.

Peter Auer. Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research*, 3(Nov):397–422, 2002.

Adrià Puigdomènech Badia, Pablo Sprechmann, Alex Vitvitskyi, Daniel Guo, Bilal Piot, Steven Kapturowski, Olivier Tieleman, Martín Arjovsky, Alexander Pritzel, Andew Bolt, et al. Never give up: Learning directed exploration strategies. *The International Conference on Learning Representations*, 2020.

Marc Bellemare, Sriram Srinivasan, Georg Ostrovski, Tom Schaul, David Saxton, and Remi Munos. Unifying count-based exploration and intrinsic motivation. *Advances in neural information processing systems*, 29, 2016.

Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.

Christoforos C Charalambous and Anil A Bharath. A data augmentation methodology for training machine/deep learning gait recognition algorithms. *arXiv preprint arXiv:1610.07570*, 2016.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PMLR, 2020.

Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3): 273–297, 1995.

Mandar Dixit, Roland Kwitt, Marc Niethammer, and Nuno Vasconcelos. Aga: Attribute-guided augmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7455–7463, 2017.

Bi'an Du, Xiang Gao, Wei Hu, and Xin Li. Self-contrastive learning with hard negative sampling for self-supervised point cloud learning. In *Proceedings of the 29th ACM International Conference on Multimedia*, pp. 3133–3142, 2021.

Benjamin Eckart, Wentao Yuan, Chao Liu, and Jan Kautz. Self-supervised learning on 3d point clouds by learning discrete generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8248–8257, 2021.

Matheus Gadelha, Rui Wang, and Subhransu Maji. Multiresolution tree networks for 3d point cloud processing. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 103–118, 2018.

Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013.

Nick Haber, Damian Mrowca, Stephanie Wang, Li F Fei-Fei, and Daniel L Yamins. Learning to play with intrinsically-motivated, self-aware agents. *Advances in neural information processing systems*, 31, 2018.

Zhizhong Han, Mingyang Shang, Yu-Shen Liu, and Matthias Zwicker. View inter-prediction gan: Unsupervised representation learning for 3d shapes by learning global shape memories to support local view predictions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 8376–8384, 2019a.

Zhizhong Han, Xiyang Wang, Yu-Shen Liu, and Matthias Zwicker. 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 *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 10441–10450. IEEE, 2019b.

Kaveh Hassani and Mike Haley. Unsupervised multi-task feature learning on point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8160–8171, 2019.

Søren Hauberg, Oren Freifeld, Anders Boesen Lindbo Larsen, John Fisher, and Lars Hansen. Dreaming more data: Class-dependent distributions over diffeomorphisms for learned data augmentation. In *Artificial Intelligence and Statistics*, pp. 342–350. PMLR, 2016.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9729–9738, 2020.

Rein Houthooft, Xi Chen, Yan Duan, John Schulman, Filip De Turck, and Pieter Abbeel. Vime: Variational information maximizing exploration. *Advances in neural information processing systems*, 29, 2016.

Siyuan Huang, Yichen Xie, Song-Chun Zhu, and Yixin Zhu. Spatio-temporal self-supervised representation learning for 3d point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6535–6545, 2021.

Jiaxin Li, Ben M Chen, and Gim Hee Lee. So-net: Self-organizing network for point cloud analysis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 9397–9406, 2018a.

Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. Pointcnn: Convolution on x-transformed points. *Advances in neural information processing systems*, 31, 2018b.

Fayao Liu, Guosheng Lin, and Chuan-Sheng Foo. Point discriminative learning for unsupervised representation learning on 3d point clouds. *arXiv preprint arXiv:2108.02104*, 2021.

Yongcheng Liu, Bin Fan, Shiming Xiang, and Chunhong Pan. Relation-shape convolutional neural network for point cloud analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8895–8904, 2019.

Erik G Miller, Nicholas E Matsakis, and Paul A Viola. Learning from one example through shared densities on transforms. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No. PR00662)*, volume 1, pp. 464–471. IEEE, 2000.

