Eurographics Workshop on 3D Object Retrieval (2024)
S. Biasotti, B. Bustos, T. Schreck, I. Sipiran, and R. C. Veltkamp (Editors)

Self-distillation for Efficient Object-level Point Cloud Learning

Lucas Oyarzún[1] and Ivan Sipiran[1,2] and José M.Saavedra[3]

[1]Department of Computer Science, University of Chile, Chile
[2]National Center for Artificial Intelligence (CENIA), Chile
[3]Faculty of Engineering and Applied Sciences, Universidad de los Andes, Chile

Abstract

The emerging accessibility of 3D point cloud data has catalyzed the evolution of deep-learning methodologies for analysis and processing of 3D data. However, the efficacy of neural networks in this domain is often inhibited by the necessity for extensively labelled datasets. In this study, we investigate the application of self-distillation techniques based on Siamese networks, BYOL and SIMSIAM, to pre-train encoders designed for 3D point cloud processing. These pre-training regimes enable encoders to generate data representations without label reliance, potentially supporting network performance in downstream tasks. The efficacy of these learned representations was assessed using the established evaluation methodologies for pre-training: linear probing and finetuning. We also incorporate an analysis of self-supervised features in a retrieval scenario. Furthermore, the impact of these representations on subsequent applications was evaluated via transfer learning by employing pre-trained models as a foundation for standard test datasets.

CCS Concepts

• ***Computing methodologies*** → ***Point-based models; Shape analysis; Shape representations;***

1. Introduction

Deep learning models have completely revolutionized image processing, natural language understanding, and 3D data analysis by excelling in complex tasks through the learning by using large datasets. This paper focuses on the application of neural networks to analyze 3D point clouds, which are essential in many applications such as robotics, virtual reality, and the automotive industry. Furthermore, point clouds play a crucial role in tasks such as object detection and reconstruction in computer vision and robotics.

However, the unstructured nature of the point clouds introduces substantial computational challenges. Processing these data requires specialized algorithms capable of handling noise, missing data, and variations in point density. Despite these obstacles, point clouds are invaluable for providing detailed insights into the geometry and spatial relationships of objects, making them indispensable for analyzing complex three-dimensional scenes [Yil20, BDTD*16]. The proliferation of 3D data and the development of extensive datasets, such as ShapeNet [CFG*15], ModelNet [WSK*15], and ScanNet [DCS*17] have been crucial for training neural networks on 3D tasks. These datasets facilitate the exploration of the capabilities of deep learning models in understanding and interacting with three-dimensional environments. Nonetheless, acquiring new 3D data samples remains a significant challenge because manual labeling is a resource-intensive and time-consuming task.

Self-supervised learning (SSL) has emerged as a promising approach to these challenges, offering a methodology for training models without manually labeled datasets. By generating pseudo-labels based on the inherent characteristics of the data, SSL enables models to autonomously learn features, facilitating the transfer of this knowledge to perform subsequent tasks, such as classification and segmentation [EGLH22].

Self-supervised learning in point clouds has mainly been addressed using approaches based on masked autoencoders. These approaches divide a point cloud into groups that are often processed using transformer-like architectures. The neural network encodes the input groups and attempts to decode them into the original point cloud. The masking strategy consists of hiding groups in the neural network and training the network to reconstruct the original point cloud. However, given the nature of processing the information in groups, a transformer architecture seems to be an ideal candidate to learn the relations between the partial point cloud and the required reconstruction. The disadvantage of this approach is that it requires a large network to train with the subsequent time and memory footprint to work properly.

In this context, our research aims to explore the potential of other self-supervised learning techniques, particularly the use of Siamese network architectures trained with a self-distillation regime such as BYOL [GSA*20] and SIMSIAM [CH21], to enhance the processing and understanding of 3D point clouds. These models, which have shown remarkable success in image-related tasks, prioritize contrastive learning and employ simpler neural network ar-

DOI: 10.2312/3dor.20241174

chitectures, offering a fresh perspective on learning from three-dimensional data without the extensive requirement for labeled datasets and the resource-consuming use of large neural networks. By adopting architectures inspired by SIMSIAM and BYOL, we aim at optimizing efficiency, reducing data requirements, and minimizing resource consumption and training time, thereby setting a new benchmark for state-of-the-art methods in the processing of 3D point clouds.

The rest of the paper is organized as follows. Section 2 describes the related work on point cloud learning. Section 2.1 presents the background on self-supervised learning. Section 3 focuses on our proposal for siamese networks for point cloud learning. Section 4 shows our experiments and results. Finally, Section 5 concludes our work.

2. Related work

Since the emergence of deep learning, much research has been carried out to try to learn from 3D data, and more specifically from point clouds. The tasks investigated have also been diverse: classification, segmentation, object detection in scenes, to name a few. In this section we present the most relevant works for our study. For a more comprehensive presentation, we recommend the reader to review the survey by Guo et al. [GWH*20] on deep learning in point clouds.

A pioneering approach was introduced by PointNet [QSMG17] that models individual points and aggregates the features in a single embedding to represent the entire object. Subsequently, PointNet++ [QYSG17] included a mechanism to aggregate features hierarchically, allowing adaptive learning of the geometric features of an object at different scales. On the other hand, DGCNN [WSL*19] models a point cloud as a $k-$nearest neighbour graph and learns features by dynamically applying convolutions on the graph. More recently, and due to the success of Transformer-like architectures, PointTransformer architectures [ZJJ*21, WLJ*22, WJW*24] were proposed to represent a point cloud as a sequence of tokens that are transformed by the neural network, in the same way as text and patch-based representation is modelled in images. Also, as an alternative to transformers and their high processing load, PointMLP [MQY*22] was proposed as a pure MLP architecture offering high performance with a focus on reducing the computational load and complexity of models to process point clouds.

