Few-shot Learning of Part-specific Probability Space for 3D Shape Segmentation

Lingjing Wang* Xiang Li* Yi Fang†
NYU Multimedia and Visual Computing Lab
New York University Abu Dhabi, Abu Dhabi, UAE
New York University, New York, USA
{lw1474, xl845, yfang}@nyu.edu

Abstract

Recently, deep neural networks are introduced as supervised discriminative models for the learning of 3D point cloud segmentation. Most previous supervised methods require a large number of training data with human annotation part labels to guide the training process to ensure the model's generalization abilities on test data. In comparison, we propose a novel 3D shape segmentation method that requires few labeled data for training. Given an input 3D shape, the training of our model starts with identifying a similar 3D shape with part annotations from a mini-pool of shape templates (e.g. 10 shapes). With the selected template shape, a novel Coherent Point Transformer is proposed to fully leverage the power of a deep neural network to smoothly morph the template shape towards the input shape. Then, based on the transformed template shapes with part labels, a newly proposed Part-specific Density Estimator is developed to learn a continuous part-specific probability distribution function on the entire 3D space with a batch consistency regularization term. With the learned part-specific probability distribution, our model is able to predict the part labels of a new input 3D shape in an end-to-end manner. We demonstrate that our proposed method can achieve remarkable segmentation results on the ShapeNet dataset with few shots, compared to previous supervised learning approaches.

1. Introduction

3D shapes can usually be described with three representations: voxel-based representations [16, 31, 2, 18, 28, 29], mesh-based representations [30, 32, 22], and point cloud-based representations [15, 17, 10, 26]. In this paper, we mainly discuss the segmentation of 3D point cloud data as a simple 3D representation. Given a 3D point could, the segmentation approach concerns the assignment of each point with a semantic part label description. Recently, data-driven and deep learning models have gained popularity and success on learning 3D shape segmentation [15, 17, 10, 20] via supervised training on an annotated shape segmentation dataset [3]. In contrast to traditional hand-craft shape segmentation approaches [36, 19, 4], learning-based models can generalize their ability from training data to resolve ambiguities (e.g. structural variation, geometric complexity, incompleteness, occlusions) in 3D shapes to reliably segment the 3D shape into meaningful parts [15, 23, 13].

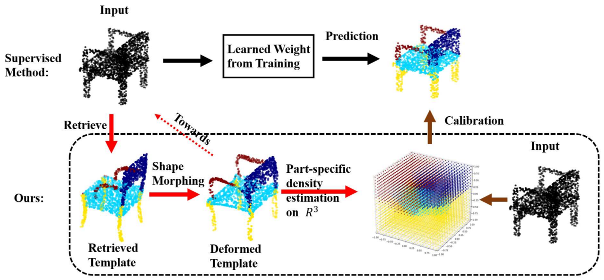

Figure 1. In comparison to supervised method which predicts the label of each point of input shape using a trained network, our model aims to transfer the labels from the retrieved template to the input shape based on the shape morphing and part-specific density estimation modules.

The promise of deep learning naturally motivates researchers to start with learning deep neural networks as a classifier for 3D points towards supervised 3D segmentation [15]. Those methods leverage the power of deep neural network to firstly encode 3D points into a high-dimensional geometric feature space, and then map the feature to the semantic part label [15, 17, 10]. To generalize the segmentation ability, the deep learning models are usually trained with a large scale well-annotated dataset in order to optimize the parameters of a model through minimizing the expectation of a categorical part classification loss across

*Equal contribution
†Corresponding author

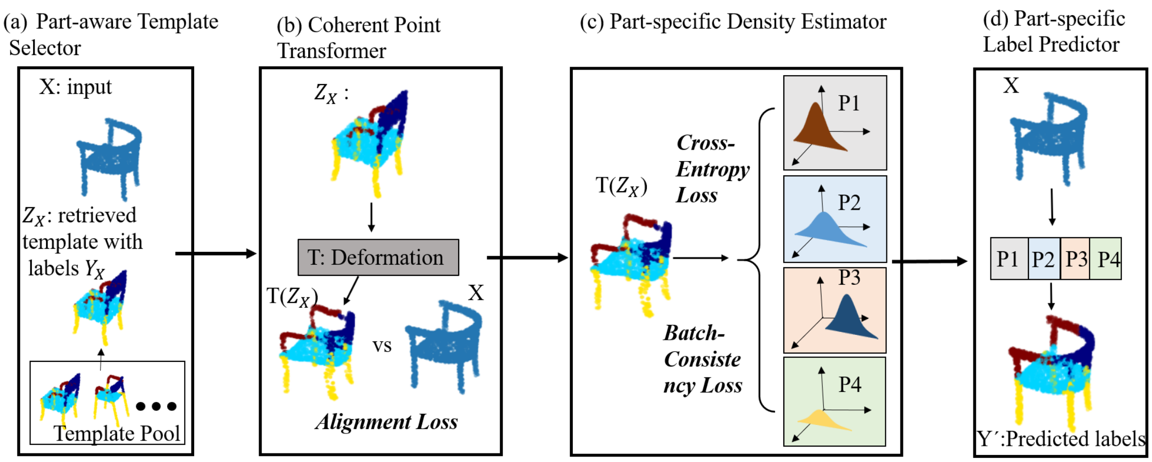
(a) Part-aware Template Selector
X: input
Z_X: retrieved template with labels Y_X
Template Pool
(b) Coherent Point Transformer
Z_X :
T: Deformation
$T(Z_X)$
X
vs
Alignment Loss
(c) Part-specific Density Estimator
$T(Z_X)$
Cross-Entropy Loss
Batch-Consistency Loss
P1
P2
P3
P4
(d) Part-specific Label Predictor
X
P1 P2 P3 P4
Y′:Predicted labels

Figure 2. Our pipeline. For a given input 3D point cloud $\mathbf{X}$, a template shape $\mathbf{Z_X}$ firstly is retrieved from a template pool by template selector (a). Coherent point transformer (b) morphs the retrieved template towards input shape. In (c), the Part-specific Density Estimator takes the points of deformed templates as input to compute the continuous probability distribution function in 3D space. In (d), for each point of input shape, its label can be predicted by Part-specific label predictor.

the entire training dataset [3]. PointNet and its variants [17, 10, 8] demonstrated that supervised deep neural networks are efficient and effective for 3D point cloud segmentation with experimental tests on various benchmark datasets. However, it is practically challenging and costly to annotate a large-scale training 3D point cloud dataset with high-quality semantic part labels, which limits the applicability of supervised learning based approaches in broader 3D segmentation applications.

