Machine Intelligence Research
www.mi-research.net

20(6), December 2023, 872-883
DOI: 10.1007/s11633-023-1430-z

MVContrast: Unsupervised Pretraining for Multi-view 3D Object Recognition

Luequan Wang Hongbin Xu Wenxiong Kang

School of Automation Science and Engineering, South China University of Technology, Guangzhou 510641, China

Abstract: 3D shape recognition has drawn much attention in recent years. The view-based approach performs best of all. However, the current multi-view methods are almost all fully supervised, and the pretraining models are almost all based on ImageNet. Although the pretraining results of ImageNet are quite impressive, there is still a significant discrepancy between multi-view datasets and ImageNet. Multi-view datasets naturally retain rich 3D information. In addition, large-scale datasets such as ImageNet require considerable cleaning and annotation work, so it is difficult to regenerate a second dataset. In contrast, unsupervised learning methods can learn general feature representations without any extra annotation. To this end, we propose a three-stage unsupervised joint pretraining model. Specifically, we decouple the final representations into three fine-grained representations. Data augmentation is utilized to obtain pixel-level representations within each view. And we boost the spatial invariant features from the view level. Finally, we exploit global information at the shape level through a novel extract-and-swap module. Experimental results demonstrate that the proposed method gains significantly in 3D object classification and retrieval tasks, and shows generalization to cross-dataset tasks.

Keywords: Multi view, unsupervised pretraining, contrastive learning, 3D vision, shape recognition.

Citation: L. Wang, H. Xu, W. Kang. MVContrast: Unsupervised pretraining for multi-view 3D object recognition. *Machine Intelligence Research*, vol.20, no.6, pp.872–883, 2023. http://doi.org/10.1007/s11633-023-1430-z

1 Introduction

With the development of AR/VR/MR and autonomous driving, 3D object recognition has received more attention. At present, 3D object representations in industrial projects and scientific research mainly focus on point clouds, voxels, and multi-view images. Point clouds are a set of unordered points that can be obtained by radar scanning, but their sparse structure will bring information loss. In addition, point cloud labelling is still a major challenge. Voxels are a kind of data structure that uses a fixed size cube as the smallest unit to represent a 3D object. The acquirement of voxels is not easy at the present stage, not to mention the relatively strict requirements for resources. By contrast, multi-view data can be easily captured just with a camera placed at different angles. As 2D images, multi-view data can retain richer texture detail without complicated data cleaning or preprocessing work. It can be directly handled end-to-end with any current mature 2D model such as VGG[1] and ResNet[2]. Moreover, the acquisition of multi-view data also accords with the imaging process of human perception of the external environment, so it is most likely to be utilized to mimic the human visual system. The above makes multi-view-based methods take the lead in 3D object recognition in several available datasets. Pretraining in 2D visual learning is widely used because it can accelerate the convergence process, perfect the training model′s generalization, and bring evident performance improvement. Consequently well-reasoned pretraining can also be deduced to the exceptional "2D visual learning" of multi-view learning.

The pretrained model of ImageNet is standard in various visual tasks, and sufficient experimental analyses have proven the impressive improvement in performance. The pretraining models for multi-view learning at present are all ImageNet based. However, there are several problems when it is applied in a multi-view learning task:

1) The ImageNet dataset is enormous, which requires considerable staffing and material resources to annotate, so it is not easy to copy the same operation on a second dataset.

2) The distribution shift between ImageNet and the target dataset may degrade the performance. According to [3], the exact solution to the nonlinear dynamics of learning in deep linear neural networks, only if the statistical structure of the input is consistent with the mapping structure of the information and output to be learned, the pretraining will bring ideal results.

3) The ImageNet pretraining scheme lacks the implicit constraint of multi-view consistency. Because multi-view images naturally have spatial complementarity, the

Research Article
Manuscript received on November 1, 2022; accepted on February 24, 2023; published online on May 10, 2023
Recommended by Associate Editor Chun-Hua Shen
Colored figures are available in the online version at https://link.springer.com/journal/11633

adoption of the ImageNet pretraining model treats and aligns each view independently. As a result, losses of relevance within and across classes of the multi-view dataset are inevitable.

Current supervised learning (pretraining) pipelines heavily rely on high-quality annotations, which requires label-intensive work and enormous computational resources. Let alone a relatively more expensive 3D annotation task. This leads the community to explore alternatives in an unsupervised manner. Pretraining is affected to some extent by the size of the dataset. Exactly unsupervised pretraining can take advantage of a nearly infinite set of unlabelled training data.

To tackle the problems existing in the pretraining model of current multi-view learning and make full use of multi-view′s rich texture and unique spatial character, we propose a three-stage self-supervised pretraining method in this paper, including AugViews, CrossViews, and RobustViews. The alignment of the final feature descriptor in the feature domain is completed through the weighted joint training of the above three tasks. According to SimCLR[4], multiple data augmentation operations are crucial in designing contrastive proxy tasks that produce effective representations. During the AugViews stage, just with some simple data augmentation operations, the network can perceive the invariant characteristics[5] and then obtain a more substantial representation power. In the CrossViews stage, a classic method of contrastive learning method is adopted to try to conduct all the view features under each shape and aggregate the view features under the same shape to increase the differentiation between them under different shapes with an InfoNCE loss. However, a problem arises that various subjects might be of the same class. The chances are that the focus on local features results in a loss of generalization as too much attention is given to several specific samples. Then in the RobustViews stage, we propose a novel structure to compensate for too much focus on the specific feature. MVCNN[6] proved through experiments that the view feature with the highest response is the largest one or even the only one that contributes to the final shape feature. To improve the representation capacity of the shape feature, we construct positive sample pairs by a frame extract-and-swap process so that the shape feature can perceive the knowledge brought by different views. Decoupling the association between a shape feature and a single view feature can improve the generalization of the pretraining model. An illustration of our three stage end to end unsupervised pretraining mechanism is provided in Fig. 1

The main contributions of this paper can be summarized as follows:

1) We propose a three-stage pretraining model that balances local and global features based on full use of the data, different from the previous self-supervised models that focus only on one aspect.

2) We show that there is much more to explore with raw data, and our proposed method can learn a good representation without introducing any additional data and annotations.

3) We evaluate the quality of our approach as a pretraining step on ModelNet40 and a more challenging dataset Shrec17. The experimental results show significant improvement over a model trained from scratch.

2 Related works

2.1 Unsupervised pretraining

Unsupervised pretraining models typically require training through proxy tasks on large datasets to introduce extra information, and then finetuning on downstream tasks to achieve final parameter alignment. At present, most classical visual tasks use ImageNet as a pretraining default setting, or other much larger datasets. However, considering the feature refinement requirements of high-level visual tasks, the performance improvement of finetune is limited even if an extremely large classification dataset is adopted[3].

