Parts4Feature: Learning 3D Global Features from Generally Semantic Parts in Multiple Views

Zhizhong Han[1,2], Xinhai Liu[1], Yu-Shen Liu[1]*, Matthias Zwicker[2]
[1]School of Software, Tsinghua University, Beijing, China
Beijing National Research Center for Information Science and Technology (BNRist)
[2]Department of Computer Science, University of Maryland, College Park, USA
h312h@umd.edu, lxh17@mails.tsinghua.edu.cn, liuyushen@tsinghua.edu.cn, zwicker@cs.umd.edu

Abstract

Deep learning has achieved remarkable results in 3D shape analysis by learning global shape features from the pixel-level over multiple views. Previous methods, however, compute low-level features for entire views without considering part-level information. In contrast, we propose a deep neural network, called *Parts4Feature*, to learn 3D global features from part-level information in multiple views. We introduce a novel definition of generally semantic parts, which Parts4Feature learns to detect in multiple views from different 3D shape segmentation benchmarks. A key idea of our architecture is that it transfers the ability to detect semantically meaningful parts in multiple views to learn 3D global features. Parts4Feature achieves this by combining a local part detection branch and a global feature learning branch with a shared region proposal module. The global feature learning branch aggregates the detected parts in terms of learned part patterns with a novel multi-attention mechanism, while the region proposal module enables locally and globally discriminative information to be promoted by each other. We demonstrate that Parts4Feature outperforms the state-of-the-art under three large-scale 3D shape benchmarks.

1 Introduction

Learning 3D global features from multiple views is an effective approach for 3D shape understanding. A widely adopted strategy is to leverage deep neural networks to aggregate features hierarchically extracted from pixel-level information in each view. However, current approaches can not employ part-level information. In this paper, we show for the first time how extracting part-level information over multiple views can be leveraged to learn 3D global features. We demonstrate that this approach further increases the discriminability of 3D global features and outperforms the state-of-the-art methods on large scale 3D shape benchmarks.

It is intuitive that learning to detect and localize semantic parts could help classify shapes more accurately. Previous studies on fine-grained image recognition also employ this intuition by combining local part detection and global feature learning together. To learn highly discriminative features to distinguish subordinate categories, these methods try to first detect important parts, such as heads, wings and tails of birds, and then collect these part features into a global feature. However, these methods do not tackle the challenges that we are facing in the 3D domain. First, these methods require ground truth parts with specified semantic labels, while 3D shape classification benchmarks do not provide such kind of labels. Second, the part detection knowledge learned by these methods cannot be transferred for general purpose use, such as non-fine-grained image classification, since it is specified for particular shape classes. Third, these methods are not designed to aggregate part information from multiple images, corresponding to multiple views of a 3D shape in our scenario. Therefore, simultaneously learning part detection and further aggregating part-level information from multiple views become a unique challenge in 3D global feature learning.

To address these issues, we propose *Parts4Feature*, a deep neural network to learn 3D global features from semantic parts in multiple views. With a novel definition of generally semantic parts (GSPs), Parts4Feature learns to detect GSPs in multiple views from different 3D shape segmentation benchmarks. Moreover, it learns a 3D global feature from shape classification data sets, by transferring the learned knowledge of part detection, and leveraging the detected GSPs in multiple views. Specifically, Parts4Feature is mainly composed of a local part detection branch and a global feature learning branch. Both branches share a region proposal module, which enables locally and globally discriminative information to get promoted by each other.

The local part detection branch employs a novel neural network derived from Fast R-CNN [Girshick, 2015] to learn to detect and localize GSPs in multiple views. In addition, the global feature learning branch incrementally aggregates the detected parts in terms of learned part patterns with multi-attention. We propose a novel multi-attention mechanism to further increase the discriminability of learned features by not only highlighting the distinctive parts and part patterns but also depressing the ambiguous ones. Our novel view aggregation based on semantic parts prevents information loss caused by the widely used pooling, and it can understand each detected part in a more detailed manner. In summary, our contribu-

---

tions are as follows:

i) We propose Parts4Feature, a novel deep neural network to learn 3D global features from semantic parts in multiple views, by combining part detection and global feature learning together.

ii) We show that the novel structure of Parts4Feature is capable of learning and transferring universal knowledge of part detection, which allows Parts4Feature to leverage discriminative information from another source (3D shape segmentation) for 3D global feature learning.

iii) Our global feature learning branch introduces a novel view aggregation based on semantic parts, where the proposed multi-attention further improves the discriminability of learned features.

2 Related work

Mesh-based deep learning models. To directly learn 3D features from 3D meshes, different novel concepts, such as circle convolution [Han and others, 2016], mesh convolution [Han and others, 2017] were proposed to perform in deep learning models. These methods aim to learn global or local features from the geometry and spatial information on meshes to understand 3D shapes.

Voxel-based deep learning models. Similar to images, voxels have regular structure to be learned by deep learning models, such as CRBM [Wu and others, 2015], fully convolutional denoising autoencoders [Sharma *et al.*, 2016], C-NNs [Qi *et al.*, 2016], GAN [Wu and others, 2016]. These methods usually employ 3D convolution to better capture the contextual information in local regions. Moreover, Tags2Parts [Muralikrishnan *et al.*, 2018] discovered semantic regions that strongly correlate with user-prescribed tags by learning from voxels using a novel U-Net.

