Unsupervised Learning of Intrinsic Structural Representation Points

Nenglun Chen[1], Lingjie Liu[2], Zhiming Cui[1], Runnan Chen[1], Duygu Ceylan[3],
Changhe Tu[4], Wenping Wang[1]

[1]The University of Hong Kong   [2]Max Planck Institute for Informatics
[3]Adobe Research   [4]Shandong University

Abstract

Learning structures of 3D shapes is a fundamental problem in the field of computer graphics and geometry processing. We present a simple yet interpretable unsupervised method for learning a new structural representation in the form of 3D structure points. The 3D structure points produced by our method encode the shape structure intrinsically and exhibit semantic consistency across all the shape instances with similar structures. This is a challenging goal that has not fully been achieved by other methods. Specifically, our method takes a 3D point cloud as input and encodes it as a set of local features. The local features are then passed through a novel point integration module to produce a set of 3D structure points. The chamfer distance is used as reconstruction loss to ensure the structure points lie close to the input point cloud. Extensive experiments have shown that our method outperforms the state-of-the-art on the semantic shape correspondence task and achieves comparable performance with the state-of-the-art on the segmentation label transfer task. Moreover, the PCA based shape embedding built upon consistent structure points demonstrates good performance in preserving the shape structures. Code is available at `https://github.com/NolenChen/3DStructurePoints`

1. Introduction

Analyzing 3D shape structures is a fundamental problem in the field of computer graphics and geometry processing. One common way is to co-analyze a large collection of shapes, such as shape co-segmentation [4, 48], shape correspondence estimation [14], shape abstraction [37, 22, 33] and 3D keypoint discovery [35]. An important requirement for such co-analysis is to leverage the semantic consistency among different shapes to discover semantically consistent features or structures that can facilitate different

tasks. Quite a few structural representations have been proposed for 3D shapes, such as medial axis, curve skeleton and keypoints, which are designed for different tasks. Early works mainly use hand-crafted features and formulate it as optimization problems. While, they usually rely on parameter tuning, and are designed for specific tasks or datasets. Recently, deep learning techniques have been emerged to address these problems[13, 26]. In this work, we propose a method to learn a new structural representation for establishing semantic correspondence for 3D point clouds.
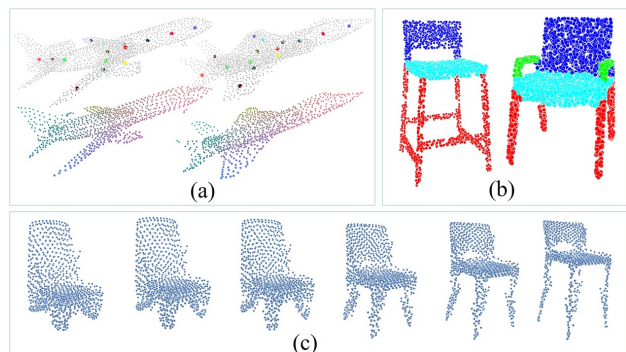

Figure 1: (a) Semantic correspondence. (b) Segmentation label transfer. (c) Visualization of one principal component of PCA embedding space.

There has recently been active research on deep learning techniques for processing 3D point clouds. PointNet [28] is a pioneer work in this direction. After that, a number of network structures have been proposed and have achieved great success in various tasks. Most existing works mainly focus on generating accurate reconstruction of the target shapes, but the generated point clouds lack of structure information. Recently, several works [36, 47, 41, 8, 31] have been proposed to generate structured point clouds, but with no guarantee for point-wise correspondence.

To address this issue, we propose a novel unsupervised method to learn semantically consistent structure points for

3D shapes in the same category. Given a 3D shape represented as a point cloud, the structure points generated by our network are a set of ordered points on the surface of the shape that provide a good abstraction and approximation of the input shape. Our architecture is simple but directly interpretable. Extensive experiments show that our method is robust to different point samplings and can be generalized to real scanned point clouds unseen in the training stage. Our method achieves state-of-the-art performance on the semantic shape correspondence task and the segmentation label transfer task. Moreover, based on the high consistency of the structure points, we build a PCA based shape embedding that can preserve shape structures well. Figure 1 illustrates the related applications.

Our key contributions are summarized as follows:

- A simple yet interpretable unsupervised architecture for extracting semantically meaningful structure points for 3D shapes represented as point clouds. It is robust to different sampling of point clouds and is applicable to real scanned data.

- Our method outperforms the state-of-the-art on the semantic shape correspondence task and achieves comparable performance with state-of-the-art methods on the segmentation label transfer task.

- A PCA based shape embedding built upon consistent structure points is able to preserve the shape structures well and has the potential to be used in several important tasks, such as shape reconstruction and shape completion.

2. Related Work

2.1. Shape Structure Analysis

Recently, quite a few works have been proposed for learning keypoints as structural shape representations. Several unsupervised methods have been proposed to learn keypoints in 2D image domain. [16, 24, 46] disentangle the structure and appearance of 2D images for keypoints discovery. For keypoint detection on 3D shapes, KeyPointNet [35] utilizes multi-view consistency to discover a sparse set of geometrically and semantically consistent keypoints across different shapes in the same category.

Unlike previous works, we conduct unsupervised learning of either sparse or dense consistent structure points as structural representation directly on 3D point clouds. Our method can be easily generalized to real scanned data.

2.2. Deep Learning on Point Clouds

Recent advances in deep neural networks have shown its great success in processing point clouds. PointNet [28], a pioneering work, utilizes point-wise MLP layers together with symmetric and permutation-invariant functions to aggregate information from all the points. To aggregate both local and global information, PointNet++ [29] introduces a hierarchical structure that applies PointNet recursively on the partitioned point cloud. So-Net [21] uses Self-Organizing Map to model the spatial distribution of a point cloud to extract hierarchical features. Inspired by 2D convolution, [32, 39, 38] reformulate the convolution operation and adapt it to point clouds. Our approach is built upon PointNet++ and it can also be built on other point cloud processing networks.

