Hierarchical View Predictor: Unsupervised 3D Global Feature Learning through Hierarchical Prediction among Unordered Views

Zhizhong Han
h312h@wayne.edu
Tsinghua University, Beijing, China and Wayne State University
Detroit, Michigan, USA

Xiyang Wang
ssdutwxy@gmail.com
School of Software, BNRist, Tsinghua University
Beijing, P. R. China

Yu-Shen Liu*
liuyushen@tsinghua.edu.cn
School of Software, BNRist, Tsinghua University
Beijing, P. R. China

Matthias Zwicker
zwicker@cs.umd.edu
University of Maryland
College Park, Maryland, USA

ABSTRACT

Unsupervised learning of global features for 3D shape analysis is an important research challenge because it avoids manual effort for supervised information collection. In this paper, we propose a view-based deep learning model called *Hierarchical View Predictor* (HVP) to learn 3D shape features from unordered views in an unsupervised manner. To mine highly discriminative information from unordered views, HVP performs a novel hierarchical view prediction over a view pair, and aggregates the knowledge learned from the predictions in all view pairs into a global feature. In a view pair, we pose hierarchical view prediction as the task of hierarchically predicting a set of image patches in a current view from its complementary set of patches, and in addition, completing the current view and its opposite from any one of the two sets of patches. Hierarchical prediction, in patches to patches, patches to view and view to view, facilitates HVP to effectively learn the structure of 3D shapes from the correlation between patches in the same view and the correlation between a pair of complementary views. In addition, the employed implicit aggregation over all view pairs enables HVP to learn global features from unordered views. Our results show that HVP can outperform state-of-the-art methods under large-scale 3D shape benchmarks in shape classification and retrieval.

*Corresponding author. This work was supported by National Key R&D Program of China (2020YFF0304100), the National Natural Science Foundation of China (62072268), and in part by Tsinghua-Kuaishou Institute of Future Media Data, and NSF (award 1813583).

MM '21, October 20–24, 2021, Virtual Event, China.

ACM ISBN 978-1-4503-8651-7/21/10…$15.00
https://doi.org/10.1145/3474085.3475172

CCS CONCEPTS

• **Computing methodologies** → **Artificial intelligence**.

KEYWORDS

3D feature learning, Unsupervised learning, 3D shape classification, Multiple views, CNN, RNN

ACM Reference Format:
Zhizhong Han, Xiyang Wang, Yu-Shen Liu, and Matthias Zwicker. 2021. Hierarchical View Predictor: Unsupervised 3D Global Feature Learning through Hierarchical Prediction among Unordered Views. In *Proceedings of the 29th ACM Int'l Conference on Multimedia (MM '21), Oct. 20–24, 2021, Virtual Event, China.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3474085.3475172

1 INTRODUCTION

Learning discriminative global features is important for 3D shape analysis tasks such as classification [1, 16–19, 23, 58, 71, 77], retrieval [1, 16–19, 58, 71, 77], correspondence [16–19, 31], segmentation [48, 51, 67], and reconstruction [4, 14, 21, 22, 30, 33, 43, 65, 68, 69, 73]. With supervised information [31, 40, 48, 51], recent deep learning based methods have achieved remarkable results under large-scale 3D benchmarks. However, intense manual labeling effort is required to obtain supervised information. In contrast, unsupervised 3D feature learning offers a more promising research challenge that avoids the manual labeling effort.

Several studies have addressed this challenge recently [1, 6, 13, 16–19, 23, 53, 58, 71, 76, 77] by extracting "supervised" information in an unsupervised scenario for the training of deep learning models. Extracting self-supervised information is usually achieved by posing different prediction tasks, such as the prediction of a shape from itself by minimizing reconstruction error [1, 17, 58, 71, 77] or embedded energy [16, 18], the prediction of a 3D shape from its context given by 2D views of the shape [6, 23, 76] or local shape features [19], or the prediction of a shape from views and itself together [13, 53]. Among all these methods, multiple sequential views are usually employed to provide a holistic

context of 3D shapes, however, unordered views can still not be leveraged to learn.

In this paper, we propose a novel model for 3D shape feature learning using a self-supervised view-based prediction task, which is formulated using unordered views and not restricted to sequential views. As we demonstrate in our results, this leads to highly discriminative 3D shape features and state-of-the-art performance in 3D shape analysis tasks such as classification and retrieval.

A key idea of our deep learning model, called the *Hierarchical View Predictor* (HVP), is that it operates on pairs of opposing views of each shape. HVP learns to hierarchically make local view predictions in each view pair, i.e., patches to patches, patches to view, and view to view, and then aggregates the knowledge learned from the predictions in all view pairs into global features. Specifically, unordered views are taken around a 3D shape on a sphere, where a current view and its opposite view form a view pair. Splitting unordered views into multiple view pairs enables HVP to handle the lack of order among views. In each view pair, given a set of patches of the current view, HVP performs hierarchical view prediction by first predicting the complementary set of patches in the current view, and then completing the whole current view and its opposite from any one of the two sets of patches. Hierarchical view prediction aims to learn the structure of 3D shapes from the correlation between patches in the same view and the correlation between appearances in the pair of complementary views. In addition, HVP employs an effective aggregation technique that implicitly aggregates the knowledge learned in the hierarchical view predictions of all pairs of opposing views. In summary, our significant contributions are as follows:

- We propose HVP as a novel deep learning model to perform unsupervised 3D global feature learning through hierarchical view prediction, which leads to state-of-the-art results in classification and retrieval.
- Hierarchical view prediction enables predictions among unordered views or ordered views, which facilitates HVP to comprehensively understand a 3D shape by hierarchically capturing the correlation between parts in the same view and the correlation between appearances in the pair of complementary views.
- With simultaneously mining "supervised" information inside a view and between two views, HVP eliminates the requirements of learning from dense neighboring view set, which enables HVP to achieve high performance under sparse neighboring view set.

2 RELATED WORK

Supervised 3D feature learning. With class labels, various deep learning models have been proposed to learn 3D features by capturing the distribution patterns among voxels [50, 64, 72], meshes [19], points clouds [39, 40, 48, 51, 66] and views [2, 15, 20, 25, 27, 31, 34, 35, 41, 55, 57, 59–61, 63]. Among these methods, multi-view based methods perform the best, where pooling is widely used for view aggregation.

Unsupervised 3D feature learning. To mine "supervised" information in unsupervised scenario, deep learning based methods adopted different prediction strategies, such as the prediction of a shape from itself by minimizing reconstruction error [1, 17, 58, 71, 77] or embedded energy [16, 18], the prediction of a shape from context [6, 19, 76], or the prediction of a shape from context and itself together [13, 53]. These methods employ different kinds of 3D raw representations, such as voxels [13, 17, 53, 58, 71, 76], meshes [16, 18, 19] or point clouds [1, 8, 11, 26, 28, 42, 47, 54, 77, 78], and accordingly, different kinds of context, such as spatial context of virtual words [19] or views [6, 12, 13, 23, 53, 76], are employed. Different from these methods, HVP employs a novel hierarchical view prediction among unordered views to mine more and finer "supervised" information.

View synthesis. Early works teach deep learning models to predict novel views according to input views and transformation parameters [7]. To generate views with more detail and less geometric distortions, external image sets [9] or geometric constraints [32, 80] are further employed. Similarly, the information of multiple past frames is aggregated in video prediction [70]. However, these methods cannot aggregate the knowledge learned in each prediction for the discriminability of global features.

3 HIERARCHICAL VIEW PREDICTOR

The rationale of hierarchical view prediction is to mimic human perception and understanding of 3D shapes. If a human knows a 3D shape, based on observing one part of a view, they can easily imagine the other part of the view, the full view, and even the opposite view. Therefore, HVP mimics this perception by hierarchical view prediction covering patches to patches, patches to view, and view to view, to learn the structure of a 3D shape from the correlation between parts in the same view and the correlation between appearances in the pair of complementary views.