Ishan Misra, Rohit Girdhar, and Armand Joulin. An end-to-end transformer model for 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2906–2917, 2021.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

Junhyuk Oh, Xiaoxiao Guo, Honglak Lee, Richard L Lewis, and Satinder Singh. Action-conditional video prediction using deep networks in atari games. *Advances in neural information processing systems*, 28, 2015.

Georg Ostrovski, Marc G Bellemare, Aäron Oord, and Rémi Munos. Count-based exploration with neural density models. In *International conference on machine learning*, pp. 2721–2730. PMLR, 2017.

Pierre-Yves Oudeyer and Frederic Kaplan. How can we define intrinsic motivation? In *the 8th International Conference on Epigenetic Robotics: Modeling Cognitive Development in Robotic Systems*. Lund University Cognitive Studies, Lund: LUCS, Brighton, 2008.

Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *International conference on machine learning*, pp. 2778–2787. PMLR, 2017.

Xingchao Peng, Baochen Sun, Karim Ali, and Kate Saenko. Learning deep object detectors from 3d models. In *Proceedings of the IEEE international conference on computer vision*, pp. 1278–1286, 2015.

Omid Poursaeed, Tianxing Jiang, Han Qiao, Nayun Xu, and Vladimir G Kim. Self-supervised learning of point clouds via orientation estimation. In *2020 International Conference on 3D Vision (3DV)*, pp. 1018–1028. IEEE, 2020.

Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 652–660, 2017a.

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, 2017b.

Yongming Rao, Jiwen Lu, and Jie Zhou. Global-local bidirectional reasoning for unsupervised representation learning of 3d point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5376–5385, 2020.

Grégory Rogez and Cordelia Schmid. Mocap-guided data augmentation for 3d pose estimation in the wild. *Advances in neural information processing systems*, 29, 2016.

Aditya Sanghi. Info3d: Representation learning on 3d objects using mutual information maximization and contrastive learning. In *European Conference on Computer Vision*, pp. 626–642. Springer, 2020.

Jonathan Sauder and Bjarne Sievers. Self-supervised deep learning on point clouds by reconstructing space. *Advances in Neural Information Processing Systems*, 32, 2019.

Charu Sharma and Manohar Kaul. Self-supervised few-shot learning on point clouds. *Advances in Neural Information Processing Systems*, 33:7212–7221, 2020.

Leslie N Smith. Cyclical learning rates for training neural networks. In *2017 IEEE winter conference on applications of computer vision (WACV)*, pp. 464–472. IEEE, 2017.

Bradly C Stadie, Sergey Levine, and Pieter Abbeel. Incentivizing exploration in reinforcement learning with deep predictive models. *arXiv preprint arXiv:1507.00814*, 2015.

Hao Su, Charles R Qi, Yangyan Li, and Leonidas J Guibas. Render for cnn: Viewpoint estimation in images using cnns trained with rendered 3d model views. In *Proceedings of the IEEE international conference on computer vision*, pp. 2686–2694, 2015.

Chao Sun, Zhedong Zheng, Xiaohan Wang, Mingliang Xu, and Yi Yang. Point cloud pre-training by mixing and disentangling. *arXiv preprint arXiv:2109.00452*, 2021.

Richard S Sutton. Integrated architectures for learning, planning, and reacting based on approximating dynamic programming. In *Machine learning proceedings 1990*, pp. 216–224. Elsevier, 1990.

Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J Guibas. Kpconv: Flexible and deformable convolution for point clouds. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 6411–6420, 2019.

Hanchen Wang, Qi Liu, Xiangyu Yue, Joan Lasenby, and Matt J Kusner. Unsupervised point cloud pre-training via occlusion completion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9782–9792, 2021.

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.

Christopher John Cornish Hellaby Watkins. Learning from delayed rewards. 1989.

Jiajun Wu, Chengkai Zhang, Tianfan Xue, Bill Freeman, and Josh Tenenbaum. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. *Advances in neural information processing systems*, 29, 2016.