In this paper, we firstly propose a novel model, named Weakly Supervised Point Cloud Segmentation Networks (WPS-Net), to realize the 3D point segmentation task assuming the existence of a few labeled training data. As shown in Figure 1, in contrast to the supervised learning of a discriminative model that firstly learns a high-dimensional point geometric feature and then maps it to a discrete semantic part label, our model aims to directly calibrate a spatially continuous probability function based on a deformed retrieved template to encode part semantic of a 3D point at infinite resolution. Figure 2 illustrates the pipeline of the proposed WPS-Net which consists of four major components. The first component is “Template Selector.” In this component, given an input 3D shape, WPS-Net starts with identifying a similar 3D shape with part annotation from a mini-pool of shape templates (e.g. 10 shapes). The second component is “Coherent Point Transformer”. In this component, a novel Coherent Shape Transformer is proposed to smoothly morph the selected template shape towards the input shape. The third component is “Part-specific Density Estimator”. In this component, based on the transformed template shape with part labels, a newly proposed Part-specific Density Estimator is developed to learn a spatially continuous probability function to encode part semantic of a 3D point at infinite resolution with a batch consistency regularization term. The fourth component is “Part-specific Label Predictor.” In this component, for a given input shape, the learned Part-specific density estimator is used to assign the part label to each point on the shape. The WPS-Net is able to train and predict the semantic part labels of a 3D shape in an end-to-end manner. The main contributions of our proposed method are listed as follows:

- We introduce a novel weakly supervised learning approach for 3D point cloud segmentation. To the best of our knowledge, WPS-Net is a pioneering attempt to 3D point cloud segmentation through a weakly supervised learning paradigm.
- We introduce a coherent point transformer for this task that can learn to predict a smooth geometric transformation field to morph the template 3D shape towards the input shape, which provides the possibility of transferring labels from the retrieved template to input shape.
- We introduce a part-specific density estimator that can calibrate a spatially continuous probability function to encode part semantic of a 3D point at infinite resolutions based on the deformed template, allowing us to further fill points from input shape into this density function for labelling.
- We introduce a novel batch-consistency regularization term in WPS-Net. Batch consistency term naturally

enforces in-part similarity and inter-part dissimilarity over the batch of input 3D shapes during the part-specific density calibration process.

2. Related Works

Supervised learning-based 3D shape segmentation. In recent years, deep learning-based methods have achieved great access in various field [12, 1, 27, 40, 11]. Much research attention have been paid to 3D point cloud segmentation using learning-based methods. These methods mainly focused on learning a robust point signature using deep neural networks, followed by a classification network to generate semantic part labels. Earlier efforts directly partitioned the input 3D point space into gridded voxels and apply 3D CNN on regular voxels [39, 16] to learn point features. To take the advantage of traditional 2D CNN for feature learning, researches [24, 25, 9] proposed to render 3D point cloud into 2D images to facilitate 2D CNN for representative feature learning. More recently, PointNet [15] firstly proposed to directly learn point features on unordered 3D point sets. Following researches such as PointNet++ [17], SO-Net [10], SplatNet [23], PointCNN [13] and D-FCN [34] focused on improving the performance by incorporating neighborhood information. Nevertheless, these methods mostly directly predict 3D shape segmentation in a supervised training process, thus require a large number of well-labeled dataset for model training. By contrast, our model does not directly determine the part label for each point but tries to predict a part-specific continuous probability distribution function in the 3D space using a set of sampled points in a weakly supervised way.

Weakly supervised learning-based 3D shape segmentation. Previous researches have explored co-analysis of a group of 3D shapes for co-segmentation [5, 6, 33, 21]. These methods tried to extract common geometric features from a group of shapes in a data-driven process, followed by clustering algorithms to group the primitive patches of all shapes into the similar part. Early researches [6, 14, 21, 36] are mainly focused on extracting low-level features from shape over-segmentation. Commonly used features include scale-invariant heat kernel signatures (SIHKS) [36], shape diameter function (SDF) [19], Gaussian curvature (GC) [4] and so on. Recent efforts also tried deep neural network for representative geometric feature extraction. For example, Shu et al. [20] proposed an unsupervised 3D shape segmentation algorithm by leveraging an Auto-Encoder model to learn high-level features from the low-level ones. These methods mainly focus on a combination of hand-craft/learning-based features with clustering algorithms to achieve unsupervised segmentation. Yuan et al. [38] proposed ROSS for one-shot learning of 3D mesh shape segmentation. In contrast, we propose a weakly supervised learning paradigm for end-to-end 3D point cloud segmentation. Moreover, group consistence has been proved in these methods to be an effective constraint to improve the segmentation performance [33]. In our method, to further boost the segmentation performance, a batch-consistency regularization term is formulated to enforce in-part similarity and inter-part dissimilarity over the batch of input 3D shapes on training.

3. Our Approach

In this section, we introduce three modules for our method. First, we introduce the coherent point transformer in section 3.1. In section 3.2, we illustrate the process of estimating the continuous part-specific probability distribution function. Section 3.3 discusses the details of batch-consistency regularization term. The model configurations and the settings for training are described in section 3.4.