Rather than learning from context by predicting a single patch using CNN in unsupervised image feature learning method [46], HVP employs an RNN based architecture to predict patch sequence. Since there is only one freely rotated object with the rest of empty background in a rendered view, the context which can be leveraged to learn is much less than in a natural image which contains multiple up-oriented objects. Thus, we want to capture more spatial relationship among parts in a view to remedy the scarce context.

Overview. The framework of HVP is illustrated in Fig. 1. By hierarchical view prediction, HVP aims to learn global feature $\boldsymbol{F}$ of a 3D shape m from V unordered views v_i ($i \in [1, V]$) taken around m on a sphere. We place the cameras at the 20 vertices of a regular dodecahedron, such that the $V = 20$ views are uniformly distributed. The F dimensional feature vector $\boldsymbol{F}$ is learned for each shape similarly as in [23] via gradient descent together with training the other parameters in HVP, starting from random initialization.

For each shape m in Fig. 1(a), each view v_i and its opposite view v_i' form a view pair $\boldsymbol{s}_i$ in Fig. 1(b). In $\boldsymbol{s}_i$, v_i and v_i' are

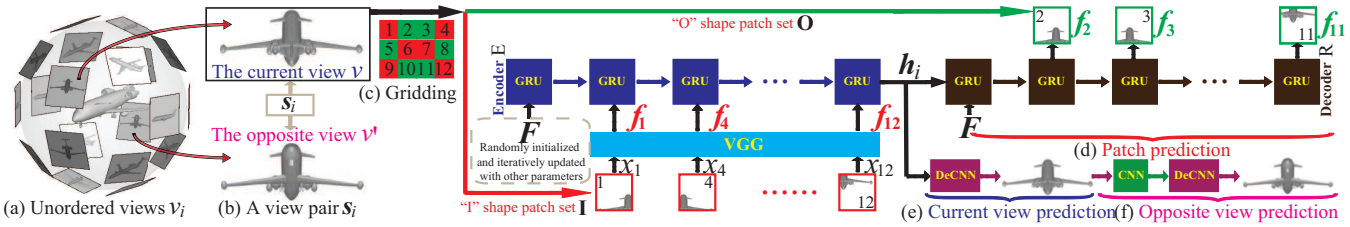


Figure 1: The framework of HVP is demonstrated by learning from a view pair of an plane from (a) to (f).

respectively denoted as v and v' for short. Using a 3×4 grid, we divide the current view into 12 overlapping patches x_j, which are further split into two subsets, an "O" like shape patch set **O** and an "I" like shape patch set **I**, as shown in Fig. 1(c). Each patch is a square of size $H \times H$.

In a pair $\boldsymbol{s}_i$, hierarchical view prediction consists of three prediction tasks in different spaces. Given either one of the two patch sets, HVP first predicts the other patch set in a feature space computed using a VGG19 network as shown in Fig. 1(d). Then, the current view is predicted in pixel space in Fig. 1(e). Finally, the opposite view is further predicted based on the predicted current view in Fig. 1(f).

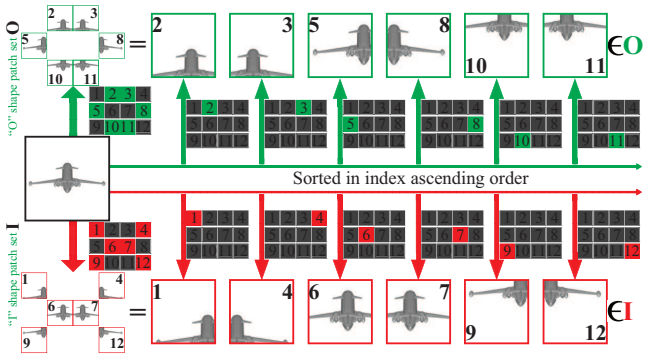


Figure 2: The gridding procedure splits a view into two patch sets O and I.

Gridding. In a view pair $\boldsymbol{s}_i$, the 224×224 current view v is divided into 12 overlapping patches x_j of size $H \times H$, as demonstrated in Fig. 2. The overlapping patches x_j are obtained by uniformly moving a $H \times H$ window across v in vertical and horizontal direction, as demonstrated in Fig. 3. This 3×4 gridding indexes x_j from 1 to 12 in order from left to right and top to bottom. We split the set of 12 patches x_j into an "O" like set **O** and an "I" like set **I** of six patches each, i.e., patches $[2, 3, 5, 8, 10, 11]$ belong **O** while the rest patches $[1, 4, 6, 7, 9, 12]$ belong **I**, where the patches in **O** or **I** are sorted into a patch sequence in index ascending order, respectively, as illustrated in Fig. 2. We formulate patch prediction as a bidirectional task. That is, both the prediction of **O** from **I** and the prediction of **I** from **O** can be conducted using the same network structure (we will take the former for example in the following description).

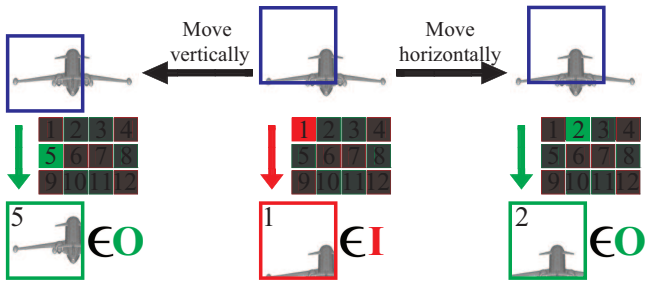


Figure 3: The gridding procedure is demonstrated by moving a blue window on a current view in vertical and horizontal direction.

Our strategy to define and split the patch set is motivated as follows: First, we want to split the patches of the current view into two equally sized subsets to facilitate bidirectional prediction using the same symmetrical structure in HVP. Second, the two patch sets should be distributed over the current view in a similar manner, which eliminates the bias of one directional prediction over the other. Third, each subset should contain a small number of patches to avoid redundancy and increasing computational cost.

Patch prediction. Patch prediction is performed to capture the correlation between parts in the same view. It predicts the feature $\boldsymbol{f}_j$ of patch x_j in set **O** (or **I**) from the features of patches in set **I** (or **O**). According to the indexes established in gridding, we sort all patches in either **O** or **I** into a sequence in ascending order. This leads us to use a seq2seq model to implement patch prediction as shown in Fig. 1(d).

We first extract a 4096 dimensional feature $\boldsymbol{f}_j$ of each patch x_j by the last fully connected layer of a VGG19 pre-trained under ImageNet. Then, we use an encoder RNN E to encode all the $\boldsymbol{f}_j$ in **I** (denoted as $\boldsymbol{f}_\mathbf{I}$) with their spatial relationship. We provide the global feature $\boldsymbol{F}$ of shape m, our learning target, at the first step of the encoder E. A key characteristic of our approach is that $\boldsymbol{F}$ is shared among all view prediction tasks for each shape m. Hence it serves as a knowledge container that keeps incorporating the knowledge derived from each hierarchical view prediction performed on m. Different from pooling, which is widely used as an explicit view aggregation, our implicit aggregation enables HVP to mine more and finer information from views of m. E encodes $\boldsymbol{f}_\mathbf{I}$ as a 4096 dimensional hidden state $\boldsymbol{h}_i = \mathrm{E}(\boldsymbol{f}_\mathbf{I})$ of the last step. Finally, based on $\boldsymbol{h}_i$, a decoder RNN R is

employ to predict the features of patches in $\mathbf{O}$ (denoted as $\boldsymbol{f}_{\mathbf{O}}$) with their spatial relationship. Similar to the encoder E, we provide the global feature $\boldsymbol{F}$ at the first step of decoder R, which is regarded as a reference for the following patch feature predictions. For each view, we measure the patch prediction performance of HVP using L-2 loss in feature space, denoted as loss L_{R},

$$L_{\mathrm{R}} = \|\mathrm{R}(\mathrm{E}(\boldsymbol{f}_{\mathbf{I}})) - \boldsymbol{f}_{\mathbf{O}}\|_2^2 + \|\mathrm{R}(\mathrm{E}(\boldsymbol{f}_{\mathbf{O}})) - \boldsymbol{f}_{\mathbf{I}}\|_2^2. \quad (1)$$

Current view prediction. Similar to patch prediction, current view prediction also aims to capture the correlation between parts in the same view, but in pixel space, which forces HVP to understand a 3D shape from the current view in different spaces. HVP predicts the full current view covering both patch sets $\mathbf{I}$ and $\mathbf{O}$ based on the encoding $\boldsymbol{h}_i$ of the patch features in set $\mathbf{I}$ or set $\mathbf{O}$. We employ a deconvolutional network U to predict the current view v in pixel space from $\boldsymbol{h}_i$, as shown in Fig. 1(e).