2.1. Self-supervised Point Cloud Learning

Self-supervised learning has emerged as an effective alternative that takes advantage of large amounts of data without the need for labels. These methods work by trying to find intrinsic relationships in the data guided by a pretext task to be optimised. From this perspective, there are two ways of categorising existing methods: by the pretext task, or by the way intrinsic relationships are searched. Existing point-cloud SSL surveys [XHG*23, ZWN*24] focus on categorising and analysing existing methods from the perspective of the pretext task. However, in this section, we follow Balestriero et al. [BIS*23], who proposed to classify the methods depending on how the self-supervision mechanism learns from the data. We identified that point cloud methods concentrate on contrastive metric learning and masking modelling approaches. Contrastive methods include DepthContrast [ZGJM21], STLR [HXZZ21], Conclu [MHL*22], and ContrastMPCT [WY22]. These methods use a contrastive loss to leverage the instance discrimination between samples. Masking-based models include Point-BERT [YTR*22], Point-MAE [JLZ*23], Point-DAE [ZLL*22], Point-M2AE [ZGG*22], and Point-MA2E [ZLL*23]. These methods tokenize the point clouds into smaller groups and applies token masking along with a reconstruction loss for learning.

3. Self-distillation for 3D Point Clouds

Self-distillation is a method for self-supervision that uses two neural networks that try to produce similar embeddings under different transformed version of an input. In simple words, the method tries to learn the transformation equivariance of the inputs. The interesting result is that embeddings produced by networks pre-trained with this approach are effective for downstream tasks.

Two methods on this approach are BYOL and SimSiam (see Fig. 1). BYOL consists of an online network (composed of an encoder f_ϕ followed by a predictor Π) that produces a first embedding α_P, and a target network (composed only of an encoder f_ψ) that produces a second embedding β_P. The loss function of BYOL is the L_2 distance between the normalized embeddings; that is $\mathcal{L}(\alpha_P, \beta_P) = \|\overline{\alpha_P} - \overline{\beta_P}\|_2^2$. To avoid embedding collapse, the weight update is performed assymetrically where the online network uses typical backpropagation and target network uses a exponential moving average of online network.

On the other hand, SimSiam uses shared weights for the online and target network where the backpropagation is performed assymmetrically on the output embeddings. This behavior is obtained by applying a stop-gradient operation on the network branches. In this way, each backpropagation step operates only on the operation that produces one embedding. Therefore the loss function is

$$\mathcal{L}(\alpha_P, \beta_P) = \frac{1}{2}\mathcal{D}(\alpha_P, sg(\beta_P)) + \frac{1}{2}\mathcal{D}(sg(\alpha_P), \beta_P) \qquad (1)$$

where $\mathcal{D}$ is the negative cosine similarity between two embeddings and sg is the stop-gradient operation.

3.1. Adaptation to Point Clouds

Adapting techniques from 2D image processing to 3D point cloud analysis requires careful consideration of transformations and their relevance or applicability in a new domain. Generalizable transformations such as translation, flipping, cropping, rotation, scaling, and noise are reinterpreted for 3D points $\mathcal{P} = \{x, y, z\}$. Networks processing 3D data must adapt to these changes, potentially through data augmentation techniques during training or employing symmetric functions like max pooling to foster invariance to these transformations.

In our data preprocessing pipeline, we applied a series of transformations to enhance model robustness and generalizability: random scaling within [2/3, 3/2] range, rotations along X, Y, Z axes, translations of ±0.2 units across each axis, jittering with a standard

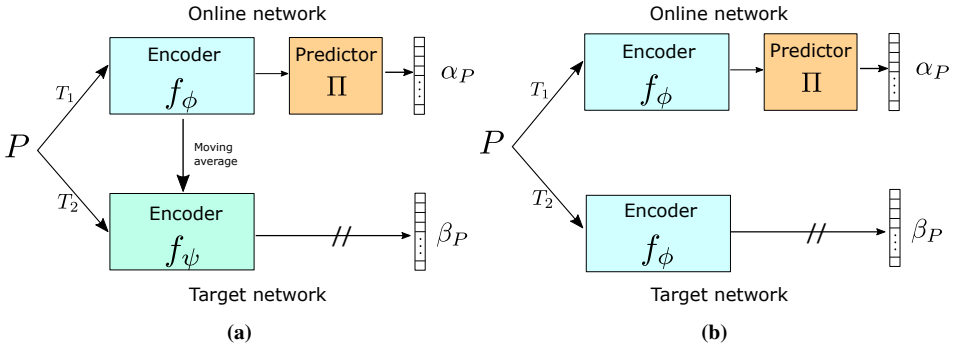

Figure 1: *Left: BYOL architecture. Right: SimSiam architecture.*

deviation of 0.01 and at 0.05, followed by an additional layer of scaling and translations within the same bounds.

This sequence concludes with a random masking operation that eliminates 60% of the data points. This operation is tailored to the specific encoder used. For PointNet, DGCNN, and PointMLP, we employ the farthest point sampling algorithm combined with k-nearest neighbors (kNN) with $k = 32$ to form potentially overlapping point clusters. From these clusters, % are randomly selected, and all points within them are removed. For Transformer encoders, similar to the approach in Point-MAE, we generate tokens for the clusters created using farthest point sampling and kNN with $k = 32$. Subsequently, 60% of these generated tokens are eliminated.