Due to the irregular structures of point clouds, designing a decoder for generating point clouds is more difficult than that for 2D images. Previous works such as [1, 10, 44] mainly use MLP to generate point clouds from the encoded embedding. The point clouds generated by these methods lack structure information. FoldingNet [42] and its variants [7, 8, 12, 41] propose to generate the point clouds by deforming primitive structures. [23] proposes a structural point cloud generator based on predicting structures at multiple viewpoints. [31, 36] use tree structure to design decoder network for generating structured point clouds. [47] introduced capsule network for processing point clouds, in which shape structure information is encoded in the latent capsules. Our network, as a new kind of structured point cloud auto-encoder, produces structure points that exhibit point-wise consistency. Moreover, our PCA based structure points embedding also has the potential to be used in some important tasks like shape reconstruction and completion.

2.3. 3D Shape Correspondence

Computing correspondences for 3D shapes is a fundamental task in geometric processings. Earlier methods [15, 18, 19, 30, 40] mainly rely on hand-crafted descriptors or optimization among a collection of shapes to compute the correspondence. The functional map and its variants [14, 27] provide a framework for representing maps between spaces of real-valued functions defined on shapes.

Recently, deep learning techniques have been widely used in learning correspondence for 3D shapes. A series of methods [3, 5, 11, 13] have been proposed to learn local descriptors for registrating point clouds. ShapeUnicode [26] proposes a way to learn unified embedding for 3D shapes in different representations, and demonstrates its ability in establishing correspondence among 3D shapes. However, these methods require labeled pairwise correspondences in the training stage. Deep functional map [34] aims at building functional bases with deep learning, while indicator functions are still needed in the training stage. [7, 45] use self supervised learning to learn local descriptors about point clouds, and demonstrate good performance in registration of 3D scans.

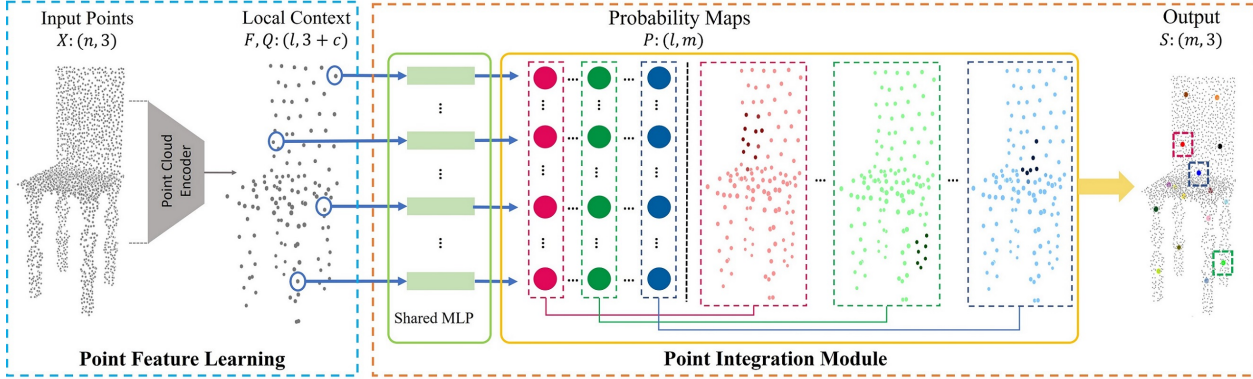Unlike previous methods, our approach learns consistent

Figure 2: Pipeline of our network: Given a point cloud, PointNet++ is first used to extract the local contextual features F and their corresponding points Q. Then, the features F and the sample points Q are further passed through the point integration module to produce the structure points S.

structure points across all shapes in a category in an unsupervised way. Based on the high consistency of the structure points, we can achieve good performance in transferring segmentation labels, and build the embedding space for point clouds directly with PCA.

3. Method

In this section, we introduce our end-to-end framework for learning intrinsic structure points from point clouds without explicit supervision. As illustrated in Figure 2, given a point cloud, $X = \{x_1, x_2, ..., x_n\}$ with $x_i \in \mathbb{R}^3$, our model predicts an ordered list of structure points, $S = \{s_1, s_2, ..., s_m\}$ with $s_i \in \mathbb{R}^3$. The network is trained for a collection of 3D shapes in the same category in an unsupervised manner. In the following, we first describe our network architecture which is composed of a PointNet++ encoder and a point integration module. Then we introduce a reconstruction loss for training. Finally, we show that the produced structure points exhibit semantic consistency across all the shapes in the same category, which is an essential property for shape co-analysis.

3.1. Network Architecture

Our network architecture is summarized in Figure 2. It can be seen as an encoder-decoder based architecture consisting of two modules, PointNet++ encoder and Point integration module. In the following, we provide more details.

PointNet++ Encoder Given a point cloud, $X = \{x_1, x_2, ..., x_n\}$ with $x_i \in \mathbb{R}^3$, the first step is to extract sample points $Q = \{q_1, q_2, ..., q_l\}(q_i \in \mathbb{R}^3)$ with the local contextual features $F = \{f_1, f_2, ..., f_l\}(f_i \in \mathbb{R}^c)$, where l indicates the number of sample points and c is the dimension of the feature representation.

We build our architecture upon the PointNet++ [29] encoder, which extracts point cloud features by adaptively

combining features from multiple scales. The PointNet++ encoder is composed of multiple set abstraction levels. Each level consists of three key layers: sampling layer, multi-scale grouping layer and PointNet layer for processing and abstracting the input points in a hierarchical way. We refer the reader to the original paper for more details.

Figure 3: The probability maps of two structure points (in red and green boxes) are shown in second and third columns. Darker color indicates higher probability.

