Shape Self-Correction for Unsupervised Point Cloud Understanding

Ye Chen[1*], Jinxian Liu[1,2*], Bingbing Ni[1,2†], Hang Wang[1,2], Jiancheng Yang[1], Ning Liu[1†], Teng Li[3], Qi Tian[4]
[1]Shanghai Jiao Tong University, Shanghai 200240, China
[2]Huawei Hisilicon, [3]Huawei Car BU, [4]Huawei Cloud & AI
{chenye123, liujinxian, nibingbing, wang--hang, jekyll4168, ningliu}@sjtu.edu.cn
liteng@ahu.edu.cn tian.qi1@huawei.com

Abstract

We develop a novel self-supervised learning method named Shape Self-Correction for point cloud analysis. Our method is motivated by the principle that a good shape representation should be able to find distorted parts of a shape and correct them. To learn strong shape representations in an unsupervised manner, we first design a shape-disorganizing module to destroy certain local shape parts of an object. Then the destroyed shape and the normal shape are sent into a point cloud network to get representations, which are employed to segment points that belong to distorted parts and further reconstruct them to restore the shape to normal. To perform better in these two associated pretext tasks, the network is constrained to capture useful shape features from the object, which indicates that the point cloud network encodes rich geometric and contextual information. The learned feature extractor transfers well to downstream classification and segmentation tasks. Experimental results on ModelNet, ScanNet and ShapeNet-Part demonstrate that our method achieves state-of-the-art performance among unsupervised methods. Our framework can be applied to a wide range of deep learning networks for point cloud analysis and we show experimentally that pre-training with our framework significantly boosts the performance of supervised models.

1. Introduction

3D shape understanding is in tremendous demand due to many important tasks like autonomous driving. Point cloud is a simple but effective representation of 3D data, which makes it popular for 3D vision analysis. With the help of extensive manually-labeled supervised information, many ingenious works [21, 22, 26, 40, 17, 42, 20, 14, 18] are proposed to directly consume point clouds and achieve remarkable performance on 3D vision tasks like classification, detection and segmentation. However, an enormous amount of 3D point cloud data has not been effectively utilized because of the expensive labeling. Hence utilizing these unlabeled data to perform effective representation learning is an important opportunity for 3D analysis.

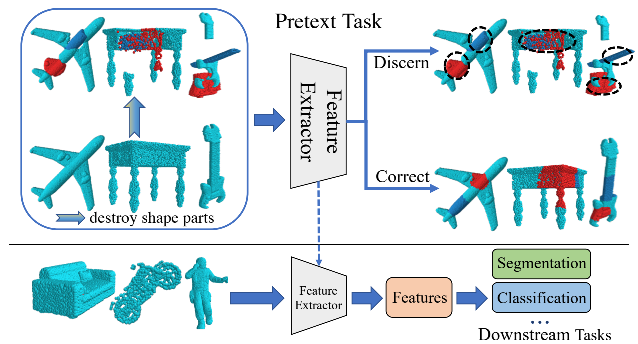

Figure 1: **Illustration of our main idea.** As shown, we destroy the shape parts with certain heuristic methods and there is a huge mismatch between the distorted parts and the normal parts. We can easily distinguish the distorted parts because we know the geometric characteristics of the object. Hence we think a strong representation which encodes effective structure information should also have the ability. We destroy shape parts and train a network to distinguish the destroyed parts and restore them to normal unsupervisedly in a pretext task. Success in the pretext task indicates the network captures strong shape representations, which can transfer well to downstream tasks.

Unsupervised learning on point clouds aims to learn useful information and representations from points without manually-labeled supervised information, which opens up the possibility to take advantage of unlabeled data. Several works focus on 3D unsupervised feature learning through using autoencoders and generative adversarial networks [2, 8, 15, 43, 47, 7]. Certain recently devoted self-supervised works [10, 25, 11, 1, 31] design target related

†Co-corresponding authors:Bingbing Ni, Ning Liu
*Equal contributions

pretext tasks to encourage the network to capture structural and low-level information. PointGLR [24] effectively captures the underlying high-level semantic knowledge through bidirectional reasoning between the local structures and the global shape and achieves superior performance on classification tasks. Nevertheless, PointGLR relies on the hierarchical local features and it is not suitable for networks like PointNet [21] and DGCNN [34]. The goal of our work is to explore a backbone-agnostic self-supervised framework that is capable of fully utilizing local structure of shape parts and boosting the performance of unsupervised learning.

Each 3D shape can be divided into several shape parts/primitives in an unsupervised manner and all the shape parts are closely related through geometric constraints. The geometric constraints reflect robust geometric characteristics and imply local structure information and semantic knowledge of the object. Hence, one can easily distinguish distorted parts of a shape if he knows the geometric structure of such shape. Motivated by such principle, we think that a good shape representation which encodes effective structural and semantic information should also have the ability to find distorted parts of a shape and correct them.

Inspired by such observations, we propose a self-supervised framework for learning strong representations of 3D shapes by destroying local parts of a 3D shape and encouraging the network to distinguish the destroyed shape parts and then restore them to normal. For success in this pretext task, the network is constrained to capture richer geometric and structural information of the 3D point cloud. The overview of our main idea is shown in Figure 1. Our proposed framework is agnostic of point-based networks like PointNet, KPConv [28], and RSCNN [20]. In this paper, we modify PointNet and RSCNN as our feature extractor respectively to evaluate our proposed method. We concatenate the normal shape and disorganized shape as the input of the backbone network during pre-training. With the features of the normal shape, accurate structure information is obtained so that the network is capable of performing well on the pretext tasks. In addition to the backbone network, our proposed framework has three other components , which can be summarized as: 1) Shape-disorganizing module: we design a cluster of heuristic methods to effectively destroy the geometric structure of normal shape parts; 2) Distinguishing Branch: we implement a point-wise classifier to segment points that belong to the distorted parts; 3) Restoring Branch: we also design a self-reconstruction module to correct the distorted shape based on the segmentation results of the Distinguishing branch. Notably,we propose an approach cluster in Shape-disorganizing module and a wide range of methods that destroy geometric structure of shape parts can be included in.