By reshaping the 4096 dimensional $\boldsymbol{h}_i$ into 256 feature maps of size 4×4, the deconvolutional network U starts generating v with a resolution of 64×64 through two deconvolutional layers. The two deconvolutional layers employ 128 and 3 kernels, respectively, and each kernel has size 4×4 and a stride of 4, where an ReLu and a tanh are followed in each layer as nonlinear activation functions respectively. For each view, we utilize the L-2 loss between the predicted current view $\mathrm{U}(\mathrm{E}(\boldsymbol{f}_{\mathbf{I}})) = \tilde{v}$ (or $\mathrm{U}(\mathrm{E}(\boldsymbol{f}_{\mathbf{O}})) = \tilde{v}$) and the ground truth current view v to measure the current view prediction performance of HVP, denoted as loss L_{U},

$$L_{\mathrm{U}} = \|\mathrm{U}(\mathrm{E}(\boldsymbol{f}_{\mathbf{I}})) - v\|_2^2 + \|\mathrm{U}(\mathrm{E}(\boldsymbol{f}_{\mathbf{O}})) - v\|_2^2. \quad (2)$$

Opposite view prediction. We further perform opposite view prediction to capture the correlation between appearances in the pair of complementary views. This helps HVP to bridge one individual view to another and encode their relationship into the global feature. Based on the predicted current view $\tilde{v}$, HVP predicts the opposite view of current view in each view pair $\boldsymbol{s}_i$, which is the most challenging task in the three predictions in HVP. This is because the opposite view can only be correctly predicted if the predicted current view $\tilde{v}$ is very close to the ground truth current view v, and no other clue is available to use. This challenging criterion pushes HVP to comprehensively learn the intrinsic structure of a 3D shape.

We employ a convolutional network C and the same deconvolutional network U to implement the opposite view prediction, as shown in Fig. 1(f). The convolutional network C abstracts the 64×64 predicted current view $\tilde{v}$ into a 4096 dimensional feature through three convolutional blocks and one fully connected layer. Each of the three blocks includes 1, 2, and 3 convolutional layers, respectively, which is followed by a maxpool with size of 2×2. All convolutional layers in the three blocks employ kernels with size of 3×3 and a stride of 1, where the ReLu is used as the nolinear activation function. Then, the deconvolutional network U generates the predicted opposite views $\tilde{v}'$ based on the 4096 dimensional feature of $\tilde{v}$. For each view, we also utilize the L-2 loss between the predicted opposite view $\mathrm{U}(\mathrm{C}(\tilde{v})) = \tilde{v}'$ and the ground truth opposite view v' to measure the opposite view prediction performance of HVP, denoted as loss L'_{U},

$$L'_{\mathrm{U}} = \|\mathrm{U}(\mathrm{C}(\tilde{v})) - v'\|_2^2. \quad (3)$$

Objective function. Finally, in each view pair $\boldsymbol{s}_i$, HVP is trained to minimize all the aforementioned losses involved in the three prediction tasks. Therefore, we define the objective function of HVP by combining the three losses as in Eq. (4), where the weights α and β are used to control the balance among them,

$$L = \alpha L_{\mathrm{R}} + L_{\mathrm{U}} + \beta L'_{\mathrm{U}}. \quad (4)$$

Note that simultaneously with the other network parameters, we also optimize the learning target $\boldsymbol{F}$ by minimizing L. We use a standard gradient descent approach by iteratively updating $\boldsymbol{F}$ as Eq. (5), where ε is the learning rate,

$$\boldsymbol{F} \leftarrow \boldsymbol{F} - \varepsilon \times \partial L / \partial \boldsymbol{F}. \quad (5)$$

Testing modes. There are two typical modes of unsupervised learning of features $\boldsymbol{F}$ of 3D shapes for testing, which we call the known-test mode and the unknown-test mode. In known-test mode, the test shapes are given with the training shapes at the same time, such that the features of test shapes can be learned with the features of training shapes together. In unknown-test mode, HVP is first pretrained using the set of training shapes only. At test time, we then iteratively learn the features $\mathbf{F}$ of test shapes by minimizing Eq. (4) while fixing the other pretrained parameters of HVP.

4 EXPERIMENTAL RESULTS AND ANALYSIS

In this section, the performance of HVP is evaluated and analyzed. First we discuss the setup of parameters involved in HVP. These parameters are tuned to demonstrate how they affect the discriminability of learned features in shape classification under ModelNet10 [72]. Then, some ablation studies are presented to show the effectiveness of some important elements involved in HVP. Finally, HVP is compared with state-of-the-art methods in shape classification and retrieval under ModelNet10 [72] and ModelNet40 [72]. In addition, some generated current views and opposite views are also visualized to justify HVP better. Note that all classification is conducted by training a linear SVM under the global features learned by HVP.

Dataset and evaluations. The training and testing sets of ModelNet40 consist of 9,843 and 2,468 shapes, respectively. In addition, the training and testing sets of ModelNet10 consist of 3,991 and 908 shapes, respectively.

We employ both average instance accuracy (InsACC) and average class accuracy (ClaACC) to evaluate the classification results. Moreover, we use mAP and precision and recall (PR) curves as metrics in shape retrieval.

Parameter setup. Initially, the dimension F of global feature $\boldsymbol{F}$ is 4096 which is the same as the dimension of $\boldsymbol{f}_j$, and the $V = 20$ views of all 3D shapes under ModelNet10 are employed to train HVP in known-test mode with a learning

Table 1: The effect of $\varepsilon(\times 0.0001)$ under ModelNet10. $\alpha = 1$, $\beta = 1$, $H = 128$.

ε	2	3	4	5	6	7
InsACC	91.74	92.29	92.51	**93.61**	92.73	92.84
ClaACC	91.48	92.05	92.15	**93.25**	92.35	92.45

Table 2: The effect of α under ModelNet10. $\beta = 1$, $H = 128$.

α	0.25	0.5	1	2	4	Avg
InsACC	92.18	92.84	**93.61**	92.18	92.51	92.66
ClaACC	91.85	92.75	**93.25**	91.85	92.11	92.36

Table 3: The effect of β under ModelNet10. $\alpha = 1$, $H = 128$.

β	0.25	0.5	1	2	4	Avg
InsACC	92.40	92.84	**93.61**	93.39	92.73	92.99
ClaACC	92.25	92.43	**93.25**	92.81	92.31	92.61

rate of $\varepsilon = 0.0002$. Both α and β are set to 1, which makes the initial values of loss L_{R}, L_{U} and L'_{U} comparable to each other, where a normal distribution with mean of 0 and standard deviation of 0.02 is used to initialize the parameters involved in HVP. In addition, the patch width H is 128, and both the prediction of **O** from **I** and the prediction of **I** from **O** are performed.

