# Enhancing 2D Representation via Adjacent Views for 3D Shape Retrieval

Cheng Xu[1], Zhaoqun Li[1], Qiang Qiu[2], Biao Leng[1,3,4]*, Jingfei Jiang[5]
[1]School of Computer Science & Engineering, Beihang University, [2]Duke University
[3]Beijing Advanced Innovation Center for Big Data and Brain Computing, Beihang University
[4]State Key Laboratory of Virtual Reality Technology and Systems, Beihang University
[5]National Laboratory for Parallel and Distributed Processing, National University of Defense Technology
{cxu, lizhaoqun, lengbiao}@buaa.edu.cn, qiang.qiu@duke.edu, jingfeijiang@nudt.edu.cn

## Abstract

*Multi-view shape descriptors obtained from various 2D images are commonly adopted in 3D shape retrieval. One major challenge is that significant shape information is discarded during 2D view rendering through projection. In this paper, we propose a convolutional neural network based method, Neighbor-Center Enhanced Network, to enhance each 2D view using its neighboring ones. By exploiting cross-view correlations, Neighbor-Center Enhanced Network learns how adjacent views can be maximally incorporated for an enhanced 2D representation to effectively describe shapes. We observe that a very small amount of, e.g., six, enhanced 2D views, are already sufficient for panoramic shape description. Thus, by simply aggregating features from six enhanced 2D views, we arrive at a highly compact yet discriminative shape descriptor. The proposed shape descriptor significantly outperforms state-of-the-art 3D shape retrieval methods on the ModelNet and ShapeNetCore55 benchmarks, and also exhibits robustness against object occlusion.*

## 1. Introduction

3D shape retrieval is widely applied in fields such as biological analysis, virtual reality, and medical imaging. Recent years have witnessed significant progress in 3D shape retrieval [25, 29, 6, 14]. By mimicking the human visual perception of 3D shapes with 2D observations, 2D view-based 3D shape retrieval methods have shown impressive performance.

The state-of-the-art view-based methods adopt deep learning. These methods first apply deep convolutional neural networks (CNNs) [12, 18] over rendered views from 3D shapes to obtain a set of discriminative features, and then assemble multiple visual features across the spatial dimension using methods such as the max-pooling operation [23] or an active concatenation technique [10]. We note that the discriminative visual representation of each projected view is vital in existing methods. However, due to the information deficiency in projection, certain views contain insufficient shape information, as shown in Figure 1 (a), to effectively identify a shape, restricting the discrimination of the features. Moreover, this deleterious effect is even more significant under object occlusion in real applications of 3D shape recognition.

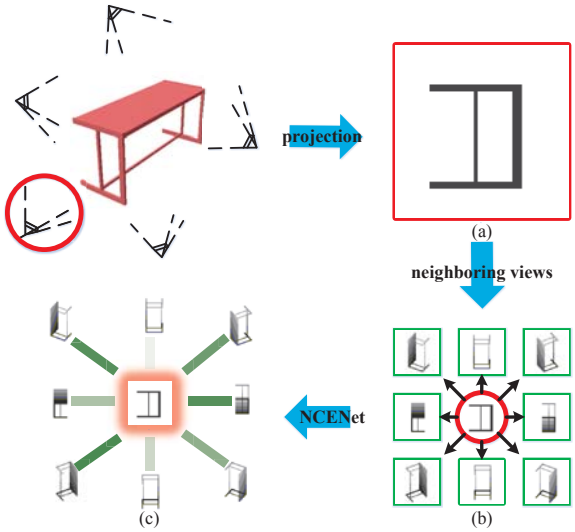



Figure 1. (a) The original 2D view shows ineffective in identifying a shape. (b) This 2D view can be enhanced from a set of neighboring views by exploring cross-view correlations. (c) Our method learns how adjacent views can be maximally incorporated for an enhanced shape representation, and the intensity of the green line indicates the value of the learned correlation attention.

*Corresponding author.

To address the above challenges, we propose to describe a 3D shape by incorporating in each single 2D view a set of neighboring views, as shown in Figure 1 (a) and Figure 1 (b). It is observed that neighboring views can supply necessary complementary geometry information to their center view. Therefore, we seek for an efficient view-based method that uses adjacent views to enhance the discrimination of shape representations at the feature level. We note that different neighboring views have different effects on the enhancement of the center view. On one hand, some views are similar to the center view and provide limited additional shape information to the center view. On the other hand, certain views may be occluded, producing a negative effect on the shape features in the presence of object occlusion. Therefore, it is necessary to explore the content relationship among neighboring views and center views in the process of enhancement.

In this work, we propose a 3D shape retrieval framework, named the Neighbor-Center Enhanced Network (NCENet), to enhance the discrimination of each 2D view feature by exploiting correlations across its adjacent views. In our network, the visual feature of each 2D view is first extracted from a CNN and then grouped with its neighboring view features to conduct center feature enhancement. Note that we refer to the 2D view to be enhanced as the center view due to its central position relative to the adjacent views to be exploited.

In our method, a correlation unit is designed to guide the feature enhancement of the center view, and such module takes into account both the discriminability and complementarity of neighboring view features. Moreover, two types of enhanced structures are presented to utilize the inherent topological order between the center view and its adjacent views: The parallel structure considers all of the adjacent features independently and simultaneously, and the serial structure incorporates the adjacent features in sequential order.