The practice of random masking follows the principle that a uniformly masked point cloud should still be identifiable by the encoder as akin to another masked version of the same object. This assertion underlines the non-uniform semantic value of points within a cloud, where corners and curves contain more information than intermediate points, allowing for the accurate characterization of point clouds based on the most significant points [PWT*22].

To complement the exploration, various neural networks were selected to act as the structural backbone. The encoders chosen represent a spectrum of network types: PointNet [QSMG17] as a point-wise network, DGCNN [WSL*19] for graph-based processing, a Transformer [VSP*17] backbone following Point-BERT [YTR*22], and Point-MLP [MQY*22], noted for its state-of-the-art performance despite its simplicity, relying solely on MLP networks. We name the resulting networks as PointBYOL and PointSimSiam.

4. Experiments

We conduct several experiments to show the effectiveness of our proposed PointBYOL and PointSIMSIAM. Our evaluation follows an established protocol to evaluate self-supervised techniques. First, we pre-train the networks using a general purpose dataset for point clouds, the ShapeNet55 dataset. Second, we test the effectiveness of pre-trained networks in several downstream tasks such as classification, part segmentation and few-shot learning. For the case of classification, it is common to evaluate the pre-trained networks in two scenarios: linear probing and fine-tuning. Linear probing shows the suitability of feature space to be classified with a linear method. Finetuning shows the effectiveness of classifying features with a non-linear classifier. Finally, we evaluate the suitability of pre-trained features in a retrieval scenario.

4.1. Pretraining Setup

The ShapeNet55 dataset, as curated by Yu et al. [YRW*21], was used for network pre-training. Adhering to the benchmark established by Point-MAE, 1024 points per model were extracted using Farthest-Point-Sampling for consistency with prior art. Preprocessing includes linear transformations and masking. AdamW optimizer [LH17], coupled with cosine learning rate decay [LH16], facilitates training over 300 epochs, batch size of 128, initializing learning rate at 0.001 and weight decay at 0.05, masking 60% points per cloud.

This study scrutinizes multiple variants of PointSIMSIAM and PointBYOL models, integrated with encoders such as PointNet, DGCNN, Transformer, and Point-MLP. These encoders, adapted to the linear and masking transformations specified in Section 3.1, generate varied implementations of the base models.

Modifications to the original encoders were explored to enhance their pre-training effectiveness. For PointNet, the elimination of T-Net modules yielded 'PointNet(w/o ST)', focusing on invariance to linear transformations. The Transformer encoder was tested in two variants: one using defined transformations T and another, named Transformer(TM), excluding the last point masking transformation in favor of Point-MAE's tokenization approach.

Additionally, Point-MLP was evaluated in its standard form and a simplified 'Point-MLP elite'. Encoder details, including inference times and parameter counts, are detailed in Table 1, following the analysis by Xu Ma et al. [MQY*22]. Notably, the PointTransformer stands out for its parameter complexity, while DGCNN exhibits longer inference times due to its computation-intensive tasks. In contrast, PointNet and Point-MLP elite showcase exceptionally low inference times, underscoring their suitability for practical 3D model handling applications.

Encoder	Inference time [s]	# parameters
PointNet	1.0	3.1M
PointNet (w/o ST)	0.8	0.4M
DGCNN	264.0	1.2M
Transformer	15.0	22.1M
Point-MLP	5.3	12.9M
Point-MLP elite	0.1	0.6M

Table 1: *Inference times of different encoders for ModelNet40 train set (subsampled to 1024 points) and the number of trainable parameters for each model's encoder.*

4.2. Downstream Tasks

4.2.1. Classification with Linear Probing

After the pre-training, we applied linear probing to evaluate the effectiveness of our methods with features extracted directly from the frozen encoders. The features are directly classified by a linear SVM, or a kNN classifier with $k = 40$. The relevance of this experiment is attained to the evaluation of the embedding quality. The idea behind this evaluation is that representative embeddings obtained from self-supervised encoders should be effectively classified with a simple classifier.

For this experiment, we use the test set of ModelNet40 dataset. We compare our results against autoencoder-based models (Point-DAE [ZLL*22], Point-BERT [YTR*22], Point-MAE [PWT*22], Point-M2AE [ZGG*22], and Point-MA2E [ZLL*23]) and contrastive learning-based models (DepthContrast [ZGJM21], STLR [HXZZ21], GISR, DCGLR, Conclu [MHL*22]). Results are shown in Table 2.

4.2.2. ModelNet Classification with Finetuning

After pre-training, the encoder is coupled with a multi-layer perceptron (MLP) and finetuned for classifying the ModelNet40 dataset. Table 3 shows the effectiveness of classifying the test split in the ModelNet40 dataset. In this table, we also incorporate ContrastMPCT [WY22] method in the comparison.

In contrast to linear probing, the evaluation on finetuning shows the capacity of embeddings to be transformed by a non-linear classifier to boost the accuracy.

4.2.3. ScanObjectNN Classification with Finetuning

The pre-trained network is coupled with a MLP classifier and finetuned to classify the ScanObjectNN dataset. This dataset consists of real-world objects scanned from indoor scenes and it contains approximately 15,000 objects categorized in 15 classes. The dataset has three variants: OBJ-ONLY, OBJ-BG, and PB-T50-RS. Variant OBJ-ONLY contains only objects isolated from any background or context information from the scene. Variant OBJ-BG contains objects with surrounding background. Finally, variant PB-T50-RS contains objects with allowed partiality as a consequence of translating up to 50% of the bounding box with random scaling and rotation, before applying the crop of the object from the scene. Note that we described the variants in ascending order of complexity. Table 4 shows the results on ScanObjectNN dataset. In this evaluation, we also include recent results from MAE3D [JLZ*23] method.