Deep learning models for point clouds. As a series of pioneering work, PointNet++ [Qi and others, 2017] inspired various supervised methods to understand point clouds. Through self-reconstruction, FoldingNet [Yang *et al.*, 2018] and LatentGAN [Achlioptas and others, 2018] learned global features with different unsupervised strategies.

View-based deep learning models. Similar to the light field descriptor (LFD), GIFT [Bai and others, 2017] measured the difference between two 3D shapes using their corresponding view feature sets. Moreover, pooling panorama views [Shi and others, 2015; Sfikas and others, 2017] or rendered views [Su and others, 2015; Han *et al.*, 2019] is more widely used to learn global features. Different improvements from camera trajectories [Johns *et al.*, 2016], view aggregation [Wang *et al.*, 2017; Han and others, 2019a], pose estimation [Kanezaki *et al.*, 2018] are also presented. However, these methods can not leverage part-level information. In contrast, Parts4Feature learns and transfers universal knowledge of part detection to facilitate 3D global feature learning.

3 Parts4Feature

Overview. Parts4Feature consists of three main components as shown in Fig. 1: a local part detection branch L, a global feature learning branch G, and a region proposal module R,

where R is shared by L and G and receives multiple views of a 3D shape as input. We train Parts4Feature simultaneously under a local part detection benchmark $\mathbf{\Phi}$ and a global feature learning benchmark $\mathbf{\Psi}$. The local part detection branch L learns to identify GSPs in multiple views under $\mathbf{\Phi}$, while G learns a global feature from the detected GSPs in multiple views under $\mathbf{\Psi}$.

For a 3D shape m in either $\mathbf{\Phi}$ or $\mathbf{\Psi}$, we capture V views v^i around it, forming a view sequence $\boldsymbol{v} = \{v^i | i \in [1, V]\}$. First, the region proposal module R provides the features $\boldsymbol{f}_j^i$ of regions r_j^i proposed in each view v^i, where $j \in [1, J]$. Then, by analyzing the region features $\boldsymbol{f}_j^i$ in $\boldsymbol{v}$, branch L learns to predict what and where GSPs are in multiple views. Finally, by aggregating the features $\boldsymbol{f}_k^i$ of the top K region proposals r_j^i in each v^i in $\boldsymbol{v}$, the global feature learning branch G produces the global feature $\boldsymbol{f}$ of shape m. Our approach to aggregating region proposal features is based on N semantic part patterns $\boldsymbol{\theta}_n$ with multi-attention for 3D shape classification, where $\boldsymbol{\theta}_n$ are learned across all views in the global feature learning benchmark $\mathbf{\Psi}$.

Generally semantic parts. We define a GSP as a local part in any semantic part category of any shape class, such as engines of airplanes or wheels of cars. Although our concept of GSPs simplifies all semantic part categories into a binary label by only determining whether a part is semantic or not, this allows us to exploit discriminative, part-level information from several different 3D shape segmentation benchmarks for global feature learning.

We use three 3D shape segmentation benchmarks involved in [Kalogerakis and others, 2017], including ShapeNetCore, Labeled-PSB, and COSEG to construct the local part detection benchmark $\mathbf{\Phi}$ and provide ground truth GSPs. We also split the 3D shapes in each segmentation benchmark into training and test sets according to [Kalogerakis and others, 2017]. Fig. 2 shows the construction of ground truth GSPs. For each view v^i of a 3D shape m shown in Fig. 2(a), we obtain its ground truth segmentation visualized in Fig. 2(b) from the shape segmentation benchmark. Then, we can isolate each part category to precisely locate GSPs, as shown from Fig. 2(c) to Fig. 2(f). We emphasize each isolated part category in blue, where we locate the corresponding GSPs by computing the bounding box (red) of the colored regions. Finally, we show all GSPs in view v^i in Fig. 2(g). We collect all GSPs of shape m by repeating these procedures in all its V views. Note that we omit small GSPs (for example the landing gear in Fig. 2(f)) whose bounding boxes are smaller than 0.45 of the max bounding box in the same part category.

Region proposal module R. R provides region proposals r_j^i in all views v^i and their features $\boldsymbol{f}_j^i$, which are then forwarded to the local part detection and global feature learning branches. Shared by all views v^i in $\boldsymbol{v}$, R is composed of a Deep Convolutional Network (DCN), and a Region Proposal Network (RPN) with Region of Interest (RoI) pooling [Girshick, 2015].