In this paper, we utilize the ShapeNet [3] dataset as our source set for self-supervised pre-training and evaluate the learned features on two important 3D understanding tasks, *i.e.*, shape classification and segmentation. Experimental results on several datasets indicate that our method achieves state-of-the-art performance among unsupervised models on both classification and segmentation tasks. Note that our model achieves remarkable performance on a real-world scanned dataset (ScanNet [4]), which demonstrates the transferability and robustness of learned features. We also show experimentally that pre-training with our framework significantly boosts the performance of supervised models. On the segmentation task, we also explore the effectiveness of the learned features in a semi-supervised setting and our method outperforms previous methods [39, 11] , especially when labels are most limited. In addition, our per-trained model achieves competitive results on downstream tasks when only using PointNet as the backbone network, which demonstrates the strong feature learning ability of our framework.

2. Related Work

Deep Learning on Point Cloud Understanding. PointNet [21] is a pioneering work to directly consume unordered and unstructured 3D point clouds, where MLPs and global max-pooling are utilized to obtain both point-wise features and global structure information. Despite PointNet well handles order invariances of input data and achieves strong performance, it fails to aggregate point-wise embeddings and capture local contextual information among points. PointNet++ [22] mitigates this issue by proposing a hierarchical learning architecture, where multi-scale local point embeddings are grouped. Several subsequent works [41, 9, 13, 32, 37, 33, 35] employ methods similar to CNNs to aggregate the contributions of neighbor points and capture local structure. All of the mentioned methods achieve remarkable performance on 3D point cloud understanding tasks with the help of labeled data. Our proposed self-supervised feature learning framework is suited for most of these methods and can learn strong representations without any human annotations.

Unsupervised Point Cloud Understanding. Unsupervised point cloud understanding aims to capture effective information from unlabeled point cloud data and utilize the learned features to handle downstream tasks. Classic methods perform unsupervised point cloud feature learning mainly based on auto-encoders [2, 5, 27, 47, 43] and generative adversarial networks [15, 2, 29]. Despite the promising performance on several specific tasks, these methods suffer from lacking local structural supervision, which limits the feature learning ability and transferability. Certain recent efforts focus on learning both structure information and semantic knowledge by defining pretext tasks [25, 11, 10, 24, 1]. RS [25] splits the shape into 3x3x3 voxels and trains the network to reconstruct the shape whose parts have been randomly rearranged by finding correct voxel assignment. The way RS uses to displace shape parts can be employed in our framework. However, RS restores the shape by simply rearranging shape parts according to predicted voxel assignment. Thus many methods that distort the shape do not apply to RS but they work well in our framework. PointGLR [24] explores high-level semantic knowledge contained in point clouds by bidirectional reasoning between local representations at different abstraction hierarchies in a network and global representation of the 3D object, which achieves extraordinary performance on classification tasks. Under this perspective, we propose a new scheme called Shape Self-Correction, which simultaneously employs local and global self-supervision and captures effective features that outperform other unsupervised methods on downstream tasks.

Point Cloud Denoising. Deep denoising approaches [6, 23, 45, 46] require pairs of clean and noisy point clouds, which in practice are produced by adding noise to original point clouds [12]. The formulation of our method is similar to point cloud denoising in that they both try to find and eliminate outliers. However, simply adding noise to the point cloud does not effectively alter geometric characteristics of the original shape. Thus the unsupervised model based on the denoising task is not able to extract effective geometric information of the object, which is demonstrated in Section 4.4. In contrast, our method destroys the geometric structure of shape parts and encourages the model to utilize geometric features to discern and restore the distortion. Through training with this pretext task, the network is constrained to capture useful structure information of the shape.

3. Methodology

To learn discriminative, robust and generalizable shape representations from unlabeled point cloud data and enhance the network's ability in 3D point cloud understanding, we propose a novel self-supervised framework named Shape Self-Correction. Our method enables the model to capture effective structural and contextual information by destroying the local shape parts and constraining the network to distinguish and restore them to normal.

3.1. Overview

Our framework contains a Shape-disorganizing module, a point cloud Encoder, a Distinguishing Branch D and a Restoring Branch R, as illustrated in Figure 2. Firstly, we disorganize the 3D shape and destroy the geometric structure of the normal shape. Then we use the encoder to generate features of both the normal shape and the disorganized one. The features are concatenated as the input of branch D and R to distinguish the disorganized shape parts and restore them to normal. Here, the features of the normal shape are utilized as the template to provide accurate structure information so that the network is capable of performing well on the pretext tasks, which enables the model to exploit effective features. Notably, Branch R does not utilize the results of Branch D as inputs so that we can arrange them in parallel.

Assume a shape $S = \{\boldsymbol{s_1}, \boldsymbol{s_2}, ..., \boldsymbol{s_N}\}$ is a point set with N points, the Shape-disorganizing module randomly samples two parts P, Q and then utilizes a combination of various approaches to distort the sampled parts. We define the points of distorted parts as incorrect points. The incorrect points together with the parts that are not selected form a new shape S^*. Intuitively, the new shape may not conform the geometric characteristics of the original shape. Considering the geometric characteristics explicitly represent the relationships among different shape parts and imply semantic knowledge of the shape, we design the Distinguishing branch D to seek out the incorrect points that break the geometric construction of the original shape, which encourages the model to better understand 3D shapes and learn effective structure and semantic information. Based on the distinguishing results, if the model is able to move the incorrect points to correct positions and restore the geometric characteristics of the normal shape, we can conclude that the model explores more fine-grained geometric and contextual features of input shapes. Hence the Restoring Branch R is designed to reconstruct input shapes. To succeed in such pretext task, the encoder is constrained to fully exploit useful shape information.