Point Integration Module The input to this module is the sample points Q with their local contextual features F of size $l \times (3 + c)$ obtained by the encoder. The outputs are m structure points S. Detailly, given the local contextual features F, we first apply a shared multi-layer perceptron (MLP) block followed by softmax as activation function to produce the probability maps $P = \{p_1, p_2, ..., p_m\}$. The element p_i^j in the probability map p_i indicates the probability of the point q_j being the structure point s_i. Therefore, the structure points S can be computed as follows:

$$s_i = \sum_{j=1}^{l} q_j p_i^j \quad \text{with} \quad \sum_{j=1}^{l} p_i^j = 1 \quad \text{for each} \quad i \quad (1)$$

Equation 1 is a convex combination of the points Q, ensuring that the predicted structure points S are located within the convex hull of the points Q.

In Figure 3, we visualize the learned probability map for different structure points. The first column shows the in-

put point clouds (in grey) and their corresponding 16 structure points (in color). The probability map of two structure points marked with red and green boxes are illustrated in the second and third columns. Note that, the learned probability maps have concentration effect, though we do not explicitly enforce them to be concentrated in local regions.

3.2. Reconstruction Loss

To train the network in an unsupervised manner, we define our reconstruction loss based on the Chamfer distance (CD) [10] to constrain the predicted structure points to be close to the input point cloud. Specifically, the reconstruction loss is the Chamfer distance between the predicted structure points S and the input points X:

$$L_{rec}(S, X) = \sum_{s_i \in S} \min_{x_j \in X} \|s_i - x_j\|_2^2 + \sum_{x_j \in X} \min_{s_i \in S} \|s_i - x_j\|_2^2 \tag{2}$$

3.3. Cross-Object Consistency

We do not explicitly enforce the structure points to be consistent for different instances, but the network is able to generate semantically consistent probability maps P for the objects in the same category, leading to the consistency of the produced structure points as shown in Figure 4. In the following, we give a detailed explanation about the consistency of structure points.

To simplify discussion, we first consider two identical shapes A and B defined by the same point cloud X, except that these points are processed in possibly different orders for A and B. After passing through PointNet++ feature extractor and the shared MLP, a set of sample points of shape A are extracted with the corresponding m-dimensional probability vectors. Let us denote these sample points of A by the ordered sequence of points $(a_1, a_2, ..., a_\ell)$, with the corresponding probability vectors $\zeta_i = (p_1^i, p_2^i, ..., p_m^i)$, $i = 1, 2, \ldots, \ell$. Since B is identical to A, we may suppose that the sample points $(b_1, b_2, ..., b_\ell)$ obtained in the same way for B are a permutation of the sample points $(a_1, a_2, ..., a_\ell)$ for A. So we may denote $b_i = a_{\pi(i)}$, where $\pi(.)$ is a permutation function of $\{1, 2, .., \ell\}$. Then the probability vectors of the sample points b_i of shape B are $\zeta_{\pi(i)} = (p_1^{\pi(i)}, p_2^{\pi(i)}, ...p_m^{\pi(i)})$, $i = 1, 2, \ldots, \ell$. And the j-th structure point for shape A is given by:

$$s_j^A = \sum_{i=1}^{l} p_j^i a_i \tag{3}$$

Then for the j-th structure point s_j^B for shape B, we have:

$$s_j^B = \sum_{i=1}^{\ell} p_j^{\pi(i)} b_i = \sum_{i=1}^{\ell} p_j^{\pi(i)} a_{\pi(i)} = \sum_{i=1}^{\ell} p_j^i a_i = s_j^A \tag{4}$$

That is, the j-th structure point for shape A is the same as the j-th structure point for shape B. When two shapes A and B are similar, because the neural networks in the pipeline are continuous mappings, the structure points $\{s_j^A\}_{j=1}^m$ of A will be individually close to their corresponding structure points $\{s_j^B\}_{j=1}^m$ of B.

4. Experiments

4.1. Datasets

We conduct most of our experiments on the ShapeNet [43] dataset. In the 3D semantic correspondence task, we use the same datasets in [13], where ShapeNet [43] and the BHCP benchmark [18] are used for training and testing respectively. For segmentation label transfer task, ShapeNet part dataset [43] is used. Farthest point sampling [9] is applied to sample point clouds from 3D shapes.

4.2. Implementation Details

The PointNet++ encoder we used is composed of two set abstraction levels with 512 and 128 grouping centers respectively. Multi-scale grouping (MSG) is used to combine the multi-scale features. The MSG layers in two levels contain scales (0.1, 0.2, 0.4) and (0.2, 0.4, 0.8) respectively. The output of the PointNet++ encoder contains $l = 128$ sample points, with each sample point having 640 dimensional local contextual features. The configuration of the MLP block in the point integration module depends on the number of structure points. Specifically, for $m = 512$ structure points, the MLP block contains 3 layers with the neuron numbers (640, 512, 512). The dropout ratio is set to 0.2 to avoid over-fitting of the MLP block. Adam[20] is used as the optimizer. We train our network on a single NVIDIA GTX 1080Ti GPU with less than 1 hour per category.

4.3. Consistency Across Objects

The structure points produced for different shapes are visualized in Figure 4, where the input point clouds (grey) and the sparse structure points (colored) are shown in the 1st and 3rd rows, and the corresponding dense structure points are illustrated in the 2rd and 4th rows. Corresponding structure points are visualized in the same color. It can be seen that our approach can generate sparse and dense structure points in a consistent manner for the shapes with similar structures. Note that such correspondence may not exist in the regions with significant structure differences. An example is the chairs with and without armrest in Fig. 4.