DCN is modified from a VGG_CNN_M_1024 network [Chatfield *et al.*, 2014], and it produces feature $\boldsymbol{f}^i$ for each view v^i as 512 feature maps of size 12×12. Based on $\boldsymbol{f}^i$, RPN then proposes regions r_j^i in a sliding-window man-
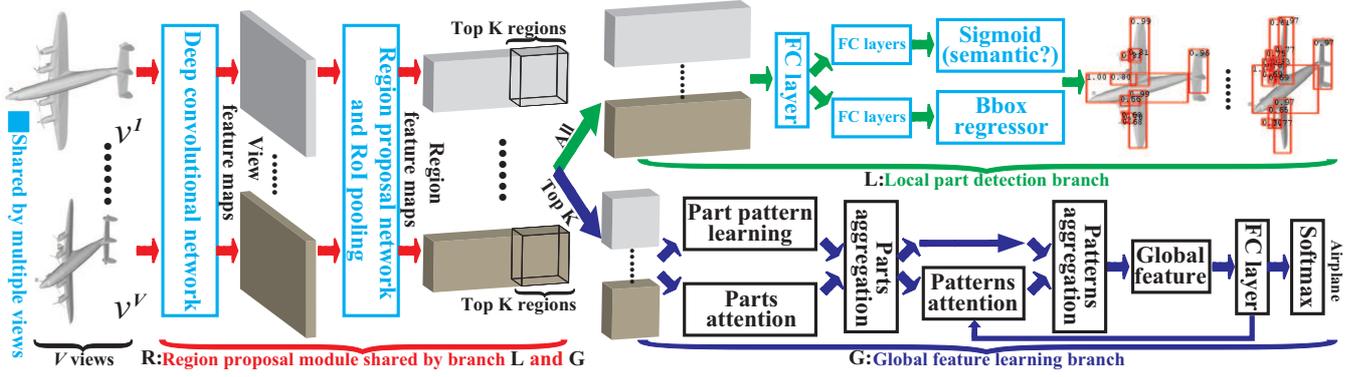
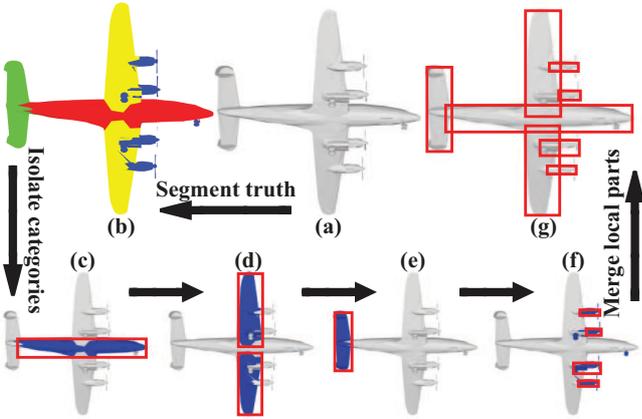Figure 1: The demonstration of Parts4Feature framework.

Figure 2: The illustration of ground truth GSPs construction.

ner. At each sliding-window location centered at each pixel of $\boldsymbol{f}^i$, a region r_j^i is proposed by determining its location t_R and predicting GSP probabilities p_R with an anchor. The location t_R is a four dimensional vector representing center, width and height of the bounding box. We use 6 scales and 3 aspect ratios to yield $6 \times 3 = 18$ anchors, which ensures a wide range of sizes to accommodate region proposals for GSPs that may be partially occluded in some views. The 6 scales relative to the size of the views are $[1, 2, 4, 8, 16, 32]$, and the 3 aspect ratios are $1 : 1$, $1 : 2$, and $2 : 1$. Altogether, this leads to $J = 2592 = 12 \times 12 \times 18$ regions r_j^i in each view v^i.

To train RPN to predict GSP probabilities p_R, we assign a binary label to each region r_j^i indicating whether r_j^i is a GSP. We assign a positive label if the Intersection-over-Union (IoU) overlap between r_j^i and any ground-truth GSP in v^i is higher than a threshold S_R, and we use a negative label otherwise. In each view v^i we apply RoI pooling over regions given by $\{t_R\}$ on feature maps $\boldsymbol{f}^i$. Hence, the features $\boldsymbol{f}_j^i$ of all J region proposals r_j^i are $512 \times 7 \times 7$ dimensional vectors, which we forward to the local part detection branch L. In addition, we provide the features $\boldsymbol{f}_k^i$ of the top K regions r_j^i according to their GSP probability p_R to the global feature learning branch G.

Local part detection branch L. The purpose of this branch

is to detect GSPs from the J region proposals r_j^i in each view v^i. We employ L as an enhancer of RPN, where L aims to learn what and where GSPs are in v^i without anchors in a more precise manner. The intuition is that this in turn pushes RPN to propose more GSP-like regions, which we provide to the global feature learning branch G.

We feed the region features $\boldsymbol{f}_j^i$ of r_j^i into a sequence of fully connected layers followed by two output layers. The first one estimates the GSP probability p_L that r_j^i is a GSP using a sigmoid function as an indicator. The second one predicts the corresponding part location t_L using a bounding box regressor, where t_L represents the same bounding box parameters as t_R in RPN. Similar to the threshold S_R in R, L employs another threshold S_L to assign positive and negative labels for training. Denoting the ground truth probabilities and locations of positive and negative samples in RPN R as p' and t', and similarly for L as p'' and t'', the objective function of Parts4Feature for GSP detection is formed by the loss in R and L, which is defined for each region proposal as follows,

$$
\begin{aligned}
O_{R+L}&(p_R, p_L, p', p'', t_R, t_L, t', t'') = \\
&O_p(p_R, p') + \lambda O_t(t_R, t') + O_p(p_L, p'') + \lambda O_t(t_L, t''),
\end{aligned}
\tag{1}
$$

where O_p measures the accuracy in terms of GSP probability by the cross-entropy function of positive labels, while O_t measures the accuracy in terms of location by the robust L_1 function as in [Ren and others, 2015]. The parameter λ balances O_p and O_t in both R and L. It works well in our experiments with a value of 1. In summary, Parts4Feature has the powerful ability to detect GSPs by simultaneously leveraging the view-level features $\boldsymbol{f}^i$ in R and the part-level features $\boldsymbol{f}_j^i$ in L, which addresses the difficulty of GSP detection from multiple views caused by rotation and occlusion effects.