Method	Backbone	SVM Acc.	kNN Acc.
Point-DAE	PointNet	89.3	-
Point-DAE	DGCNN	91.9	-
Point-BERT	Transformer	87.4	-
Point-MAE	Transformer	91.0	-
Point-M2AE	Transformer	92.9	-
Point-MA2E	PointNet	89.4	-
Point-MA2E	DGCNN	92.1	-
Point-MA2E	Transformer	**93.1**	-
DepthContrast	PointNet++	85.4	-
STLR	PointNet	88.3	-
STLR	DGCNN	90.9	-
GISR	DGCNN	90.4	-
DCGLR	3D-ViT	91.3	-
DCGLR	PCT	91.4	-
ConClu	DGCNN	91.6	-
PointSIMSIAM	PointNet	67.9	60.6
PointBYOL	PointNet	85.3	76.9
PointSIMSIAM	PointNet(w/o ST)	88.0	79.0
PointBYOL	PointNet(w/o ST)	88.3	79.7
PointSIMSIAM	DGCNN	89.6	81.7
PointBYOL	DGCNN	**91.7**	**85.5**
PointSIMSIAM	Transformer	87.3	82.6
PointBYOL	Transformer	85.7	80.5
PointSIMSIAM	Transformer(TM)	88.3	82.3
PointBYOL	Transformer(TM)	76.0	74.7
PointSIMSIAM	Point-MLP	88.8	83.6
PointBYOL	Point-MLP	88.2	83.7
PointSIMSIAM	Point-MLP elite	87.5	83.2
PointBYOL	Point-MLP elite	88.4	82.7

Table 2: *Linear evaluation on ModelNet40 [WSK*15] using SVM and kNN. Methods are categorised according to their self-supervised learning (SSL) methodology. The best of the proposed models and the best state-of-the-art model are highlighted in bold. Models are organised by methodology: first, denoising autoencoder models; then, contrastive learning-based models; and finally, the proposed models.*

4.2.4. Few-shot Learning

We follow the protocol used in previous works to evaluate the few-shot learning ability of our pre-trained networks. We adopt the *n*-way, *m*-shot setting on the ModelNet40 dataset. The protocol consists of randomly selecting n classes from the dataset and randomly selecting m objects from each class for training. For testing, the evaluation randomly samples 20 objects from each of the n selected classes. The set of objects in testing is entirely different to the set of objects selected for training in each class.

Results for few-shot learning can be seen in Table 5. We experiment with standard values of $n \in \{5, 10\}$ and $m \in \{10, 20\}$ for a fair comparison with previous methods in the literature. In addition, the results show the mean accuracy and deviation for 10 experiments. We include the results from the Cover-tree method [SK20], the pioneering method in proposing the protocol for few-shot learning on the ModelNet dataset.

Method	Backbone	Acc.	Voting Acc.
Point-DAE	PointNet		90.6
Point-DAE	DGCNN		93.3
Point-BERT	Transformer		93.2
Point-MAE	Transformer		93.8
Point-M2AE	Transformer		**94.0**
DepthContrast	PointNet++		85.4
ContrastMPCT	Transformer	**93.3**	
STRL	PointNet		88.6
STRL	DGCNN		90.8
PointSIMSIAM	PointNet	91.1	90.9
PointBYOL	PointNet	91.3	91.2
PointSIMSIAM	PointNet(w/o ST)	89.9	90.0
PointBYOL	PointNet(w/o ST)	90.0	90.3
PointSIMSIAM	DGCNN	92.3	92.6
PointBYOL	DGCNN	93.0	93.0
PointSIMSIAM	Transformer	92.5	92.5
PointBYOL	Transformer	92.7	93.1
PointSIMSIAM	Transformer(TM)	91.9	92.5
PointBYOL	Transformer(TM)	92.5	92.9
PointSIMSIAM	Point-MLP	92.9	93.0
PointBYOL	Point-MLP	93.0	**93.6**
PointSIMSIAM	Point-MLP elite	92.7	92.3
PointBYOL	Point-MLP elite	**93.3**	93.1

Table 3: *Shape classification on ModelNet40. 'Acc (%)' denotes overall accuracy, and 'Voting Acc (%)' indicates voting accuracy. The results are listed starting with autoencoder-based models, followed by models primarily using contrastive learning, and finally, the results of the proposed models are presented.*

Method	Backbone	OBJ ONLY	OBJ BG	PB T50 RS
Point-DAE	PointNet		80.2	
Point-DAE	DGCNN		92.1	
Point-BERT	Transformer	88.12	87.43	83.07
Point-MAE	Transformer	88.29	90.02	85.18
Point-M2AE	Transformer	88.81	91.22	86.43
MAE3D	DGCNN			83.21
Point-MA2E	PointNet		80.2	
Point-MA2E	DGCNN		92.1	
Point-MA2E	Transformer	**93.07**	**93.86**	**89.31**
ContrastMPCT	Transformer	90.42	90.15	85.50
PointSIMSIAM	PointNet	77.97	78.31	69.00
PointBYOL	PointNet	82.44	81.58	69.81
PointSIMSIAM	PointNet(w/o ST)	79.69	78.83	69.04
PointBYOL	PointNet(w/o ST)	80.89	79.00	68.98
PointSIMSIAM	DGCNN	88.81	88.98	81.57
PointBYOL	DGCNN	89.15	88.81	82.27
PointSIMSIAM	Transformer	85.71	86.57	81.57
PointBYOL	Transformer	88.64	89.33	84.73
PointSIMSIAM	Transformer(TM)	87.95	88.64	83.48
PointBYOL	Transformer(TM)	86.92	85.37	80.22
PointSIMSIAM	Point-MLP	90.19	92.08	87.95
PointBYOL	Point-MLP	90.71	**94.15**	**89.00**
PointSIMSIAM	Point-MLP elite	89.67	91.37	85.60
PointBYOL	Point-MLP elite	**91.74**	91.05	87.02