4.4. 3D Semantic Correspondence

To evaluate the cross-object consistency of the structure points, we test our approach on the task of computing 3D shape semantic correspondence and compare it with the state-of-the-art methods that have been specifically tuned
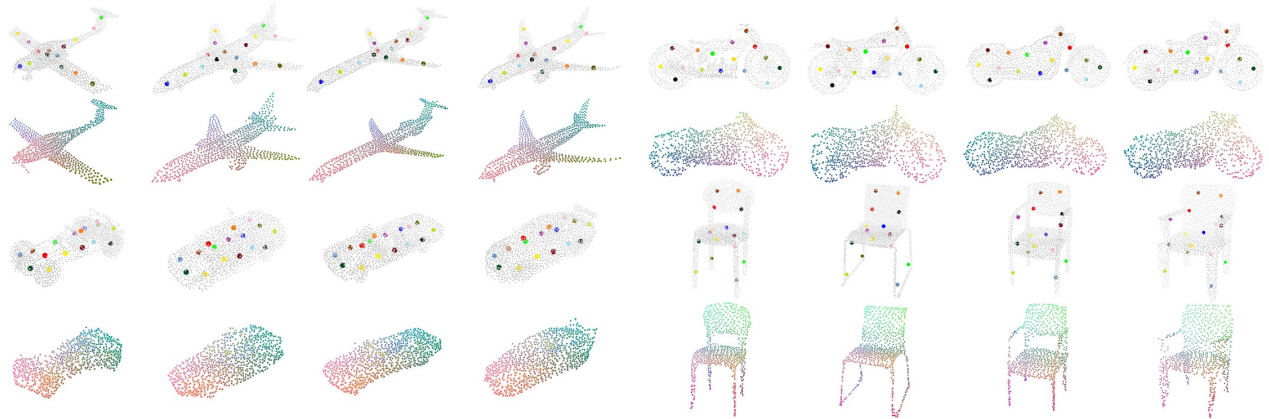
Figure 4: Consistency across different shapes. The produced 16 structure points (colored) for different input point clouds (grey) are shown in 1st and 3rd rows, and the produced 1024 structure points are in 2nd and 4th rows. Corresponding structure points are in the same color.

for this task including LMVCNN [13], ShapeUnicode [26] and EdgeNet [3], as well as some point cloud encoder-decoder architectures: AtlasNet2 [8], FoldingNet [42], Top-Net [36] and 3DPointCapsule [47]. We also use the critical points in PointNet [28] and the FCN based decoder described in [1] as baseline comparisons. Since LMVCNN is rotation-invariant, for fair comparison with it, we train our method with rotation augmentation and test on the point clouds with arbitrary rotations.

Specifically, to make the network applicable to a rotated point cloud, we perform a PCA (Principle Components Analysis) [17] based augmentation for training. We first compute three main axes for each shape in the training dataset. In each training iteration the main axes of each shape are randomly swapped and the shapes are aligned according to the swapped axes. The consistency of the structure points for original and augmented shape are enforced with Mean Square Error (MSE) loss. During testing, we also compute the three main axes for each shape, and align it accordingly. Compared to data augmentation with random rotations, the PCA based scheme can reduce the rotation space, and makes the network converge more efficiently.

We use the datasets used in [13], where ShapeNet [43] and the BHCP benchmark [18] are used for training and testing respectively. Note that we use airplanes to train the network for helicopters in the same way it did in [13] since helicopters are not included in the training data.

Our network is trained on a collection of aligned shapes represented by 2048 points with 512 structure points as output. During testing, given a point x_q on a 3D shape, we first find its closest structure point s_q, and then use the corresponding structure point s_q' on the target shape as the corresponding point of x_q. The correspondence accuracy is measured by the fraction of the correspondences that are correctly predicted with the error below given Euclidean thresholds.
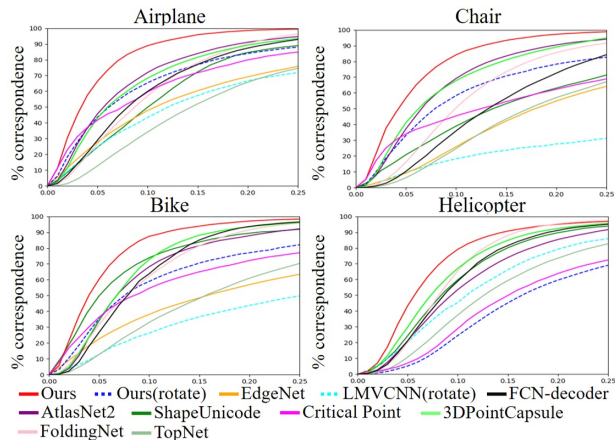

Figure 5: Correspondence accuracy for each category in the BHCP benchmark.

In Figure 5, the solid lines show the results tested on the aligned data and the dotted lines denote the results tested on the unaligned data. We can clearly see that our method significantly outperforms other state-of-the-art methods. Moreover, we demonstrate the good generalization to unseen categories (e.g. train with airplanes and test with helicopters). However, in this case, the performance of our method on the rotated data is not as good as the aligned one. That is because the PCA of helicopters is quite different from that of airplanes, therefore more difficult for the network to adapt to the unseen category of helicopters.

4.5. Example based Label Transfer

We further evaluate the quality of the structure points by transferring segmentation labels from few examples, and compare our average IOU with a state-of-the-art shape segmentation method BAE-NET [4] on shapenet part[43] dataset. BAE-NET has different settings(e.g. unsupervised, one-shot and few-shot), we compare our method with it's

few-shot setting. Specifically, our network is trained on the training set without any label. During testing, to label a point x_q on a shape, we find its closest structure point s_q as well as a list of corresponding structure points $E = \{s_q^1, s_q^2, ..., s_q^k\}$ on k example shapes, and the label of structure point $s_q' \in E$ which has the most similar features with s_q is transferred to x_q.
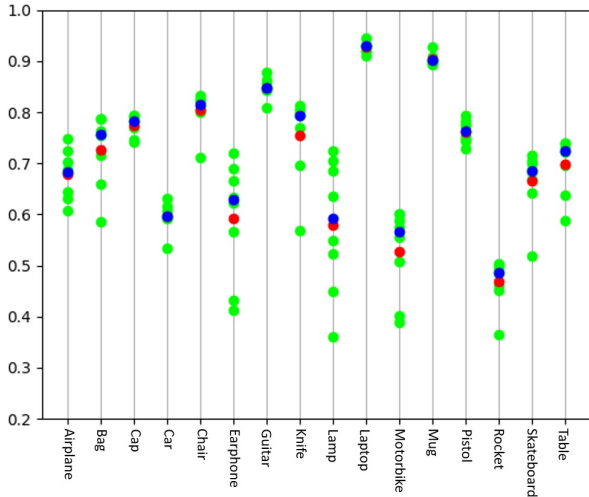

Figure 6: Label transfer results based on 3 examples. For each category, the average IOUs by 8 randomly selected example sets (green dots) are illustrated. The mean and median are shown in red and blue.