3.1. Coherent Point Transformer

For a given 3D point cloud $\mathbf{X} \subset \mathbb{R}^3$, and a pool of S templates $\{\mathbf{Z_i}\}_{i=1,2,...,S}$, we denote $\mathbf{Z_X}$ as a retrieved template for input shape $\mathbf{X}$, where $dist(\mathbf{X}, \mathbf{Z_X}) = min\{dist(\mathbf{X}, \mathbf{Z_i})\}_{i=1,2,...,S}$. $dist(*, *) : \mathbb{R}^3 \times \mathbb{R}^3 \rightarrow R$, is a pre-set distance function defined on two point clouds. In this paper, we use Chamfer Distance for template retrieval. The coherent point transformer includes two parts: learning shape descriptor and coherent point morphing architecture as shown in Figure 3.

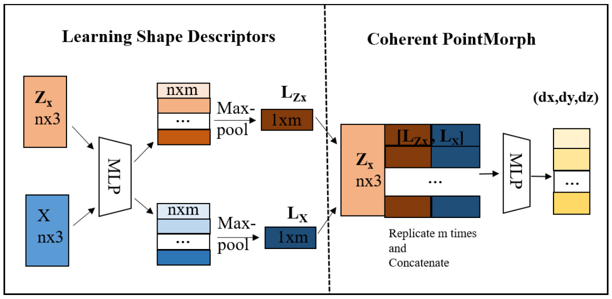

Figure 3. Coherent point transformer structure. For each point in shape X, we concatenate two shape descriptors to its coordinate for coherent drifts prediction, which is essential for keeping reasonable part-categorical labels for each point during the deformation process.

Learning Shape Descriptor. Given an input 3D shape $\mathbf{X}$ with its retrieved template $\mathbf{Z_X}$, we firstly learn the shape descriptors directly from the coordinates of 3D points. Let $(\mathbf{L_X}, \mathbf{L_{Z_x}})$ denotes the shape descriptors for input $(\mathbf{X}, \mathbf{Z_x})$, where $\mathbf{L_X}, \mathbf{L_{Z_x}} \subset \mathbb{R}^m$, as shown in Figure 3. Due to the irregularity of point cloud, we leverage the following encoder network for feature embedding, which includes t successive multi-layer perceptrons (MLP) with ReLu activation function $\{f_i\}_{i=1,2,...,t}$, such that: $f_i : \mathbb{R}^{\psi_i} \rightarrow \mathbb{R}^{\psi_{i+1}}$, where ψ_i

and ψ_{i+1} are the dimensions of the input and the output of the layer, respectively. The encoder network is defined as: $\forall(\mathbf{X}, \mathbf{Z_x})$,

$$\mathbf{L_X} = Maxpool\{f_t f_{t-1} ... f_1(\mathbf{x_j})\}_{\mathbf{x_j} \in \mathbf{X}} \tag{1}$$

$$\mathbf{L_{Z_X}} = Maxpool\{f_t f_{t-1} ... f_1(\mathbf{x_j})\}_{\mathbf{x_j} \in \mathbf{Z_X}} \tag{2}$$

Here, the symmetrical Maxpool function ensures an order-invariant feature learning [15].

Coherent point morphing architecture. For the next step, we define a deep neural network architecture for learning the coherent point drifts to align the template shape with the input shape as shown in Figure 3. This architecture includes successive multi-layer perceptrons (MLP) with ReLu activation function: $\{g_i\}_{i=1,2,...,s}$, such that: $g_i : \mathbb{R}^{\eta_i} \rightarrow \mathbb{R}^{\eta_{i+1}}$, where η_i and η_{i+1} are the dimensions of the input and the output of the layer. For each point $\mathbf{w_i} \in \mathbf{Z_x}$, the predicted drift vector $\mathbf{dw_i}$ is calculated as,

$$\mathbf{dw_i} = g_s g_{s-1} ... g_1([\mathbf{w_i}, \mathbf{L_X}, \mathbf{L_{Z_X}}]) \tag{3}$$

where [*,*] denotes the vector concatenation operation and the deformed template shape $\mathbf{Z'_X}$ is formulated as,

$$\mathbf{Z_x}' = T(\mathbf{Z_x}) = \{\mathbf{w_i} + \mathbf{dw_i}\}_{\mathbf{w_i} \in \mathbf{Z_X}} \tag{4}$$

In our coherent deformation transformer network, the coherency of predicted drifts is essential for preserving the label correspondence between the template shape and the deformed one, contributing to the weakly supervised nature of our method. Our intention is to deform the template to be as close as the target shape, while preserving reasonable part categorical label on the template shape. Our network can achieve this goal by concatenating the common global shape descriptors $[\mathbf{L_x}, \mathbf{L_{Z_X}}]$ with the coordinates of points on the original template [37]. We define the similarity measure between the input point cloud $\mathbf{X}$ and the transformed template point set $\mathbf{Z'_X}$ as the alignment loss function:

$$\mathcal{L}_{Alignment} = \sum_{x \in \mathbf{X}} \min_{y \in \mathbf{Z'_X}} ||x - y||_2^2 + \sum_{y \in \mathbf{Z'_X}} \min_{x \in \mathbf{X}} ||x - y||_2^2 \tag{5}$$

Ideally, the deformed template $\mathbf{Z'_X}$ with part annotations can geometrically align well with the input shape $\mathbf{X}$. Therefore, we use all deformed shapes with part labels for the estimation of the following part-specific probability distribution function.