In unsupervised pretraining, the proxy task is just a

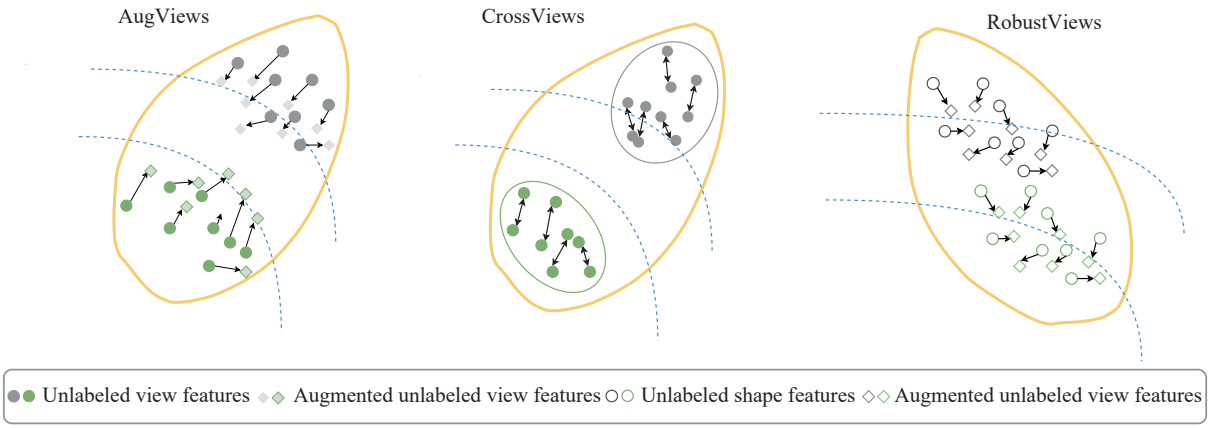

Fig. 1 Conceptual descriptions of the feature alignment approaches. The three domains describe the proposed AugViews, CrossViews, and RobustViews schemes, which are explained in Section 3 in detail.

trampline, and we care more about the representation of the learned feature descriptor than the final performance of the proxy task. Reconstruction-based tasks and contrast-based tasks are two classical directions of unsupervised proxy tasks. Reconstruction-based tasks mainly explore learning a good representation by computing pixel-level reconstruction or generation tasks, including Inpainting[7], Colorization[8, 9], Puzzle[5], GAN[10], and super resolution[11]. However, these efforts are more illuminating but have limited performance gains. Contrast-based learning is the current mainstream direction. It introduces a large number of negative samples, divides positive and negative samples through different strategies, and completes the alignment of positive and negative samples in the feature space. Positive sample pairs are closer, while negative sample pairs are farther away.

Contrast-based learning is more beneficial to the backbone to establish spatial correlation and characterize the complex structure of objects[12, 13]. The breakthrough of the contrastive-based method is SimCLR, which introduces noise and rotation transformation to each image in batch to obtain positive sample pairs of the original images. The network then employs InfoLoss to achieve a gradient decrease. Due to the large batch size and a large number of positive and negative sample pairs, the linear classifier trained by SimCLR can be comparable to ResNet50. In later work, MOCO[14, 15] reduces memory consumption through memory banks and introduced momentum updates. BYOL[16] adds prediction head, and only positive sample pairs are used for training.

2.2 Multi-view learning

Multi-view learning of all kinds has been a hot issue[17]. For 3D models, multi-views are one of the classic representations. Compared with points, meshes and voxels, multi-views contain more texture information of 3D models. The capture of multiple views more simulates the process of human eyes′ perception of the real world. In addition, compared with the point-based and mesh based methods, the view-based method takes a lead in the performance of each dataset.

A pioneering of view-based method is MVCNN. MVCNN encodes each view through CNNs with shared weight. It creatively proposes to aggregate the final global feature descriptor by using max pooling. However, MVCNN treats each view individually and does not make effective use of the relationship between different views. The subsequent works gradually mine the correlation between views and intra views. GVCNN employs FCN to calculate a weight of views respectively. The views are grouped according to their weights, and then global features are aggregated through a three-stage feature extractor. MHBN[18] divides each view into blocks to realize the feature abstraction process from a patchwise perspective. In addition, general max-pooling is replaced with bilinear-pooling to take the second-order statistics into consideration. View-NGram[19] refers to the N-Gram idea in natural language processing to group views, and further adopts blocks similar to self-attention to complete the aggregation of local features. In recent years, there have also been some works focusing on graph topological association of the multi-view[20, 21]. 3DViewGraph[22] takes into account that the pooling operation will lose the information between views. It builds a graph structure of all the views and abstracts the disordered view nodes with a delicate attention module. View-GCN[23] also regards the unordered view as a graph structure and uses GCN to extract the global characteristics of the graph layer by layer.

3 Method

3.1 Overview

We develop a scalable three-stage pretraining method, MVContrast, for 3D representations that utilize the raw data without any extra annotations. Our method, illustrated in Fig. 2, comprises three submodules. AugViews distill the information within a view (Section 3.2). CrossViews explores the correlation between different views of the same shape (Section 3.3). RobustViews proposes a novel module from the global perspective to discover the uniqueness of category features (Section 3.4).

3.2 AugViews

The AugViews module aims to obtain pixel-level representations within each view which is illustrated in Fig. 2 "AugViews". Given a dataset $D = \{s\}_{i=1}^{S}$, containing S shape samples, we wish to learn a function $G(s)$ that for the design of a proxy task to capture the individual feature of each view with relatively fine-grained data. We regard each view under each shape as an independent view v_i^j, and get augment view $v_i^{j'}$ through some data augment strategies. Input v_i^j and $v_i^{j'}$ into the encoder and obtain view featurse z_i^j and $z_i^{j'}$ respectively. With InfoNCE loss, z_i^j and $z_i^{j'}$ are drawn closer and z_i^j and z_i^k, which are from different shapes, are pushed farther.

$$\ell_A(i,j) = -\log \frac{\exp\left(\text{sim}\left(\boldsymbol{z}_i^j, \boldsymbol{z}_i^{j'}\right)/\tau\right)}{\sum\limits_{k=1,[k\neq i]}^{2N} \exp\left(\text{sim}\left(\boldsymbol{z}_i^j, \boldsymbol{z}_i^k\right)/\tau\right)}. \quad (1)$$

In (1), τ can guarantee the smoothness of the softmax process, and sim measures the cosine distance of the pair. That is, we wish that features from the same raw view are more similar, while the features from different views are more diverse. The model is able to learn invariant view features by aligning view features in the latent