Saining Xie, Jiatao Gu, Demi Guo, Charles R Qi, Leonidas Guibas, and Or Litany. Pointcontrast: Unsupervised pre-training for 3d point cloud understanding. In *European conference on computer vision*, pp. 574–591. Springer, 2020.

Juyoung Yang, Pyunghwan Ahn, Doyeon Kim, Haeil Lee, and Junmo Kim. Progressive seed generation auto-encoder for unsupervised point cloud learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6413–6422, 2021.

Yaoqing Yang, Chen Feng, Yiru Shen, and Dong Tian. Foldingnet: Point cloud auto-encoder via deep grid deformation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 206–215, 2018.

Li Yi, Vladimir G Kim, Duygu Ceylan, I-Chao Shen, Mengyan Yan, Hao Su, Cewu Lu, Qixing Huang, Alla Sheffer, and Leonidas Guibas. A scalable active framework for region annotation in 3d shape collections. *ACM Transactions on Graphics (ToG)*, 35(6):1–12, 2016.

Ling Zhang and Zhigang Zhu. Unsupervised feature learning for point cloud understanding by contrasting and clustering using graph convolutional neural networks. In *2019 International Conference on 3D Vision (3DV)*, pp. 395–404. IEEE, 2019.

Zaiwei Zhang, Rohit Girdhar, Armand Joulin, and Ishan Misra. Self-supervised pretraining of 3d features on any point-cloud. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10252–10263, 2021.

Yongheng Zhao, Tolga Birdal, Haowen Deng, and Federico Tombari. 3d point capsule networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1009–1018, 2019.

Rui Zhu, Bingchen Zhao, Jingen Liu, Zhenglong Sun, and Chang Wen Chen. Improving contrastive learning by visualizing feature transformation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10306–10315, 2021.

Appendices

A Pretraining

We pretrain on the Shapenet dataset (Chang et al., 2015) using the proposed contrastive self-supervised approach, similar to other reported benchmarks (Afham et al., 2022; Sauder & Sievers, 2019; Huang et al., 2021). The dataset has 55 different classes with a total of 57,386 CAD models. We sample 2048 points before performing augmentation and 1024 after applying all augmentations as mentioned in section 3.2.b. The pretrained model is benchmarked on three downstream tasks.

B Implementation Details

For all of the experiments, we use cyclic learning rates (Smith, 2017) for 3 cycles with each cycle for 100 epochs and a cosine annealing based learning scheduler. We employ Adam optimizer with a learning rate of 10^{-3} and a weight decay of 10^{-4}. PNet (Qi et al., 2017a) and DGCNN (Wang et al., 2019) are utilized for point cloud encoding. We apply augmentation ranges for translation, rotation, and scaling as mentioned in section 3.2.a. For jittering, we apply Gaussian noise of 1cm standard deviation. For cropping, we randomly select a point and crop 30% of the points that are closer to the selected point. Three downstream tasks are benchmarked in this paper, i.e., classification, few-shot learning, and object part semantic segmentation. For training, we use two NVIDIA RTX 6000 GPUs with a batch size B of 32. For Equation 2 in the main manuscript, we use $\alpha_R = 1, \alpha_T = 1, \alpha_S = 1$.

C Guided Augmentation sampling vs Random sampling

We visualize the convergence in performance over 40 epochs for two different sampling techniques in Figure 5. During self-supervised training on the ShapeNet dataset, the performance (accuracy) is evaluated on the validation set of ModelNet-40 after every epoch.