3.2. Part-specific Density Estimator

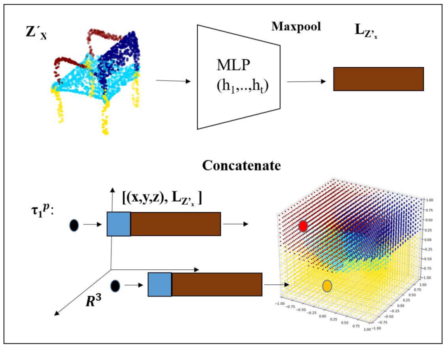

Figure 4. Illustration of part-specific density fitting process. Based on a given deformed template, we map each point in R^3 to a probability in two steps. Firstly, we leverage a MLP to learn a global descriptor from the deformed template. Secondly, we concatenate each point's coordinates to the learned global descriptor as input for further label prediction.

We propose a part-specific continuous probability distribution function f for every possible 3D point $\mathbf{x} \in \mathbb{R}^3$ instead of only for points in the target shape. We use the resulting function

$$\mathcal{F} : \mathbb{R}^3 \rightarrow \{0, 1, ..., K - 1\} \tag{6}$$

as the part-specific continuous function of the 3D object, where K indicates the number of different part categories. This function $\mathcal{F}$ can be implemented with a neural network structure that assigns every location $\mathbf{x} \in R^3$ a probability vector $\mathbf{v} \in [0, 1]^K$ for all K part categories. Our solution utilizes the deep neural network including multiple MLP layers with ReLu activation function $\{h_i\}_{i=1,2,...,q}$, such that: $h_i : \mathbb{R}^{\zeta_i} \rightarrow \mathbb{R}^{\zeta_{i+1}}$, where ζ_i and ζ_{i+1} are the dimension of the input and the output of the layer, to learn the global descriptor $L_{\mathbf{Z'_X}} \in \Omega$ (space of global shape descriptors of deformed templates) from the deformed template $\mathbf{Z_X}'$:

$$\mathbf{L_{Z'_X}} = Maxpool\{h_q ... h_2 h_1(\mathbf{x_j})\}_{\mathbf{x_j} \in \mathbf{Z'_X}} \tag{7}$$

Based on the deformed template shape $\mathbf{Z'_X}$ with part labels $\mathbf{Y_X}$ and the learned shape vector $\mathbf{L_{Z'_X}}$ as reference, we define the following continuous probability distribution function estimator τ,

$$\tau : \mathbb{R}^3 \times \Omega \rightarrow [0, 1]^K \tag{8}$$

In this paper, we implement a deep neural network structure to realize this function. We use the MLP layers with ReLu activation function $\{\tau_i\}_{i=1,2,...,r}$, such that: $\tau_i : \mathbb{R}^{\omega_i} \rightarrow \mathbb{R}^{\omega_{i+1}}$, where ω_i and ω_{i+1} are the dimensions of the input and the output of the layer. $\forall x \in \mathbb{R}^3$, $\mathbf{L_{Z'_{X_i}}} \in R^{t_q}$,

$$\tau(\mathbf{x}, \mathbf{L_{Z'_X}}) = \tau_r ... \tau_2([\mathbf{x}, \tau_1([\mathbf{x}, \mathbf{L_{Z'_X}}])]) \tag{9}$$

, where [*,*] represents the vector concatenation operation. We concatenate high dimensional feature with the coordinate of 3D points as the input for each neuron layer τ_i of our density estimator model. In this way, our density estimator can theoretically learn a spatially continuous probability function to encode part semantic of a 3D point at infinite resolutions. The "Continuity" of probability function consequently guarantee the continuous part label assignment $\tau(\mathbf{x_i}), \tau(\mathbf{x_j}) \in \mathbb{R}^K$ for neighbor points $\mathbf{x_i}, \mathbf{x_j} \in \mathbb{R}^3$. For the i-th sample $\mathbf{Z'_{X_i}}$ in a training batch, $i \in \{1, 2, ..., |\mathcal{B}|\}$, and its corresponding global descriptor $\mathbf{L_{Z'_X}}$, we evaluate mini-batch loss at each observed sample point $\mathbf{x_{ij}} \in \mathbf{Z'_{X_i}}$, $j \in \{1, 2, ..., N\}$, with corresponding label $\mathbf{y_{ij}} \in \mathbf{Y_{X_i}}$:

$$\mathcal{L}_{cross-entropy} = \frac{1}{|\mathcal{B}|} \sum_{i=1}^{|\mathcal{B}|} \sum_{j=1}^{N} \mathcal{L}(\tau(\mathbf{x_{ij}}), \mathbf{y_{ij}}) \tag{10}$$

, where $\mathcal{L}(*, *)$ represents the cross-entropy classification loss.

3.3. Batch-consistency Regularization

The similarity-based retrieval of a shape template in the "Template Selector" component cannot guarantee the part semantic consistency between the input shape and the template shape. For example, the retrieved template shape might have more part categories than the input shape, as shown in the third columns in Figure 8. This inconsistency of 3D point cloud segmentation unavoidably causes the errors of probability density estimation, consequently lead to the errors in part label assignment. To address this issue, in this section, we introduce a batch consistency loss to regularize our probability distribution estimator network, as shown in Figure 5. For each part category k and each point from $\{\mathbf{x_{ij}} \in \mathbf{Z'_{X_i}} | \mathbf{y_{ij}} = \mathbf{k}\}$, we extract the part-specific features by aggregating all point features in this category, calculated as,

$$\mathbf{L_i^k} = Maxpool\{h_q...h_2h_1(\mathbf{x_{ij}})|\mathbf{y_{ij}} = k\}_{\mathbf{x_{ij}} \in Z'_{X_i}} \tag{11}$$