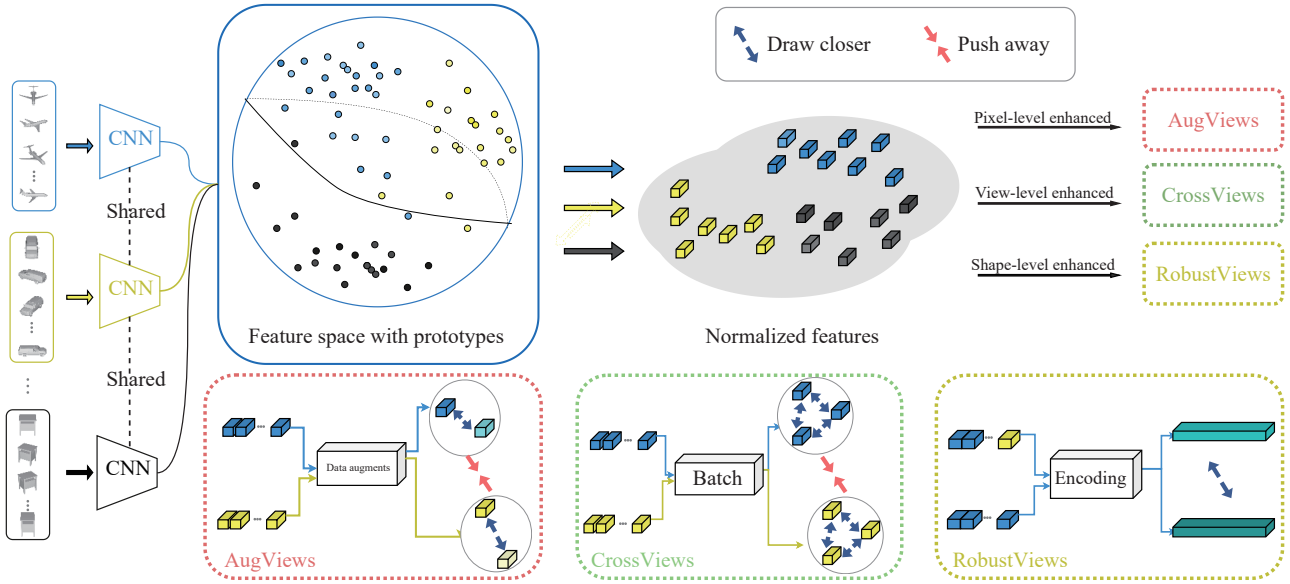

Fig. 2 Illustration of the proposed framework. After encoding different views, the alignment in latent space is completed through three stages of head.

space. It is worth noting that compared with [4], we only employ ResizedCrop, RandomHorizontalFlip and other data augmentation strategies trying to focus on capturing high-level geometric information. Considering that reconstruction tasks such as color gamut transformation and denoising are too fine-grained, they are not conducive to knowledge transfer for shape recognition tasks.

In addition, considering the importance of batch size and negative sample amount for contrastive learning, we adopted the same configuration as SIMCLR[4].

3.3 CrossViews

The CrossView module is designed to capture spatially invariant features from the view level as illustrated in Fig. 2 "CrossViews". Given a shape $S = \{V\}_{i=1}^{N}$ containing N views of a shape. The purpose of CrossViews is to learn the relationship between views under the same shape and those under different shapes as well as mine the spatial invariance feature.

The input shapes are $S1$ and $S2$, and the views $\{V_1\}_{i=1}^{N}$ and $\{V_2\}_{i=1}^{N}$ are generated by the projection from different views. The views are encoded by backbone to obtain the view features, and we can get an InfoNCE (2) of CrossViews by the extension of (1).

$$\ell_C(i,j) = -\log \frac{\sum_{v=1,[v\neq j]}^{N} \exp\left(\text{sim}\left(z_i^j, z_v^j\right)/\tau\right)}{\sum_{k=1,[k\neq i]}^{S}\sum_{v=1}^{N} \exp\left(\text{sim}\left(z_i^j, z_v^k\right)/\tau\right)}. \quad (2)$$

When the shape to which view v_1 belongs is different from the shape to which view v_2 belongs, the features of v_1 and v_2 should be more disparate, and vice versa. Through the simple design, we can see that the experimental results are significantly improved in the subsequent experiments, which also indicates that the design of CrossViews makes the pretraining model better represent the most important spatial features of multi-view data.

3.4 RobustViews

The RoubstView module is proposed to support global information at the shape level as illustrated in Fig. 2 "RobustViews". MVCNN shows that the feature descriptor obtained by conducting a global pooling of view features can best represent the shape feature, that is, the final shape feature shares only one view feature content. Although different views contribute to the final shape feature to different degrees, human eyes can still easily distinguish shape from a "bad" view in real life. This means that even the information from a less than "good" view may be enough to represent the shape. The design of RobustViews decouples the dependence of the final shape feature on a fixed view and obtains a more robust shape representation through perceptual learning of the global views. By replacing different views under a shape with random frames, a new view set can be built to represent the same shape. The detailed process is shown in Fig. 2. By extending (1), we can obtain an InfoNCE loss (3) of shape feature.

$$\ell_R(i) = -\log \frac{\exp\left(\text{sim}\left(f_i, f'_i\right)/\tau\right)}{\sum_{j=1,[j\neq i]}^{2S} \exp\left(\text{sim}\left(f_i, f_j\right)/\tau\right)}. \quad (3)$$

In (3), f_i represents the origin feature descriptor, and f_i' represents the descriptor of the new set. By shortening the distance between the new shape and the raw

shape in the feature space, and making the shape feature of different shapes farther away makes the model have a better understanding of a "good" shape feature, making the model more discriminative.

3.5 Training process

Let us denote the AugViews, CrossViews, and RobustViews objectives by L_A, L_C, and L_R, respectively. We define the multi-task objective as a linear combination of these objectives:

$$L = \lambda_1 L_A + \lambda_2 L_C + \lambda_3 L_R. \tag{4}$$

In (4), to limit the three losses to the same scale λ_1, λ_2 and λ_3 are initialized as 1, 0.25 and 0.01, respectively. Moreover, we use early stopping to prevent overfitting or a dominating task.

4 Experiment

4.1 Implementation details

Datasets. ModelNet40 is an open-source 3D CAD model dataset from Princeton and a commonly used 3D model dataset in recent years. It consists of 12 311 shapes from 40 categories, including 9 843 training samples and 2 468 test samples. SHREC17, a large-scale 3D retrieval competition dataset based on ShapeNet, is a very challenging classic benchmark. It consists of 51 162 shapes in 55 categories and 203 subclasses, including 35 764 training samples, 5 133 validation samples, and 10 265 test samples. The number of shapes varies significantly among different classes. Our experiments are conducted on a typical dataset, all shapes of which have been prealigned and normalized.

For each 3D shape, we render a series of 256×256 2D grayscale images according to the same strategy as MVCNN. We place a virtual camera around the object at 30-degree intervals and end up with 12 view images. Before inputting them into the network, we scale the 256×256 images to 224×224.

During self-supervised pretraining, the backbone is a ResNet18 in default. We adopt an SGD optimizer with an initial learning rate of 0.01, momentum of 0.9 and weight decay of 0.000 1. To normalize task weights to the same scale, we reweight AugViews, CrossViews, and RobustViews to 1.0, 0.25, and 0.01 respectively. And to reduce overfitting, we train the model for 100 epochs with a batch size of 24.

The pretrained feature extractor (e.g., ResNet18) is finetuned on the same dataset as that in pretraining to make the feature extractor, as well as a new classifier, perform better. In the inference time, we extract a 1 024-dimensional embedding as the 3D shape descriptor.

We implement our method using PyTorch with the CUDA backend. All our experiments are conducted on a GeForce GTX 1 080 Ti graphics card.

4.2 Experiments on 3D shape classification

Compared with the current methods that employ overfitting training as well as data preprocessing to perform better (MVTN[24] introduces a new network to learn some specific viewpoints), we use the classic and recurring MVCNN[6] as our baseline. To verify the validity of our method, we first use the common self-supervised learning test method (finetuning the unsupervised learned representation with a linear SVM on the training set and test on the test set) to demonstrate the representation of our model directly. In our experiments, we first compare with a series of unsupervised state-of-the-art learning approaches on ModelNet40. The experimental results are demonstrated in Table 1. These methods include the voxel-based methods TL Network[25], VConv-DAE[26], 3DGAN[27], VSL[28], point-based methods LGAN[29], FoldingNet[30], 3D-PointCapsNet[31], MAP-VAE[32] and the view-based method VIPGAN[33]. Our self-supervised approach achieves a comparable 90.24% with the full-supervised baseline 91.94% and outperforms most unsupervised learning methods. In addition, it should be noted that VIPGAN adopts a delicately designed GAN structure and the starting point is completely different from our method.