In summary, our main contributions are as follows:

- We introduce a compact, discriminative and robust 2D view-based 3D shape descriptor, by enhancing each 2D view with its neighboring ones through CNNs.
- We design network modules and structures to exploit the discriminability and complementarity of neighboring views for optimized 2D view enhancement.
- Our enhanced 2D views show both discriminative for shape description and robust to object occlusion, significantly outperforming the state-of-the-art methods on both ModelNet and ShapeNetCore55 benchmarks.

## 2. Related Work

Our method is related to prior work on 3D shape representation learning for 3D shape retrieval. There has been much insightful research on developing 3D shape representation learning methods [8, 24]. These approaches can be classified into two categories: model-based methods that directly extract the features from the raw 3D representations of 3D objects, such as polygon meshes, point clouds, geometric-based shape distributions, or graph-based topological structures, and view-based methods that leverage the shape information in a collection of rendered views.

Many model-based shape descriptors have been proposed for 3D shapes. Previously, model-based shape descriptors were largely constructed by hand, which was time-consuming. For instance, 3D shapes can be represented with a heat kernel signature with heat diffusion on polygon meshes [1] or a sequence of radii of the maximal balls at the skeleton points [13]. The recently developed deep learning techniques fall into this category, and shape descriptors can be learned from a volumetric occupancy grid representation with a convolutional neural network [16], a "geometry image" transformed from a 3D shape using CNNs [22], point sets [17], and the probability distribution of HKS through a deep auto-encoder [28]. Although model-based shape descriptors can effectively capture the discriminative geometric characteristics, they face several serious challenges. First, model-based shape descriptors tend to be very high-dimensional, imposing a high computational burden of the distance measurement computation between different 3D shapes. Second, naive 3D representations of recently organized 3D object datasets face many obstacles, such as noise, incompleteness or occlusions, severely hindering the development of model-based algorithms.

Regarding view-based shape representation learning, a 3D model is represented by a set of rendered views. These have many desirable properties: they are convenient for applying deep learning models to regular structures and vertex topologies and are efficient in computing and robust in handling naive 3D shape representation, such as holes, ambiguous orientation on surfaces and numerical noise. The visual similarity between the views of two models is regarded as the model difference. A typical example of a view-based technique is the LightField descriptor [3] that extracts the representations of 3D shapes by utilizing geometric and Fourier descriptors from rendered images. [2] extracted visual features using GPU acceleration and adopted an efficient context-based re-ranking technique. Compared to a single image, a multi-view image sequence provides a much richer capacity for 3D shape retrieval. In recent years, CNN architectures have been extended to play an important role in retrieval and recognition from image sequences by transforming a 3D shape into a panoramic view and max pooling across each row [20], max pooling across

all viewpoints [23], or utilizing the spatial correlation information among multiple views with RNN [4]. However, these methods ignore the intrinsic content relationship between the views and fail to make full use of the information of all of the views. In our method, the content correlation among views is leveraged to enhance all 2D view features. Recently, Feng *et al.* [6] propose a similar framework (GVCNN) to learn the relationship among views using a group strategy. However, our method comes from a very different motivation, that [6] is motivated by exploiting the group-level relationship, while our inspiration comes from each 2D view and its neighboring views.

## 3. Method

In this section, we introduce the proposed neighbor-center enhanced network (NCENet) in detail. To produce discriminative 3D shape representation, NCENet enhances the discrimination of each 2D view feature via exploring the intrinsic correlation across its adjacent views. Specifically, multiple 2D view features are firstly extracted from a CNN. Then, the feature of each 2D view image is combined with the features of its neighboring views, and inputted into the center feature enhanced module. In this step, the present view to be enhanced is referred to as the ***center view*** because of its central position relative to the adjacent views.

We take the following two issues into consideration while designing the enhanced module. First, even though adjacent views are complementary to the center view, they can be redundant due to cross-view similarity. We present a correlation unit, modeling the intrinsic correlation between views, to help the center view effectively capture adjacent discriminative information. Second, the adjacent views present a specific and inherent order, according to their positions relative to the center view. We adopt two strategies to address the order of the neighboring views in our network, *i.e.*, the parallel and serial structures, and then we compare their retrieval performances.

Figure 2 illustrates the detailed flowchart of our proposed method. NCENet employs GoogLeNet with batch normalization [9] as the base architecture. The "CNN" denotes the whole GoogLeNet architecture. Softmax loss and center loss [26] are adopted to jointly supervise the NCENet. Center loss can give effective supervision of NCENet, and the analysis of the impact of the center loss on NCENet is given in the experiment.

### 3.1. Neighboring Views and Center View

We use the Phong reflection model [15] to render the views of 3D shapes in a unit spherical coordinate system, as shown in the left of Figure 3. The images are projected in the depth buffer at each combination of latitude $\theta_{la}$ and longitude $\theta_{lo}$. Consider a shape $X$, where $I_i$ denotes the projected image assigned to the

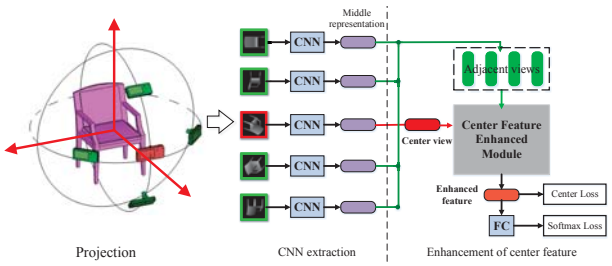



Figure 2. Neighbor-center enhanced network (NCENet). NCENet extracts the feature of each 2D view using a CNN architecture and enhances each view representation with neighboring ones. Red: Center view. Green: Neighboring views. For better visualization, only four neighboring views (up, down, left and right) are displayed (Best viewed in color).