And we can compute the part-specific average in-group feature for each part as: $\mathbf{L^k} = \frac{1}{|\mathcal{B}|}\sum_{i=1}^{|\mathcal{B}|} \mathbf{L_i^k}$. Our correlation loss is defined as:

$$\begin{aligned} \mathcal{L}_{corr} = & \sum_{i=1}^{|\mathcal{B}|} \sum_{k_1=1}^{K} \sum_{k_2 \neq k_1} corr(\mathbf{L^{k_1}}, \mathbf{L_i^{k_2}}) \\ & - \sum_{i=1}^{|\mathcal{B}|} \sum_{k=1}^{K} corr(\mathbf{L^k}, \mathbf{L_i^k}) \end{aligned} \tag{12}$$

, where corr(*,*) represents the correlation between two vectors. The first correlation term in the loss encourages the shapes from different part categories to have a lower correlation and the second term encourages the shapes from same part categories to have a higher correlation. To put them all together, our coherent point transformer network and part-specific probability distribution estimator network with batch-consistency regularization can be trained in an end-to-end manner by minimizing the following loss function:

$$\mathcal{L} = \mathcal{L}_{cross-entropy} + \lambda_1 * \mathcal{L}_{alignment} + \lambda_2 * \mathcal{L}_{corr} \tag{13}$$

where λ_1 and λ_2 are hyper-parameters to control the balance of different loss terms.

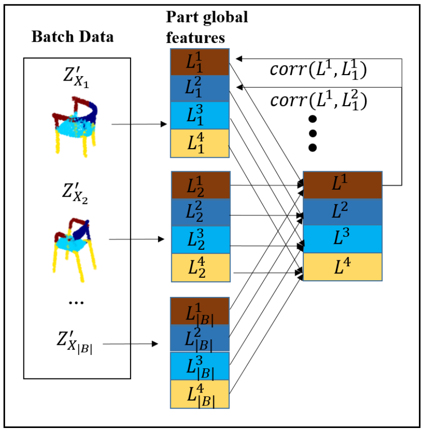

Figure 5. Illustration of batch-consistency. For deformed templates in a batch, we accumulate their features for each part category as a group of features. By comparing individual's part features with group's part features, consistency is expected to be hold. This loss helps adjusting the density estimation process to not simply rely on the deformed templates.

3.4. Training Paradigm

To learn the shape descriptor for input shape and template shape as described in section 3.1, we use 5 MLP layers with dimensions $(16, 64, 128, 256, 512)$ and a Maxpool layer to convert it to a 512-dimensional descriptor. To learn the coherent point drifts for aligning the template shape with the input shape, we use 3 MLP layers with dimensions $(256, 128, 3)$. In our Part-Specific Density Estimator module, we use another 5 MLP layers with dimensions $(16, 64, 128, 256, 512)$ to build our part-specific density estimator network. We use ReLU activation function and implement batch normalization [7] for every layer except the output layer. In our experiments, we set λ_1 to 100 with an exponential decay of 0.999, and we set λ_2 to 0.5. Adam optimizer with an initial learning rate of 1e-3 and an exponential decay of 0.9995 was applied for model optimization. Our model was implemented using Tensorflow framework. It took about 6 hours to train our model per category on a single Nvidia Tesla K80 GPU.

4. Experiments

In this section, we carry out experiments to demonstrate the effectiveness of the modules in our method and evaluate the 3D point cloud segmentation performance of WPS-

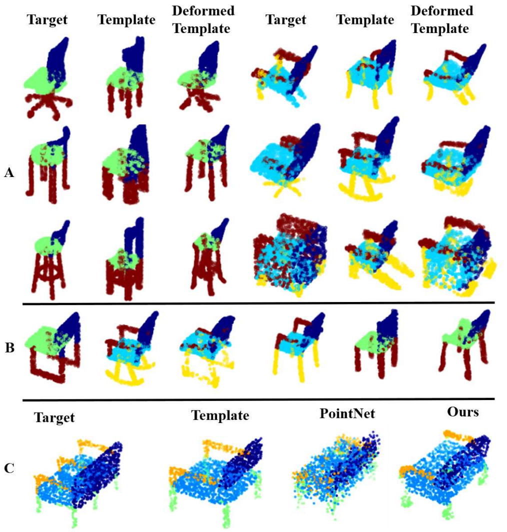

Figure 6. Deformation results. For part A and B, we demonstrate the deformed template shapes in comparison with their target shapes. For part B, we select two typical cases which might increase the difficulty of further fitting part-specific density function as "failed" cases. In part C, we compare our deformation result with the PointNet based auto-encoder.

Net. In section 4.1, we describe the dataset in our experiments. Section 4.2 validates the effectiveness of our coherent point transformer. We demonstrate the superiority of the proposed part-specific density estimator in section 4.3. In section 4.4, we further show performance boost from batch-consistency regularization. Comparison with supervised methods is demonstrated in section 4.6.

4.1. Dataset

We evaluate the proposed method on ShapeNet part dataset [3]. This dataset includes 16,881 shapes from 16 object categories. For each category, we randomly choose 10 objects with annotated labels to form a mini-pool of shape templates. For a fair comparison, we report the performance of our method on the test dataset following the official train test split. The mean IoU (Intersection-over-Union) of each category is calculated as an average value over all shapes in that category.

4.2. Validation of Deformation network

Experiment Setting: In this part, we conduct experiments to demonstrate the effectiveness of the coherent point transformer for 3D point cloud segmentation. As we mentioned above, our part-specific probability density estimation network is trained on the deformed template shapes with part labels transferred from original templates. To guarantee the label quality on these deformed shapes, our deformation network is designed with the ability to produce deformed shapes which can not only be aligned with original input shape but also preserve reasonable correspondence between deformed templates and original templates. To prove the superiority of our coherent point transformer network, we compared our method with a PointNet-based auto-encoder network [3]. Both networks share the same encoder part. We evaluated their deformation quality (Chamfer distance between target shapes with transformed shapes) and the coherency of the deformation process. Figure 6 shows selected examples of the deformation results of both methods.