Global feature learning branch G. This branch learns to map the features $\boldsymbol{f}_k^i$ of the top K region proposals r_k^i in each view v^i in $\boldsymbol{v}$ to the 3D global feature $\boldsymbol{f}$. To avoid information loss caused by widely used pooling for aggregation, G incrementally aggregates all $V \times K$ region features $\boldsymbol{f}_k^i$ in terms of semantic part patterns $\boldsymbol{\theta}_n$ with multi-attention, where we learn the patterns $\boldsymbol{\theta}_n$ across all training data in the global feature learning benchmark $\boldsymbol{\Psi}$. The motivation for learning part patterns to aggregate regions is that the appearance of GSPs

is so various that it would limit the discriminability of global features $\boldsymbol{f}$. Our multi-attention mechanism includes attention weights for view aggregation on the part-level and the part-pattern-level, denoted by $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, respectively. Here, $\boldsymbol{\alpha}$ models how each of the N patterns $\boldsymbol{\theta}_n$ weights each of the $V \times K$ regions r_k^i, while $\boldsymbol{\beta}$ measures how the final, global feature $\boldsymbol{f}$ weights each of the N patterns $\boldsymbol{\theta}_n$.

Specifically, we employ a single-layer perceptron to learn $\boldsymbol{\theta}_n$, where $\boldsymbol{\theta}_n$ has the same dimension as $\boldsymbol{f}_k^i$. $\boldsymbol{\alpha}$ is a $(V \times K) \times N$ matrix, where each entry $\boldsymbol{\alpha}((i,k),n)$ is the attention paid to each of the $(V \times K)$ regions r_k^i by the n-th pattern $\boldsymbol{\theta}_n$. $\boldsymbol{\alpha}((i,k),n)$ is measured by a softmax function as $exp(\boldsymbol{w}_n^T \boldsymbol{f}_k^i + b_n)/\sum_{n' \in [1,N]} exp(\boldsymbol{w}_{n'}^T \boldsymbol{f}_k^i + b_{n'})$. With $\boldsymbol{\alpha}$, we first aggregate all $(V \times K)$ region features $\boldsymbol{f}_k^i$ into a pattern specific aggregation $\boldsymbol{\varphi}_n$ in terms of each pattern $\boldsymbol{\theta}_n$ by computing $\sum_{i \in [1,V], k \in [1,K]} \boldsymbol{\alpha}((i,k),n)(\boldsymbol{\theta}_n - \boldsymbol{f}_k^i)$. Then, we further aggregate all N pattern specific aggregations $\boldsymbol{\varphi}_n$ into the final, global feature $\boldsymbol{f}$ of 3D shape m. This is performed by linear weighting with the N dimensional vector $\boldsymbol{\beta}$, such that $\boldsymbol{f} = \sum_{n \in [1,N]} \boldsymbol{\beta}(n)\boldsymbol{\varphi}_n$. For clarity of exposition, we explain the details of how we obtain $\boldsymbol{\beta}$ further below.

Finally, we use $\boldsymbol{f}$ to classify m into one of C shape classes by a softmax function after a fully connected layer, where the softmax function outputs the classification probabilities $\boldsymbol{p}$, such that each probability $\boldsymbol{p}(c)$ is defined as $exp(\boldsymbol{u}_c^T \boldsymbol{f} + a_c)/\sum_{c' \in [1,C]} exp(\boldsymbol{u}_{c'}^T \boldsymbol{f} + a_{c'})$. The objective function of G is the cross entropy between $\boldsymbol{p}$ and the ground truth probability $\boldsymbol{p}'$,

$$O_{\text{G}}(\boldsymbol{p}, \boldsymbol{p}') = -\sum_{c \in [1,C]} \boldsymbol{p}'(c) log \boldsymbol{p}(c). \tag{2}$$

The intuition behind modelling part-pattern-level attention is to enable Parts4Feature to weight the pattern specific aggregations $\boldsymbol{\varphi}_n$ according to the 3D shape characteristics that it has learned. This leads Parts4Feature to differentiate shapes in detail. To implement this, $\boldsymbol{\beta}$ is designed to capture the similarities between each of the N pattern specific aggregations $\boldsymbol{\varphi}_n$ and the C shape classes. To represent the characteristics of C shape classes, we propose to employ the weights $\boldsymbol{u}_c$ in the fully connected layer before the last softmax function, as illustrated in Fig. 1. We first project $\boldsymbol{\varphi}_n$ and $\boldsymbol{u}_c$ into a common space using matrices $\boldsymbol{W}_1$ and $\boldsymbol{W}_2$. Then we compute normalized similarities using a linear mapping with $\boldsymbol{w}$ and $\boldsymbol{g}$ as follows, $\boldsymbol{\beta} = softmax((\boldsymbol{W}_1[\boldsymbol{\varphi}_n^T]_N + \boldsymbol{W}_2[\boldsymbol{u}_c^T]_C)\boldsymbol{w} + \boldsymbol{g})$, where learnable parameters $\boldsymbol{W}_1$ and $\boldsymbol{W}_2$ are $N \times N$ and $N \times C$ dimensional matrices, $\boldsymbol{w}$ and $\boldsymbol{g}$ are d and N dimensional vectors, $[\bullet]_\circ$ means stacking all $\circ$ vectors $\bullet$ into a matrix row by row.