The standard deviation for Guided Augmentation from Table 6 under multiple runs (5) is +/-0.12, compared to the random selection process +/-0.37

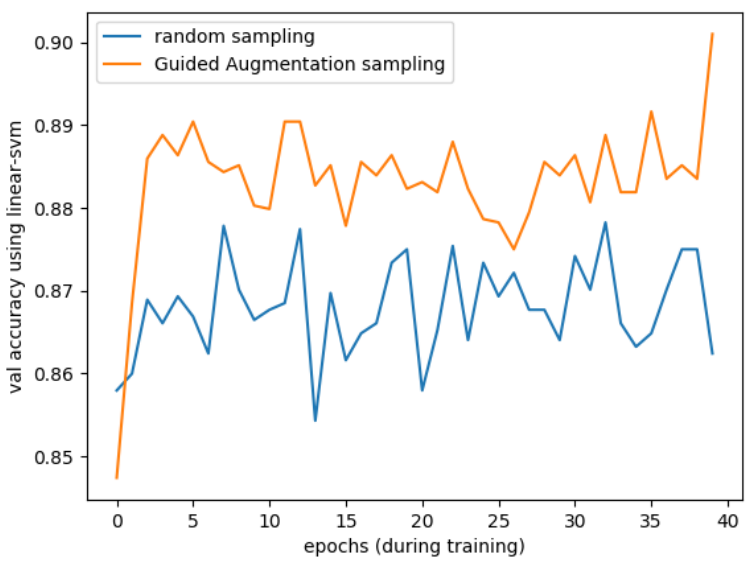


Figure 5: Validation Accuracy on ModelNet-40 using LinearSVM, during self-supervised training with random and guided augmentation sampling.

D LIMITATIONS

The proposed GA module uses a very effective memory mechanism, but it might not be memory efficient with many augmentation samples. It takes 3 minutes and 30 seconds for 35000 augmentations (around the sample size of shapenet dataset), without any advanced libraries (only using the NumPy library with naive implementation) and the storage memory footprint is 2.52 MB (with 8 bytes per element in the array). Please note that we train linear-SVM on the features on tasks (classification/few-sot learning) for both datasets (ModelNet-40/ScanNet), because of this the memory limitation only applies to the pretrained dataset.

E DISCUSSIONS

E.1 MEMORY SIZE ON DIFFERENT DATASETS

We trained only one dataset for self-supervised learning (ShapeNet dataset) even though there are different tasks/datasets that are tested using Linear-SVM. So in our experiments, it doesn't change with tasks/datasets that are tested on. But without memory, there is a performance degradation of 0.8%, as seen in Table 6. We chose memory based on the size of the dataset it is pretrained on.

E.2 WHY GUIDED FEATURE MAPPING WHEN THERE IS A POOLING OPERATION?

The pooling operation is performed on the encoded features and before latent feature projection. But because of cropping the same point cloud can resemble being coming from two different classes, as mentioned in the Introduction section. So we hypothesize that only pooling features that have similar structural similarities will result in an effective contrastive learning, which is also observed in our empirical results. To study the effectiveness of the Latent features, in the main manuscript we also show t-SNE plots in Figure 4.