As discussed in [4], the accuracy of the segmentation results will be affected by the selection of the example shapes in the few-shot setting. Empirically, the exemplars should contain all the segmentation labels and represent the variations of the dataset. In Figure 6, we show all the results of our segmentation label transfer by using 8 randomly selected exemplar sets, each set contains 3 exemplars. For the categories with regular shapes (e.g. Laptop), the results are consistent. While, the accuracy may vary a lot for the categories with large shape variations (e.g. 'Earphone'). In Table 1, the best performing results are reported for both methods, and Figure 7 demonstrates some qualitative results. We can see that, our label transfer results are on par with BAE-Net in most categories with structured shapes. For some categories such as 'Bag', 'Cap', and 'Earphone', due to their large variations, 3 exemplars are not enough for our method. For the 'Car' category, since it contains flat surface that BAE-Net cannot separate, we do not report its result. Moreover, in BAE-NET, the segmentation labels are either pre-defined by the user (few-shot setting) or implicitly defined by the network (unsupervised setting); once the training is done, the labels cannot change. In contrast, our work achieves the few-shot label transfer by transferring segmentation labels directly from exemplars, thus has the potential of transferring arbitrary labels (e.g., labels with

different hierarchies) after training. (More results are in the supplementary material)
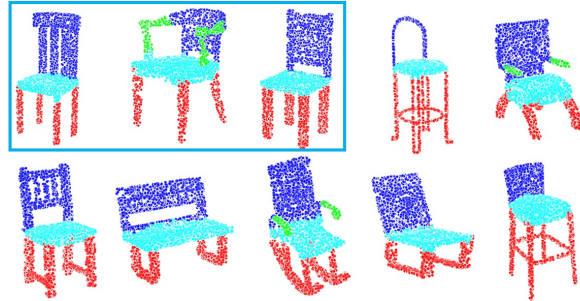

Figure 7: Qualitative results for label transfer, the example shapes are in the blue box.

4.6. PCA based Shape Embedding

Given high quality shape correspondences, we can build a shape embedding space based on PCA. Specifically, given a collection of point clouds, we produce m structure points $S = \{s_1^1, s_1^2, s_1^3, ..., s_m^1, s_m^2, s_m^3\}$ for each point cloud with the proposed network. Where $\{s_i^1, s_i^2, s_i^3\} \in \mathbb{R}^3$ denotes the location of ith structure point. Then a shape morphable model [2] can be constructed based on the learnt structure points. Thus, a new shape X can be be represented as a linear combination of the principal components:

$$X = \bar{S} + \sum_{i=1}^{k} \alpha_i c_i \qquad (5)$$

Where $\bar{S}$ is the mean shape of the structure points, k is the number of principal components, c_i and α_i denote the ith principal component and the corresponding coefficient respectively. Figure 8(a) shows the PCA embedding space by adding the first two PCA principal components to the mean shape with ratio from -3σ to 3σ, where σ^2 denotes the eigen value of the corresponding principal component. Figure 8(b) shows some reconstruction results with only 50 principal components. We can see that the structures of the shapes can be well preserved with our PCA based embedding. (More results are in supplementary material)

5. Ablations and Visualizations

5.1. Feature Embedding Visualization

To better understand the mechanism of the proposed method, we extract and visualize the latent features learned by the network. The local contextual features F produced by PointNet++ are first weighted by the probability map P to get per-point features $H = \{h_1, h_2, ..., h_m\}$ with $h_i \in \mathbb{R}^c$ for each structure point:

	Airplane	Bag	Cap	Car	Chair	Earphone	Guitar	Knife	Lamp	Laptop	Motorbike	Mug	Pistol	Rocket	Skateboard	Table
BAE-Net	74.7	**83.9**	**85.5**	-	**86.0**	**76.2**	87.8	**83.6**	70.1	**94.8**	**64.6**	**94.8**	78.7	**52.1**	**74.2**	73.3
Ours	**74.9**	78.6	79.3	**60.6**	84.5	69.0	**87.9**	80.6	**72.4**	94.4	60.0	92.7	**79.4**	50.3	70.4	**73.9**

Table 1: Comparison of our label transfer results against BAE-NET with 3 labeled exemplars on the ShapeNet part dataset[43] measured with average IOU(%)
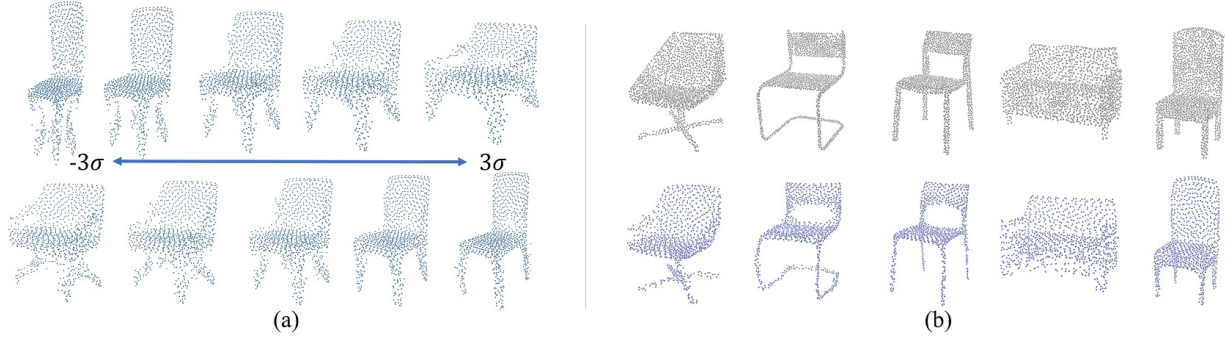