Training. We train R and L together under a local part detection benchmark $\boldsymbol{\Phi}$, and G under a global feature learning benchmark $\boldsymbol{\Psi}$. The Parts4Feature objective is to simultaneously minimize Eq. 1 and Eq. 2, which leads to the loss

$$O = \frac{1}{\|\boldsymbol{\Phi}\|} \sum_{(p,t) \in \boldsymbol{\Phi}} O_{\text{R+L}}(p_\text{R}, p_\text{L}, p', p'', t_\text{R}, t_\text{L}, t', t'')$$
$$+ \frac{\eta}{\|\boldsymbol{\Psi}\|} \sum_{\boldsymbol{p} \in \boldsymbol{\Psi}} O_{\text{G}}(\boldsymbol{p}, \boldsymbol{p}'), \tag{3}$$
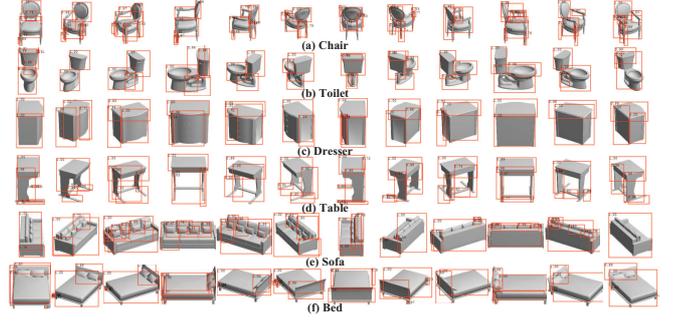

Figure 3: The detected GSPs with $p_\text{L} > 0.6$ in red boxes.

where the number of samples $\|\cdot\|$ is a normalization factor and η is a balance parameter. Since R and L are based on the object detection architecture of Fast R-CNN [Girshick, 2015], we adopt the approximate approach in [Ren and others, 2015] to jointly train R and L fast. In addition, we simultaneously update $\boldsymbol{u}_c$ in the softmax classifier in G by $\partial O_{\text{G}}/\partial \boldsymbol{u}_c$ and $\partial \boldsymbol{\beta}/\partial \boldsymbol{u}_c$. This enables $\boldsymbol{u}_c$ to be learned more flexibly for optimization convergence, which is a connection across G. For the $\boldsymbol{\Phi} = \boldsymbol{\Psi}$ case, parameters in R, L and G can be simultaneously updated, otherwise, they are updated alternatively. For example, parameters in R and L are first updated under $\boldsymbol{\Phi}$, then, parameters in R (except RPN) and G are updated under $\boldsymbol{\Psi}$, and this process is iterated until convergence. In our following experiments we use $\eta = 1$.

4 Experiments and analysis

Parameters. We investigate how some important parameters affect Parts4Feature in shape classification under ModelNet [Wu and others, 2015].

We first explore the IoU thresholds S_R in R and S_L in L that are used to establish positive GSP samples using ModelNet40 [Wu and others, 2015] as $\boldsymbol{\Psi}$, as shown in Table 1, where we initially use $V = 12$ views, $K = 20$ regions, and $N = 256$ patterns. With $S_\text{R} = 0.7$ and increasing S_L from 0.5 to 0.8, the mean Average Presicion (mAP) under the test set of $\boldsymbol{\Phi}$ decreases, and accordingly, the average instance accuracy under the test set of $\boldsymbol{\Psi}$ decreases, compared to the highest classification accuracy 93.40%. With $S_\text{L} = 0.5$, we also decrease S_R to 0.5 and increase it to 0.8 respectively. The mAP only slightly drops from 77.28 to 75.39 and 72.32, although the corresponding accuracy decreases too. However, the mAP and the accuracy are not strictly positive correlated, as shown by "(0.6,0.6)", which has lower mAP but higher accuracy than "(0.8,0.5)" and "(0.5,0.5)". This comparison also implies that S_L affects part detection more than S_R.

Next, we apply the parameters setting "(0.7,0.5)" under ModelNet10 [Wu and others, 2015], as shown by the first accuracy of 95.26% in Table 2. Increasing S_L to 0.7 leads to an even better result of 96.15%. We also find the slight effect of K, N, and V on the performance.

We visualize part detection and multi-attention involved in our best result under ModelNet10 in Fig. 3 and Fig. 4, respectively. Although there are no ground truth GSPs under ModelNet10, Parts4Feature still successfully transfers the part de-

Table 1: The effects of S_R and S_L on the performance of Parts4Feature under ModelNet40.

Metrics	$(0.7, 0.5)$	$(0.7, 0.6)$	$(0.7, 0.7)$	$(0.7, 0.8)$	$(0.5, 0.5)$	$(0.8, 0.5)$	$(0.6, 0.6)$
mAP	**77.28**	69.35	66.12	56.97	75.39	72.32	69.51
Acc	**93.40**	93.15	92.38	92.50	92.67	92.71	92.95

Table 2: The effects of K, N and d on the performance of Parts4Feature under ModelNet10.