First, we conduct experiments to explore how the learning rate affects the performance of HVP, as shown in Table 1. We employ different learning rates ε with initial parameters mentioned in the former paragraph. Besides the initial setting, we iteratively use $\varepsilon = \{0.0002, 0.0003, 0.0004, 0.0005, 0.0006, 0.0007\}$ for training. We achieve the best instance accuracy of 93.61% with $\varepsilon = 0.0003$.

Next, we explore the balance weights α and β. We summarize the results in Table 2 and Table 3, which shows that the weights are important for the performance of HVP. With $\beta = 1$, we explore the effect of α by iteratively setting α to $\{0.25, 0.5, 1, 2, 4\}$ in Table 2. The instance accuracy increases to a best of 93.61% with $\alpha = 1$, and then, decreases gradually. We observe a similar phenomenon in the exploration of the effect of β in Table 3. With $\alpha = 1$ obtaining the best result in Table 2, we iteratively set β to $\{0.25, 0.5, 1, 2, 4\}$. The instance accuracy also achieves up to 93.61% with $\beta = 1$, and then, decreases a little bit. These results show that both under-fitted and over-fitted patch prediction and opposite view prediction would also affect the discriminability of learned features. In addition, in terms of row averaged accuracies in Table 2 and Table 3, HVP is affected more by patch prediction than by opposite view prediction, since the row averaged accuracies of α drop more than the ones of β from the same highest accuracies.

Finally, we explore how the patch size H affects the performance of HVP, as shown in Table 4. Beside the patch size $H = 128$ involved in the former experiments, we also iteratively employ patches with size of $\{160, 180, 200\}$. Based on a view with size of 224 in our experiments, the increasing patch size H makes the results achieve up to 94.16% with $H = 160$, while decreasing gradually for even larger patches. These results show that both a lack of semantic information in small patches and too much redundant information among big overlapping patches are not helpful to increase the discriminability of learned features. Since smaller sizes make the features of the patches meaningless, while big sizes provide too much redundant information, such that HVP can easily solve the patch prediction, without needing to store information in the global feature **F** that is shared among all predictions for each shape.

Table 4: The effect of patch size H under ModelNet10. $\alpha = 1$, $\beta = 1$.

Size	128	160	180	200
InsACC	93.61	**94.16**	93.94	93.73
ClaACC	93.25	**93.80**	93.60	93.36

Table 5: The contribution of each prediction involved in hierarchical view prediction under ModelNet10. $\alpha = 1$, $\beta = 1$,$H = 160$.

Loss	L_{U}	$L_{\mathrm{U}} + \beta L'_{\mathrm{U}}$	$L_{\mathrm{U}} + \alpha L_{\mathrm{R}}$	L
InsACC	86.78	88.66	90.31	**94.16**
ClaACC	86.41	87.80	90.10	**93.80**

Ablation studies. Based on the former experiments, we further explore the contribution of each prediction involved in hierarchical view prediction, as highlighted by results in Table 5. We first train HVP only using the current view prediction loss (L_{U}), then, we incrementally add opposite view prediction loss ($\beta L'_{\mathrm{U}}$) or patch prediction loss (αL_{R}), finally, we compare these results with our best results employing all the three losses ($L_{\mathrm{U}} + \beta L'_{\mathrm{U}} + \alpha L_{\mathrm{R}}$).

Compared to the results of "L_{U}", each incrementally added loss can improve the performance of HVP in terms of both averaged instance accuracy and averaged class accuracy. In addition, the patch prediction loss can help HVP improve more than the opposite view prediction loss. To better visualize the effect of these losses on HVP, we show the generated current view or generated opposite view involved in the experiments in Table 5. In Fig. 4, the generated views and their distances to the ground truth are shown. The added opposite view prediction loss $\beta L'_{\mathrm{U}}$ can degenerate the generated current view compared to the one with only L_{U} or the one with $L_{\mathrm{U}} + \alpha L_{\mathrm{R}}$. While the added patch prediction loss αL_{R} can improve the generated current view or the generated opposite view, as shown by the comparison between L_{U} and $L_{\mathrm{U}} + \alpha L_{\mathrm{R}}$ and the comparison between $L_{\mathrm{U}} + \alpha L_{\mathrm{R}}$ and $L_{\mathrm{U}} + \beta L'_{\mathrm{U}} + \alpha L_{\mathrm{R}}$.

Then, we highlight the effect of prediction direction in Table 6. In all the former experiments, we employ the bidirectional prediction This could provide more data to train HVP thanks to the symmetrical structure of HVP. In the

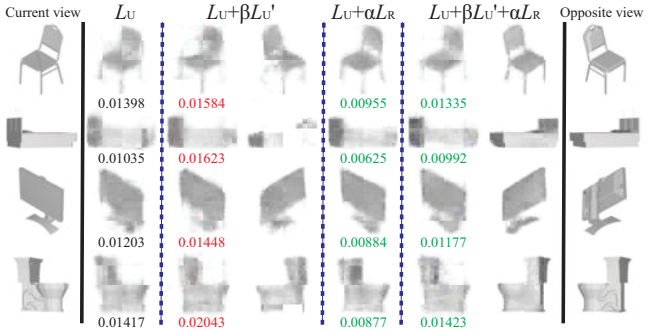


Figure 4: The effect of different losses on the generated current view and opposite view in a view pair under ModelNet10. The distance between each generated current view and ground truth is also shown under each generated current view, where the red color means the distance is larger than the distance in the first column in the same row while the green color means it becomes smaller.

Table 6: The effect of prediction direction under ModelNet10. $\alpha = 1$, $\beta = 1$, $H = 160$.

Direction	**O2I**	**I2O**	**O2I** or **I2O**	**O2I** and **I2O**
InsACC	93.83	93.61	93.72	**94.16**
ClaACC	93.38	93.16	93.25	**93.80**

following experiments, we train HVP using different kinds of single directional data respectively, such as from **O** to **I** (shown as **O2I**), from **I** to **O** (shown as **I2O**), and randomly selected single direction for hierarchical view prediction on each view (shown as **O2I** or **I2O**). As shown by these results, bidirectional prediction could learn better features than single direction prediction.

Finally, we justify our view aggregation method in Table 7. We first train HVP without $\boldsymbol{F}$. To obtain a global shape feature without $\boldsymbol{F}$, we use pooling on the features $\boldsymbol{h}_i$ for all view pairs $\boldsymbol{s}_i$ for each shape (recall that we use the $\boldsymbol{h}_i$ to solve the view prediction tasks as shown in Fig. 1). The results using mean and max pooling without $\boldsymbol{F}$ in the first column in Table 7 exhibit a large drop in performance compared to our approach. To better understand this, we perform a second experiment where we train HVP with $\boldsymbol{F}$ as described previously, but we again use the mean- or max-pooled features $\boldsymbol{h}_i$ as the global shape feature. The results in the second column in Table 7 (first and second row) show that using $\boldsymbol{F}$ improves the performance of the mean- and max-pooled features $\boldsymbol{h}_i$. However, our approach that uses $\boldsymbol{F}$ itself as the shape feature (third row) achieves even better performance, indicating that HVP's implicit pooling is superior to explicit schemes such as mean- or max-pooling. Intuitively, this is because max pooling can lose some information in each view pair, while mean pooling weights all view pairs equally. Hence

Table 7: The effect of view aggregation under ModelNet10. $\alpha = 1$, $\beta = 1$, $H = 160$.

	Without $\boldsymbol{F}$		With $\boldsymbol{F}$	
Methods	InsACC	ClaACC	InsACC	ClaACC
MeanPool	89.65	87.96	90.75	90.58
MaxPool	90.31	89.84	91.19	90.92
Our	-	-	**94.16**	**93.80**

Table 8: Classification comparison under MN10.