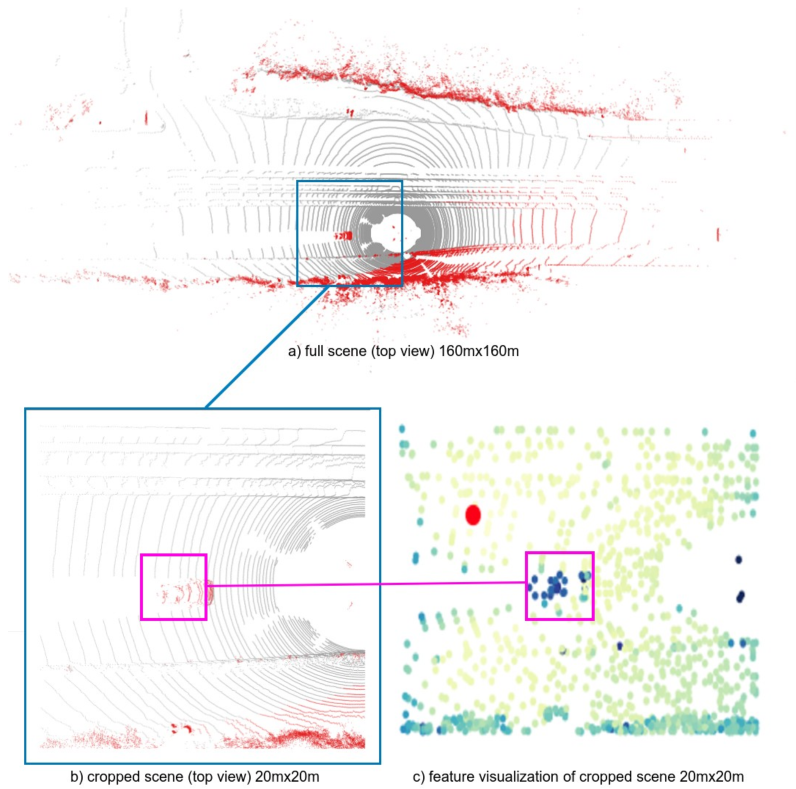
E.3 EXTENSION TO OTHER SENSOR MODALITIES

This is an interesting future direction that can be explored. Based on our understanding, our approach can be applied to such works, as crosspoint. To ensure efficient cross-modal embedding, we also need to search for the right approaches for images. That is not the focus of this paper, so we leave it to future work.

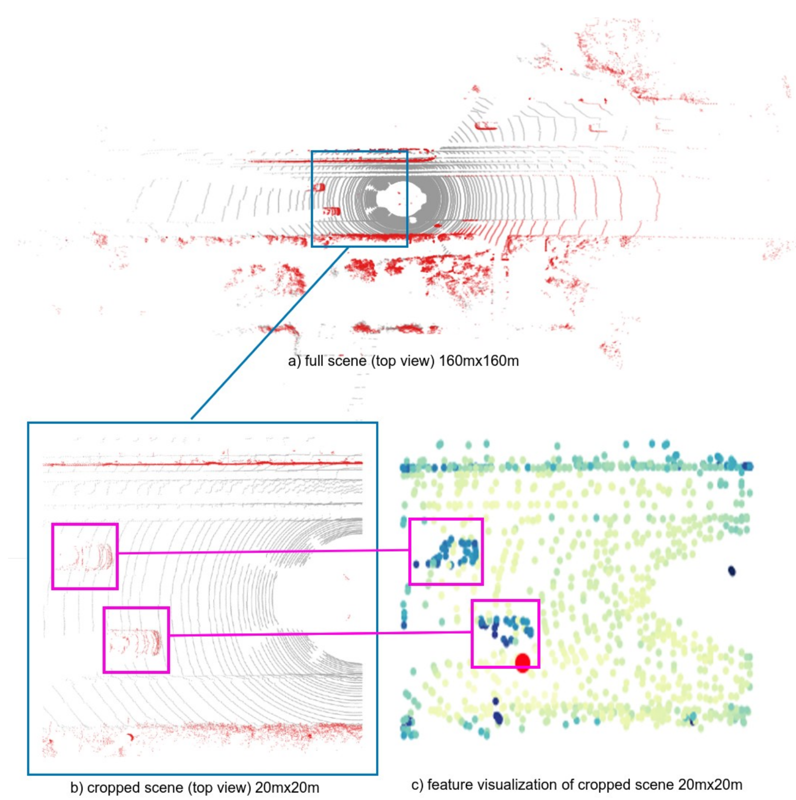
F QUALITATIVE RESULTS (KITTI)

In order to understand the generalization of the proposed unsupervised approach to real world application or datasets, we perform feature visualization of two driving scenarios from KITTI dataset (Geiger et al., 2013) in Figure 6. The full scene contains 80 meters on all directions to the ego-vehicle (160m x 160m) is show in (a) as a top down image. In (a) the gray color is used for ground and red color is used for non-ground or obstacles. The separation is done using -1.5 meters in height axis of the pointcloud data or velodyne sensor. Blue box is the region of interest which is zoomed in subfigure (b), which is 20m x 20m region. This is subsampled to around 4000 points using voxel based sampling with 0.3 meter voxel length in all three axes. 1024 points are randomly selected and passed to feature encoder. The features are visualized in subfigure (c). The color scale is same as Figure 6 in main manuscript, Yellow-Green-Blue. The closest feature in the feature space is yellow, and the farthest is blue with respect to a randomly selected point (colored in red).