(a)   (b)

Figure 8: PCA based shape embedding. (a) Visualization of the PCA embedding space by adding the PCA principle components to the mean shape with ratio from -3σ to 3σ. The first two components are used in the 1st and 2rd row respectively (b) Input point clouds (first row) and the corresponding reconstructions (second row) with 50 principal components.

$$h_i = \sum_{j=1}^{l} f_j p_i^j \quad \text{with} \quad \sum_{j=1}^{l} p_i^j = 1 \quad \text{for each} \quad i \quad (6)$$

The per-point features H for all the shapes in the ShapeNet testing dataset with the same category are then embedded in a 2-dimensional embedding space using t-SNE [25] for visualization.
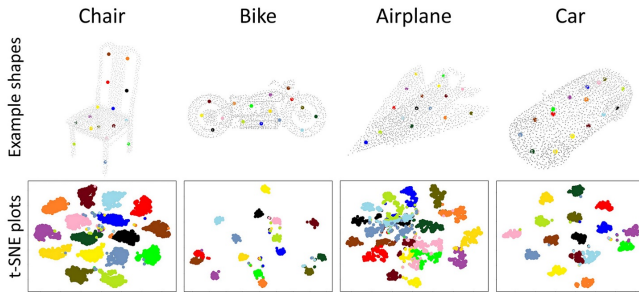

Figure 9: The T-SNE embedding of the learned features. The first row shows examples of structure points overlaid with the input point clouds, and the second row shows the corresponding t-SNE embeddings.

Figure 9 illustrates the t-SNE results of the embedding space computed on four categories when 16 structure points are predicted. The 2D points in the plot are colored with 16 unique colors each corresponding to a specific structure point. Note that, the learned features are well clustered, which means the structure points with the same semantic location tend to have similar features, though we do not explicitly enforce the consistency of the latent features. One may note that symmetry structure points on the same shape

do not have similar features, this is because the PointNet++ encoder we used is not symmetric invariant. One may consider adding symmetry constraints into the loss or using a symmetry invariant feature encoder to make the structure points symmetric invariant.

5.2. Robustness to Sampling Densities

To evaluate the robustness of our approach to input point clouds with different densities, we train our network on 2048 points sampled on each shape, and test the network on input point clouds with different densities.

We use the point-wise average Euclidean distance to measure the stability of the produced structure points with different input densities in comparison with the structure points produced from 2048 input points. To generate non-uniformly sampled points, we first randomly sample a relatively small number of seed points from the initial point cloud, and remove points near the seeds with certain probability. This creates a set of non-uniformly distributed points with missing points (or "holes") around the seed points.

Sample Num	Average Distance(%)			
	256	512	1024	4096
Chair	0.7346	0.1308	0.0200	0.0027
Airplane	0.1209	0.0169	0.0022	0.0001
Car	0.5470	0.1348	0.0099	0.0017
Motorbike	0.3070	0.0610	0.0195	0.0111

Table 2: Stability of the network for producing 1024 structure points with input points of different uniform densities.

Table 2 shows the stability of the structure points with different numbers of uniformly sampled points as inputs. For non-uniformly sampled inputs, the average stability of

these categories is 0.3499%. We also demonstrate some qualitative results in Figure 10 and 11. As shown by qualitative and quantitative results, our approach is not sensitive to the point sampling density. That is because the PointNet++ [29] we used to encode local features is not sensitive to the sampling of point clouds and our point integration module also maintains such property.
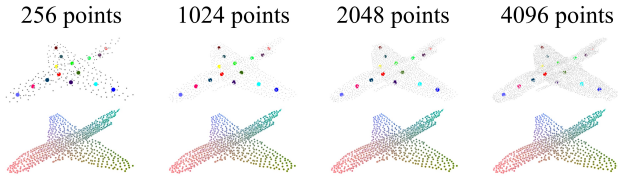

Figure 10: Robustness to different samplings. The 1st row shows 16 structure points and different numbers of input point clouds. The 2nd row shows 1024 structure points. Corresponding structure points have the same color.
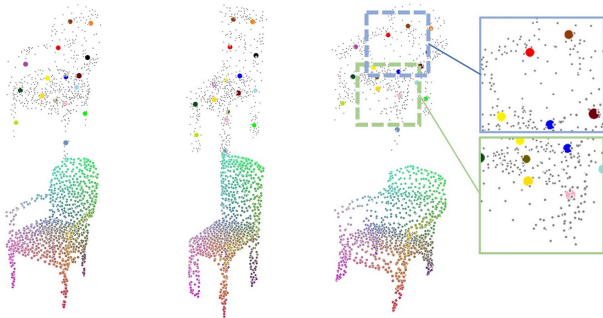

Figure 11: Robustness to non-uniform sampling. The 1st row shows 16 structure points and non-uniform input point clouds. The 2nd row shows 1024 structure points. Corresponding structure points have the same color.

5.3. Testing with Real Scanned Data

We show the good generalization of our method by testing on real scanned data. Specifically, we train our network on the ShapeNet dataset with rotation augmentation, and test the trained network on real scanned point clouds [6]. As illustrated in Figure 12, even though the real scanned point clouds are noisy and not seen during training, our network can still produce semantically consistent structure points.