Table 4: *Accuracy (%) results of fine-tuning on ScanObjectNN classification. Models are categorised based on their pre-training methodology: first, models under the autoencoder methodology; then, state-of-the-art models following contrastive learning; and finally, the models proposed in this research.*

4.2.5. Segmentation

After pre-training on ShapeNet55 dataset, we finetune our model to perform part segmentation using the ShapeNetPart dataset [YKC*16]. Table 6 shows the results for part segmentation. The metric for evaluation is the mean intersection over union (mIoU). In this evaluation, we also consider the results from PointContrast method [XGG*20].

4.2.6. Retrieval Techniques

Retrieval performance is assessed through a L_2 distance matrix of ModelNet40 feature vectors, analyzing model performance via nearest neighbor (NN) and mean average precision (mAP) metrics. After pre-training the proposed network with the ShapeNet55 dataset, we compute the embedding for each test model in ModelNet40 dataset. We thus compute the distance matrix between test models against training models and simulate a retrieval scenario, where we finally compute the retrieval metrics. Table 7 shows the results for the retrieval experiment. In addition, Fig. 2 shoes some results of a retrieval task.

5. Conclusions

This study embarks on the exploration of novel methodologies for the processing and interpretation of 3D point cloud data, leveraging the principles of self-supervised learning to address the inherent challenges of 3D data analysis. By adopting Siamese network-based approaches, this research aims to mitigate the dependency on extensive, labeled datasets, a common bottleneck in the field of 3D data processing. The utility and effectiveness of these self-supervised learning techniques were rigorously assessed across a spectrum of evaluation methods, including linear probing and transfer learning through fine tuning. These methodologies have shown promising results, particularly in enhancing the performance of downstream tasks such as classification and segmentation within 3D point clouds.

The empirical evidence gathered through the experiments conducted as part of this study underscores the effectiveness of the Siamese network methodology for pre-training models specialized in 3D point cloud analysis. Notably, the proposed models demonstrated their capacity to exceed the benchmarks set by existing state-of-the-art techniques in self-supervised learning of 3D representations for specific tasks. Although masked autoencoders have been identified as the premier methodology for pre-training networks across a range of tasks, the findings from this research position contrastive learning, facilitated by Siamese networks, as a suitable and competitive pre-training strategy.

However, it is imperative to note that Siamese networks did not universally outperform the leading masked autoencoder models in every task, particularly in segmentation and classification tasks using the ModelNet40 dataset. Given ModelNet40's origin as

Method	Backbone	5w10s	5w20s	10w10s	10w20s
Point-DAE	PointNet	93.0 ± 3.7	94.9 ± 3.3	86.7 ± 5.8	92.1 ± 4.6
Point-DAE	DGCNN	96.7 ± 2.5	97.7 ± 1.6	93.0 ± 3.8	$\mathbf{95.6 \pm 2.6}$
Point-BERT	Transformer	94.6 ± 3.1	96.3 ± 2.7	91.0 ± 5.4	92.7 ± 5.1
Point-MAE	Transformer	96.3 ± 2.5	97.8 ± 1.8	92.6 ± 4.1	95.0 ± 3.0
Point-M2AE	Transformer	$\mathbf{96.8 \pm 1.8}$	98.3 ± 4.5	92.3 ± 4.5	95.0 ± 3.0
MAE3D	Transformer	95.2 ± 3.1	97.9 ± 1.6	91.1 ± 4.6	94.2 ± 3.8
ContrastMPCT	Transformer	96.5 ± 1.7	$\mathbf{98.5 \pm 1.7}$	$\mathbf{93.0 \pm 2.4}$	95.2 ± 2.0
Cover-tree	PointNet	63.2 ± 10.7	68.9 ± 9.4	49.2 ± 6.1	50.1 ± 5.0
Cover-tree	DGCNN	60.0 ± 8.9	65.7 ± 8.4	48.5 ± 5.6	53.0 ± 4.1
PointSIMSIAM	PointNet	90.8 ± 6.6	93.8 ± 5.1	84.3 ± 6.4	90.8 ± 5.1
PointBYOL	PointNet	93.3 ± 3.4	96.4 ± 1.9	87.9 ± 5.2	92.6 ± 4.5
PointSIMSIAM	PointNet(w/o ST)	93.2 ± 3.4	94.9 ± 3.2	87.5 ± 6.2	92.1 ± 4.1
PointBYOL	PointNet(w/o ST)	93.7 ± 3.0	94.9 ± 3.0	87.7 ± 5.9	92.0 ± 4.2
PointSIMSIAM	DGCNN	96.8 ± 2.3	$\mathbf{98.8 \pm 1.4}$	92.6 ± 4.5	95.3 ± 2.8
PointBYOL	DGCNN	$\mathbf{97.1 \pm 2.1}$	98.4 ± 0.9	92.3 ± 4.0	$\mathbf{95.5 \pm 2.8}$
PointSIMSIAM	Transformer	94.7 ± 3.1	97.0 ± 2.5	90.2 ± 5.2	94.3 ± 3.0
PointBYOL	Transformer	95.7 ± 3.6	98.4 ± 1.2	92.4 ± 4.5	95.1 ± 2.8
PointSIMSIAM	Transformer(TM)	95.5 ± 2.8	96.9 ± 1.9	91.2 ± 4.6	94.5 ± 2.4
PointBYOL	Transformer(TM)	93.8 ± 2.7	95.7 ± 2.5	88.9 ± 5.2	93.1 ± 4.0
PointSIMSIAM	Point-MLP	95.2 ± 3.0	98.3 ± 1.6	91.9 ± 4.8	94.9 ± 2.6
PointBYOL	Point-MLP	96.7 ± 2.3	98.0 ± 1.9	$\mathbf{92.6 \pm 4.1}$	95.1 ± 2.5
PointSIMSIAM	Point-MLP elite	95.1 ± 1.8	98.1 ± 1.8	92.0 ± 4.9	94.5 ± 3.0
PointBYOL	Point-MLP elite	95.8 ± 2.8	97.4 ± 2.3	91.7 ± 4.3	95.0 ± 2.5