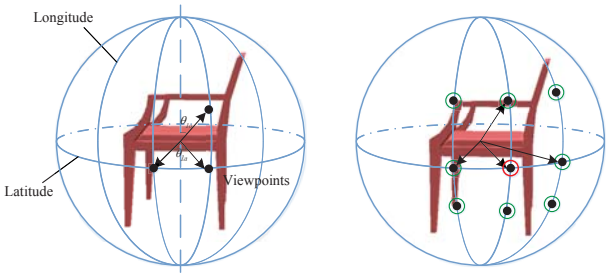



Figure 3. Left: views are rendered in a unit spherical coordinate system. Right: a present view to be enhanced (red circle) and its neighboring views (green circle). Since this view is at the center of the adjacent views, we refer to it as the center view in our network.

$i$-th view of shape $X$, and we can find its neighboring view set $Neighbor(I_i) = \{I_{i\to j}\}_{j\in Pos}$, where $Pos = \{up, down, left, right, upper\_left, upper\_right, bottom\_left, bottom\_right\}$, denoting the eight adjacent directions relative to $I_i$. The right side of Figure 3 shows $I_i$ and its neighboring set $Neighbor(I_i)$. It is observed that $I_i$ is in the center of the adjacent views. Again, we denote $I_i$ as the ***center view*** in the enhancement process. Note that each view is enhanced as the center view with its neighboring views in the training stage. Since some viewpoints are on the boundary of the projection system, these views can have missing adjacent views in certain directions. And we use the feature of the center view to replace missing adjacent features.

### 3.2. Correlation Unit

The correlation unit aims to describe the relationship between neighboring views and the center view to assist in maximizing the enhancement of the center feature. Specifically, the correlation unit learns to generate a correlation weight for each neighboring feature, and then, the weighted neighboring feature is used as an auxiliary feature to en-

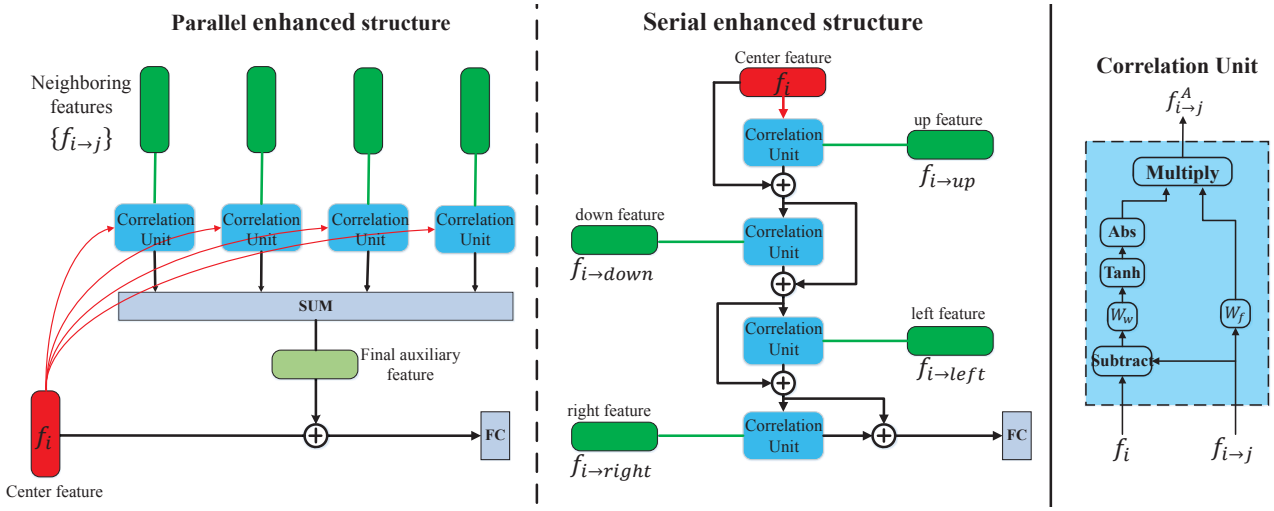



Figure 4. Two enhanced feature structures in NCENet. Left: parallel enhanced structure. Right: serial enhanced structure. They contain correlation units for modeling the correlation among the views, and the computation of the correlation unit is given in the blue box. Not all of the neighboring features are shown between the input and parallel (serial) structure for clarity. The figure is best viewed in color.

hance the center feature. The correlation weight of each neighboring feature is determined based on two criteria: (1) the discriminability of the neighboring view and (2) the information redundancy between the center view and its neighboring view, *i.e.*, certain neighboring views similar to the center can contain limited auxiliary information.