Methods	Supervised	Instance%	Class%
ORION[56]	Yes	-	93.80
3DDescriptorNet[74]	Yes	-	92.40
Pairwise[34]	Yes	-	92.80
GIFT[2]	Yes	-	91.50
VoxNet[44]	Yes	-	92.00
VRN[3]	Yes	93.80	-
PANORAMA[57]	Yes	-	91.12
LFD[5]	No	79.90	-
Vconv-DAE[58]	No	-	80.50
3DGAN[71]	No	-	91.00
VSL[38]	No	91.00	-
NSampler[52]	no	88.70	95.30
LGAN[1]	No	95.30	-
LGAN[1](MN10)	No	92.18	-
FNet[77]	No	94.40	-
FNet[77](MN10)	No	91.85	-
VIPGAN[23]	No	94.05	93.71
Our	No	**94.16**	**93.80**
Our(MN40)	No	**92.18**	**91.57**

it fails to give more weight to certain highly distinct view pairs.

Classification. We compare HVP with the state-of-the-art methods in classification under ModelNet10 and ModelNet40 in Table 8 and Table 9, respectively. The parameters under ModelNet40 are the same ones with our best results under ModelNet10 in Table 7. Under ModelNet10, HVP outperforms all its unsupervised competitors under ModelNet10, as shown by "Our", which is also the best result compared to eight top ranked supervised methods. Under ModelNet40, HVP achieves the state-of-the-art results among all the unsupervised and supervised competitors. For fair comparison, the result of VRN [3] is presented without ensemble learning. The result of RotationNet [35] is presented with views taken by the default camera system orientation that is identical to the others. Although the results of LGAN, FNet and NSampler are better than our results under ModelNet10, it is inconclusive whether they are better than ours. This is because these methods are trained under a version of ShapeNet55 that contains more than 57,000 3D shapes, including a number of 3D point clouds. However, there are only 51,679 3D shapes from ShapeNet55 that are available for public download. Therefore, we cannot use the same amount of training data to train HVP to compare with them. To perform fair

Table 9: Classification comparison under MN40.

Methods	Supervised	Instance%	Class%
MVCNN[49]	Yes	92.0	89.7
MVCNN-Sphere[49]	Yes	89.5	86.6
Pairwise[34]	Yes	-	90.70
GIFT[2]	Yes	-	89.50
PointNet++[51]	Yes	91.90	-
VRN[3]	Yes	91.33	-
RotationNet[35]	Yes	92.37	-
PANORAMA[57]	Yes	-	90.70
T-L Network[13]	No	-	74.40
Vconv-DAE[58]	No	-	75.50
3DGAN[71]	No	-	83.30
VSL[38]	No	84.50	-
LGAN[1]	No	85.70	-
LGAN[1](MN40)	No	87.27	-
FNet[77]	No	88.40	-
FNet[77](MN40)	No	84.36	-
NSampler[52]	no	88.70	-
MRTNet[10]	No	86.40	-
3DCapsule[79]	No	88.90	-
PointGrow[62]	No	85.80	-
PCGAN[37]	No	87.80	-
MAPVAE[26]	No	90.15	-
OrientNet[47]	No	90.75	-
VIPGAN[23]	No	91.98	-
VIPGAN(Two)[23]	No	4.05	2.50
Our	No	**90.72**	**87.95**
Our(Two)	No	**88.33**	**83.53**

comparison with "Our", we use the codes of LGAN and FNet to conduct experiments only using shapes in ModelNet, as shown by "LGAN()" and "FNet()", which employs the same training data as ours. Our performing results show that our method is superior to these methods.

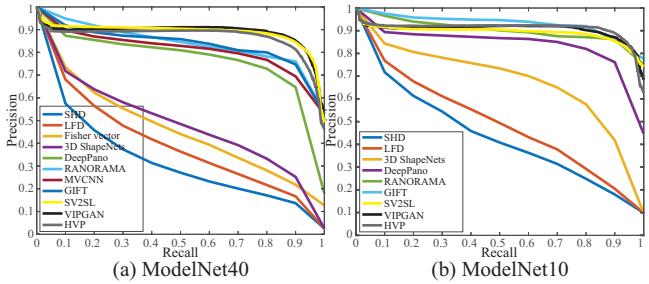
With the ability of mining correlation inside each view, HVP does not rely on dense neighboring views to learn. To justify this, we train HVP using two views of each 3D shape from ModelNet40. Although VIPGAN achieves the best results with 12 views under ModelNet40, HVP works much better than VIPGAN under sparse neighboring view set, as shown by the comparison between "Our(Two)" and "VIPGAN(Two)" in Table 9.

Moreover, we evaluate HVP in unknown-test mode by learning features of ModelNet10 using parameters pretrained under ModelNet40 ("Our" in Table 9). As shown by "Our(MN40)" in Table 8, HVP can still produce good results. These results show that HVP is with remarkable transfer learning ability based on comprehensive 3D shapes understanding, which is benefited by mining correlation inside a view and between complementary views.

Retrieval. We further evaluate HVP in shape retrieval under the ModelNet40 and ModelNet10 by comparing with the state-of-the-art methods in Table 10. These experiments are conducted under the test set, where each 3D shape is used as a query to retrieve from the rest of the shapes, and the retrieval performance is evaluated by mAP. In addition, we employ the same parameters with our best classification results in Table 9 and Table 8 to extract the global features for the retrieval experiments under ModelNet40 and ModelNet10, respectively. As shown in Table 10, our results outperform all the compared results under ModelNet10 and achieve state-of-the-art under ModelNet40. In addition, the available PR curves under ModelNet40 and ModelNet10 are also compared in Fig. 5, which also demonstrates our outperforming results in shape retrieval.

Table 10: The comparison of retrieval in terms of mAP under ModelNet40 and ModelNet10.

Methods	Range	MN40	MN10
SHD [36]	Test-Test	33.26	44.05
LFD [5]	Test-Test	40.91	49.82
3DShapeNets [72]	Test-Test	49.23	68.26
GeomImage [60]	Test-Test	51.30	74.90
DeepPano [59]	Test-Test	76.81	84.18
MVCNN [61]	Test-Test	79.50	-
PANORAMA [57]	Test-Test	83.45	87.39
GIFT [2]	Random	81.94	91.12
Triplet [29]	Test-Test	88.00	-
SliceVoxel [45]	Test-Test	77.48	85.34
SV2SL [24]	Test-Test	89.00	89.55
VIPGAN [23]	Test-Test	89.23	90.69
Serial [75]	Test-Test	87.05	-
Ours	Test-Test	**87.13**	**91.19**



(a) ModelNet40 (b) ModelNet10

Figure 5: The PR curve comparison under ModelNet40 and ModelNet10.

Visualization. For our classification results, we show the generated current views and the generated opposite views of all ten shape classes under ModelNet10, where two shapes are involved in each shape class, as shown in Fig. 6. These results show that HVP can generate plausible views based on the comprehensive understanding of 3D shapes. Another interesting observation is that the opposite view can be generated better than the current view. In addition, we show the confusion matrix of our classification results under ModelNet10 and ModelNet40 in Fig. 7 and Fig. 8, respectively. In each confusion matrix, an element in the diagonal line means the

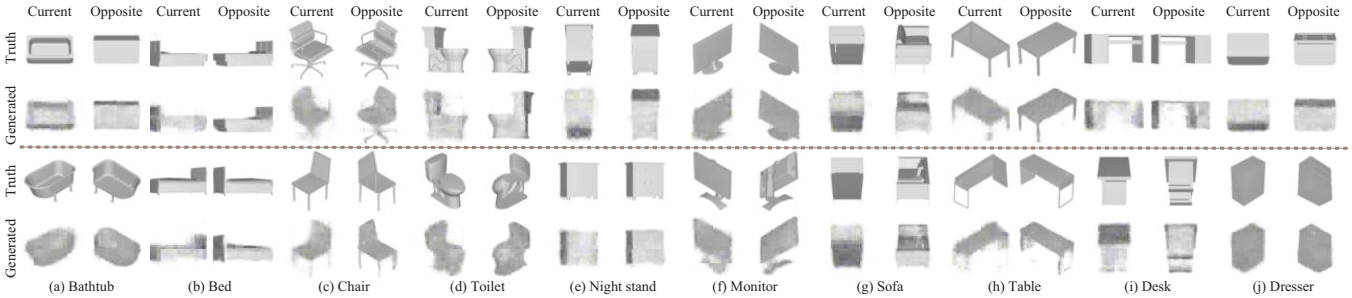


Figure 6: The generated current view and generated opposite view of a shape are visualized.

percentage of how many 3D shapes are correctly classified, while other elements in the same row means the percentage of 3D shapes wrongly classified into other shape classes. The large diagonal elements in each confusion matrix show that HVP is able to learn highly discriminative features for 3D shapes, which facilitates HVP to achieve high performance in classifying large-scale 3D shapes.