Metric	$S_\text{L} = 0.5$	0.7	0.8	$K = 10$	30	$N = 128$	512	$V = 3$	6
Acc	95.26	**96.15**	94.38	94.38	94.93	94.49	95.04	94.27	94.93

Table 3: The view aggregation and attention comparison.

Pooling		Attention	
Methods	Acc	Methods	Acc
MaxPool	90.97	NoAtt	92.84
MeanPool	92.18	PtAtt	93.17
NetVLAD	93.50	PnAtt	93.72
No L	93.61	MultiAtt	**96.15**
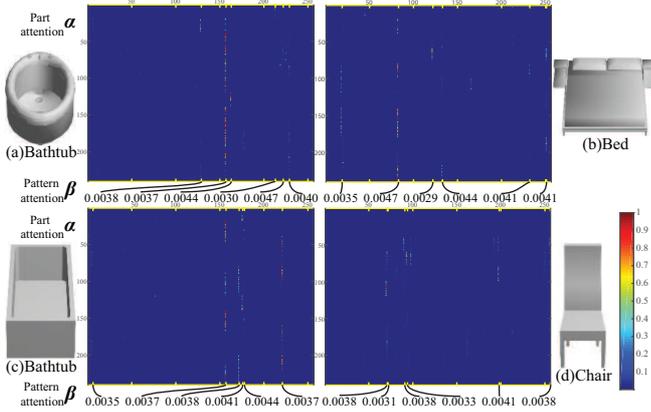
Figure 4: Multi-attention is visualized under the test set of Model-Net10. α and β are shown as matrices and numbers.

Figure 5: The classification confusion matrix under ModelNet10.

tection knowledge learned from Φ to detect GSPs in multiple views. Moreover, β is learned to focus on the patterns with high part attentions in α, where the top-6 patterns with high part attentions in α are shown below for clarity.

Ablation study. Finally, in Table 3 we highlight our semantic part based view aggregation and multi-attention method in branch G under ModelNet10. We replace our view aggregation with max pooling, mean pooling, and NetVLAD, where we aggregate $V \times K$ region features $\boldsymbol{f}_k^i$ for classification. Although these results are good, our novel aggregation with multi-attention can further improve the results. For evaluating multi-attention, we keep G unchanged and set all entries in α and β to 1 ("NoAtt"). This leads to significantly worse performance compared to our "MultiAtt". Next, we employ α and β separately. We find that both of part attention and part pattern attention improve "NoAtt", but α ("PtAtt") contributes less than β ("PnAtt"). Moreover, we highlight the effect of branch L as an enhancer of module R by removing L ("No L") from Parts4Feature, which is also justified by the degenerated results.

Classification. Table 4 compares Parts4Feature with the state-of-the-art in 3D shape classification under ModelNet. The comparison are conducted under the same condition[1].

Under both benchmarks, Parts4Feature outperforms all its competitors at the same condition, where "Our" are obtained with the parameters of our best accuracy under ModelNet40 in Table 1 and the ones under ModelNet10 in Table 2. This comparison shows that Parts4Feature effectively employs part-level information to significantly improve the discriminability of learned features. Parts4Feature is also outperforming under ShapeNet55 with the same parameters of our best results under ModelNet10, as shown by the comparison in the last three rows in Table 7.

To better demonstrate our classification results, we visualize the confusion matrix of our classification result under ModelNet10 and ShapeNet55 in Fig. 5 and Fig. 6, respectively. In each confusion matrix, an element in the diagonal line means the classification accuracy in a class, while other elements in the same row means the misclassification accuracy. The large diagonal elements shows that Parts4Feature is good at classifying large-scale 3D shapes.

We also conduct experiments with reduced number of segmented shapes for training under ModelNet10. As shown in Table 5, trained by randomly sampled $\{0\%, 1\%, 5\%, 10\%, 25\%, 50\%\}$ of 6,386 shapes, our results

---

[1] We use the same modality of views from the same camera system for the comparison, where the results of RotationNet are from Fig.4 (a) and (b) in https://arxiv.org/pdf/1603.06208.pdf. Moreover, the benchmarks are with the standard training and test split.

Table 4: The classification comparison ModelNet.

| Methods	Raw	MN40	MN10
MVCN[Su and others, 2015]	View	90.10	-
MVVC[Qi et al., 2016]	Voxel	91.40	-
3DDt[Xie and others, 2018]	Voxel	-	92.40
PaiV[Johns et al., 2016]	View	90.70	92.80
Sphe[Cao et al., 2017]	View	93.31	-
GIFT[Bai and others, 2017]	View	89.50	91.50
RAMA[Sfikas and others, 2017]	View	90.70	91.12
VRN[Brock et al., 2016]	Voxel	91.33	93.80
RNet[Kanezaki et al., 2018]	View	90.65	93.84
PNetP[Qi and others, 2017]	Point	91.90	-
DSet[Wang et al., 2017]	View	92.20	-
VGAN[Wu and others, 2016]	Voxel	83.30	91.00
LAN[Achlioptas and others, 2018]	Point	85.70	95.30
FNet[Yang et al., 2018]	Point	88.40	94.40
SVSL[Han and others, 2019a]	View	93.31	94.82
VIPG[Han and others, 2019b]	View	91.98	94.05
Our	View	**93.40**	**96.15**
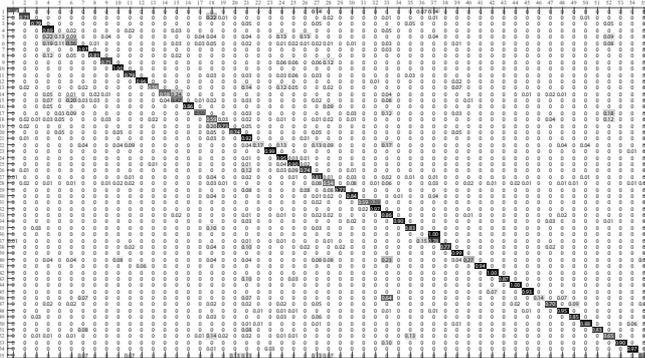
Figure 6: The classification confusion matrix under ShapeNet55.