In scenario 1 the single vehicle has distinct features from the road, which is highlighted in pink box. Similarly in scenario 2 the two vehicles have similar features distinct from the ground, which are highlighted in pink boxes.



scenario 1



scenario 2

Figure 6: Feature visualization of unseen **driving scene** selected from the KITTI dataset.

G Qualitative Results (ModelNet40)

We visualize feature representations (learned from the proposed CLR-GAM) of each point/node in an unseen object's point cloud selected from test sets of ModelNet-40 in Figure 3. The color scale is same as Figure 3 in main manuscript, Yellow-Green-Blue. The closest feature in the feature space is yellow, and the farthest is blue with respect to a selected point (colored in red). Some qualitative results and discussions of the airplane, bathtub, bed, guitar, person, vase and lamp are shown below.

G.1 airplane

In the Figure 7(a-d) we visualize four different airplanes pointcloud features. In (a,b,d) the selected points (red dot) for the three different planes are on the wings. Except the sharper wings ends or tail ends or engines or mouth of the airplane, the whole body of the plane has similar features. Similarly, in (c) when selected sharper wing end (red dot), tail wings are more closer in the feature space, along with engines and mouth of the airplane.

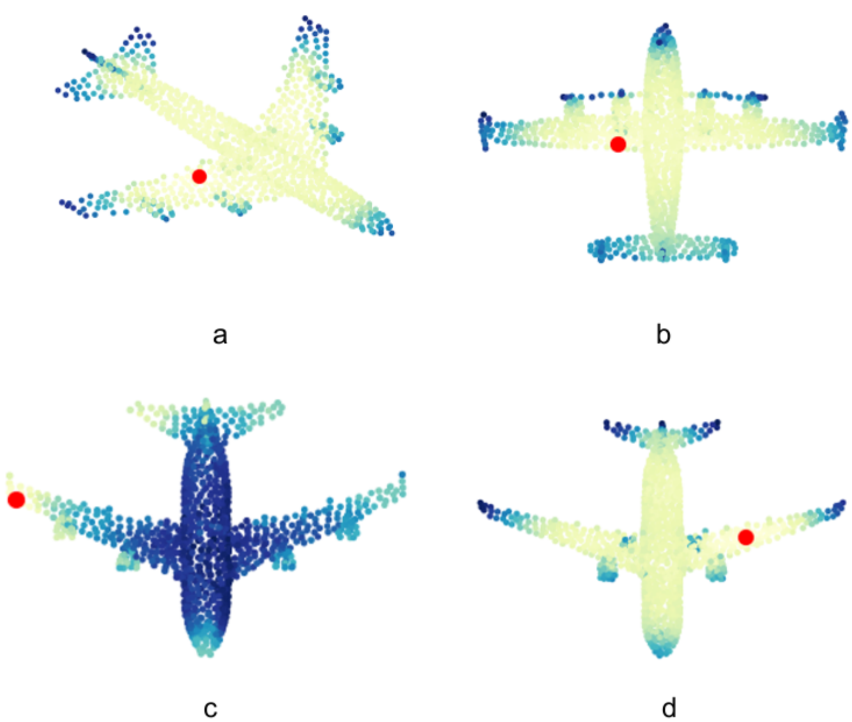


Figure 7: Feature visualization of unseen **(airplane)** objects selected from the test sets of ModelNet-40.

G.2 BATHTUB

In the Figure 8(e,f) we visualize two different bathtub pointcloud features. In (e,f) we selected points shown in red dot are on the tub. In (e) the whole symmetrical tub shape has similar features excluding the legs and top edge handle. Similarly in (f) the tap/handle, separate object and sharp corners has different features from the rest of the bath tub.

Figure 8: Feature visualization of unseen **(bathtub)** objects selected from the test sets of ModelNet-40.