We now describe the correlation unit computation. Given a set of views from a 3D shape $S = \{I_1, I_2, ..., I_N\}$, the extracted middle features of the views are denoted as $F = \{f_1, f_2, ..., f_N\}$, and the visual feature dimension is $K$. In our experiment, $f_i$ is a 1024-dimensional feature vector. For each view feature $f_i$, its neighboring view feature set is denoted as $Neighbor(f_i) = \{f_{i\to j}\}_{j\in Pos}$. The output auxiliary feature of the correlation unit $f^A_{i\to j}$ with respect to the neighboring feature $f_{i\to j}$ is computed as

$$f^A_{i\to j} = w^{i\to j} \cdot (W_f \cdot f_{i\to j}). \tag{1}$$

The output is a weighted neighboring feature linearly transformed by $W_f \in \mathbb{R}^{K\times K}$ which is implemented as a fully connected layer. The correlation weight $w^{i\to j}$ indicates the impact of a neighboring feature on the center feature, which can be computed as

$$w^{i\to j} = abs(tanh(W_w \cdot (f_i - f_{i\to j}))), \tag{2}$$

where $W_w \in \mathbb{R}^{K\times 1}$ is the parameter of the fully connected layer. The function $abs(tanh(\cdot))$ constrains the correlation weight to be in the range between 0 and 1. Eq. (2) describes the information difference between the neighboring feature and the center view. For example, the generated correlation weight is 0 when a neighboring view is identical to the center view, as this neighboring view provides no additional shape information to the center feature. Note that some common weight normalization ways, such as sigmoid and softmax operations, are inappropriate to construct correlation weight. First, if the neighbor feature is identical to center feature, $sig(f_i - f_{i\to j}) = 0.5$. Second, softmax operation cannot be used in the following serial structure because weights are generated sequentially.

The correlation unit has the same input and output dimensions, and we use the correlation unit as a basic block in our NCENet architecture.

### 3.3. Parallel & Serial Structure

To address the order of adjacent views according to their locations relative to the center view, we propose to access two different structures in our network: the parallel enhanced structure (Figure 4 left) and the serial enhanced structure (Figure 4 right). In this subsection, we will introduce these two enhanced structures.

**Parallel structure.** As shown in the left of Figure 4, we adopt a parallel order to conduct the enhancement, where all adjacent features are inputted into the network independently. The auxiliary features of the neighboring features $\{f^A_{i\to j}\}$ are generated from different correlation units. To make different weighted neighboring features comparable, the different correlation units are forced to share the same weight. Then, different auxiliary features of the neighboring views and the center feature are aggregated via addition and are then learned by a fully connected layer to produce the K-dimensional final enhanced feature vector $f^e_i$, which

can be computed as

$$f_i^e = W_e \cdot (f_i + \sum_j f_{i\to j}^A). \quad (3)$$

**Serial structure.** In this structure, neighboring features are arranged to enhance the center feature sequentially, and we use a fixed order of the neighboring views: up, down, left, right, upper left, upper right, bottom left and bottom right. The entire enhancement procedure can be divided into several blocks corresponding to different adjacent features. Considering the $k$-th block, the center input $o^{k-1}$ and neighboring feature $f_{i\to k}$ are inputted into the correlation unit to produce the auxiliary feature $f_{i\to k}^A$. The output of the $k$-th enhanced block is computed as

$$o^k = W_o \cdot (o^{k-1} + f_{i\to k}^A), \quad (4)$$

where $W_o \in \mathbb{R}^{K\times K}$ and $o^0$ is initialized as the feature of the center view. The produced $K$- dimensional feature vector $o^k$ is used as the center input for the subsequent enhanced block. We can express the serial enhanced structure as a sequence of enhanced blocks, and the output of the sequence varies with the order of the enhanced blocks.

### 3.4. Objective Function

NCENet adopts the softmax loss regularized with the center loss [26], which can be represented as

$$\mathcal{L} = \mathcal{L}_S + \lambda \mathcal{L}_C, \quad (5)$$

where $\lambda$ is a hyper-parameter used to balance the softmax loss and center loss. In our case, $\lambda$ is set to 0.01. Softmax loss $\mathcal{L}_S$ is used for the classification task for the discrimination of visual features. The center loss $\mathcal{L}_C$ is adopted to guide the feature enhancement by optimizing the category center, which is expressed as

$$\mathcal{L}_C = \frac{1}{2}\sum_{i=1}^{s} ||f_i^e - c_{y_i}||_2^2, \quad (6)$$

where $f_i^e$ denotes the enhanced center representation. $c_{y_i}$ denotes the center representation of the $y_i$-th class, which is computed as the average of all view features from the $y_i$-th class, and $c_{y_i}$ is updated independently. The mini-batch size is $s$.

## 4. Experiments

In this section, we first evaluate the NCENet on different kinds of 3D shape retrieval tasks. Then, we evaluate the discriminative capacity of our method for poor quality views, and the robustness of NCENet in the case of object occlusion. In the last part, we further discuss the effect of the number of center views, the number of neighbor views, the learned correlation weights among the views and center views, and the center loss on NCENet.

### 4.1. Implementation Details

Each 3D shape is rendered to yield 224 $\times$ 224 depth images from 36 virtual cameras.