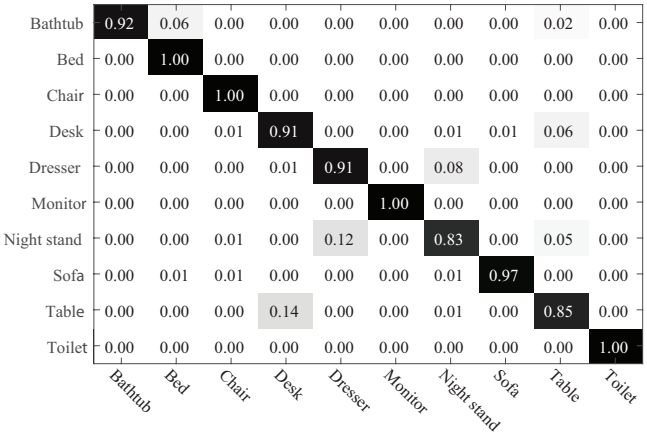


Figure 7: The confusion matrix of our results in 3D shape classification under ModelNet10.

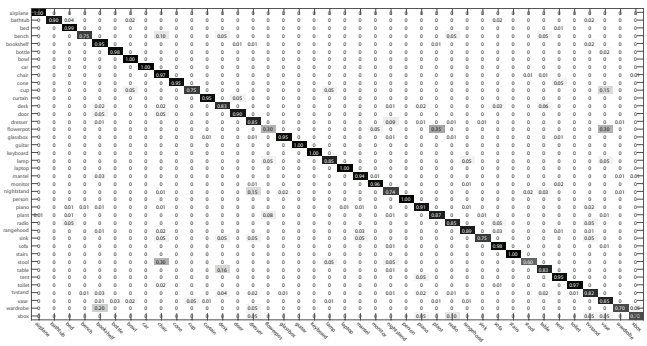


Figure 8: The confusion matrix of our results in 3D shape classification under ModelNet40.

For our retrieval results, we show the top 5 retrieved 3D shapes for some queries from ModelNet10 and ModelNet40 in Fig. 9. According to the distance between the query and each retrieved shape (shown under each retrieved shape), we find HVP is capable of learning features to distinguish 3D shapes in detail, which enables retrieving 3D shapes with similar structures.

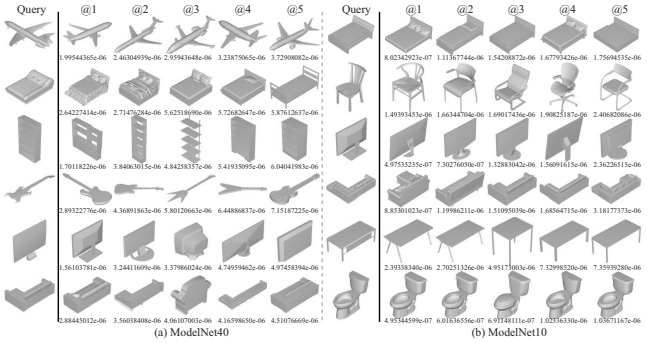


Figure 9: The top 5 retrieved 3D shapes for some queries from (a) ModelNet40 and (b) ModelNet10 are demonstrated. The distance between the query and each retrieved shape is also shown.

5 CONCLUSIONS

We proposed HVP for unsupervised 3D global feature learning from unordered views of 3D shapes. By implementing a novel hierarchical view prediction, HVP successfully mines highly discriminative information among unordered views in an unsupervised manner. Our results show that HVP effectively learns to hierarchically make patch predictions, current view prediction and opposite view prediction in each view pair, and then, comprehensively aggregates the knowledge learned from the predictions in all view pairs into global features. HVP can not only learn from both unordered and ordered view set, but also work well under sparse neighboring view sets, which eliminates the requirement of mining "supervised" information from dense neighboring views. Our results show that HVP outperforms its unsupervised counterparts, as well as some top ranked supervised methods under large-scale benchmarks in shape classification and retrieval.

REFERENCES

[1] Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas J. Guibas. 2018. Learning Representations and Generative Models for 3D Point Clouds. In *International Conference on Machine Learning*. 40–49.

[2] Song Bai, Xiang Bai, Zhichao Zhou, Zhaoxiang Zhang, and Longin Jan Latecki. 2017. GIFT: Towards Scalable 3D Shape Retrieval. *IEEE Transaction on Multimedia* 19, 6 (2017), 1257–1271.

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

[4] Chao Chen, Zhizhong Han, Yu-Shen Liu, and Matthias Zwicker. 2021. Unsupervised Learning of Fine Structure Generation for 3D Point Clouds by 2D Projections Matching. In *IEEE International Conference on Computer Vision*.

[5] Dingyun Chen, Xiaopei Tian, Yute Shen, and Ming Ouhyoung. 2003. On visual similarity based 3D model retrieval. *Computer Graphics Forum* 22, 3 (2003), 223–232.

[6] Christopher Bongsoo Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 2016. 3D-R2N2: A Unified Approach for Single and Multi-view 3D Object Reconstruction. In *European Conference on Computer Vision*. 628–644.

[7] Alexey Dosovitskiy, Jost Tobias Springenberg, Maxim Tatarchenko, and Thomas Brox. 2017. Learning to Generate Chairs, Tables and Cars with Convolutional Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39, 4 (2017), 692–705.

[8] Bi'an Du, Xiang Gao, Wei Hu, and Xin Li. 2021. Self-Contrastive Learning with Hard Negative Sampling for Self-supervised Point Cloud Learning. *CoRR* abs/2107.01886 (2021).

[9] John Flynn, Ivan Neulander, James Philbin, and Noah Snavely. 2016. DeepStereo: Learning to Predict New Views From the World's Imagery. In *The IEEE Conference on Computer Vision and Pattern Recognition*.

[10] Matheus Gadelha, Rui Wang, and Subhransu Maji. 2018. Multiresolution Tree Networks for 3D Point Cloud Processing. In *European Conference on Computer vision*.

[11] Xiang Gao, Wei Hu, and Guo-Jun Qi. 2020. GraphTER: Unsupervised Learning of Graph Transformation Equivariant Representations via Auto-Encoding Node-wise Transformations. In *IEEE Conference on Computer Vision and Pattern Recognition*.

[12] Xiang Gao, Wei Hu, and Guo-Jun Qi. 2021. Self-Supervised Multi-View Learning via Auto-Encoding 3D Transformations. *ArXiv* abs/2103.00787 (2021).

[13] Rohit Girdhar, David F. Fouhey, Mikel Rodriguez, and Abhinav Gupta. 2016. Learning a Predictable and Generative Vector Representation for Objects. In *Proceedings of European Conference on Computer Vision*. 484–499.

[14] Zhizhong Han, Chao Chen, Yu-Shen Liu, and Matthias Zwicker. 2020. DRWR: A Differentiable Renderer without Rendering for Unsupervised 3D Structure Learning from Silhouette Images. In *International Conference on Machine Learning*.