With fully supervised training, our method achieves a classification accuracy of 92.54%. It has gained 2.44% and 0.60% compared with the vanilla MVCNN and the MVCNN with a ResNet18[2] as the backbone. To show that our method can be generalized for various networks, we also take a re-implement GVCNN[34] as another baseline on the ModelNet40. Pretraining in our self-supervised manner, GVCNN can reach a result of 92.99% which boosts the effect given in the paper (under 12 views) by 0.39%. The above experiments indicate that our method can extract more representative features and when compared with the conventional method that pretrains through a super large dataset (such as ImageNet[35]), our method can optimize the initial model in a more favorable direction.

We then conduct a comparison experiment within the model on a more challenging dataset SHREC17. The experimental results are shown in Table 2. It can be seen that in the absence of any supervising signals, our model can still achieve 72.69% in overall accuracy (Acc), but only 38.34% in avg-Acc. This is largely due to the highly unequal data distribution (the largest category has 150 times more samples than the smallest category). This data distribution problem is also quite unfriendly to unsupervised learning. Pretraining with our model, it can be seen that the model′s performance has been significantly improved, and it has gained 1.26% and 5.45% in Acc and

Table 1 Comparison of 3D shape classification against methods on the ModelNet40 dataset

	Input	Supervised	Acc (%)
TL Network[25]	Voxel	×	74.40
VConv-DAE[26]		×	75.50
3DGAN[27]		×	83.30
VSL[28]		×	84.50
LGAN[29]	Points	×	85.70
FoldingNet[30]		×	84.36
3D-PointCapsNet[31]		×	88.90
MAP-VAE[32]		×	90.15
VIPGAN[33]	Views	×	91.98
MVCNN+ImageNet		√	90.10
MVCNN (ResNet18)		√	90.23
MVCNN (ResNet18)+ImageNet		√	91.94
GVCNN[34]+ImageNet		√	92.60
MVTN[24]+ImageNet+ViewPoints		√	93.80
Ours w/o MVCNN	Views	×	90.24
Ours w/ MVCNN		√	92.54
Ours w/ GVCNN		√	**92.99**

Table 2 Comparison of 3D shape classification against methods on the SHREC17 dataset

Method	Results	
	Acc (%)	Avg-Acc (%)
MVCNN (ResNet18)	86.14	71.21
Ours w/o MVCNN	72.69	38.34
Ours w/ MVCNN	**87.40**	**76.66**

avg-Acc, respectively. This also shows that our approach is applicable even in more challenging and large-scale datasets.

In Table 3, we evaluate our model alongside mainstream self-supervised pretraining models. Our method performs better than self-supervised methods designed under 2D modalities in both unsupervised pretraining and fine-tuning tasks. We compare the effect of pretraining SimCLR in ImageNet, STL10 and BYOL in ImageNet, respectively. SimCLR's self-supervised learning in STL10 achieves less than 80% accuracy on the classification task, which is much worse than our approach. This is even worse than our first stage AugViews in Table 4. Since the STL10 dataset contains only 10 categories, the amount of information contained is much less than that of ModelNet40. It can be seen that the pretraining results of SimCLR on ModelNet40 are significantly improved. However, it is still not as good as our AugViews. This is mainly due to our retuning of the proxy tasks for such a relatively simple textured dataset. We can see that SimCLR and BYOL perform better than AugViews due to the use of

Table 3 Comparison of 3D shape classification with different self-supervised pretraining methods on ModelNet40

Pretrain	Supervised	Overall Acc (%)
SimCLR+STL10	×	79.20
SimCLR+ModelNet40	×	86.38
SimCLR+ImageNet	×	88.73
Byol+ImageNet	×	89.21
Byol+ImageNet	√	92.06
Ours	×	90.24
Ours	√	92.54

Table 4 Impacts of different stages of the classification accuracy on ModelNet40

AugViews	CrossViews	RobustViews	Results	
			Acc (%)	mAP (%)
×	×	×	48.5	47.1
×	×	√	79.4	48.6
×	√	×	85.3	50.9
√	×	×	88.5	64.3
√	√	×	89.4	69.2
√	√	√	90.2	70.8

massive data from ImageNet, but our final model outperforms them by 1.5% and 1% due to fully exploiting the spatial geometry information of the 3D multi-view. Moreover, after fully supervised fine-tuning, our model still outperforms the self-supervised approach for 2D images.

In Table 5, we show the experimental results on different backbones. We choose VGGM and larger backbones such as ResNet18 and ResNet50 under different pretraining methods. It can be seen that our model is significantly boosted under different backbones. Our model achieves a 1.62% improvement in classification accuracy with VGGM. And when it comes to ResNet, our model also outperforms by more than 0.6%. More importantly, as the backbone goes deeper in ImageNet pretraining mode, the final performance improvement is limited. When switching from ResNet18 to ResNet50, the improvement is only 0.22, while our model almost doubles the improvement. This is mainly because our pretraining approach is able to capture more effective information. As the network deepens, the representational power of the backbone becomes stronger.

4.3 Experiments on 3D shape retrieval

To validate the effectiveness of our method, we conduct experiments on a common dataset Modelnet40 and a more challenging dataset SHREC17, respectively. The measurement metrics on ModelNet40 adopt the open-

Table 5 Comparison of 3D shape classification with different backbones on ModelNet40

Backbone	Pretrain	Overall Acc (%)
VGG M	ImageNet	90.10
ResNet18	ImageNet	91.94
ResNet50	ImageNet	92.16
VGG M	Ours	91.72
ResNet18	Ours	92.54
ResNet50	Ours	92.97

source test method in PVRNet[36]. The comparison experiments are listed in Table 6. In the experiments, we compare three voxel-based methods SPH[37], 3DShapeNet[38], and DLAN[39], and several view-based methods DeepPano[40], GIFT[41], RAMA[42], GVCNN[34], TCL[43], and VNN[19]. (Superscript ξ represents low-rank Mahalanobis metric learning). Our method reaches 70.8% in mAP without any supervision, which is even better than the vanilla MVCNN in the case of full supervision. Pretraining with our method, it can be seen that the final result can reach 87% in mAP, which boosts 17% in mAP more than vanilla MVCNN and even 7% more in mAP than MVCNN+Metric. In comparison, GVCNN achieves a result of 90% in mAP, which has gained of 4% in mAP compared with the original model and even 1% in mAP compared with the state-of-the-art VNN.

Table 6 Comparison of 3D shape retrieval against methods on the ModelNet40 dataset

Method	Input	mAP (%)
SPH[37]	Voxel	33.3
3DShapeNet[38]		49.2
DLAN[39]		85.0
DeepPano[40]	Views	76.8
MVCNN		70.1
MVCNN$^{\xi}$		80.2
GIFT[41]		81.9
RAMA[42]		83.5
GVCNN[34]		85.7
TCL[43]		88.0
VNN[19]		89.3
Ours w/o MVCNN	Views	70.8
Ours w/ MVCNN		87.1
Ours w/ GVCNN		**90.3**

In the experiments on SHREC17, we follow the rules of the track of SHREC17 for large-scale 3D shape retrieval on ShapeNet Core55[44]. For each query object, all objects with the same predictive label are taken as the retrieval shape.