3.2. Shape Disorganizing

The Shape-disorganizing module is designed to destroy the geometric structure of input 3D shapes by disorganizing the shape parts. In our method, we design a method cluster to disorganize the input shape, including (1) randomly rotate the sampled part along X,Y or Z axis; (2) randomly translate sampled points to new positions; (3) randomly scale the sampled part; (4) crop the sampled part and replace it with a random sphere; and (5) exchange the coordinates of two sampled parts. For the input shape, this module randomly samples two shape parts and then randomly selects certain distortion approaches from the cluster to generate the disorganized shape. Specifically, from the input points $S = \{\boldsymbol{s_1}, \boldsymbol{s_2}, ..., \boldsymbol{s_N}\}$, we randomly select two center points $\boldsymbol{s_i} = (x_i, y_i, z_i)$ and $\boldsymbol{s_j} = (x_j, y_j, z_j)$. Following the grouping layer in PointNet++, we employ ball query to sample two clusters of points $P = \{\boldsymbol{p_1}, \boldsymbol{p_1}, ..., \boldsymbol{p_K}\}$ and $Q = \{\boldsymbol{q_1}, \boldsymbol{q_2}, ..., \boldsymbol{q_K}\}$ from S, where all points in P/Q are within a radius to $\boldsymbol{s_i}$/$\boldsymbol{s_j}$ (an upper limit of K is set in our implementation). $S^{'} = S \setminus \{P \cup Q\}$ denotes the point set that is not sampled. For the sampled parts P and Q, a combination of distortion approaches is utilized to generate distorted versions P^* and Q^*. Then the new shape S^* can be expressed as $S^* = S^{'} \cup P^* \cup Q^*$, which denotes

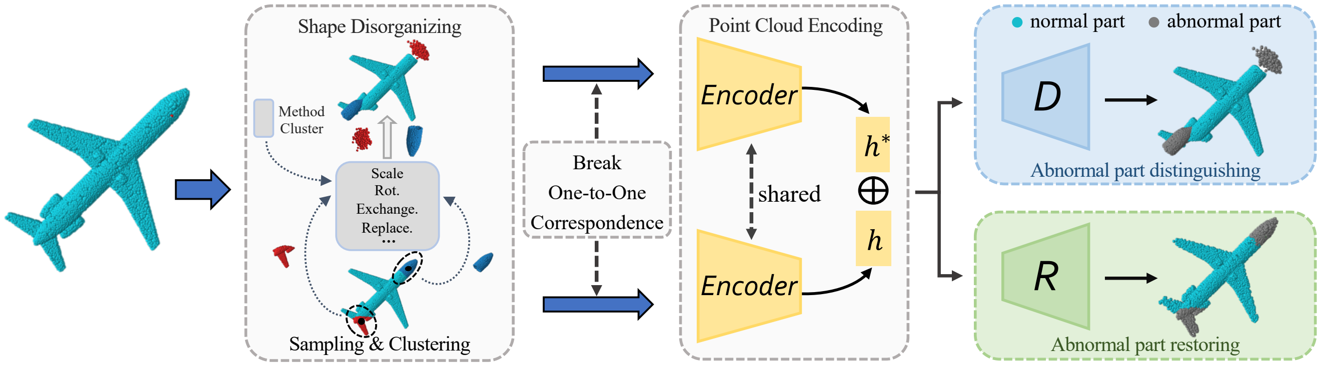

Figure 2: **Framework of the proposed self-supervised method Shape Self-Correction.** The framework consists of a shape-disorganizing module, a point cloud encoding network and two task-related branches. We design a cluster of methods to distort shape parts. Abnormal part distinguishing branch and abnormal part restoring branch are designed to segment points that belong to destroyed parts and restore the disorganized shape to normal respectively.

the disorganized shape that breaks the original geometric structure. The visual examples of normal shapes and disorganized shapes are shown in Figure 3.

As shown in Figure 2, to encourage the network to better understand the geometric characteristics of the correct shape, we employ the original shape as a template and the encoder extracts high-dimensional features of both the new shape and original shape. Intuitively, if the two shapes have a point-to-point correspondence, the Distinguishing Branch tends to learn point transformation and gives trivial solutions. To avoid such correspondence in coordinates, we use random sampling to choose two subsets of points $T = \{\boldsymbol{t_1}, \boldsymbol{t_2}, ..., \boldsymbol{t_{N'}}\}$, $T^* = \{\boldsymbol{t_1^*}, \boldsymbol{t_2^*}, ..., \boldsymbol{t_{N'}^*}\}$ from S and S^* respectively, where $N' = N/2$. Moreover, we perform simple random data augmentation on both T and T^* for the purpose of better representation learning, which further breaks the point-to-point correspondence between normal shapes and disorganized shapes. In the meanwhile, Shape-disorganizing module generates pseudo-labels for T^*. We express it as $\mathcal{Y} = \{y_1, y_2, ..., y_{N'}\}$ such that $y_i \in \{0, 1\}$, where $y_i = 1$ means the corresponding point belongs to distorted parts (*i.e.*, P^* and Q^*). The output of Shape-disorganizing can be expressed as a tuple $\boldsymbol{t} = [T, T^*, \mathcal{Y}]$.

3.3. Point Cloud Encoding

Any learning-based network that takes point clouds as the input and outputs high-dimensional features can be utilized as the encoder of Shape Self-Correction. In our implementation, we employ RSCNN and PointNet as the encoder that maps input point sets from Euclidean space $\mathbb{R}^{n\times 3}$ into the latent space $\mathcal{Z} \in \mathbb{R}^{n\times d}$. Specifically, for each shape T^*, the encoder extracts its point-wise features $\boldsymbol{l^*} \in \mathbb{R}^{n\times d_l}$ and global feature $\boldsymbol{g^*} \in \mathbb{R}^{1\times d_g}$ to encode richer local and global information than the original space. When using PointNet as the encoder, global and point-wise features are defined

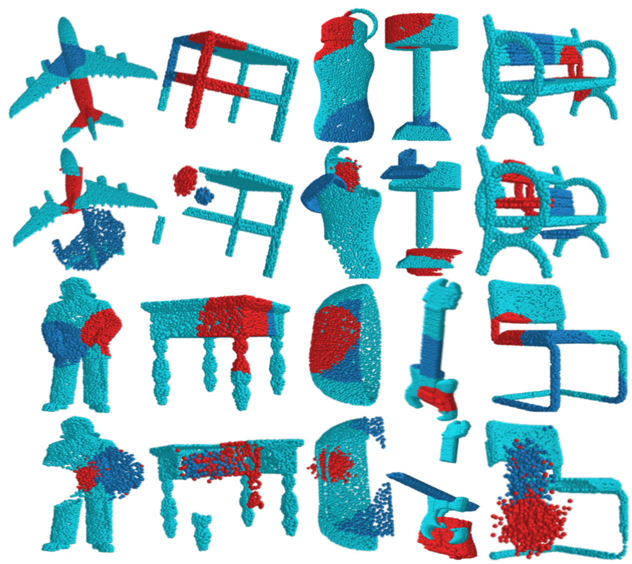

Figure 3: Visualization of normal shapes (row 1 and 3) and abnormal shapes disorganized by Shape Disorganizing module (row 2 and 4).