increase accordingly. The good results with 0% segmented shapes show that we not only learn from pixel-level information in 3D classification benchmarks, similar to existing methods, but also improve performance further by absorbing part-level information from 3D segmentation benchmark.

Retrieval. We further evaluate Parts4Feature in shape retrieval under ModelNet and ShapeNetCore55 by comparing with the state-of-the-art methods in Table 6 and Table 7. These experiments are conducted under the test set, where each 3D shape is used as a query to retrieve from the rest of the shapes, and the retrieval performance is evaluated by mAP. The compared results include LFD, SHD, Fisher vector, 3D ShapeNets [Wu and others, 2015], Pano [Shi and others, 2015], MVC-N [Su and others, 2015], GIFT [Bai and others, 2017], RA-

Table 5: The effect of less segmented shapes for training.

Acc	0%	1%	5%	10%	25%	50%	100%
Instance	93.0	93.5	93.8	93.8	94.1	94.3	**96.15**
Class	92.7	93.1	93.4	93.6	94.0	94.1	**96.14**

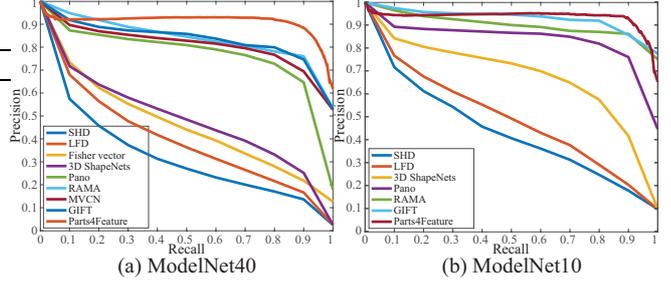(a) ModelNet40   (b) ModelNet10

Figure 7: The PR curve comparison under ModelNet.

Table 6: The retrieval (mAP) comparison under ModelNet.

Data	Pano	MVCN	GIFT	RAMA	Trip	Ours
MN40	76.8	79.5	81.9	83.5	88.0	**91.5**
MN10	84.2	-	91.1	87.4	-	**93.8**

MA [Sfikas and others, 2017] and Trip [He et al., 2018] under ModelNet.

As shown in Table 6, Table 7, our results outperform all the compared results in each benchmark. Besides Taco [Cohen et al., 2018] in Table 7, the compared micro-averaged results in Table 7 are from SHREC2017 shape retrieval contest [Savva and others, 2017] with the same names. In addition, the available PR curves under ModelNet40 and ModelNet10 are also compared in Fig. 7, which also demonstrates our outperforming results in shape retrieval.

Table 7: Retrieval and classification comparison in terms of Micro-averaged metrics under ShapeNetCore55.

Methods	Micro				
	P	R	F1	mAP	NDCG
Kanezaki	81.0	80.1	**79.8**	77.2	86.5
Zhou	78.6	77.3	76.7	72.2	82.7
Tatsuma	76.5	80.3	77.2	74.9	82.8
Furuya	**81.8**	68.9	71.2	66.3	76.2
Thermos	74.3	67.7	69.2	62.2	73.2
Deng	41.8	71.7	47.9	54.0	65.4
Li	53.5	25.6	28.2	19.9	33.0
Mk	79.3	21.1	25.3	19.2	27.7
Su	77.0	77.0	76.4	73.5	81.5
Bai	70.6	69.5	68.9	64.0	76.5
Taco	70.1	71.1	69.9	67.6	75.6
Our	62.0	**80.4**	62.2	**85.9**	**90.2**
SVSL[Han and others, 2019a]				85.5	
VIPG[Han and others, 2019b]				83.0	
Our classification				**86.9**	

5 Conclusions

Parts4Feature is proposed to learn 3D global features from part-level information in a semantic way. It successfully learns universal knowledge of generally semantic part detection from 3D segmentation benchmarks, and effectively transfers the knowledge to other shape analysis benchmarks

by learning 3D global features from detected parts in multiple views. Parts4Feature makes it feasible to improve 3D global feature learning by leveraging discriminative information from another source. Moreover, our novel view aggregation with multi-attention can also benefit Parts4Feature to learn more discriminative features than widely used aggregation procedures. Our outperforming results show that Parts4Feature is superior to other state-of-the-art methods.

6 Acknowledgments

References

[Achlioptas and others, 2018] Panos Achlioptas et al. Learning representations and generative models for 3D point clouds. In *The International Conference on Machine Learning*, pages 40–49, 2018.