The retrieval order is determined according to the ranking of the similarity score. A retrieval list is required to provide at most 1 000 results. In the experiment, we compare our method with some of the approaches used in the competition, including the voxel-based approaches ZFDR[45], DeepVoxNet[46], and DLAN, as well as the view-based approaches CM-VGG5-6DB, GIFT, ReVGG, MVFusionNet, and RotationNet[47]. It can be seen from the Table 7 that compared with vanilla MVCNN and MVCNN (ResNet18), the model pretrained with our method has a great improvement under micro-average standard (an average without reweighting based on category size) in P@N, R@N, F1@N, mAP, and NDCG. This also proves the validity of our model. It is noteworthy that under the Marco-average standard (an unweighted average over the entire dataset), our model even exceeds RotationNet in P@N, R@N, F1@N, and NDCG. This also indicates that our model is more robust for diverse objects than methods that overfit the distribution of specific datasets.

4.4 Ablation study

Different stages

To verify the actual effectiveness of our designs, we present ablation experiments of the impacts of different stages on the unsupervised learned features. All the contrast experiments still follow the MVCNN structure. Detailed results on ModelNet40 are summarized in Table 4. The baseline model is a vanilla MVCNN using ResNet18 as a feature extractor. We adopt classification accuracy (Acc) and mean average precision (mAP) as evaluation criteria for the models. To obtain Acc, the feature extractor is fixed, and only the parameters of the classification layer are updated during the training process. While mAP is used to directly evaluate the performance of unsupervised features without any extra supervised information.

The baseline model can only achieve a result of 48.5% and 47.1%. Moreover, during the training process, the convergence rate of the baseline model is so slow that it takes approximately 15 epochs to determine the ultimate optimization direction. We see that the model with AugViews can significantly improve the baseline by 50% and 17%. AugViews adopts the approach of SimCLR[4] to characterize the features of a single view. Concerning the universality and stability of the application of data augmentation methods in self-supervised learning, we decide to take AugViews as a primary loss. We can see that the model assembles only CrossViews or RobustViews can also enable the experimental results to reach 85.3% and 50.9% respectively. Consider that CrossViews tries to learn the differences between views, while RobustViews attempts to distinguish the representations between objects (categories). As the proxy tasks become more brutal, the information that a simple backbone can learn is

Table 7 Comparison of 3D shape retrieval against methods on the ShapeNet Core55 dataset

Method	MicroALL					MacroALL				
	P@N (%)	R@N (%)	F1@N (%)	mAP (%)	NDCG (%)	P@N (%)	R@N (%)	F1@N (%)	mAP (%)	NDCG (%)
ZFDR[45]	53.5	25.6	28.2	19.9	33.0	21.9	40.9	19.7	25.5	37.7
DeepVoxNet[46]	79.3	21.1	25.3	19.2	27.7	59.8	28.3	25.8	23.2	33.7
DLAN	81.8	68.9	71.2	66.3	76.2	61.8	53.3	50.5	47.7	56.3
CM-VGG5-6DB	41.8	71.7	47.9	54.0	65.4	12.2	66.7	16.6	33.9	40.4
GIFT	70.6	69.5	68.9	64.0	76.5	44.4	53.1	45.4	44.7	54.8
ReVGG	76.5	80.3	77.2	74.9	82.8	51.8	60.1	51.9	49.6	55.9
MVFusionNet	74.3	67.7	69.2	62.2	73.2	52.3	49.4	48.4	41.8	50.2
MVCNN	77.0	77.0	76.4	73.5	81.5	57.1	62.5	57.5	56.6	64.0
MVCNN (ResNet18)	78.7	79.1	78.3	74.2	83.7	58.1	59.8	56.7	52.6	61.8
RotationNet	**81.0**	**80.1**	**79.8**	**77.2**	**86.5**	60.2	63.9	59.0	**58.3**	65.6
Ours w/o MVCNN	54.7	59.4	55.5	46.9	60.2	25.6	39.0	25.8	23.2	34.1
Ours w/ MVCNN	79.8	79.9	79.4	75.6	84.9	**62.1**	**65.4**	**62.1**	58.2	**66.3**

limited, so the result is understandably lower than AugViews. However, as all stages are incorporated, and knowledge learned from diverse tasks is introduced, the final model yields a performance of 90.2% and 70.8%.

Number of views

To investigate the influence of different views on the model, we adjust the number of views introduced during the self-supervised training process. The results are presented in Table 8. When the number of views is 4, the model achieves a result of 91.61% and 89.96% in Acc and avg-Acc, respectively. With the increase in the number of views, the classification performance improves gradually, and the Acc performance is the best when it comes to 12 views. It is reasonable that too few views may lead to a loss of useful information approximately 3D shapes. Nevertheless, it can be seen that as the number of views grows, the improvement on avg-Acc is relatively limited (e.g., 90.24% for 12 views and 90.25% for eight views). This may be because an increase in the number of views can not provide enough information to make the model more discriminative on some hard categories.

Table 8 Comparison of different numbers of views during the test process

Number of views	Results	
	Acc (%)	Avg-Acc (%)
4	91.61	89.96
8	92.02	90.25
12	92.54	90.24

Robustness towards missing views

In the real world, it is difficult to avoid missing views, and we expect that our model can effectively cope with similar situations. In this section, we conduct experiments on the robustness of MVCNN and MVCNN with our pretraining method to the missing views during test time. The results are drawn in Fig. 3. From the curve, we can see that 4-views is an inflection point, and the performance of both models declines sharply when the number of views is less than 4. It is reasonable that too few views lose much information about the 3D shape. It can be seen that the MVCNN without any pretraining declines earlier, and the degradation is more serious. When the number of views is only 1, it tends to make a random prediction of the input. It is demonstrated that our pretraining method is more robust to the problem of missing views.

4.5 Visualization

Feature distribution. To intuitively understand the impact of different losses on our model, we map the high-dimensional abstract unsupervised features on the ModelNet40 test set to two-dimensional space for a decoupling visualization. The visualization results are presented in Fig. 4 by t-SNE. It can be seen in Fig. 4 that the learned features gradually show cluster-friendly behavior with the introduction of losses at each level. In addition, it is noted that in comparison with the point-based unsupervised SOTA method[48], our method has a distinct enhancement in feature identification, which also reflects the strong discriminative power of our representation.