the same as proposed in [21]. For RSCNN [20], we utilize the architecture for classification (single-scale neighborhood version) as our backbone and generate point-wise features by attaching certain feature propagation layers. For the purpose of guiding the network to correctly discern those disorganized parts, we also extract the global feature $\boldsymbol{g}$ of the original shape T. The concatenation of $\boldsymbol{g}$, $\boldsymbol{g^*}$ and $\boldsymbol{l^*}$ is fed into the Distinguishing Branch D and Restoring Branch R simultaneously. Through the task of discerning the disorganized parts and restoring the original shape, the encoder is encouraged to generate strong shape representations that facilitate high-quality classification, segmentation, and other 3D point cloud understanding tasks.

3.4. Abnormal Part Distinguishing

For a disorganized shape, the task of distinguishing the parts that make the shape violate the geometric construction enables the model to better understand 3D shapes and capture more effective shape features. Hence the Distinguishing Branch is designed to seek out all incorrect points of the disorganized shape. We formulate the task as a point-wise classification. This task is defined as $\mathcal{F}_\zeta : \mathcal{Z} \in \mathbb{R}^{N' \times d} \mapsto \mathcal{Y} \in \mathbb{R}^{N' \times 2}$, which maps the high-dimensional features extracted by the point cloud encoder into predicted categories, *i.e.*, the corresponding point belongs to distorted parts or not. In our method, we use RSCNN/PointNet as the encoder, we concatenate the global features $\boldsymbol{g}^*, \boldsymbol{g} \in \mathbb{R}^{1 \times d_g}$ and the point-wise features $\boldsymbol{l}^* \in \mathbb{R}^{N' \times d_l}$ as the input of this branch. The classification is formed by several MLP layers. The output of Distinguishing Branch is denoted as $\hat{\mathcal{Y}} = \{\hat{y_1}, \hat{y_2}, ..., \hat{y_{N'}}\}$, where $\hat{y_i}$ represents the probability distribution formulated by softmax function.

3.5. Abnormal Part Restoring

Paralleled with Distinguishing Branch, we develop a Restoring Branch and encourage the model to restore the original shape, which constrains the encoder to capture more contextual and geometric information contained in point clouds. Thus the Restoring Branch is naturally designed to move the incorrect points to original locations. We formulate the task as reconstruction. We define the function of Restoring Branch as $\mathcal{R}_\phi : \mathcal{Z} \in \mathbb{R}^{N' \times d} \mapsto \mathcal{P} \in \mathbb{R}^{N' \times 3}$. Through decoding the high-dimensional features extracted by the encoder, the Restoring Branch performs point-wise displacement prediction and tries to output a point cloud $\bar{T}$ as similar as possible to the original point set T by the function $\mathcal{R}$. Here, we use Chamfer Distance (CD) to measure the distance between the reconstructed $\bar{T}$ and the original T. The Chamfer Distance is often applied as the cost of the reconstruction task, which finds the nearest neighbour of each point and computes their Euclidean distance in a bidirectional way between two point sets. In our method, considering the disorganized parts dominate the performance of reconstruction, we modify the Chamfer Distance and attach larger weights to the predicted incorrect points than the correct ones, which is written as:

$$\mathcal{L}_c = \sum_{p \in T} \lambda_{\bar{p}} \min_{\bar{p} \in \bar{T}} \|p - \bar{p}\|_2^2 + \sum_{\bar{p} \in \bar{T}} \lambda_{\bar{p}} \min_{p \in T} \|p - \bar{p}\|_2^2, \quad (1)$$

where $\lambda_{\bar{p}}$ denotes the weight attached to each point in the reconstructed set. Here, we set $\lambda_{\bar{p}} \in \{0.5, 1.0\}$, where $\lambda_{\bar{p}}$ is set to 0.5 and 1.0 for points that belong to normal and distorted parts respectively.

To accurately restore the coordinates of incorrect points, the point-wise local features $\boldsymbol{l}^*$ and global feature $\boldsymbol{g}^*$ are utilized because features of the correct points are favorable for the network to exploit the point relation information and then find proper locations of incorrect points. The same as Distinguishing Branch, we employ the global feature of the original shape $\boldsymbol{g}$ as a template. Thus the input of Restoring Branch is the concatenation of $\boldsymbol{l}^*$, $\boldsymbol{g}^*$ and $\boldsymbol{g}$. The output is a reconstructed point set $\bar{T} \in \mathbb{R}^{N' \times 3}$.

3.6. Objective Function

The Distinguishing Branch is trained by classical cross-entropy loss and supervised by the pseudo-labels $\mathcal{Y} = \{y_1, y_2, ..., y_{N'}\}$, which is written as:

$$\mathcal{L}_s = -\frac{1}{N'} \sum_{i=1}^{N'} y_i \log \hat{y_i}, \quad (2)$$

where $y_i \in \mathcal{Y}$ and $\hat{y_i}$ denotes the output probability distribution formulated by softmax function. We train the Restoring Branch with a modified Chamfer Distance Loss as formulated in Equation (1).

The two branches are jointly optimized and the overall objective function of Shape Self-Correction scheme is a combination of two losses:

$$\mathcal{L} = \mathcal{L}_s + \beta \mathcal{L}_c, \quad (3)$$

where β is used to balance contributions of the two terms such that two branches contribute equally to the whole network.