[Bai and others, 2017] Song Bai et al. GIFT: Towards scalable 3D shape retrieval. *IEEE Transaction on Multimedia*, 19(6):1257–1271, 2017.

[Brock et al., 2016] Andrew Brock, Theodore Lim, J.M. Ritchie, and Nick Weston. Generative and discriminative voxel modeling with convolutional neural networks. In *3D deep learning workshop (NIPS)*, 2016.

[Cao et al., 2017] Zhangjie Cao, Qixing Huang, and Karthik Ramani. 3D object classification via spherical projections. In *International Conference on 3D Vision*. 2017.

[Chatfield et al., 2014] Ken Chatfield, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Return of the devil in the details: Delving deep into convolutional nets. In *British Machine Vision Conference*, 2014.

[Cohen et al., 2018] Taco S. Cohen, Mario Geiger, Jonas Köhler, and Max Welling. Spherical CNNs. In *International Conference on Learning Representations*, 2018.

[Girshick, 2015] Ross Girshick. Fast R-CNN. In *IEEE International Conference on Computer Vision*, pages 1440–1448, 2015.

[Han and others, 2016] Zhizhong Han et al. Unsupervised 3D local feature learning by circle convolutional restricted boltzmann machine. *IEEE Transactions on Image Processing*, 25(11):5331–5344, 2016.

[Han and others, 2017] Zhizhong Han et al. Mesh convolutional restricted boltzmann machines for unsupervised learning of features with structure preservation on 3D meshes. *IEEE Transactions on Neural Network and Learning Systems*, 28(10):2268 – 2281, 2017.

[Han and others, 2019a] Zhizhong Han et al. Seqviews 2seqlabels: Learning 3D global features via aggregating sequential views by rnn with attention. *IEEE Transactions on Image Processing*, 28(2):1941–0042, 2019.

[Han and others, 2019b] Zhizhong Han et al. View interprediction gan: Unsupervised representation learning for 3D shapes by learning global shape memories to support local view predictions. In *AAAI*, 2019.

[Han et al., 2019] Zhizhong Han, Mingyang Shang, Xiyang Wang, Yu-Shen Liu, and Matthias Zwicker. Y2seq2seq: Cross-modal representation learning for 3D shape and text by joint reconstruction and prediction of view and word sequences. In *AAAI*, 2019.

[He et al., 2018] Xinwei He, Yang Zhou, Zhichao Zhou, Song Bai, and Xiang Bai. Triplet-center loss for multiview 3D object retrieval. In *The IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[Johns et al., 2016] Edward Johns, Stefan Leutenegger, and Andrew J. Davison. Pairwise decomposition of image sequences for active multi-view recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3813–3822, 2016.

[Kalogerakis and others, 2017] Evangelos Kalogerakis et al. 3D shape segmentation with projective convolutional networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 6630–6639, 2017.

[Kanezaki et al., 2018] Asako Kanezaki, Yasuyuki Matsushita, and Yoshifumi Nishida. Rotationnet: Joint object categorization and pose estimation using multiviews from unsupervised viewpoints. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[Muralikrishnan et al., 2018] Sanjeev Muralikrishnan, Vladimir G. Kim, and Siddhartha Chaudhuri. Tags2parts: Discovering semantic regions from shape tags. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2926–2935, 2018.

[Qi and others, 2017] Charles Qi et al. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems*, pages 5105–5114, 2017.

[Qi et al., 2016] C R Qi, H Su, and M Niebner. Volumetric and multi-view cnns for object classification on 3D data. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5648–5656, 2016.

[Ren and others, 2015] Shaoqing Ren et al. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, pages 91–99, 2015.

[Savva and others, 2017] Manolis Savva et al. SHREC'17 Large-Scale 3D Shape Retrieval from ShapeNet Core55. In *Eurographics Workshop on 3D Object Retrieval*, 2017.

[Sfikas and others, 2017] Konstantinos Sfikas et al. Exploiting the PANORAMA Representation for Convolutional Neural Network Classification and Retrieval. In *EG Workshop on 3D Object Retrieval*, pages 1–7, 2017.

[Sharma et al., 2016] Abhishek Sharma, Oliver Grau, and Mario Fritz. VConv-DAE: Deep volumetric shape learning without object labels. In *Proceedings of European Conference on Computer Vision*, pages 236–250, 2016.

[Shi and others, 2015] B. Shi et al. Deeppano: Deep panoramic representation for 3D shape recognition. *IEEE Signal Processing Letters*, 22(12):2339–2343, 2015.

[Su and others, 2015] Hang Su et al. Multi-view convolutional neural networks for 3D shape recognition. In *International Conference on Computer Vision*, pages 945–953, 2015.

[Wang *et al.*, 2017] 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*, 2017.

[Wu and others, 2015] Zhirong Wu et al. 3D ShapeNets: A deep representation for volumetric shapes. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 1912–1920, 2015.

[Wu and others, 2016] Jiajun Wu et al. Learning a probabilistic latent space of object shapes via 3D generative-adversarial modeling. In *Advances in Neural Information Processing Systems*, pages 82–90, 2016.

[Xie and others, 2018] Jianwen Xie et al. Learning descriptor networks for 3D shape synthesis and analysis. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[Yang *et al.*, 2018] Yaoqing Yang, Chen Feng, Yiru Shen, and Dong Tian. Foldingnet: Point cloud auto-encoder via deep grid deformation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.