**Training.** Center views are randomly selected from 36 views of each shape. Then, the center views with their adjacent views in 8 directions are inputted into the network. For the multi-view feature extraction procedure, the CNN is fine-tuned on a specific 3D dataset, with pre-training conducted on the ImageNet $1k$ dataset [5]. For the feature enhancement, we use features from the pooling layer $Pool5/7\times7\_s1$ as the visual representation for the enhancement, for which the feature dimension is 1024.

**Testing.** Each shape selects 6 views from fixed viewpoints as the center views to conduct the enhancement, and we find that the retrieval results no longer improve when the number of the center views exceeds 6. The 6 enhanced features are average pooled into a compact feature as the final 3D shape representation. We adopt the cosine distance to measure the similarity between the shapes.

### 4.2. Retrieval on ModelNet

We evaluate the performance of the NCENet on the Princeton ModelNet dataset [21]. ModelNet is composed of 127,915 3D CAD models from 662 categories. We use two subsets, ModelNet40 and ModelNet10, for evaluation. The first subset contains 12,311 models, and the second contains 4,899 models. We follow [27, 2] to conduct the training/testing split, where 100 unique shapes are randomly selected per category from the subset and the first 80 shapes are used for training and the rest for testing.

Our NCENet is compared against the DeepPano [20], MVCNN [23], GIFT [2], CNN-BiLSTM [4], GVCNN [6] and TCL [7] methods. We also set two baseline methods NCENet_Max and NCENet_Ave. NCENet_Ave denotes adjacent features are directly average pooled to enhance the center view. NCENet_Max denotes adjacent features are directly max pooled to enhance the center view.

The performances of different methods are shown in Table 1. Although TCL assisted by softmax loss achieves higher MAP, TCL is more sensitive to the margin design and it is difficult to search an ideal margin. Compared to TCL, our method utilizes natural adjacent information and is easy to implement. Moreover, our method outperforms by $0.35\%$ mAP over using only TCL loss, referring to $86.7\%$ mAP in [7]. Compared to GVCNN, Our-Serial gains $1.35\%$ improvement of mAP on ModelNet40.

Note that NCENet employs GoogLeNet-bn as the base architecture, which differs from MVCNN. To maintain a fair comparison, we further implement MVCNN with 36

Table 1. Comparison of performance on ModelNet40 with state-of-the-art methods

| Method | ModelNet40 | |
|---|---|---|
| | AUC | mAP |
| DeepPano [20] | 77.63% | 76.81% |
| MVCNN [23] | - | 80.20% |
| GIFT [2] | 83.10% | 81.90% |
| CNN-BiLSTM [4] | - | 83.30% |
| GVCNN [6] | - | 85.70% |
| TCL+softmax [7] | **89.00%** | **88.00%** |
| MVCNN (GoogLeNet) | 86.58% | 85.49% |
| NCENet_Max | 85.23% | 84.12% |
| NCENet_Ave | 82.50% | 81.29% |
| Our-Parallel | 87.28% | 86.27% |
| Our-Serial | 88.04% | 87.05% |

views based on GoogLeNet-bn and the low-rank Mahalanobis metric learning is also adopted, corresponding to MVCNN (GoogLeNet) in Table 1. It is shown that the use of GoogLeNet can improve the retrieval performance. Using the same base network, NCENet-Serial gained of 1.56% mAP on ModelNet40 compared to MVCNN (GoogLeNet). Furthermore, since GoogLeNet has fewer parameters and higher discrimination, we use ***GoogLeNet*** as the basic network of our method in all of the subsequent experiments.

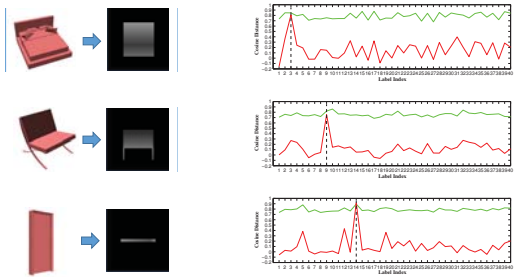
Compared with basic aggregation mechanisms, our enhanced structure shows better performance than NCENet_Max and NCENet_Ave. The superior performance of our method is attributed to the following. NCENet contains a correlation unit that identifies the relationship between the neighboring view and the center view. The correlation unit generates a weight for each neighboring view to identify whether it is significant for the center view. Thus, all of the features of the views are effectively improved by exploiting the intrinsic correlation between the views. The maximum aggregation operation in MVCNN is limited by treating all views equally, and the correlation between the views cannot be well-exploited. Compared to GIFT, our method can be regarded as a feature-level enhancement that is better than the re-ranking method in GIFT, which can be regarded as enhancement on the metric level.

### 4.3. Retrieval on ShapeNetCore55

To test the scalability of NCENet, we choose large-scale 3D Shape Retrieval from the ShapeNetCore55 track [19] for a comprehensive evaluation. This dataset contains 51,190 3D shapes over 55 common categories. In our experiment, we adopt the official training and testing split method [19] and split the database into three parts, with 70% of shapes used for training, 10% of shapes used as validation data and the remaining 20% used for testing. Additionally, to test the robustness of our method, we choose the more challenging perturbed dataset where all of the shapes are randomly rotated.