Our common goal is to encourage the encoder to learn more discriminative shape features through training it with the Shape Self-Correction tasks. We define the encoder as $\mathcal{E}_\theta : \mathcal{P} \in \mathbb{R}^{N' \times 3} \mapsto \mathcal{Z} \in \mathbb{R}^{N' \times d}$ and any parametric non-linear function parameterized by θ can be used as the encoder. Hence the optimal problem of Shape Self-Correction can be expressed as:

$$\min_{\{\theta, \zeta, \phi\}} \mathcal{L}_s + \beta \mathcal{L}_c. \quad (4)$$

After optimization, the encoder generates more effective features and performs better on specific downstream tasks like shape classification and segmentation.

4. Experiments

In this section, we evaluate the proposed Shape Self-Correction framework qualitatively on two of the most important 3D tasks, *i.e.*, classification and segmentation. Specifically, the encoder trained with Shape Self-Correction scheme can be used as a pre-trained model for the two downstream tasks. Our framework is general and we modify PointNet and RSCNN as our encoder respectively. For training and evaluation regarding the tasks, we use multiple benchmark datasets, *i.e.*, ShapeNet [3], ShapeNetPart [44], ModelNet [38] and ScanNet [4].

4.1. Experimental setups

Datasets. ShapeNet [3] contains more than 50,000 3D shapes across 55 categories of man-made objects. ShapeNetPart dataset [44] contains 16,681 objects from 16 categories of ShapeNet dataset. Each category contains 2-6 parts and there are 50 parts in total. ModelNet dataset [38] has two variants, *i.e.*, ModelNet40 and ModelNet10, comprising 9832/3991 training objects and 2468/908 test objects in 40 and 10 classes respectively. ScanNet [4] contains 1513 scanned and reconstructed real-world indoor scenes. We follow the practice in [17, 24] to obtain point clouds from ScanNet according to the semantic voxel labels, which contain 17 categories.

Evaluation Metrics. For the classification task on ModelNet and ScanNet, we use the classification accuracy as the metric. On ShapeNetPart dataset, we evaluate our scheme with part classification accuracy and mean Intersection-over-Union (mIoU). For each sample, IoU is computed for each part that belongs to that object category. The mean of all part IoUs is regarded as the IoU for that sample.

Model Pre-Training. Following the experimental protocol introduced in [2], we pre-train the encoder with our proposed scheme across all categories of the ShapeNet dataset, and then transfer the pre-trained model to the downstream tasks (*i.e.*, classification on ModelNet&ScanNet and part segmentation on ShapeNetPart). We take PointNet and RSCNN as our backbone. The Shape-disorganizing module, Distinguishing Branch and Restoring Branch are all discarded and only the encoder is used in downstream tasks. During pre-training, each shape in ShapeNet is sampled to 2048 points initially. The Shape-disorganizing module samples two clusters of points from the input point set as stated in Section 3.2 and we set the upper limit number of part points K to 256. After disorganizing the input shape, we sample the new point set to 1024 points to weaken the point-to-point correspondence between the new shape and the original one. During pre-training, adam optimizer is used. The learning rate is set to 0.001 and the loss weight coefficient β for $\mathcal{L}_c$ is set to 4.0. Notably, only 3D coordinates are used during self-supervised training.

4.2. Shape Classification

To evaluate the performance of the Shape Self-Correction scheme on shape feature learning, we first conduct transfer experiments from ShapeNet to ModelNet/ScanNet dataset. Following [2, 11], we extract the shape features of the ModelNet/ScanNet samples with the pre-trained model without any parameter fine-tuning. Then we train a linear SVM on the embeddings of ModelNet/ScanNet train split and report the classification accuracy on the ModelNet/ScanNet test split. Each point cloud contains 1024 points and we only use the coordinates as the input. Results on ModelNet/ScanNet are shown in Table 1&2 ("Unsupervised Transfer Learning"). To perform fair comparisons, we reproduce PointGLR [24] without using annotated normal information as unsupervised signals. Our method achieves competitive results when only using PointNet as the encoder. When utilizing RSCNN, our method outperforms all previous unsupervised counterparts and the results on ModelNet are comparable to certain

Supervision	Method	MN10(%)	MN40(%)
Supervised Learning	PointNet [21]	-	89.2
	PointNet++ [22]	-	90.7
	SpecGCN [30]	-	91.5
	DGCNN [34]	-	92.2
	DensePoint [19]	-	92.8
Unsupervised Transfer Learning	3D-GAN [36]	91.0	83.3
	FoldingNet [43]	94.4	88.4
	MAP-VAE [10]	94.8	90.2
	Multi-task [11]	-	89.1
	MT-PointNet [11]	-	86.2
	RS-PointNet [25]	91.6	87.3
	RS-DGCNN [25]	94.5	90.6
	GLR-RSCNN [24]	94.2	91.3
	Ours-PointNet	93.3	89.9
	Ours-RSCNN	95.0	**92.4**
Supervised Fine-Tuning	RI-PointNet	93.2	89.1
	Ours-PointNet	93.9**(+0.7)**	90.0(+0.9)
	RI-RSCNN	94.8	91.7
	GLR-RSCNN [24]	94.8(+0.0)	92.2(+0.5)
	Ours-RSCNN	**95.5(+0.7)**	**93.0(+1.3)**

Table 1: **Shape Classification Results on ModelNet.** Results of both supervised and unsupervised models are reported. "Unsupervised Transfer Learning" denotes the parameters of the pre-trained models are fixed on downstream tasks, while "Supervised Fine-Tuning" denotes the pre-trained models are fine-tuned on target tasks. "RI" denotes the model is trained on target dataset from scratch. Our results are measured without using tricks like voting.

Supervision	Method	Acc.%	Inc.%
Unsupervised Transfer	GLR-RSCNN [24]	88.1	-
	Ours-PointNet	84.2	-
	Ours-RSCNN	**89.0**	-
Supervised Fine-Tuning	RI-PointNet	87.8	-
	Ours-PointNet	89.7	+1.9
	RI-RSCNN	90.1	-
	GLR-RSCNN [24]	90.8	+0.7
	Ours-RSCNN	**92.9**	**+2.8**

Table 2: **Shape Classification Results on ScanNet.** The classification accuracy of our method and the state-of-the-art unsupervised method are reported. "RI" denotes the model is trained on ScanNet from scratch. We also list the increments of pre-training.

fully-supervised models. Since the pre-training of the encoder and the training of the SVM are based on different datasets, the results imply the strong transferability of our framework, which is regarded as a significant application of self-supervised representation learning. Notably, ShapeNet is a synthetic dataset sampled from CAD models and ScanNet is a scanned real-world dataset, the domain gap between these two datasets is considered to be large. Thus the superior performance on ScanNet further demonstrates that our model generalizes well to unseen categories and the learned features are robust and generic.