Table 5: *Few-shot classification results on ModelNet40. Mean accuracy (%) and standard deviation (%) from 10 independent experiments are reported.*

Method	Backbone	$mIoU_I$
Point-DAE	PointNet	84.7
Point-DAE	DGCNN	85.9
Point-BERT	Transformer	85.6
Point-MAE	Transformer	86.1
Point-M2AE	Transformer	**86.5**
Point-MA2E	PointNet	84.8
Point-MA2E	DGCNN	86.0
Point-MA2E	Transformer	86.4
PointContrast	SR-UNet	85.1
ContrastMPCT	Transformer	86.2
PointSIMSIAM	PointNet	84.4
PointBYOL	PointNet	84.6
PointSIMSIAM	PointNet(w/o ST)	84.4
PointBYOL	PointNet(w/o ST)	84.3
PointSIMSIAM	DGCNN	85.3
PointBYOL	DGCNN	85.3
PointSIMSIAM	Transformer	85.6
PointBYOL	Transformer	85.6
PointSIMSIAM	Transformer(TM)	85.6
PointBYOL	Transformer(TM)	85.7
PointSIMSIAM	Point-MLP	**85.9**
PointBYOL	Point-MLP	**85.9**
PointSIMSIAM	Point-MLP elite	85.6
PointBYOL	Point-MLP elite	85.8

Table 6: *Part Segmentation on ShapeNetPart [YKC*16]. 'mIoU$_I$' denotes the model's mean IoU across all instances in the dataset. Results are categorised based on autoencoder models, contrastive models, and proposed models.*

Method	Backbone	NN	mAP
PointSIMSIAM	PointNet	60.4	26.7
PointBYOL	PointNet	73.8	31.2
PointSIMSIAM	PointNet(w/o ST)	76.8	36.9
PointBYOL	PointNet(w/o ST)	78.2	38.2
PointSIMSIAM	DGCNN	74.8	39.4
PointBYOL	DGCNN	75.2	39.2
PointSIMSIAM	Transformer	77.1	41.1
PointBYOL	Transformer	77.2	38.3
PointSIMSIAM	Transformer(TM)	73.7	37.9
PointBYOL	Transformer(TM)	68.1	32.2
PointSIMSIAM	Point-MLP	**78.8**	**45.1**
PointBYOL	Point-MLP	77.5	37.3
PointSIMSIAM	Point-MLP elite	77.2	42.8
PointBYOL	Point-MLP elite	77.9	39.6

Table 7: ***Retrieval results***. *'NN (%)' denotes the percentage of instances where the model accurately identified the nearest neighbour. 'mAP (%)' signifies the model's mean average precision in its retrieval tasks across various levels of recall.*

a dataset comprising simplistic CAD models, a marginal accuracy discrepancy of 0.4% from the apex state-of-the-art model did not detract significantly from the value of the pre-training assessment. Conversely, the performance exhibited by ScanObjectNN, a dataset derived from real-world object scans and indicative of practical applications, highlights the robustness and applicability of Siamese networks in scenarios that reflect real-world complexities.