Table 2. Comparison of performance (%) on the ShapeNetCore55 perturbed dataset

| Method | Micro | | | Macro | | |
|---|---|---|---|---|---|---|
| | F1 | mAP | NCDG | F1 | mAP | NCDG |
| Wang [19] | 24.6 | 60.0 | 77.6 | 16.3 | 47.8 | 69.5 |
| Li [19] | 53.4 | 74.9 | 86.5 | 18.2 | 57.9 | 76.7 |
| Kd-network [11] | 45.1 | 61.7 | 81.4 | 24.1 | 48.4 | 72.6 |
| MVCNN [23] | 61.2 | 73.4 | 84.3 | 41.6 | 66.2 | 79.3 |
| GIFT [2] | 66.1 | 81.1 | 88.9 | 42.3 | 73.0 | 84.3 |
| TCL [7] | 67.9 | 84.0 | 89.5 | 43.9 | 79.3 | 86.9 |
| Our-Parallel | **73.3** | **89.6** | **92.1** | **51.3** | **85.6** | 90.5 |
| Our-Serial | 70.8 | 89.0 | 92.0 | 50.6 | 85.3 | **90.6** |



(a) (b)

Figure 5. The performance of NCENet on poor views of 3D shapes. In (a), poor views are rendered from Bed, Chair and Door objects. (b) denotes the distance comparison between the extracted features ***(green line)*** and the enhanced features ***(red line)*** of poor views in (a). The dashed line corresponds to groundtruth label index of each poor view. It is shown that the enhanced feature of poor view clearly decreases the distance between different categories and maintain high cosine distance in the same category. The figure is best viewed in color.

Table 2 presents a comprehensive comparison between NCENet and various state-of-the-art methods, for a fair comparison, two types of results are adopted, namely, macro and micro as defined in [19]. Macro aims to provide an unweighted average for the entire database, and micro addresses the influence of different model category sizes by offering a representative performance metric averaged across categories. An examination of the data presented in Table 2 shows that NCENet-Parallel and NCENet-Serial exhibit encouraging scalability and rotation invariance in the large-scale 3D competition, achieving state-of-the-art performances consistently for all evaluation metrics.

### 4.4. Discriminative Capacity for Poor Quality Views

In this experiment, we evaluate the discriminative capacity of NCENet on poor quality views of 3D shapes. We pick up some poor views, which contain little shape in-

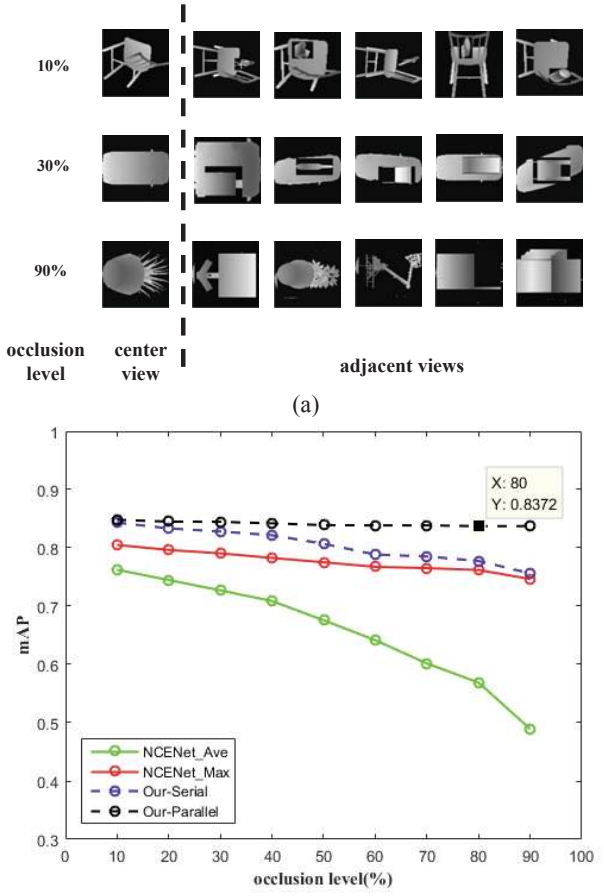



Figure 6. (a) is some examples of synthetic occluded neighboring views under different occlusion level. (b) is the performance comparison of different occlusion levels on ModelNet40 dataset.

formation and are difficult to identify, from ModelNet40. Then we compute the cosine distance between each poor view and other 3D shapes utilizing their representations. To keep comparison, both extracted features from CNN and our enhanced features from NCENet are adopted. We choose some views as examples, which are shown in Figure 5. As we can see, extracted features fail to identify the poor view because the distances between the poor view and shapes of 40 categories are relatively close. However, our enhanced feature of the poor view exhibits high similarity with its ground-truth category and the cosine distance with other categories is significantly decreased. It is shown that NCENet achieves better discriminative capacity for poor views utilizing the information of adjacent views to enrich the representation of center view, making it more suitable for retrieval tasks.