As stated in Section 2, RS [25] also disorganizes the shape and discerns the incorrect points. However, our method is motivated to offer a pipeline to destroy the geometric structure of shape parts and then distinguish and restore the distortion. We utilize a cluster of approaches to distort shape parts, which do not apply to RS. Also, we employ the features of the original shape as the template to facilitate feature learning. The Restoring Branch also contributes a lot for training the encoder, thus our method outperforms RS by a large margin.

Supervised Fine-Tuning. We think the most important application of self-supervised learning is to make full use of abundant unlabeled data and boost the performance of supervised methods. Following [39], we employ the supervised fine-tuning strategy to evaluate the effectiveness of our proposed Shape Self-Correction. Specifically, we pre-train the model with our framework and fine-tune the weights on downstream tasks and compare the results with the randomly initialized model (not pre-trained). Under this perspective, we conduct extensive experiments on ModelNet/ScanNet and the results are also shown in Table 1&2 ("Supervised Fine-Tuning"). Note that pre-training with PointGLR [24] slightly benefits the supervised tasks while our method significantly boosts the performance, especially on ScanNet. Pre-training with our framework can be utilized as a strong initializer for supervised models.

4.3. Part Segmentation

Shape part segmentation is formed as a fine-grained point-wise classification task to predict the part category label of each point in a given object. Hence we explore the learned point-wise embeddings through such task. In this section, we evaluate the learned features on ShapeNetPart dataset and report part classification accuracy and mIoU.

Following [47, 11], we first conduct the shape segmentation experiments in a semi-supervised manner, *i.e.*, we randomly sample 1% and 5% of the ShapeNetPart train set as training data. We use the pre-trained model to extract the point features of all samples **without any parameter fine-tuning**, and then train a 4-layer MLP-based [2048,4096,1024,50] classifier on the sampled training set. The evaluation is conducted on the whole test set.

Model	1% of train data		5% of train data	
	Accuracy	**IoU**	**Accuracy**	**IoU**
SO-Net [16]	78.0	64.0	84.0	69.0
PointCapsNet [47]	85.0	67.0	86.0	70.0
Multi-task [11]	88.6	68.2	93.7	77.7
Ours-PointNet	84.9	69.7	88.1	74.0
Ours-RSCNN	**89.8**	**74.1**	**94.3**	**80.1**

Table 3: **Shape part segmentation results without fine-tuning.** Part classification accuracy and Ins.mIoU on ShapeNetPart datast are reported. All compared methods are evaluated in a semi-supervised manner (*i.e.*, 1%, 5% of training data is sampled), where the parameters of pre-trained models are **fixed**.

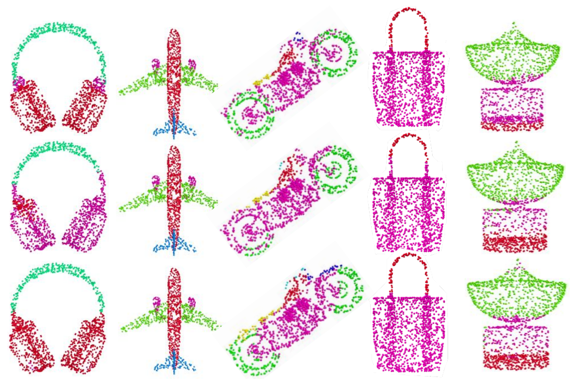

Figure 4: **The segmentation results on ShapeNetPart dataset.** Row 1: Ground Truth. Row 2/3: Results predicted by the model trained on 1%/5% data with encoder **fixed**.

The results are shown in Table 3. Our method significantly outperforms other unsupervised models, which shows that our pre-trained model captures more effective point embeddings that transfer well to segmentation tasks. Especially when using only 1% of training data, our RSCNN model outperforms all previous methods by a large margin. Considering Multi-Task [11] employs a heavier graph-based backbone, our PointNet model is also competitive. The results demonstrate that, through pre-training with the proposed Shape Self-Correction scheme, a very small number of labelled samples are sufficient to achieve strong performance on the downstream task. Some results are visualized in Figure 4. Despite the training data is limited, our model segments the fine-grained details well.

Supervised Fine-Tuning. The shape segmentation experiments under supervised fine-tuning strategy are also conducted. We report mIoU under several training-data sampling strategies (*i.e.*, 1%, 5%, 100%) and make comparisons with PointContrast [39] in Table 4. As shown, our RSCNN model fine-tuned on 5% labeled samples achieves a Ins.mIoU that is only 3.9% less than the fully-supervised model trained from scratch. Compared to the randomly initialized model, our pre-trained model achieves remarkable performance improvements, especially when only 1%

Model	IoU (1%)	IoU (5%)	IoU (100%)
PContrast (RI) [39]	71.8	79.3	84.7
PContrast (FT) [39]	74.0 (+2.2)	79.9 (+0.6)	85.1 (+0.4)
Ours-PointNet (RI)	68.6	76.9	83.2
Ours-PointNet (FT)	72.9 **(+4.3)**	78.5 **(+1.6)**	84.1 **(+0.9)**
Ours-RSCNN (RI)	71.6	79.4	84.3
Ours-RSCNN (FT)	**74.3** (+2.7)	**80.4** (+1.0)	**85.2 (+0.9)**

Table 4: **Shape part segmentation results with fine-tuning strategy**. "RI" denotes the model is not pre-trained. "FT" denotes the model is pre-trained with the corresponding unsupervised scheme and fine-tuned on target task.