G.3 BED

In the Figure 9(g,h) we visualize two different bed pointcloud features. In (g) the selected point (red dot) is on box spring, the whole part has similar features excluding legs and head board . In (h) the selected point is close to foot board, since there is no separate foot board in this pointcloud the whole box spring has similar features excluding legs and head board.

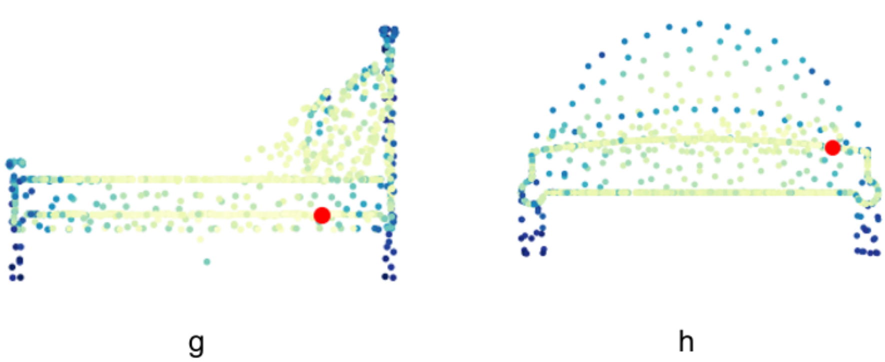


Figure 9: Feature visualization of unseen **(bed)** objects selected from the test sets of ModelNet-40.

G.4 GUITAR

In the Figure 10(i,j) we visualize two different guitar pointcloud features. In (i) the selected point (red dot) is on the nut, the whole finger board and head stock has same features excluding the body (since the head stock doesn't have any varied design as shown in (j)). In (j) the selected point is on head stock, only head stock and nut has similar features, finger board and body have different features.

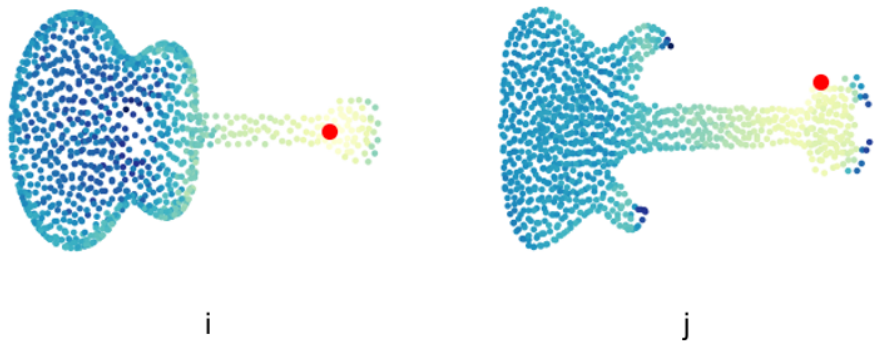


Figure 10: Feature visualization of unseen **(guitar)** objects selected from the test sets of ModelNet-40.

G.5 PERSON

In the Figure 11(k,l) we visualize two different person pointcloud features. In (k) the selected point (red dot) is on the leg, both the legs have same features excluding the feet and the upper body. Similary in (l) the selected point is on the ball and the person is catching the ball in this pointcloud. The person's head and the ball have same features because they are round in shape.

Figure 11: Feature visualization of unseen **(person)** objects selected from the test sets of ModelNet-40.

G.6 VASE

In the Figure 12(m,n) we visualize two different vase pointcloud features. In (m) the selected point (red dot) is on the body of the vase, the whole body has similar features excluding the lip, foot and neck. In (n) the selected point is also on the body. Even though the body shape is complicated the whole body has similar features, excluding the lip.