### 4.5. Robust Enhancement for Object Occlusion

In this section, we investigate the enhancement of the center view when neighboring views are occluded, which often occurs in real-world indoor scenes. In order to imitate the object occlusion, we use fixed size image patches to randomly cover the neighboring views, and the content of image patches comes from views of other shapes. We note that all of the methods are trained with clean projected views and are directly tested with noisy views. During the testing, each center view is clean and all its adjacent views are occluded in order to clearly analyze the robustness of the central view to the adjacent occlusion information. We compare different methods under multiple occlusion levels, the size of the occluded image patch ranging from 10% to 90% of the view, and the occlusion is added to all adjacent views. As shown in Figure 6, NCENet obtains the best performances, especially Our-parallel. Compared to the clean adjacent views, the mAP decreases slightly by $2.55\%$ for the parallel structure when the occlusion level is $80\%$, demonstrating the robustness of our center enhancement. Our-serial shows worse robustness than Our-parallel because of its sequential structure, where the later the view is entered the enhancement, the greater the impact on the final output representation. *More object occlusion experiments are shown in supplementary material.*

Table 3. The retrieval mAP of different numbers of center views on ModelNet40 dataset.

| Methods | The number of center view | | | | | |
|---|---|---|---|---|---|---|
| | 1 | 2 | 4 | 6 | 8 | 10 |
| Our-Parallel | 0.819 | 0.849 | 0.867 | 0.863 | 0.863 | 0.861 |
| Our-Serial | 0.838 | 0.862 | 0.868 | 0.871 | 0.870 | 0.868 |

Table 4. The retrieval mAP of different numbers of neighbor views on ModelNet40 dataset.

| Methods | The number of neighbor view | | | | |
|---|---|---|---|---|---|
| | 1 | 2 | 4 | 6 | 8 |
| Our-Parallel | 0.852 | 0.856 | 0.861 | 0.863 | 0.863 |
| Our-Serial | 0.853 | 0.860 | 0.867 | 0.868 | 0.871 |

### 4.6. Discussion & Analysis

**Number of center views.** To investigate the effect of the number of center views, we test Our-Parallel and Our-Serial on the ModelNet40 with different numbers of center views. The results are shown in Table 3. As we can see, the performance converges in general as the number of center views increases and the performance does not have much differ-

ence when the number of center views exceeds 6. Therefore, we use 6 center views for each shape in the testing.

**Number of neighbor views.** To evaluate the performance of our methods with different numbers of neighboring views, we train and test Our-Parallel and Our-Serial with one, two, four, six and eight neighboring views. Table 4 presents the results for the performance in terms of MAP on the ModelNet40 dataset. The performance gradually improves as the number of neighboring views increases. It is shown that more adjacent complementary information can significantly enhance the center feature. Therefore, we use the nearest eight neighboring views in our method.

**Correlation weight visualization** To investigate the proposed correlation unit, we have visualized some examples in Figure 7. The task of the correlation unit is to identify the neighbor views for which the content is significant for improving the center feature. It is expected that the neighboring view with a higher discriminability and a greater information difference between it and the center feature will have a higher correlation weight.

The correlation unit evaluates the discrimination of the views. For example, in Figure 7 (a), compared to other neighbor views, the upper_left adjacent views has lower weights equal to $0.26$, which may be because this view contains less shape information and it can not provide efficient discriminative information for center view to identify the target 3D shape.

Another important property of the correlation unit is the measurement of the information difference between neighboring views and the center view. In Figure 7 (d), the left and right neighboring views have lower weights equal to $0.02$ and $0.15$, respectively. Their appearances are similar to that of the center view, and thus, they fail to provide additional shape information to the center feature.

**Effect of the center loss on NCENet** We give an ablation study on center loss by evaluating the performance of NCENet with and without center loss on ModelNet40, which is shown in Table 5. NCENet-Parallel and NCENet-Serial obtain the improvements of $2.80\%$ and $4.01\%$, respectively, in mAP with center loss. We further investigate the learned correlation weights without center loss, and they fail to distinguish the content of different views compared to more distinguishable weights learned with the center loss. It is shown that center loss can enable the correlation unit to effectively filter the information from neighboring views to decrease intra-distance and increase inter-class distance, while softmax loss does not provide such strong distance monitoring signals.

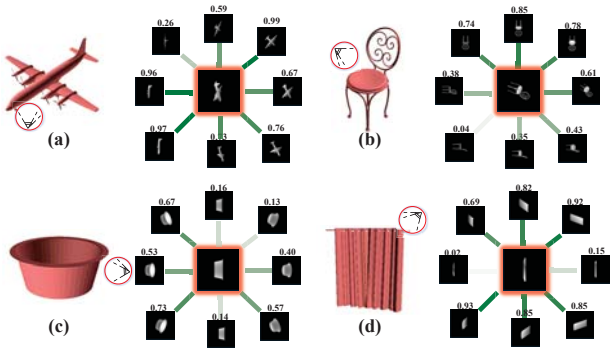



Figure 7. In this figure, four groups of correlation weights between the center view and its neighboring views are given. The red image is the center view, which is surrounded by its neighboring views from 8 directions. The intensity of the green line indicates the value of the corresponding weight. The correlation weight of the view is also shown in the upper part of each neighboring view. The figure is best viewed in color.

Table 5. Comparison of retrieval results in terms of center loss.

| Method | Softmax loss | | + Center loss | |
|---|---|---|---|---|
| | AUC | mAP | AUC | mAP |
| Our-Parallel | $84.57\%$ | $83.47\%$ | $87.28\%$ | $86.27\%$ |
| Our-serial | $84.17\%$ | $83.04\%$ | $88.04\%$ | $87.05\%$ |

## 5. Conclusion

In this paper, we proposed NCENet with two enhanced structures, named the parallel and serial enhanced structures, to learn a powerful representation for each view. More importantly, the correlation unit is proposed to determine the content correlation between adjacent views and their center view. Final experiments show that NCENet outperforms state-of-the-art methods for two 3D shape datasets and such enhancement for center view is highly robust against object occlusion in neighboring views. Our future work will explore how to optimize the center view to automatically select a specific set of views for the best enhancement.