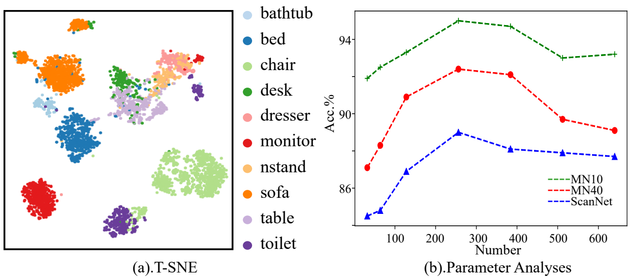

Figure 5: (a). T-SNE visualization of shape representations of ModelNet10 test data. (b). Parameter analyses on the point number of distorted parts.

labelled data is obtained (+4.3% for PointNet and +2.7% for RSCNN). We can conclude that pre-training with our framework on unlabeled data significantly boosts the performance and can be regarded as a strong initializer for supervised models, especially when labelled data is limited, which is a critical application of self-supervised learning.

4.4. Ablation Study

In this section, we explore the crucial components and hyper-parameters of Shape Self-Correction. All the experiments in this section are conducted on ModelNet40 dataset and we fix the encoder (RSCNN) after pre-training.

Branch D	**Branch R**	**Aug.& Down-sample**	**Temp Feature**	**Acc. %**
	√	√	√	88.1
√		√	√	89.3
√	√		√	90.9
√			√	88.0
	√		√	87.6
√	√	√		87.8
√	√	√	√	**92.4**

Table 5: **Component analyses**. Accuracy results on ModelNet40 are shown.

Component Analyses. We first conduct ablation study to investigate the effectiveness of each branch in Shape Self-Correction. We remove the corresponding loss when investigating the effect of such branch. Besides, we perform points down-sampling and data augmentation to break the coordinate correspondence. Hence we also conduct experiments to explore the effectiveness of such operations.

The results shown in Table 5 indicate that the Distinguishing Branch plays a more important role than the Restoring Branch, while Restoring Branch can further improve performance. We also compare the ablated version without features from the template shape and the accuracy degrades to 87.8%, which convincingly verifies the effectiveness of utilizing the features of original shapes.

A second experiment is conducted to explore how the approach cluster in shape-disorganizing module affects the performance of the scheme. The results are shown in Table 6. As shown, exchanging and replacing points are the most important distortion methods. Notably, our method achieves competitive performance by only randomly translating and rotating sampled parts. We also generate abnormal objects by only adding noise to the original shapes and the accuracy degrades to 87.2%, which proves the importance of altering geometric structure on the pre-task as illustrated in Section 2.

Rot.	**Trans.**	**Scale.**	**Exchange.**	**Replace.**	**Acc.%**
√	√				89.2
√	√	√			89.5
			√		91.1
				√	90.7
			√	√	92.0
√	√	√	√	√	**92.4**

Table 6: **Effectiveness of the distortion approaches.** Accuracy results on ModelNet40 are shown.

Parameter Analyses. We also explore how the number of incorrect points (*i.e.*, the hyper-parameter K as stated in Section 3.2) affects the performance of the model. The results are shown in Figure 5(b). We can observe that good performance is achieved when K is set to 256. No obvious improvements show up when further increasing K.

5. Conclusion

We propose an unsupervised framework for point cloud analysis named Shape Self-Correction. Experimental results on various datasets demonstrate that our method transfers well to downstream tasks and achieves state-of-the-art performance among unsupervised methods. Notably, Shape Self-Correction can be regarded as a pipeline and we provide a simple and effective implementation. For future directions, we are intending to explore more effective approaches to distort the shape parts and extend our scheme to more scenarios like point cloud completion.

Acknowledgment This work was supported by National Science Foundation of China (U20B2072, 61976137). This work was also supported by NSFC (U1908210).

References

[1] Idan Achituve, Haggai Maron, and Gal Chechik. Self-supervised learning for domain adaptation on point clouds. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 123–133, 2021.

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

[3] Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. ShapeNet: An Information-Rich 3D Model Repository. Technical Report arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago, 2015.

[4] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5828–5839, 2017.

[5] Haowen Deng, Tolga Birdal, and Slobodan Ilic. Ppf-foldnet: Unsupervised learning of rotation invariant 3d local descriptors. In *ECCV*, pages 602–618, 2018.

[6] Chaojing Duan, Siheng Chen, and Jelena Kovacevic. 3d point cloud denoising via deep neural network based local surface estimation. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8553–8557. IEEE, 2019.

[7] Gil Elbaz, Tamar Avraham, and Anath Fischer. 3d point cloud registration for localization using a deep neural network auto-encoder. In *CVPR*, pages 4631–4640, 2017.

[8] Matheus Gadelha, Rui Wang, and Subhransu Maji. Multiresolution tree networks for 3d point cloud processing. In *ECCV*, pages 103–118, 2018.

[9] Benjamin Graham, Martin Engelcke, and Laurens Van Der Maaten. 3d semantic segmentation with submanifold sparse convolutional networks. In *CVPR*, pages 9224–9232, 2018.

[10] 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-ction and half-to-half prediction. In *ICCV*, pages 10441–10450. IEEE, 2019.

[11] Kaveh Hassani and Mike Haley. Unsupervised multi-task feature learning on point clouds. In *ICCV*, pages 8160–8171, 2019.

[12] Pedro Hermosilla, Tobias Ritschel, and Timo Ropinski. Total denoising: Unsupervised learning of 3d point cloud cleaning. In *ICCV*, pages 52–60, 2019.

[13] Pedro Hermosilla, Tobias Ritschel, Pere-Pau Vázquez, Àlvar Vinacua, and Timo Ropinski. Monte carlo convolution for learning on non-uniformly sampled point clouds. *ACM Transactions on Graphics (TOG)*, 37(6):1–12, 2018.

[14] Qiangui Huang, Weiyue Wang, and Ulrich Neumann. Recurrent slice networks for 3d segmentation of point clouds. In *CVPR*, pages 2626–2635, 2018.

[15] Chun-Liang Li, Manzil Zaheer, Yang Zhang, Barnabás Póczos, and Ruslan Salakhutdinov. Point cloud GAN. In *Deep Generative Models for Highly Structured Data, ICLR 2019 Workshop, New Orleans, Louisiana, United States, May 6, 2019*, 2019.