Results and Discussion: To compare the deformation quality of PointNet-based auto-encoder and our coherent point transformer network, as indicated by the convergence Chamfer distance (0.00168 and 0.00177 respectively), both two models can reach a small Chamfer distance between the deformed templates with the target shapes. As shown in Figure 6 part A, our coherent point transformer tends to predict smooth deformation field, which results in a high-quality dense correspondence between the input template and the deformed one. In comparison, part C shows that PointNet based auto-encoder module produces noncontinuous deformation field and then, fails to preserve such correspondence. Our coherent point estimator provides our method with the ability to transfer the part labels from the template to the deformed shapes, which contributes to the weakly supervised nature of segmentation network. Part B demonstrates some cases which may cause problem for learning part-specific density.

4.3. Validation of part-specific density estimator

Experiment Setting: In this experiment, we evaluate the effectiveness of the proposed continuous density estimator module for 3D point cloud segmentation. After generating the deformed templates with part labels, the simplest way to transfer labels to target shape is by nearest neighbor searching strategy (NN). That is, for all point $\mathbf{x_{ij}} \in X_i$, we generate its label prediction by referring the label of its nearest point in the template shape $\mathbf{Z}'_{\mathbf{X_i}}$. We use this labeling strategy as a baseline for comparison with our continuous probability distribution estimator module.

Method	Chair	Table	Lamp
Nearest Neighbor Search	81.2	70.3	66.4
Ours, w/o batch consistency	82.3	71.7	68.1
Ours, w batch consistency	83.4	72.2	68.7

Table 1. Comparison of the IoU metric between NN (first row), our continuous probability estimation network (second row) and our model with batch-consistency (third row).

Results and Discussion: Table 1 lists the overall segmentation performance of our method and the NN. As indicated by the IoU value on chair category, our continuous probability estimation module achieves better segmentation performance with an improvement of 1.3%. Though our method gains the incremental IoU compared to the NN method, we observed the following interesting label prediction by our WPS-Net. As shown in Figure 7, the input shape has an arm and leg in the middle while the template shape does not have those two parts. Therefore, based on the NN's prediction, it is reasonable to see that those two parts are mis-labelled as the "seat" since they are nearest neighbors as indicated in the fourth column (two red circled regions). In contrast, our WPS-Net gains its advantage of a probability based label assignment strategy that predicts the correct labels for middle arm and leg of the input shape even though the template shape does not have the corresponding parts as indicated in last column (two red circled regions) of Figure 7.

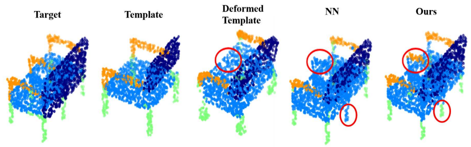

Figure 7. Comparison of the segmentation results based on the deformed template between using Nearest Neighbors method and our part-specific density estimator.

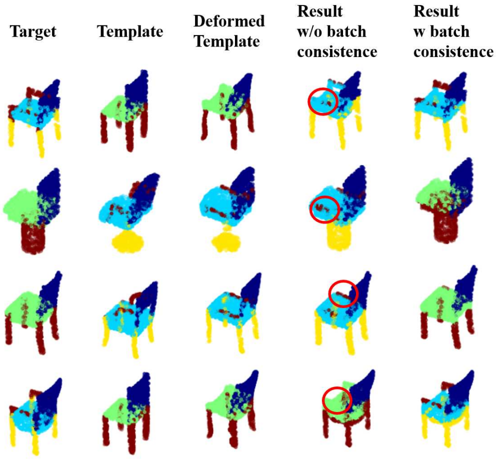

Figure 8. Comparison of the segmentation results between our method with and without batch consistency.

# points	IoU%	# templates	IoU%
512	79.6	5	77.9
1024	78.9	10	83.4
2048	83.4	20	85.7

Table 2. Comparison of the IoU metric of Chair category using shapes with different number of sampled points (left) and using different number of templates (right).

4.4. Validation of batch-consistency

Experiment Setting: In this experiment, we evaluate the functional benefit of batch consistency loss that contributes better segmentation of 3D point cloud. To demonstrate the effectiveness of batch consistency loss module, we investigate a few case studies that the template shape has more/less parts than that of input shape shown in Figure 8 and the overall quantitative result is shown in Table 1. We conduct experiments on chair, table and lamp categories.

Results and Discussion: As indicated in row 2 and 3 in Figure 8, the input shapes has three parts without arm part but the template has an additional arm part. To align the template shape with the input one, the arm part of the template chair is forced to be deformed as the seat. In this way, it transfers the "arm" label to the seat region, which consequently leads to a bias probability estimation for part label description. As shown in fourth column of Figure 8, without the rectification by batch consistency module, there are quite a few points in the seat area (red circled regions) that are mistakenly labelled as the arm. In contrast, with batch-consistency rectification, a large portion of mistaken label assignments can be rectified as shown in last column of Figure 8 (red circled regions). As shown in Table 1, the overall IoU improved from 82.3% to 83.4% with the batch-consistency loss. Regarding the improvement of our batch-consistency module, the quantitative performance looks not quite significant. However, batch consistency adjustment is specially designed for the case when we have more part categories in the template than in the target shape as shown in the first case of row B in Figure 6. During our experiments, most cases do not suffer this problem.

4.5. Studies on number of sampled points and number of templates

Experiment Setting: In this experiment, we conduct two experiments to verify our model's performance using various number of sampled points of 3D shapes and using various number of randomly chosen templates. For the first experiment on number of sampled points in 3D shape, we choose three different levels: 512, 1024 and 2048. For the second experiment on number of randomly chosen templates, we choose three different levels: 5, 10 and 20.

Results and Discussion: As shown in Table 2, when the number of sampled points in each shape decreases from 2048 to 1024, the IoU on Chair category drops 4.5%. But the performance degradation is not obvious when sampling points from 1024 to 512. When the number of templates increases, the model's performance can improve quickly.