A noteworthy distinction of Siamese networks is their profi-

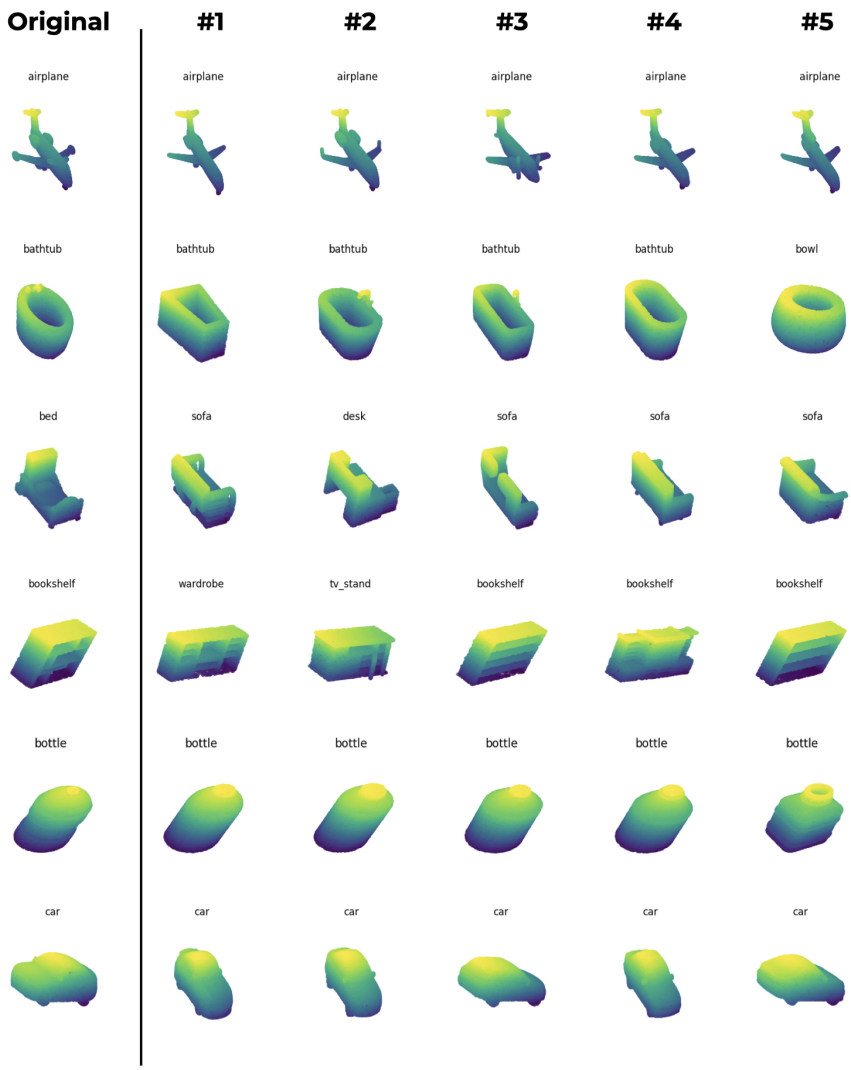

Figure 2: ***Visual Representation of retrieval****. Objects randomly selected from ModelNet are displayed alongside their 5 nearest elements in the feature space, according to output from PointSIMSIAM with Point-MLP encoder.*

ciency in few-shot learning scenarios, where models devoid of transformer architecture, such as DGCNN and Point-MLP, outshine their transformer-based counterparts. This success, achieved with comparatively simpler models, underscores the overarching objective of this study: to diminish reliance on extensive, labeled datasets necessitated by larger, more complex models.

This work demonstrates the feasibility and effectiveness of using Siamese networks for pre-training in 3D point cloud analysis, highlighting their significant role in enhancing encoder performance. Our comparative analysis shows that models pre-trained with Siamese networks outperform those trained solely on labeled data, suggesting that pre-training should be a standard practice before tackling downstream tasks. Additionally, combining pre-training strategies, even on models already pre-trained with autoencoders, can further enhance task performance when each approach's strengths are effectively leveraged.

As interest in Siamese networks for 3D point cloud processing grows, their potential to match or surpass the capabilities of leading models increases. This study marks a significant step towards that goal, offering new insights and methodologies for efficient learning in 3D point cloud analysis.

Acknowledgements

The work of Lucas Oyarzún and Ivan Sipiran has been partially supported by ANID Chile—Research Initiation Program - Grant 11220211.

References

[BDTD*16] BOJARSKI M., DEL TESTA D., DWORAKOWSKI D., FIRNER B., FLEPP B., GOYAL P., JACKEL L. D., MONFORT M., MULLER U., ZHANG J., ET AL.: End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316* (2016). 1

[BIS*23] BALESTRIERO R., IBRAHIM M., SOBAL V., MORCOS A., SHEKHAR S., GOLDSTEIN T., BORDES F., BARDES A., MIALON G., TIAN Y., SCHWARZSCHILD A., WILSON A. G., GEIPING J., GARRIDO Q., FERNANDEZ P., BAR A., PIRSIAVASH H., LECUN Y., GOLDBLUM M.: A cookbook of self-supervised learning, 2023. arXiv:2304.12210. 2

[CFG*15] CHANG A. X., FUNKHOUSER T., GUIBAS L., HANRAHAN P., HUANG Q., LI Z., SAVARESE S., SAVVA M., SONG S., SU H., ET AL.: Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012* (2015). 1

[CH21] CHEN X., HE K.: Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2021), pp. 15750–15758. 1

[DCS*17] DAI A., CHANG A. X., SAVVA M., HALBER M., FUNKHOUSER T., NIESSNER M.: Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2017), pp. 5828–5839. 1

[EGLH22] ERICSSON L., GOUK H., LOY C. C., HOSPEDALES T. M.: Self-supervised representation learning: Introduction, advances, and challenges. *IEEE Signal Processing Magazine 39*, 3 (2022), 42–62. 1

[GSA*20] GRILL J.-B., STRUB F., ALTCHÉ F., TALLEC C., RICHEMOND P., BUCHATSKAYA E., DOERSCH C., AVILA PIRES B., GUO Z., GHESHLAGHI AZAR M., ET AL.: Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems 33* (2020), 21271–21284. 1

[GWH*20] GUO Y., WANG H., HU Q., LIU H., LIU L., BENNAMOUN M.: Deep learning for 3d point clouds: A survey. *IEEE transactions on pattern analysis and machine intelligence 43*, 12 (2020), 4338–4364. 2

[HXZZ21] HUANG S., XIE Y., ZHU S.-C., ZHU Y.: Spatio-temporal self-supervised representation learning for 3d point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), pp. 6535–6545. 2, 4

[JLZ*23] JIANG J., LU X., ZHAO L., DAZALEY R., WANG M.: Masked autoencoders in 3d point cloud representation learning. *IEEE Transactions on Multimedia* (2023). 2, 4

[LH16] LOSHCHILOV I., HUTTER F.: Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983* (2016). 3

[LH17] LOSHCHILOV I., HUTTER F.: Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101* (2017). 3