5.4. Testing with Different Feature Encoder

The proposed point integration module can also be integrated with other point cloud learning architectures to learn consistent structure points. Specifically, We replace the PointNet++ feature encoder with PointConv [39], a state-of-the-art point cloud learning architecture, and evaluate the performance of semantic shape correspondence. As shown in Figure 13, both architectures can produce similar results on semantic shape correspondence accuracy.
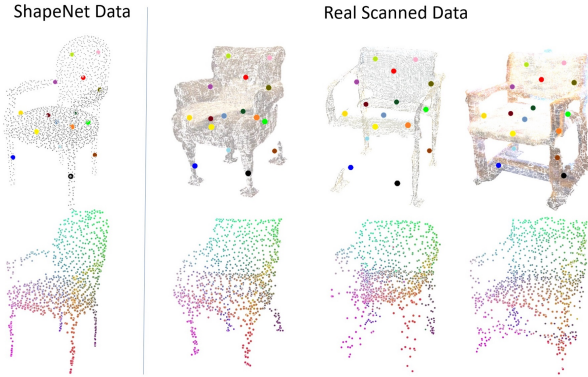

Figure 12: Structure points on real scanned data. We show the produced 16 (1st row) and 1024 (2nd row) structure points for scanned point clouds of chairs. Corresponding structure points are in the same color.
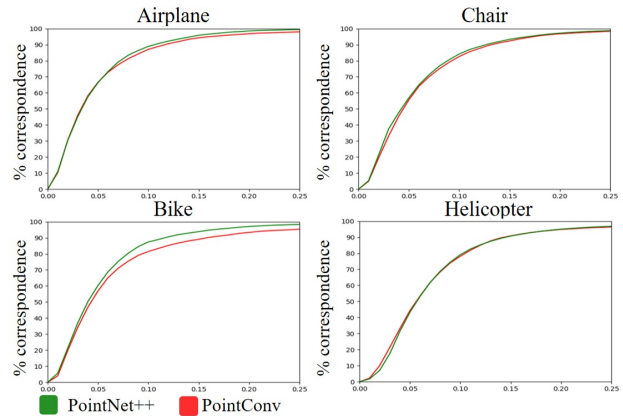

Figure 13: Testing with different point cloud encoder.

6. Conclusion

In this paper, we present a simple yet interpretable unsupervised method for learning a new structural representation in the form of 3D structure points. The produced structure points encode shape structures and exhibit semantic consistency across all the shape instances with similar shape structures. We evaluate the proposed method with extensive experiments and show state-of-the-art performance on both semantic correspondence and segmentation label transfer tasks. We also show the good generalization of our network by testing on real scanned data. Moreover, our PCA based structure points embedding also has the potential to be used in some important tasks like shape reconstruction and completion, which we would like to explore more in the future.

References

[1] Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas Guibas. Learning representations and generative models for 3d point clouds. *arXiv preprint arXiv:1707.02392*, 2017. 2, 5

[2] Volker Blanz, Thomas Vetter, et al. A morphable model for the synthesis of 3d faces. In *Siggraph*, volume 99, pages 187–194, 1999. 6

[3] Mingjia Chen, Qianfang Zou, Changbo Wang, and Ligang Liu. Edgenet: Deep metric learning for 3d shapes. *CAGD*, 72:19–33, 2019. 2, 5

[4] Zhiqin Chen, Kangxue Yin, Matthew Fisher, Siddhartha Chaudhuri, and Hao Zhang. Bae-net: Branched autoencoder for shape co-segmentation. *arXiv preprint arXiv:1903.11228*, 2019. 1, 5, 6

[5] Christopher Choy, Jaesik Park, and Vladlen Koltun. Fully convolutional geometric features. In *ICCV*, 2019. 2

[6] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017. 8

[7] Haowen Deng, Tolga Birdal, and Slobodan Ilic. Ppf-foldnet: Unsupervised learning of rotation invariant 3d local descriptors. In *ECCV*, pages 602–618, 2018. 2

[8] Theo Deprelle, Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. Learning elementary structures for 3d shape generation and matching. *arXiv preprint arXiv:1908.04725*, 2019. 1, 2, 5

[9] Yuval Eldar, Michael Lindenbaum, Moshe Porat, and Yehoshua Y Zeevi. The farthest point strategy for progressive image sampling. *TIP*, 6(9):1305–1315, 1997. 4

[10] Haoqiang Fan, Hao Su, and Leonidas J Guibas. A point set generation network for 3d object reconstruction from a single image. In *CVPR*, pages 605–613, 2017. 2, 4

[11] Zan Gojcic, Caifa Zhou, Jan D Wegner, and Andreas Wieser. The perfect match: 3d point cloud matching with smoothed densities. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5545–5554, 2019. 2

[12] Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. Atlasnet: A papier-m\^ ach\'e approach to learning 3d surface generation. *arXiv preprint arXiv:1802.05384*, 2018. 2

[13] Haibin Huang, Evangelos Kalogerakis, Siddhartha Chaudhuri, Duygu Ceylan, Vladimir G Kim, and Ersin Yumer. Learning local shape descriptors from part correspondences with multiview convolutional networks. *TOG*, 37(1):6, 2018. 1, 2, 4, 5

[14] Qixing Huang, Fan Wang, and Leonidas Guibas. Functional map networks for analyzing and exploring large shape collections. *TOG*, 33(4):36, 2014. 1, 2

[15] Qi-Xing Huang, Hao Su, and Leonidas Guibas. Fine-grained semi-supervised labeling of large shape collections. *TOG*, 32(6):190, 2013. 2

[16] Tomas Jakab, Ankush Gupta, Hakan Bilen, and Andrea Vedaldi. Conditional image generation for learning the structure of visual objects. *methods*, 43:44, 2018. 2

[17] Ian Jolliffe. *Principal component analysis*. Springer, 2011. 5

[18] Vladimir G Kim, Wilmot Li, Niloy J Mitra, Siddhartha Chaudhuri, Stephen DiVerdi, and Thomas Funkhouser. Learning part-based templates from large collections of 3d shapes. *TOG*, 32(4):70, 2013. 2, 4, 5

[19] Vladimir G Kim, Wilmot Li, Niloy J Mitra, Stephen DiVerdi, and Thomas A Funkhouser. Exploring collections of 3d models using fuzzy correspondences. *TOG*, 31(4):54–1, 2012. 2