[16] Jiaxin Li, Ben M Chen, and Gim Hee Lee. So-net: Self-organizing network for point cloud analysis. In *CVPR*, pages 9397–9406, 2018.

[17] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. Pointcnn: Convolution on x-transformed points. In *Advances in neural information processing systems*, pages 820–830, 2018.

[18] Jinxian Liu, Bingbing Ni, Caiyuan Li, Jiancheng Yang, and Qi Tian. Dynamic points agglomeration for hierarchical point sets learning. In *ICCV*, pages 7546–7555, 2019.

[19] Yongcheng Liu, Bin Fan, Gaofeng Meng, Jiwen Lu, Shiming Xiang, and Chunhong Pan. Densepoint: Learning densely contextual representation for efficient point cloud processing. In *ICCV*, pages 5239–5248, 2019.

[20] Yongcheng Liu, Bin Fan, Shiming Xiang, and Chunhong Pan. Relation-shape convolutional neural network for point cloud analysis. In *CVPR*, pages 8895–8904, 2019.

[21] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*, pages 652–660, 2017.

[22] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in neural information processing systems*, pages 5099–5108, 2017.

[23] Marie-Julie Rakotosaona, Vittorio La Barbera, Paul Guerrero, Niloy J Mitra, and Maks Ovsjanikov. Pointcleannet: Learning to denoise and remove outliers from dense point clouds. In *Computer Graphics Forum*, volume 39, pages 185–203. Wiley Online Library, 2020.

[24] Yongming Rao, Jiwen Lu, and Jie Zhou. Global-local bidirectional reasoning for unsupervised representation learning of 3d point clouds. In *CVPR*, pages 5376–5385, 2020.

[25] Jonathan Sauder and Bjarne Sievers. Self-supervised deep learning on point clouds by reconstructing space. In *Advances in Neural Information Processing Systems*, pages 12962–12972, 2019.

[26] Hang Su, Varun Jampani, Deqing Sun, Subhransu Maji, Evangelos Kalogerakis, Ming-Hsuan Yang, and Jan Kautz. Splatnet: Sparse lattice networks for point cloud processing. In *CVPR*, pages 2530–2539, 2018.

[27] Yongbin Sun, Yue Wang, Ziwei Liu, Joshua Siegel, and Sanjay Sarma. Pointgrow: Autoregressively learned point cloud generation with self-attention. In *The IEEE Winter Conference on Applications of Computer Vision*, pages 61–70, 2020.

[28] 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 *ICCV*, pages 6411–6420, 2019.

[29] Diego Valsesia, Giulia Fracastoro, and Enrico Magli. Learning localized generative models for 3d point clouds via graph convolution. In *7th International Conference on Learning

Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019. OpenReview.net, 2019.

[30] Chu Wang, Babak Samari, and Kaleem Siddiqi. Local spectral graph convolution for point set feature learning. In *ECCV*, pages 52–66, 2018.

[31] Peng-Shuai Wang, Yu-Qi Yang, Qian-Fang Zou, Zhirong Wu, Yang Liu, and Xin Tong. Unsupervised 3d learning for shape analysis via multiresolution instance discrimination. *ACM Trans. Graphic*, 2020.

[32] Shenlong Wang, Simon Suo, Wei-Chiu Ma, Andrei Pokrovsky, and Raquel Urtasun. Deep parametric continuous convolutional neural networks. In *CVPR*, pages 2589–2597, 2018.

[33] Weiyue Wang, Ronald Yu, Qiangui Huang, and Ulrich Neumann. Sgpn: Similarity group proposal network for 3d point cloud instance segmentation. In *CVPR*, pages 2569–2578, 2018.

[34] 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.

[35] Yida Wang, David Joseph Tan, Nassir Navab, and Federico Tombari. Softpoolnet: Shape descriptor for point cloud completion and classification. In *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part III*, pages 70–85, 2020.

[36] Jiajun Wu, Chengkai Zhang, Tianfan Xue, Bill Freeman, and Josh Tenenbaum. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In *Advances in neural information processing systems*, pages 82–90, 2016.

[37] Wenxuan Wu, Zhongang Qi, and Li Fuxin. Pointconv: Deep convolutional networks on 3d point clouds. In *CVPR*, pages 9621–9630, 2019.

[38] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *CVPR*, pages 1912–1920, 2015.

[39] Saining Xie, Jiatao Gu, Demi Guo, Charles R Qi, Leonidas Guibas, and Or Litany. Pointcontrast: Unsupervised pretraining for 3d point cloud understanding. In *European Conference on Computer Vision*, pages 574–591. Springer, 2020.

[40] Saining Xie, Sainan Liu, Zeyu Chen, and Zhuowen Tu. Attentional shapecontextnet for point cloud recognition. In *CVPR*, pages 4606–4615, 2018.

[41] Yifan Xu, Tianqi Fan, Mingye Xu, Long Zeng, and Yu Qiao. Spidercnn: Deep learning on point sets with parameterized convolutional filters. In *ECCV*, pages 87–102, 2018.

[42] Jiancheng Yang, Qiang Zhang, Bingbing Ni, Linguo Li, Jinxian Liu, Mengdie Zhou, and Qi Tian. Modeling point clouds with self-attention and gumbel subset sampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3323–3332, 2019.

[43] Yaoqing Yang, Chen Feng, Yiru Shen, and Dong Tian. Foldingnet: Point cloud auto-encoder via deep grid deformation. In *CVPR*, pages 206–215, 2018.

[44] 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.

[45] Wang Yifan, Shihao Wu, Hui Huang, Daniel Cohen-Or, and Olga Sorkine-Hornung. Patch-based progressive 3d point set upsampling. In *CVPR*, pages 5958–5967, 2019.

[46] Lequan Yu, Xianzhi Li, Chi-Wing Fu, Daniel Cohen-Or, and Pheng-Ann Heng. Ec-net: an edge-aware point set consolidation network. In *ECCV*, pages 386–402, 2018.

[47] Yongheng Zhao, Tolga Birdal, Haowen Deng, and Federico Tombari. 3d point capsule networks. In *CVPR*, pages 1009–1018, 2019.