4.6. Comparison with supervised methods

Experiment Setting: In this experiment, we evaluate the overall segmentation performance of our model on more

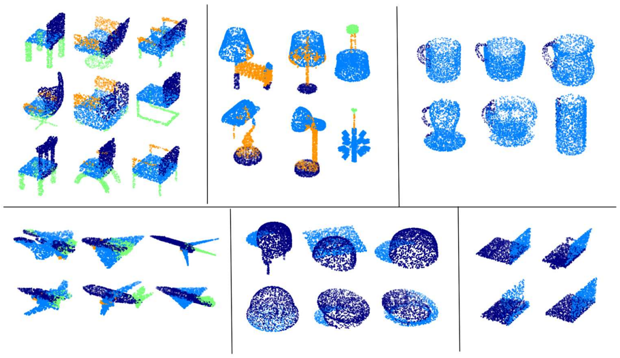

Figure 9. Randomly selected examples of the weakly supervised 3D point cloud segmentation results.

shape categories and compare the performance with current state-of-the-art supervised methods. For a fair comparison, we trained supervised model for each shape category independently using the same randomly selected 10 samples which are used as our template pool. We use all the available dataset including annotated samples with target shapes and other supervised methods use the annotated examples during training. However, our setting does not require any ad-ditional information in comparison to supervised methods. In addition, we report the performance of PointNet model trained with all samples in the original training dataset for reference. Since our method may depend on the template selection, we rerun our model with comparative methods 5 times on different randomly selected template pools for chair category as an additional experiment.

Categories	Ours	[15]	[17]	[35]	[15]
#Samples	10	10	10	10	All
#Parameters	2.6M	3.5M	1.4M	6.9M	3.5M
Airplane	**67.3**	63.3	62.3	65.1	83.4
Bag	**74.4**	64.9	67.4	68.2	78.7
Cap	**86.3**	75.2	80.0	80.7	82.5
Chair	**83.4**	73.8	61.6	66.1	89.6
Lamp	**68.7**	63.8	57.8	60.2	80.8
Laptop	**93.8**	87.3	94.2	93.7	95.3
Mug	**90.9**	80.9	83.1	86.0	93.0
Table	**74.2**	72.2	72.2	72.5	80.6
Mean	**79.8**	72.7	72.3	74.1	85.5

Table 3. Quantitative result. Comparison with supervised methods on randomly selected small training samples.

Results and Discussion: Table 3 shows the per-category IoU for each model. As one can see from Table 3, with comparable number of parameters, our model achieves better performance on all shape categories compared to the supervised method using same training examples. Our method reports an improvement of 5.7% on category mean IoU over the PointConv [35]. When compared with the PointNet using all training samples, our method falls behind with a reasonable margin. When training size is small, state-of-the-art supervised method PointConv behaves worse than other methods. The model with high complexity might overfit the small training data and loose predicting power for testing dataset. Randomly selected examples are shown in Figure 9. Table 4 confirms that our method is more stable than other methods given different templates.

Methods	Chair IoU%
PointNet [15]	73.3 ± 1.42
PointNet++ [17]	61.5 ± 1.48
PointConv [35]	67.2 ± 3.5
Ours	**82.7** ± **0.38**

Table 4. Quantitative result. Comparison with supervised methods on five randomly selected small training samples using Chair category.

5. Future work

Assuming a small number of annotated templates, a more efficient design of using label information from the pool of templates worth more analysis in the future work. In the current work, for a given target 3D shape, we define a global geometric similarity metric to only retrieve the most similar template from the pool of the annotated shapes, but we eliminate the possible useful information from other less similar templates. In practice, we notice that those less similar templates can also contribute the valuable label information based on their local parts/structures, thus further enhancing the performance of few-shot learning of part-specific probability space. In the future work, researches on the deformation of a group of templates towards the target shape to better combine the information from all templates in the annotated pool will be conducted.

6. Conclusion

We introduce a novel weakly supervised paradigm for learning 3D point cloud segmentation. For this quite challenging task, we accordingly propose a pipeline including 1) coherent point transformer to coherently morph the retrieved template towards input shape while maintaining the part correspondence, 2) a continuous probability distribution estimation network to encode part label description of a 3D point of the deformed template and 3) the batch-consistency regularization loss to further enforce in-part similarity and inter-part dissimilarity. We experimentally verified the effectiveness of each module and achieved a remarkable weakly supervised 3D point cloud segmentation result on the ShapeNet 3D point cloud segmentation dataset.

References

[1] Jianchun Chen, Lingjing Wang, Xiang Li, and Yi Fang. Arbicon-net: Arbitrary continuous geometric transformation networks for image registration. In *Advances in Neural Information Processing Systems*, pages 3410–3420, 2019.

[2] Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *European conference on computer vision*, pages 628–644. Springer, 2016.

[3] Haoqiang Fan, Hao Su, and Leonidas J Guibas. A point set generation network for 3d object reconstruction from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 605–613, 2017.

[4] Ran Gal and Daniel Cohen-Or. Salient geometric features for partial shape matching and similarity. *ACM Transactions on Graphics (TOG)*, 25(1):130–150, 2006.

[5] Ruizhen Hu, Lubin Fan, and Ligang Liu. Co-segmentation of 3d shapes via subspace clustering. In *Computer graphics forum*, volume 31, pages 1703–1713. Wiley Online Library, 2012.

[6] Qixing Huang, Vladlen Koltun, and Leonidas Guibas. Joint shape segmentation with linear programming. In *ACM transactions on graphics (TOG)*, volume 30, page 125. ACM, 2011.

[7] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[8] Mingyang Jiang, Yiran Wu, and Cewu Lu. Pointsift: A sift-like network module for 3d point cloud semantic segmentation. *arXiv preprint arXiv:1807.00652*, 2018.