[20] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 4

[21] Jiaxin Li, Ben M Chen, and Gim Hee Lee. So-net: Self-organizing network for point cloud analysis. In *CVPR*, pages 9397–9406, 2018. 2

[22] Lingxiao Li, Minhyuk Sung, Anastasia Dubrovina, Li Yi, and Leonidas J Guibas. Supervised fitting of geometric primitives to 3d point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2652–2660, 2019. 1

[23] Chen-Hsuan Lin, Chen Kong, and Simon Lucey. Learning efficient point cloud generation for dense 3d object reconstruction. In *AAAI*, 2018. 2

[24] Dominik Lorenz, Leonard Bereska, Timo Milbich, and Björn Ommer. Unsupervised part-based disentangling of object shape and appearance. *arXiv preprint arXiv:1903.06946*, 2019. 2

[25] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008. 7

[26] Sanjeev Muralikrishnan, Vladimir G Kim, Matthew Fisher, and Siddhartha Chaudhuri. Shape unicode: A unified shape representation. In *CVPR*, pages 3790–3799, 2019. 1, 2, 5

[27] Maks Ovsjanikov, Mirela Ben-Chen, Justin Solomon, Adrian Butscher, and Leonidas Guibas. Functional maps: a flexible representation of maps between shapes. *TOG*, 31(4):30, 2012. 2

[28] 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. 1, 2, 5

[29] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Neurips*, pages 5099–5108, 2017. 2, 3, 8

[30] Emanuele Rodolà, Samuel Rota Bulo, Thomas Windheuser, Matthias Vestner, and Daniel Cremers. Dense non-rigid shape correspondence using random forests. In *CVPR*, pages 4177–4184, 2014. 2

[31] Dong Wook Shu, Sung Woo Park, and Junseok Kwon. 3d point cloud generative adversarial network based on tree structured graph convolutions. *arXiv preprint arXiv:1905.06292*, 2019. 1, 2

[32] Martin Simonovsky and Nikos Komodakis. Dynamic edge-conditioned filters in convolutional neural networks on graphs. In *CVPR*, pages 3693–3702, 2017. 2

[33] Chunyu Sun, Qianfang Zou, Xin Tong, and Yang Liu. Learning adaptive hierarchical cuboid abstractions of 3d shape collections. *ACM Transactions on Graphics (SIGGRAPH Asia)*, 38(6), 2019. 1

[34] Minhyuk Sung, Hao Su, Ronald Yu, and Leonidas J Guibas. Deep functional dictionaries: Learning consistent semantic structures on 3d models from functions. In *Advances in Neural Information Processing Systems*, pages 485–495, 2018. 2

[35] Supasorn Suwajanakorn, Noah Snavely, Jonathan J Tompson, and Mohammad Norouzi. Discovery of latent 3d keypoints via end-to-end geometric reasoning. In *Neurips*, pages 2059–2070, 2018. 1, 2

[36] Lyne P Tchapmi, Vineet Kosaraju, Hamid Rezatofighi, Ian Reid, and Silvio Savarese. Topnet: Structural point cloud decoder. In *CVPR*, pages 383–392, 2019. 1, 2, 5

[37] Shubham Tulsiani, Hao Su, Leonidas J Guibas, Alexei A Efros, and Jitendra Malik. Learning shape abstractions by assembling volumetric primitives. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2635–2643, 2017. 1

[38] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *arXiv preprint arXiv:1801.07829*, 2018. 2

[39] Wenxuan Wu, Zhongang Qi, and Li Fuxin. Pointconv: Deep convolutional networks on 3d point clouds. In *CVPR*, pages 9621–9630, 2019. 2, 8

[40] Kai Xu, Vladimir G Kim, Qixing Huang, Niloy Mitra, and Evangelos Kalogerakis. Data-driven shape analysis and processing. In *SIGGRAPH ASIA*, page 4. ACM, 2016. 2

[41] Guandao Yang, Xun Huang, Zekun Hao, Ming-Yu Liu, Serge Belongie, and Bharath Hariharan. Pointflow: 3d point cloud generation with continuous normalizing flows. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4541–4550, 2019. 1, 2

[42] Yaoqing Yang, Chen Feng, Yiru Shen, and Dong Tian. Foldingnet: Point cloud auto-encoder via deep grid deformation. In *CVPR*, pages 206–215, 2018. 2, 5

[43] Li Yi, Vladimir G Kim, Duygu Ceylan, I Shen, Mengyan Yan, Hao Su, Cewu Lu, Qixing Huang, Alla Sheffer, Leonidas Guibas, et al. A scalable active framework for region annotation in 3d shape collections. *TOG*, 35(6):210, 2016. 4, 5, 7

[44] Lequan Yu, Xianzhi Li, Chi-Wing Fu, Daniel Cohen-Or, and Pheng-Ann Heng. Pu-net: Point cloud upsampling network. In *CVPR*, pages 2790–2799, 2018. 2

[45] Andy Zeng, Shuran Song, Matthias Nießner, Matthew Fisher, Jianxiong Xiao, and Thomas Funkhouser. 3dmatch: Learning local geometric descriptors from rgb-d reconstructions. In *CVPR*, pages 1802–1811, 2017. 2

[46] Yuting Zhang, Yijie Guo, Yixin Jin, Yijun Luo, Zhiyuan He, and Honglak Lee. Unsupervised discovery of object landmarks as structural representations. In *CVPR*, pages 2694–2703, 2018. 2

[47] Yongheng Zhao, Tolga Birdal, Haowen Deng, and Federico Tombari. 3d point capsule networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1009–1018, 2019. 1, 2, 5

[48] Chenyang Zhu, Kai Xu, Siddhartha Chaudhuri, Li Yi, Leonidas Guibas, and Hao Zhang. Cosegnet: Deep co-segmentation of 3d shapes with group consistency loss. *arXiv preprint arXiv:1903.10297*, 2019. 1