Feature heatmap. Gradcam is used to obtain a heatmap from the backbone after three-staged unsupervised pretraining. This heatmap can represent the contribution of different pixel regions to the final classification probability. That is to say, and it means the influence of different regions on the final feature descriptor. It can be seen in Fig. 5, the monitor pays more attention to the junction of bracket and screen, corners dominate in the tent, the airplane is mainly interested in the nose and wings, and the body part receives more attention in the

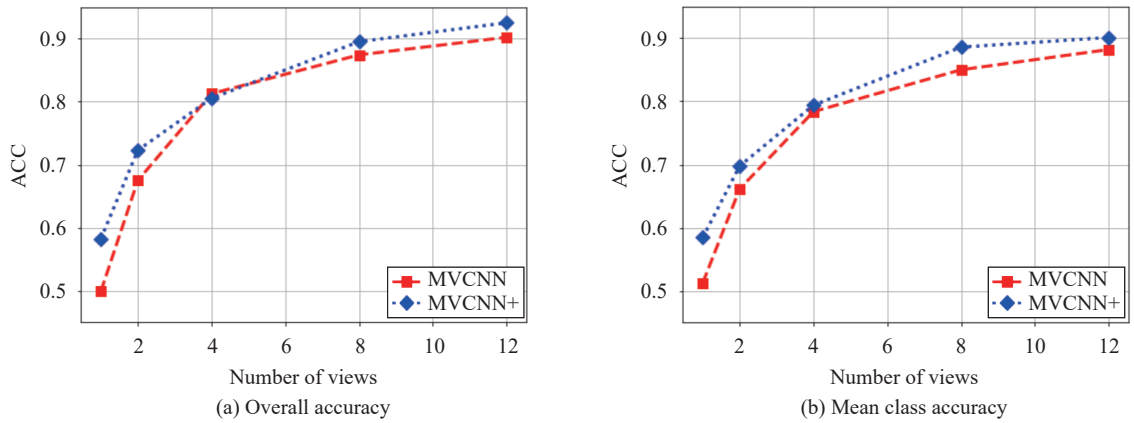

Fig. 3 Experimental results on the robustness towards missing views. (a) and (b) present the overall accuracy and mean class accuracy of MVCNN without any pretraining and MVCNN with our pretraining model (MVCNN+), respectively.

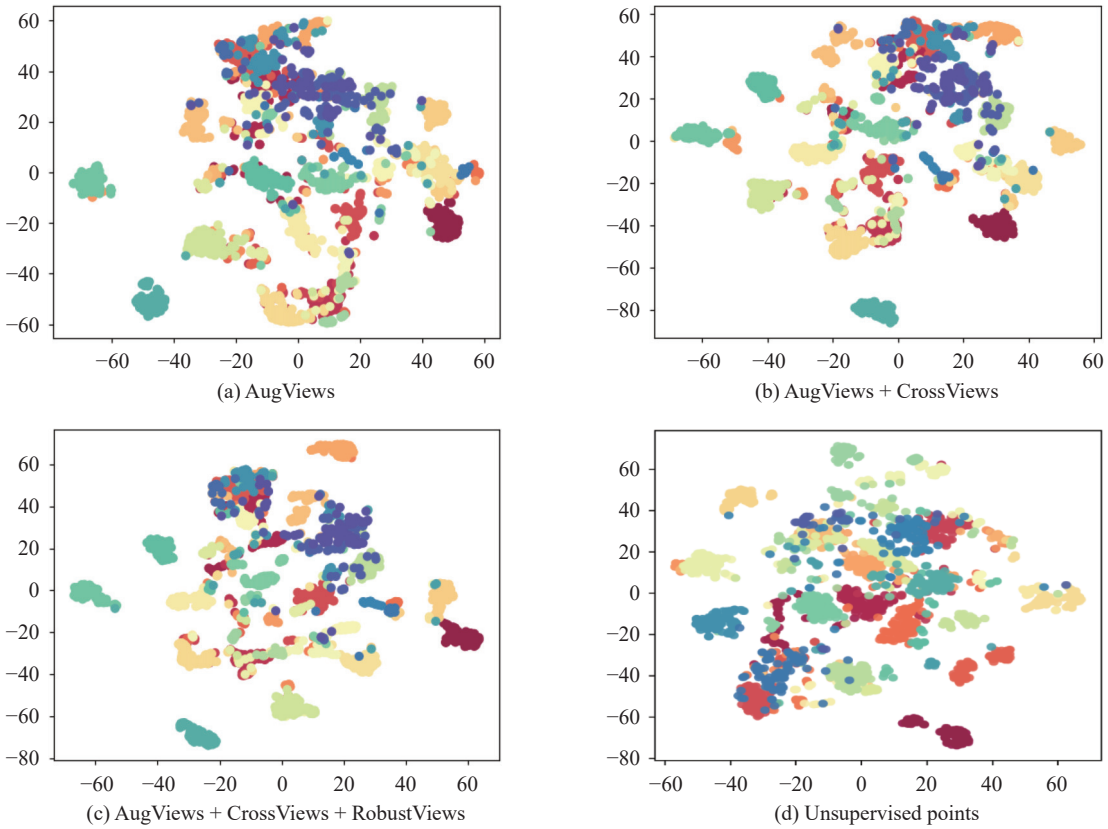

Fig. 4 Effect of tasks of the unsupervised learned features on the test set of ModelNet40 (visualized using t-SNE) and a comparison with the state-of-art unsupervised points method[48] (screenshot from the paper). With the introduction of objectives, the model extracts features that are more class-discriminative as well as domain-invariant.

guitar. In addition, it can be seen that the backbone trained without supervision is robust to the position. Although the body position of the guitar varies in the below line, it can still be accurately highlighted.

5 Conclusions

In this paper, we proposed a three-staged unsupervised end-to-end pretraining method for 3D object recognition. AugViews follows the same settings as SimCLR and treats each view individually to mine the intra-view knowledge. CrossViews approaches views of the same shape as positive samples and tries to exploit the intra-shape knowledge. RobustViews is designed to compensate for the neglect of the global representation and take advantage of the intra-categories information. Subsequent experiments on multiple datasets also demonstrate the effectiveness of our three-stage unsupervised pretraining. It is worth mentioning that our experimental improvement on complex data sets is more significant, proving our model's robustness.

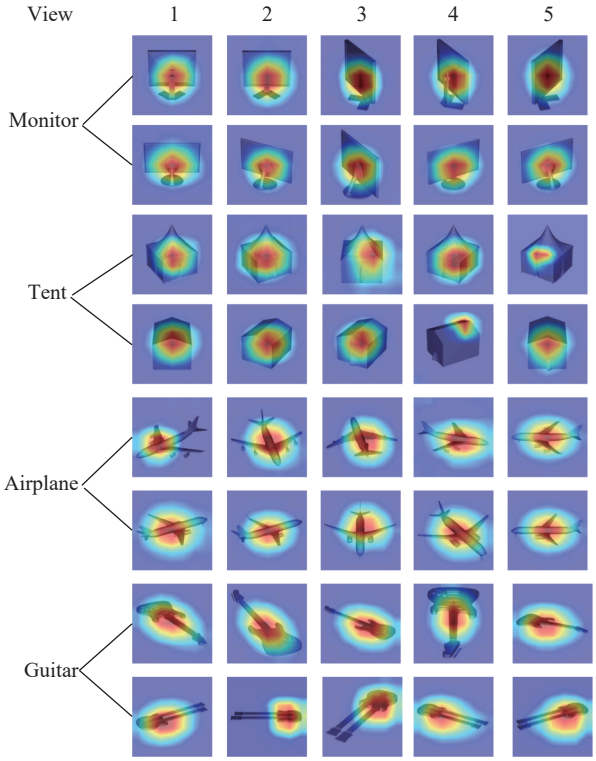

Fig. 5 Learned features visualized by GradCam

Acknowledegments

This work was supported in part by National Natural Science Foundation of China (No. 61976095) and the Science and Technology Planning Project of Guangdong Province, China (No. 2018B030323026).

Declarations of conflict of interest

The authors declared that they have no conflicts of interest to this work.