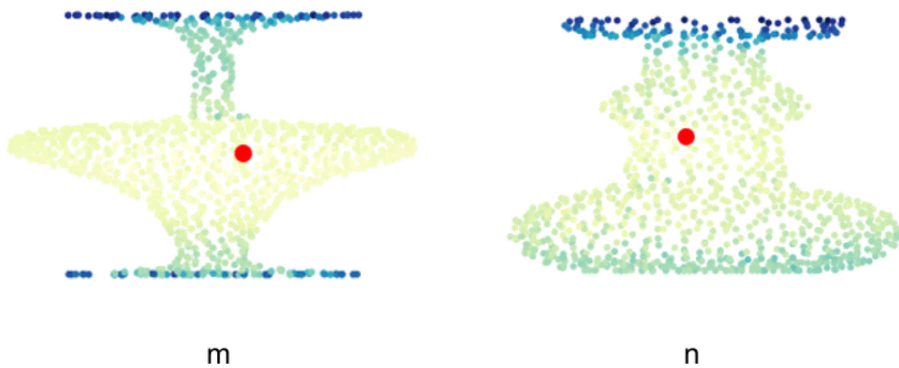


Figure 12: Feature visualization of unseen **(vase)** objects selected from the test sets of ModelNet-40.

G.7 LAMP

In the Figure 13(o,p) we visualize two different lamp pointcloud features. In both cases the selected point (red dot) is on the shade. In (o) the complete shade has same features, even tough the bulb and tube are closer they have different features. In case of (p) the shade and bridge arm have same features, excluding the base and tube.

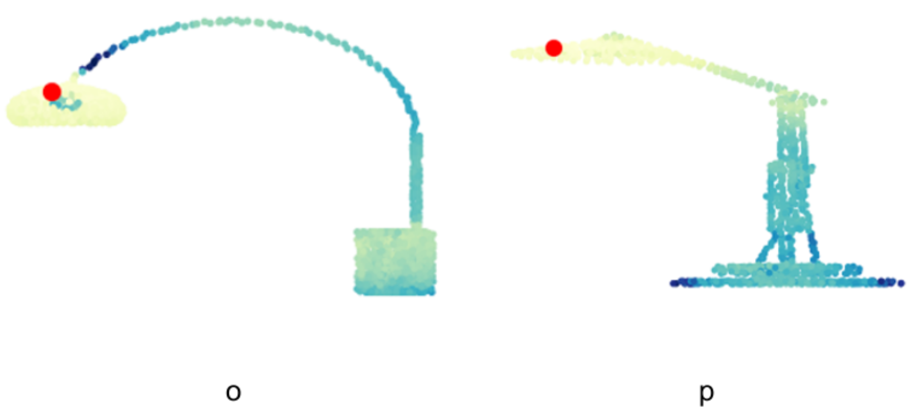


Figure 13: Feature visualization of unseen **(lamp)** objects selected from the test sets of ModelNet-40.

H Notations

Augmentations

$\mathbf{a}$	set of all augmentations
$\mathbf{a}^S$	scaling
$\mathbf{a}^T$	translation
$\mathbf{a}^R$	rotation
$\mathbf{a}^J$	jitter
$\mathbf{a}^C$	crop
$\hat{\mathbf{a}}$	randomly sampled augmentation
$\mathbf{a}^*$	novel augmentation
$\mathbf{a}^{-\mathbf{1}}$	inverse augmentation

PointCloud, Features, Memory

P	pointcloud
$\mathbf{x}$	points in pointcloud
n	number of points in the pointcloud
$\mathbb{R}$	real numbers
z	projected latent feature
N	number of randomly sampled augmentations
M	memory bank
m	index of the memory slot in memory bank
K	dirac delta kernal function
d	total distance measure
d_R	angular distance measure
c, ϵ	small values for numerical stability
S	structural index mapping
$\otimes$	augmentation operator
$\odot$	indexing operator
B	size of mini-batch
b	index of the feature in mini-batch

Indexing

P_i	sample i of pointcloud from the dataset
F_i	feature set corresponding to the sample i of pointcloud
$\mathbf{x}_j$	jth point in the pointcloud
$F(j)$	feature corresponding to the jth point in the pointcloud
$\mathbf{a}_k$	kth augmentation
P^k	pointcloud augmented with augmentation with index k
F^k	features of pointcloud with augmentation with index k
S_{12}	structural index mapping from pointcloud 1 to 2