[15] Zhizhong Han, Xinhai Liu, Yu-Shen Liu, and Matthias Zwicker. 2019. Parts4Feature: Learning 3D Global Features from Generally Semantic Parts in Multiple Views. In *IJCAI*.

[16] Zhizhong Han, Zhenbao Liu, Junwei Han, Chi-Man Vong, Shuhui Bu, and C.L.Philip Chen. 2017. Mesh Convolutional Restricted Boltzmann Machines for Unsupervised Learning of Features With Structure Preservation on 3D Meshes. *IEEE Transactions on Neural Network and Learning Systems* 28, 10 (2017), 2268 – 2281.

[17] Zhizhong Han, Zhenbao Liu, Junwei Han, Chi-Man Vong, Shuhui Bu, and C.L.P. Chen. 2019. Unsupervised Learning of 3D Local Features from Raw Voxels Based on A Novel Permutation Voxelization Strategy. *IEEE Transactions on Cybernetics* 49, 2 (2019), 481–494.

[18] Zhizhong Han, Zhenbao Liu, Junwei Han, Chi-Man Vong, Shuhui Bu, and Xuelong Li. 2016. Unsupervised 3D Local Feature Learning by Circle Convolutional Restricted Boltzmann Machine. *IEEE Transactions on Image Processing* 25, 11 (2016), 5331–5344.

[19] Zhizhong Han, Zhenbao Liu, Chi-Man Vong, Yu-Shen Liu, Shuhui Bu, Junwei Han, and CL Philip Chen. 2018. Deep Spatiality: Unsupervised Learning of Spatially-Enhanced Global and Local 3D Features by Deep Neural Network with Coupled Softmax. *IEEE Transactions on Image Processing* 27, 6 (2018), 3049–3063.

[20] Zhizhong Han, Honglei Lu, Zhenbao Liu, Chi-Man Vong, Yu-Shen Liu, Matthias Zwicker, Junwei Han, and C.L. Philip Chen. 2019. 3D2SeqViews: Aggregating Sequential Views for 3D Global Feature Learning by CNN With Hierarchical Attention Aggregation. *IEEE Transactions on Image Processing* 28, 8 (2019), 3986–3999.

[21] Zhizhong Han, Baorui Ma, Yu-Shen Liu, and Matthias Zwicker. 2020. Reconstructing 3D Shapes from Multiple Sketches using Direct Shape Optimization. *IEEE Transactions on Image Processing* 29 (2020), 8721–8734.

[22] Zhizhong Han, Guanhui Qiao, Yu-Shen Liu, and Matthias Zwicker. 2020. SeqXY2SeqZ: Structure Learning for 3D Shapes by Sequentially Predicting 1D Occupancy Segments From 2D Coordinates. In *European Conference on Computer Vision*.

[23] Zhizhong Han, Mingyang Shang, Yu-Shen Liu, and Matthias Zwicker. 2019. View Inter-Prediction GAN: Unsupervised Representation Learning for 3D Shapes by Learning Global Shape Memories to Support Local View Predictions. In *AAAI*. 8376–8384.

[24] Zhizhong Han, Mingyang Shang, Zhenbao Liu, Chi-Man Vong, Yu-Shen Liu, Matthias Zwicker, Junwei Han, and C.L. Philip Chen. 2019. SeqViews2SeqLabels: Learning 3D Global Features via Aggregating Sequential Views by RNN With Attention. *IEEE Transactions on Image Processing* 28, 2 (2019), 685–672.

[25] Zhizhong Han, Mingyang Shang, Xiyang Wang, Yu-Shen Liu, and Matthias Zwicker. 2019. Y2Seq2Seq: Cross-Modal Representation Learning for 3D Shape and Text by Joint Reconstruction and Prediction of View and Word Sequences. In *AAAI*. 126–133.

[26] Zhizhong Han, Xiyang Wang, Yu-Shen Liu, and Matthias Zwicker. 2019. 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 *IEEE International Conference on Computer Vision*.

[27] Zhizhong Han, Xiyang Wang, Chi-Man Vong, Yu-Shen Liu, Matthias Zwicker, and C.L. Philip Chen. 2019. 3DViewGraph: Learning Global Features for 3D Shapes from A Graph of Unordered Views with Attention. In *IJCAI*.

[28] Kaveh Hassani and Mike Haley. 2019. Unsupervised Multi-Task Feature Learning on Point Clouds. In *IEEE International Conference on Computer Vision*.

[29] Xinwei He, Yang Zhou, Zhichao Zhou, Song Bai, and Xiang Bai. 2018. Triplet-Center Loss for Multi-View 3D Object Retrieval. In *The IEEE Conference on Computer Vision and Pattern Recognition*.

[30] Tao Hu, Zhizhong Han, and Matthias Zwicker. 2020. 3D Shape Completion with Multi-view Consistent Inference. In *AAAI*.

[31] H. Huang, E. Kalegorakis, S. Chaudhuri, D. Ceylan, V. Kim, and E. Yumer. 2017. Learning Local Shape Descriptors with View-based Convolutional Neural Networks. *ACM Transactions on Graphics* (2017).

[32] Dinghuang Ji, Junghyun Kwon, Max McFarland, and Silvio Savarese. 2017. Deep View Morphing. *The IEEE Conference on Computer Vision and Pattern Recognition* (2017).

[33] Yue Jiang, Dantong Ji, Zhizhong Han, and Matthias Zwicker. 2020. SDFDiff: Differentiable Rendering of Signed Distance Fields for 3D Shape Optimization. In *IEEE Conference on Computer Vision and Pattern Recognition*.

[34] Edward Johns, Stefan Leutenegger, and Andrew J. Davison. 2016. Pairwise Decomposition of Image Sequences for Active Multi-view Recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*. 3813–3822.

[35] Asako Kanezaki, Yasuyuki Matsushita, and Yoshifumi Nishida. 2018. RotationNet: Joint Object Categorization and Pose Estimation Using Multiviews from Unsupervised Viewpoints. In *IEEE Conference on Computer Vision and Pattern Recognition*.

[36] Michael Kazhdan, Thomas Funkhouser, and Szymon Rusinkiewicz. 2003. Rotation invariant spherical harmonic representation of 3D shape descriptors. In *Proceedings of Eurographics Symposium on Geometry Processing*. 156–165.

[37] Chun-Liang Li, Manzil Zaheer, Yang Zhang, Barnabas Poczos, and Ruslan Salakhutdinov. 2018. Point Cloud GAN. *CoRR* abs/1810.05795 (2018).

[38] Shikun Liu, C. Lee Giles, and Alexander G. Ororbia II. 2018. Learning a Hierarchical Latent-Variable Model of 3D Shapes. In *2018 International Conference on 3D Vision (3DV)*.

[39] Xinhai Liu, Zhizhong Han, Fangzhou Hong, Yu-Shen Liu, and Matthias Zwicker. 2020. LRC-Net: Learning discriminative features on point clouds by encoding local region contexts. *Computer*

Aided Geometric Design 79 (2020), 101859.

[40] Xinhai Liu, Zhizhong Han, Yu-Shen Liu, and Matthias Zwicker. 2019. Point2Sequence: Learning the Shape Representation of 3D Point Clouds with an Attention-based Sequence to Sequence Network. In *AAAI*. 8778–8785.

[41] Xinhai Liu, Zhizhong Han, Yu-Shen Liu, and Matthias Zwicker. 2021. Fine-Grained 3D Shape Classification With Hierarchical Part-View Attention. *IEEE Transactions on Image Processing* 30 (2021), 1744–1758.

[42] Xinhai Liu, Zhizhong Han, Wen Xin, Yu-Shen Liu, and Matthias Zwicker. 2019. L2G Auto-encoder: Understanding Point Clouds by Local-to-Global Reconstruction with Hierarchical Self-Attention. In *ACM International Conference on Multimedia*.