[9] Evangelos Kalogerakis, Melinos Averkiou, Subhransu Maji, and Siddhartha Chaudhuri. 3d shape segmentation with projective convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3779–3788, 2017.

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

[11] Xiang Li, Hanzhang Cui, John-Ross Rizzo, Edward Wong, and Yi Fang. Cross-safe: A computer vision-based approach to make all intersection-related pedestrian signals accessible for the visually impaired. In *Science and Information Conference*, pages 132–146. Springer, 2019.

[12] Xiang Li, Lingjing Wang, and Yi Fang. Pc-net: Unsupervised point correspondence learning with neural networks. In *2019 International Conference on 3D Vision (3DV)*, pages 145–154. IEEE, 2019.

[13] 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.

[14] Min Meng, Jiazhi Xia, Jun Luo, and Ying He. Unsupervised co-segmentation for 3d shapes using iterative multi-label optimization. *Computer-Aided Design*, 45(2):312–320, 2013.

[15] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 1(2):4, 2017.

[16] Charles R Qi, Hao Su, Matthias Nießner, Angela Dai, Mengyuan Yan, and Leonidas J Guibas. Volumetric and multi-view cnns for object classification on 3d data. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5648–5656, 2016.

[17] 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.

[18] Danilo Jimenez Rezende, SM Ali Eslami, Shakir Mohamed, Peter Battaglia, Max Jaderberg, and Nicolas Heess. Unsupervised learning of 3d structure from images. In *Advances in Neural Information Processing Systems*, pages 4996–5004, 2016.

[19] Lior Shapira, Ariel Shamir, and Daniel Cohen-Or. Consistent mesh partitioning and skeletonisation using the shape diameter function. *The Visual Computer*, 24(4):249, 2008.

[20] Zhenyu Shu, Chengwu Qi, Shiqing Xin, Chao Hu, Li Wang, Yu Zhang, and Ligang Liu. Unsupervised 3d shape segmentation and co-segmentation via deep learning. *Computer Aided Geometric Design*, 43:39–52, 2016.

[21] Oana Sidi, Oliver van Kaick, Yanir Kleiman, Hao Zhang, and Daniel Cohen-Or. *Unsupervised co-segmentation of a set of shapes via descriptor-space spectral clustering*, volume 30. ACM, 2011.

[22] Edward Smith, Scott Fujimoto, and David Meger. Multi-view silhouette and depth decomposition for high resolution 3d object representation. In *Advances in Neural Information Processing Systems*, pages 6478–6488, 2018.

[23] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2530–2539, 2018.

[24] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 945–953, 2015.

[25] Chu Wang, Marcello Pelillo, and Kaleem Siddiqi. Dominant set clustering and pooling for multi-view 3d object recognition. In *Proceedings of British Machine Vision Conference (BMVC)*, volume 12, 2017.

[26] Lingjing Wang, Jianchun Chen, Xiang Li, and Yi Fang. Non-rigid point set registration networks. *arXiv preprint arXiv:1904.01428*, 2019.

[27] Lingjing Wang and Yi Fang. Unsupervised 3d reconstruction from a single image via adversarial learning. *arXiv preprint arXiv:1711.09312*, 2017.

[28] Lingjing Wang, Cheng Qian, Jifei Wang, and Yi Fang. Unsupervised learning of 3d model reconstruction from hand-drawn sketches. In *Proceedings of the 26th ACM international conference on Multimedia*, pages 1820–1828, 2018.

[29] Meng Wang, Lingjing Wang, and Yi Fang. 3densinet: A robust neural network architecture towards 3d volumetric object prediction from 2d image. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 961–969, 2017.

[30] Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. Pixel2mesh: Generating 3d mesh models from single rgb images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 52–67, 2018.

[31] Peng-Shuai Wang, Yang Liu, Yu-Xiao Guo, Chun-Yu Sun, and Xin Tong. O-cnn: Octree-based convolutional neural networks for 3d shape analysis. *ACM Transactions on Graphics (TOG)*, 36(4):72, 2017.

[32] Peng-Shuai Wang, Chun-Yu Sun, Yang Liu, and Xin Tong. Adaptive o-cnn: a patch-based deep representation of 3d shapes. In *SIGGRAPH Asia 2018 Technical Papers*, page 217. ACM, 2018.

[33] Yunhai Wang, Shmulik Asafi, Oliver Van Kaick, Hao Zhang, Daniel Cohen-Or, and Baoquan Chen. Active co-analysis of a set of shapes. *ACM Transactions on Graphics (TOG)*, 31(6):165, 2012.

[34] Congcong Wen, Lina Yang, Xiang Li, Ling Peng, and Tianhe Chi. Directionally constrained fully convolutional neural network for airborne lidar point cloud classification. *ISPRS Journal of Photogrammetry and Remote Sensing*, 162:50–62, 2020.

[35] Wenxuan Wu, Zhongang Qi, and Li Fuxin. Pointconv: Deep convolutional networks on 3d point clouds. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[36] Zizhao Wu, Yunhai Wang, Ruyang Shou, Baoquan Chen, and Xinguo Liu. Unsupervised co-segmentation of 3d shapes via affinity aggregation spectral clustering. *Computers & Graphics*, 37(6):628–637, 2013.

[37] Yaoqing Yang, Chen Feng, Yiru Shen, and Dong Tian. Foldingnet: Point cloud auto-encoder via deep grid deformation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 206–215, 2018.

[38] Shuaihang Yuan and Yi Fang. Ross: Robust learning of one-shot 3d shape segmentation. In *The IEEE Winter Conference on Applications of Computer Vision*, pages 1961–1969, 2020.

[39] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4490–4499, 2018.

[40] Jing Zhu and Yi Fang. Learning object-specific distance from a monocular image. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3839–3848, 2019.