[MHL*22] MEI G., HUANG X., LIU J., ZHANG J., WU Q.: Unsupervised point cloud pre-training via contrasting and clustering. In *2022 IEEE International Conference on Image Processing (ICIP)* (2022), IEEE, pp. 66–70. 2, 4

[MQY*22] MA X., QIN C., YOU H., RAN H., FU Y.: Rethinking network design and local geometry in point cloud: A simple residual mlp framework. *arXiv preprint arXiv:2202.07123* (2022). 2, 3

[PWT*22] PANG Y., WANG W., TAY F. E., LIU W., TIAN Y., YUAN L.: Masked autoencoders for point cloud self-supervised learning. In *European conference on computer vision* (2022), Springer, pp. 604–621. 3, 4

[QSMG17] QI C. R., SU H., MO K., GUIBAS L. J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2017), pp. 652–660. 2, 3

[QYSG17] QI C. R., YI L., SU H., GUIBAS L. J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems 30* (2017). 2

[SK20] SHARMA C., KAUL M.: Self-supervised few-shot learning on point clouds. *Advances in Neural Information Processing Systems 33* (2020), 7212–7221. 4

[VSP*17] VASWANI A., SHAZEER N., PARMAR N., USZKOREIT J., JONES L., GOMEZ A. N., KAISER Ł., POLOSUKHIN I.: Attention is all you need. *Advances in neural information processing systems 30* (2017). 3

[WJW*24] WU X., JIANG L., WANG P.-S., LIU Z., LIU X., QIAO Y., OUYANG W., HE T., ZHAO H.: Point transformer v3: Simpler, faster, stronger. In *CVPR* (2024). 2

[WLJ*22] WU X., LAO Y., JIANG L., LIU X., ZHAO H.: Point transformer v2: Grouped vector attention and partition-based pooling. In *NeurIPS* (2022). 2

[WSK*15] WU Z., SONG S., KHOSLA A., YU F., ZHANG L., TANG X., XIAO J.: 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2015), pp. 1912–1920. 1, 4

[WSL*19] WANG Y., SUN Y., LIU Z., SARMA S. E., BRONSTEIN M. M., SOLOMON J. M.: Dynamic graph cnn for learning on point clouds. *Acm Transactions On Graphics (tog) 38*, 5 (2019), 1–12. 2, 3

[WY22] WANG D., YANG Z.-X.: Self-supervised point cloud understanding via mask transformer and contrastive learning. *IEEE Robotics and Automation Letters 8*, 1 (2022), 184–191. 2, 4

[XGG*20] XIE S., GU J., GUO D., QI C. R., GUIBAS L., LITANY O.: Pointcontrast: Unsupervised pre-training for 3d point cloud understanding. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16* (2020), Springer, pp. 574–591. 5

[XHG*23] XIAO A., HUANG J., GUAN D., ZHANG X., LU S., SHAO L.: Unsupervised point cloud representation learning with deep neural networks: A survey. *IEEE Transactions on Pattern Analysis amp; Machine Intelligence 45*, 09 (sep 2023), 11321–11339. 2

[Yil20] YILDIRIM C.: A review of deep learning approaches to eeg-based classification of cybersickness in virtual reality. In *2020 IEEE international conference on artificial intelligence and virtual reality (AIVR)* (2020), IEEE, pp. 351–357. 1

[YKC*16] YI L., KIM V. G., CEYLAN D., SHEN I.-C., YAN M., SU H., LU C., HUANG Q., SHEFFER A., GUIBAS L.: A scalable active framework for region annotation in 3d shape collections. *ACM Transactions on Graphics (ToG) 35*, 6 (2016), 1–12. 5, 6

[YRW*21] YU X., RAO Y., WANG Z., LIU Z., LU J., ZHOU J.: Pointr: Diverse point cloud completion with geometry-aware transformers. In *Proceedings of the IEEE/CVF international conference on computer vision* (2021), pp. 12498–12507. 3

[YTR*22] YU X., TANG L., RAO Y., HUANG T., ZHOU J., LU J.: Point-bert: Pre-training 3d point cloud transformers with masked point modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), pp. 19313–19322. 2, 3, 4

[ZGG*22] ZHANG R., GUO Z., GAO P., FANG R., ZHAO B., WANG D., QIAO Y., LI H.: Point-m2ae: multi-scale masked autoencoders for hierarchical point cloud pre-training. *Advances in neural information processing systems 35* (2022), 27061–27074. 2, 4

[ZGJM21] ZHANG Z., GIRDHAR R., JOULIN A., MISRA I.: Self-supervised pretraining of 3d features on any point-cloud. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), pp. 10252–10263. 2, 4

[ZJJ*21] ZHAO H., JIANG L., JIA J., TORR P. H., KOLTUN V.: Point transformer. In *Proceedings of the IEEE/CVF international conference on computer vision* (2021), pp. 16259–16268. 2

[ZLL*22] ZHANG Y., LIN J., LI R., JIA K., ZHANG L.: Point-dae: Denoising autoencoders for self-supervised point cloud learning. *arXiv preprint arXiv:2211.06841* (2022). 2, 4

[ZLL*23] ZHANG Y., LIN J., LI R., JIA K., ZHANG L.: Point-ma2e: Masked and affine transformed autoencoder for self-supervised point cloud learning, 2023. arXiv:2211.06841. 2, 4

[ZWN*24] ZENG C., WANG W., NGUYEN A., XIAO J., YUE Y.: Self-supervised learning for point cloud data: A survey. *Expert Systems with Applications 237* (2024), 121354. 2