## 6. Acknowledgements

This work is supported by NFS, the Beijing Municipal Natural Science Foundation (No.L182014), the Open Project Program of State Key Laboratory of Virtual Reality Technology and Systems, Beihang University (No.VRLAB2019C05), foundation of Science and Technology on Parallel and Distributed Processing laboratory (PDL), and the Fundamental Research Funds for the Central Universities.

## References


[1] Mathieu Aubry, Ulrich Schlickewei, and Daniel Cremers. The wave kernel signature: A quantum mechanical approach to shape analysis. In *ICCV Workshops*, pages 1626–1633. IEEE, 2011.

[2] Song Bai, Xiang Bai, Zhichao Zhou, Zhaoxiang Zhang, and Longin Jan Latecki. Gift: A real-time and scalable 3d shape search engine. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5023–5032, 2016.

[3] Ding-Yun Chen, Xiao-Pei Tian, Yu-Te Shen, and Ming Ouhyoung. On visual similarity based 3d model retrieval. In *Computer graphics forum*, volume 22, pages 223–232. Wiley Online Library, 2003.

[4] Guoxian Dai, Jin Xie, and Yi Fang. Siamese cnn-bilstm architecture for 3d shape representation learning. In *IJCAI*, pages 670–676, 2018.

[5] Jia Deng, Wei Dong, Richard Socher, and *et al*. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255. IEEE, 2009.

[6] Yifan Feng, Zizhao Zhang, Xibin Zhao, Rongrong Ji, and Yue Gao. Gvcnn: Group-view convolutional neural networks for 3d shape recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 264–272, 2018.

[7] Xinwei He, Yang Zhou, Zhichao Zhou, Song Bai, and Xiang Bai. Triplet-center loss for multi-view 3d object retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1945–1954, 2018.

[8] Anastasia Ioannidou, Elisavet Chatzilari, Spiros Nikolopoulos, and Ioannis Kompatsiaris. Deep learning advances in computer vision with 3d data: A survey. *ACM Computing Surveys (CSUR)*, 50(2):20, 2017.

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

[10] Edward Johns, Stefan Leutenegger, and Andrew J Davison. Pairwise decomposition of image sequences for active multi-view recognition. In *CVPR*, pages 3813–3822, 2016.

[11] Roman Klokov and Victor Lempitsky. Escape from cells: Deep kd-networks for the recognition of 3d point cloud models. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 863–872, 2017.

[12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[13] Chunyuan Li and A Ben Hamza. Symmetry discovery and retrieval of nonrigid 3d shapes using geodesic skeleton paths. *Multimedia tools and applications*, 72(2):1027–1047, 2014.

[14] Zhaoqun Li, Cheng Xu, and Biao Leng. Angular triplet-center loss for multi-view 3d shape retrieval. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8682–8689, 2019.

[15] Bui Tuong Phong. Illumination for computer generated pictures. *Communications of the ACM*, 18(6):311–317, 1975.

[16] Charles R. Qi, Hao Su, Matthias Niebner, Angela Dai, Mengyuan Yan, and Leonidas J. Guibas. Volumetric and multi-view cnns for object classification on 3d data. In *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] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[19] M Savva, F Yu, Hao Su, and *et al*. Shrec16 track large-scale 3d shape retrieval from shapenet core55. In *Proceedings of the Eurographics Workshop on 3D Object Retrieval*, 2016.

[20] Baoguang Shi, Song Bai, Zhichao Zhou, and Xiang Bai. Deeppano: Deep panoramic representation for 3-d shape recognition. *IEEE Signal Processing Letters*, 22(12):2339–2343, 2015.

[21] Philip Shilane, Patrick Min, Michael Kazhdan, and Thomas Funkhouser. The princeton shape benchmark. In *Shape modeling applications*, pages 167–178. IEEE, 2004.

[22] Ayan Sinha, Jing Bai, and Karthik Ramani. Deep learning 3d shape surfaces using geometry images. In *ECCV*, pages 223–240. Springer, 2016.

[23] 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.

[24] Johan WH Tangelder and Remco C Veltkamp. A survey of content based 3d shape retrieval methods. In *Shape Modeling Applications, 2004. Proceedings*, pages 145–156. IEEE, 2004.

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

[26] Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao. A discriminative feature learning approach for deep face recognition. In *ECCV*, pages 499–515. Springer, 2016.

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

[28] Jin Xie, Yi Fang, Fan Zhu, and Edward Wong. Deepshape: Deep learned shape descriptor for 3d shape matching and retrieval. In *CVPR*, pages 1275–1283, 2015.

[29] Kai Xu, Yifei Shi, Junyu Zhang, Junyu Zhang, Min Liu, Hui Huang, Hao Su, Daniel Cohen-Or, and Baoquan Chen. 3d attention-driven depth acquisition for object identification. *ACM Transactions on Graphics*, 35(6):238, 2016.