References

[1] K. Simonyan, A. Zisserman. Very deep convolutional networks for large-scale image recognition, [Online], Available: https://arxiv.org/abs/1409.1556, 2014.

[2] K. M. He, X. Y. Zhang, S. Q. Ren, J. Sun. Deep residual learning for image recognition. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, USA, pp. 770–778, 2016. DOI: 10.1109/CVPR.2016.90.

[3] K. M. He, R. Girshick, P. Dollár. Rethinking imageNet pre-training. In *Proceedings of IEEE/CVF International Conference on Computer Vision*, IEEE, Seoul, Republic of Korea, pp. 4917–4926, 2019. DOI: 10.1109/ICCV.2019.00502.

[4] T. Chen, S. Kornblith, M. Norouzi, G. Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning*, Article number 149, 2020.

[5] I. Misra, L. van der Maaten. Self-supervised learning of pretext-invariant representations. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Seattle, USA, pp. 6706–6716, 2020. DOI: 10.1109/CVPR42600.2020.00674.

[6] H. Su, S. Maji, E. Kalogerakis, E. Learned-Miller. Multi-view convolutional neural networks for 3D shape recognition. In *Proceedings of IEEE International Conference on Computer Vision*, Santiago, Chile, pp. 945–953, 2015. DOI: 10.1109/ICCV.2015.114.

[7] D. Pathak, P. Krähenbühl, J. Donahue, T. Darrell, A. A. Efros. Context encoders: Feature learning by inpainting. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, USA, pp. 2536–2544, 2016. DOI: 10.1109/CVPR.2016.278.

[8] R. Qian, T. J. Meng, B. Q. Gong, M. H. Yang, H. S. Wang, S. Belongie, Y. Cui. Spatiotemporal contrastive video representation learning. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Nashville, USA, pp. 6960–6970, 2021. DOI: 10.1109/CVPR46437.2021.00689.

[9] R. Zhang, P. Isola, A. A. Efros. Colorful image colorization. In *Proceedings of the 14th European Conference on Computer Vision*, Springer, Amsterdam, The Netherlands, pp. 649–666, 2016. DOI: 10.1007/978-3-319-46487-9_40.

[10] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, W. Z. Shi. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, USA, pp. 105–114, 2017. DOI: 10.1109/CVPR.2017.19.

[11] J. Y. Liang, J. Z. Cao, G. L. Sun, K. Zhang, L. Van Gool, R. Timofte. SwinIR: Image restoration using swin transformer. In *Proceedings of IEEE/CVF International Conference on Computer Vision Workshops*, IEEE, Montreal, Canada, pp. 1833–1844, 2021. DOI: 10.1109/ICCVW54120.2021.00210.

[12] R. R. Zhang, Z. Y. Guo, W. Zhang, K. C. Li, X. P. Miao, B. Cui, Y. Qiao, P. Gao, H. S. Li. PointCLIP: Point cloud understanding by clip. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, New Orleans, USA, pp. 8542–8552, 2022. DOI: 10.1109/CVPR52688.2022.00836.

[13] T. Y. Huang, B. W. Dong, Y. H. Yang, X. S. Huang, R. W. H. Lau, W. L. Ouyang, W. M. Zuo. CLIP2Point: Transfer CLIP to point cloud classification with image-depth pre-training, [Online], Available: https://arxiv.org/abs/2210.01055, 2022.

[14] K. M. He, H. Q. Fan, Y. X. Wu, S. N. Xie, R. Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Seattle, USA, pp. 9726–9735, 2020. DOI: 10.1109/CVPR42600.2020.00975.

[15] X. L. Chen, H. Q. Fan, R. Girshick, K. M. He. Improved baselines with momentum contrastive learning, [Online], Available: https://arxiv.org/abs/2003.04297, 2020.

[16] J. B. Grill, F. Strub, F. Altché, C. Tallec, P. H. Richemond, E. Buchatskaya, C. Doersch, B. A. Pires, Z. D. Guo, M. G. Azar, B. Piot, K. Kavukcuoglu, R. Munos, M. Valko. Bootstrap your own latent a new approach to self-supervised learning. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, Vancouver, Canada, Article number 1786, 2020.

[17] Z. Zhang, L. Liu, F. M. Shen, H. T. Shen, L. Shao. Binary multi-view clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 7, pp. 1774–1782, 2019. DOI: 10.1109/TPAMI.2018.2847335.

[18] T. Yu, J. J. Meng, J. S. Yuan. Multi-view harmonized bilinear network for 3D object recognition. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Salt Lake City, USA, pp. 186–194, 2018. DOI: 10.1109/CVPR.2018.00027.

[19] X. W. He, T. T. Huang, S. Bai, X. Bai. View N-gram network for 3d object retrieval. In *Proceedings of IEEE/CVF International Conference on Computer Vision*, IEEE, Seoul, Republic of Korea, pp. 7514–7523, 2019. DOI: 10.1109/ICCV.2019.00761.

[20] Y. Xu, C. D. Zheng, R. T. Xu, Y. H. Quan, H. B. Ling. Multi-view 3D shape recognition via correspondence-aware deep learning. *IEEE Transactions on Image Processing*, vol. 30, pp. 5299–5312, 2021. DOI: 10.1109/TIP.2021.3082310.

[21] S. S. Mohammadi, Y. M. Wang, A. Del Bue. Pointview-GCN: 3D shape classification with multi-view point clouds. In *Proceedings of IEEE International Conference on Image Processing*, Anchorage, USA, pp. 3103–3107, 2021. DOI: 10.1109/ICIP42928.2021.9506426.

[22] Z. Z. Han, X. Y. Wang, C. M. Vong, Y. S. Liu, M. Zwicker, C. L. P. Chen. 3Dviewgraph: Learning global features for 3D shapes from a graph of unordered views with attention. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, Macao, China, pp. 758–765, 2019.

[23] X. Wei, R. X. Yu, J. Sun. View-GCN: View-based graph convolutional network for 3D shape analysis. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Seattle, USA, pp. 1847–1856, 2020. DOI: 10.1109/CVPR42600.2020.00192.

[24] A. Hamdi, S. Giancola, B. Ghanem. MVTN: Multi-view transformation network for 3D shape recognition. In *Proceedings of IEEE/CVF International Conference on Computer Vision*, IEEE, Montreal, Canada, pp. 1–11, 2021. DOI: 10.1109/ICCV48922.2021.00007.

[25] R. Girdhar, D. F. Fouhey, M. Rodriguez, A. Gupta. Learning a predictable and generative vector representation for objects. In *Proceedings of the 14th European Conference on Computer Vision*, Springer, Amsterdam, The Netherlands, pp. 484–499, 2016. DOI: 10.1007/978-3-319-46466-4_29.

[26] A. Sharma, O. Grau, M. Fritz. VConv-DAE: Deep volumetric shape learning without object labels. In *Proceedings of European Conference on Computer Vision*, Springer, Amsterdam, The Netherlands, pp. 236–250, 2016. 10.1007/978-3-319-49409-8_20.

[27] J. J. Wu, C. K. Zhang, T. F. Xue, W. T. Freeman, J. B. Tenenbaum. Learning a probabilistic latent space of object shapes via 3D generative-adversarial modeling. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, Barcelona, Spain, pp. 82–90, 2016.

[28] S. K. Liu, L. Giles, A. Ororbia. Learning a hierarchical latent-variable model of 3D shapes. In *Proceedings of International Conference on 3D Vision*, IEEE, Verona, Italy, pp. 542–551, 2018. DOI: 10.1109/3DV.2018.00068.

[29] P. Achlioptas, O. Diamanti, I. Mitliagkas, L. J. Guibas. Learning representations and generative models for 3D point clouds. In *Proceedings of the 35th International Conference on Machine Learning*, Stockholm, Sweden, pp. 40–49, 2018.

[30] Y. Q. Yang, C. Feng, Y. R. Shen, D. Tian. FoldingNet: Point cloud auto-encoder via deep grid deformation. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Salt Lake City, USA, pp. 206–215, 2018. DOI: 10.1109/CVPR.2018.00029.

[31] Y. H. Zhao, T. Birdal, H. W. Deng, F. Tombari. 3D point capsule networks. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Long Beach, USA, pp. 1009–1018, 2019. DOI: 10.1109/CVPR.2019.00110.

[32] Z. Z. Han, X. Y. Wang, Y. S. Liu, M. Zwicker. Multi-angle point cloud-VAE: Unsupervised feature learning for 3D point clouds from multiple angles by joint self-reconstruction and half-to-half prediction. In *Proceedings of IEEE/CVF International Conference on Computer Vision*, IEEE, Seoul, Republic of Korea, pp. 10441–10450, 2019. DOI: 10.1109/ICCV.2019.01054.

[33] Z. Z. Han, M. Y. Shang, Y. S. Liu, M. Zwicker. View inter-prediction GAN: Unsupervised representation learning for 3D shapes by learning global shape memories to support local view predictions. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence, the 31st Innovative Applications of Artificial Intelligence Conference and the 9th AAAI Symposium on Educational Advances in Artificial Intelligence*, Honolulu, USA, Article number 1027, 2019.

[34] Y. F. Feng, Z. Z. Zhang, X. B. Zhao, R. R. Ji, Y. Gao. GVCNN: Group-view convolutional neural networks for 3D shape recognition. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Salt Lake City, USA, pp. 264–272, 2018. DOI: 10.1109/CVPR.2018.00035.

[35] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, L. Fei-Fei. ImageNet: A large-scale hierarchical image database. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Miami, USA, pp. 248-255, 2009. DOI: 10.1109/CVPR.2009.5206848.

[36] H. X. You, Y. F. Feng, X. B. Zhao, C. Q. Zou, R. R. Ji, Y. Gao. PVRNet: Point-view relation neural network for 3D shape recognition. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence and 31st Innovative Applications of Artificial Intelligence Conference and the 9th AAAI Symposium on Educational Advances in Artificial Intelligence*, Honolulu, USA, Article number 1119, 2019. DOI: 10.1609/aaai.v33i01.33019119.

[37] M. Kazhdan, T. Funkhouser, S. Rusinkiewicz. Rotation invariant spherical harmonic representation of 3D shape descriptors. In *Proceedings of Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*, Aachen, Germany, pp. 156–164, 2003.

[38] Z. R. Wu, S. R. Song, A. Khosla, F. Yu, L. G. Zhang, X. O. Tang, J. X. Xiao. 3D shapeNets: A deep representation for volumetric shapes. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Boston, USA, pp. 1912–1920, 2015. DOI: 10.1109/CVPR.2015.7298801.

[39] T. Furuya, R. Ohbuchi. Deep aggregation of local 3D geometric features for 3D model retrieval. In *Proceedings of British Machine Vision Conference*, York, UK, Article number 8, 2016.

[40] B. G. Shi, S. Bai, Z. C. Zhou, X. Bai. DeepPano: Deep panoramic representation for 3-D shape recognition. *IEEE Signal Processing Letters*, vol. 22, no. 12, pp. 2339–2343, 2015. DOI: 10.1109/LSP.2015.2480802.

[41] S. Bai, X. Bai, Z. C. Zhou, Z. X. Zhang, L. Jan Latecki. GIFT: A real-time and scalable 3D shape search engine. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, USA, pp. 5023–5032, 2016. DOI: 10.1109/CVPR.2016.543.

[42] K. Sfikas, T. Theoharis, I. Pratikakis. Exploiting the PANORAMA representation for convolutional neural network classification and retrieval. In *Proceedings of Workshop on 3D Object Retrieval*, Lyon, France, 2017. DOI: 10.2312/3dor.20171045.

[43] X. W. He, Y. Zhou, Z. C. Zhou, S. Bai, X. Bai. Triplet-center loss for multi-view 3D object retrieval. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Salt Lake City, USA, pp. 1945–1954, 2018. DOI: 10.1109/CVPR.2018.00208.

[44] M. Savva, F. Yu, H. Su, A. Kanezaki, T. Furuya, R. Ohbuchi, Z. C. Zhou, R. Yu, S. Bai, X. Bai, M. Aono, A. Tatsuma, S. Thermos, A. Axenopoulos, G. T. Papadopoulos, P. Daras, X. Deng, Z. H. Lian, B. Li, H. Johan, Y. J. Lu, S. Mk. Large-scale 3D shape retrieval from shapeNet core55. In *Proceedings of Workshop on 3D Object Retrieval*, Lyon, France, pp. 39–50, 2017. DOI: 10.2312/3dor.20171050.

[45] B. Li, H. Johan. 3D model retrieval using hybrid features and class information. *Multimedia Tools and Applications*, vol. 62, no. 3, pp. 821–846, 2013. DOI: 10.1007/s11042-011-0873-3.

[46] D. Robben, J. Bertels, S. Willems, D. Vandermeulen, F. Maes, P. Suetens. DeepVoxNet: Voxel-Wise Prediction for 3D Images, Report No. KUL/ESAT/PSI/1801, 2018.

[47] A. Kanezaki, Y. Matsushita, Y. Nishida. RotationNet: Joint object categorization and pose estimation using multiviews from unsupervised viewpoints. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Salt Lake City, USA, pp. 5010–5019, 2018. DOI: 10.1109/CVPR.2018.00526.

[48] Y. M. Rao, J. W. Lu, J. Zhou. Global-local bidirectional reasoning for unsupervised representation learning of 3D point clouds. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Seattle, USA, pp. 5375–5384, 2020. DOI: 10.1109/CVPR42600.2020.00542.

Luequan Wang received the B. Sc. degree in automation from South China University of Technology, China in 2020. He is a master student in automation science and engineering at South China University of Technology, China.

His research interests include self-supervised learning, 3D vision and deep learning.

E-mail: 875713197@qq.com
ORCID iD: 0000-0001-9320-6873

Hongbin Xu received the M. Sc. degree from South China University of Technology, China in 2021. He is currently a Ph. D. degree candidate in automation science and engineering at South China University of Technology (SCUT), China.

His research interests include 3D vision, multi-view stereo and self-supervised learning.

E-mail: hongbinxu1013@gmail.com
ORCID iD: 0000-0002-3455-1527

Wenxiong Kang received the M. Sc. degree from Northwestern Polytechnical University, China in 2003, and the Ph. D. degree in automation science and engineering from South China University of Technology, China in 2009. He is currently a professor with School of Automation Science and Engineering, South China University of Technology, China.

His research interests include biometrics identification, image processing, pattern recognition and computer vision.

E-mail: auwxkang@scut.edu.cn (Corresponding author)
ORCID iD: 0000-0001-9023-7252