[43] Baorui Ma, Zhizhong Han, Yu-Shen Liu, and Matthias Zwicker. 2021. Neural-Pull: Learning Signed Distance Functions from Point Clouds by Learning to Pull Space onto Surfaces. In *International Conference on Machine Learning*.

[44] D. Maturana and Scherer S. 2015. VoxNet: A 3D Convolutional Neural Network for real-time object recognition. In *International Conference on Intelligent Robots and Systems*. 922–928.

[45] Ryo Miyagi and Masaki Aono. 2017. Sliced voxel representations with LSTM and CNN for 3D shape recognition. In *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference*.

[46] Deepak Pathak, Philipp Krähenbühl, Jeff Donahue, Trevor Darrell, and Alexei Efros. 2016. Context Encoders: Feature Learning by Inpainting. In *Computer Vision and Pattern Recognition*.

[47] Omid Poursaeed, Tianxing Jiang, Quintessa Qiao, Nayun Xu, and Vladimir G. Kim. 2020. Self-supervised Learning of Point Clouds via Orientation Estimation. *3DV* (2020).

[48] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. 2017. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*.

[49] C R Qi, H Su, and M Niebner. 2016. Volumetric and Multi-view CNNs for Object Classification on 3D Data. In *IEEE Conference on Computer Vision and Pattern Recognition*. 5648–5656.

[50] Charles Ruizhongtai Qi, Hao Su, Matthias Nießner, Angela Dai, Mengyuan Yan, and Leonidas Guibas. 2016. Volumetric and Multi-View CNNs for Object Classification on 3D Data. In *IEEE Conference on Computer Vision and Pattern Recognition*. 5648–5656.

[51] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J. Guibas. 2017. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. In *Advances in Neural Information Processing Systems*. 5105–5114.

[52] Edoardo Remelli, Pierre Baque, and Pascal Fua. 2019. NeuralSampler: Euclidean Point Cloud Auto-Encoder and Sampler. *CoRR* abs/1901.09394 (2019).

[53] Danilo Jimenez Rezende, S. M. Ali Eslami, Shakir Mohamed, Peter Battaglia, Max Jaderberg, and Nicolas Heess. 2016. Unsupervised Learning of 3D Structure from Images. In *Advances in Neural Information Processing Systems*. 4997–5005.

[54] Jonathan Sauder and Bjarne Sievers. 2019. Self-Supervised Deep Learning on Point Clouds by Reconstructing Space. In *Advances in Neural Information Processing Systems*. 12962–12972.

[55] M. Savva, F. Yu, Hao Su, M. Aono, B. Chen, D. Cohen-Or, W. Deng, Hang Su, S. Bai, and X. Bai. 2016. SHREC'16 Track Large-Scale 3D Shape Retrieval from ShapeNet Core55. In *EG 2016 workshop on 3D Object Recognition*.

[56] N. Sedaghat, M. Zolfaghari, E. Amiri, and T. Brox. 2017. Orientation-boosted voxel nets for 3D object recognition. In *British Machine Vision Conference*.

[57] Konstantinos Sfikas, Theoharis Theoharis, and Ioannis Pratikakis. 2017. Exploiting the PANORAMA Representation for Convolutional Neural Network Classification and Retrieval. In *Eurographics Workshop on 3D Object Retrieval*. 1–7.

[58] Abhishek Sharma, Oliver Grau, and Mario Fritz. 2016. VConv-DAE: Deep Volumetric Shape Learning Without Object Labels. In *Proceedings of European Conference on Computer Vision*. 236–250.

[59] B. Shi, S. Bai, Z. Zhou, and X. Bai. 2015. DeepPano: Deep panoramic representation for 3D shape recognition. *IEEE Signal Processing Letters* 22, 12 (2015), 2339–2343.

[60] Ayan Sinha, Jing Bai, and Karthik Ramani. 2016. Deep Learning 3D Shape Surfaces Using Geometry Images. In *European Conference on Computer Vision*. 223–240.

[61] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik G. Learned-Miller. 2015. Multi-view convolutional neural networks for 3D shape recognition. In *International Conference on Computer Vision*. 945–953.

[62] Yongbin Sun, Yue Wang, Ziwei Liu, Joshua E. Siegel, and Sanjay E. Sarma. 2018. PointGrow: Autoregressively Learned Point Cloud Generation with Self-Attention. *CoRR* abs/1810.05591 (2018). arXiv:1810.05591 http://arxiv.org/abs/1810.05591

[63] Chu Wang, Marcello Pelillo, and Kaleem Siddiqi. 2017. Dominant Set Clustering and Pooling for Multi-View 3D Object Recognition. In *Proceedings of British Machine Vision Conference*.

[64] Peng-Shuai Wang, Yang Liu, Yu-Xiao Guo, Chun-Yu Sun, and Xin Tong. 2017. O-CNN: Octree-based Convolutional Neural Networks for 3D Shape Analysis. *ACM Transactions on Graphics* 36, 4 (2017), 72:1–72:11.

[65] Xin Wen, Zhizhong Han, Yan-Pei Cao, Pengfei Wan, Wen Zheng, and Yu-Shen Liu. 2021. Cycle4Completion: Unpaired Point Cloud Completion using Cycle Transformation with Missing Region Coding. In *IEEE Conference on Computer Vision and Pattern Recognition*.

[66] Xin Wen, Zhizhong Han, Xinhai Liu, and Yu-Shen Liu. 2020. Point2SpatialCapsule: Aggregating Features and Spatial Relationships of Local Regions on Point Clouds Using Spatial-Aware Capsules. *IEEE Transactions on Image Processing* 29 (2020), 8855–8869.

[67] Xin Wen, Zhizhong Han, Geunhyuk Youk, and Yu-Shen Liu. 2020. CF-SIS: Semantic-Instance Segmentation of 3D Point Clouds by Context Fusion with Self-Attention. In *ACM International Conference on Multimedia*.

[68] Xin Wen, Tianyang Li, Zhizhong Han, and Yu-Shen Liu. 2020. Point Cloud Completion by Skip-attention Network with Hierarchical Folding. In *IEEE Conference on Computer Vision and Pattern Recognition*.

[69] Xin Wen, Peng Xiang, Zhizhong Han, Yan-Pei Cao, Pengfei Wan, Wen Zheng, and Yu-Shen Liu. 2021. PMP-Net: Point Cloud Completion by Learning Multi-step Point Moving Paths. In *IEEE Conference on Computer Vision and Pattern Recognition*.

[70] Lotter William, Kreiman Gabriel, and Cox David. 2016. Unsupervised Learning of Visual Structure using Predictive Generative Networks. In *ICLR*.

[71] Jiajun Wu, Chengkai Zhang, Tianfan Xue, Bill Freeman, and Josh Tenenbaum. 2016. Learning a Probabilistic Latent Space of Object Shapes via 3D Generative-Adversarial Modeling. In *Advances in Neural Information Processing Systems*. 82–90.

[72] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 2015. 3D ShapeNets: A Deep Representation for Volumetric Shapes. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. 1912–1920.

[73] Peng Xiang, Xin Wen, Yu-Shen Liu, Yan-Pei Cao, Pengfei Wan, Wen Zheng, and Zhizhong Han. 2021. SnowflakeNet: Point Cloud Completion by Snowflake Point Deconvolution with Skip-Transformer. In *IEEE International Conference on Computer Vision*.

[74] Jianwen Xie, Zilong Zheng, Ruiqi Gao, Wenguan Wang, Song-Chun Zhu, and Ying Nian Wu. 2018. Learning Descriptor Networks for 3D Shape Synthesis and Analysis. In *IEEE Conference on Computer Vision and Pattern Recognition*.

[75] Cheng Xu, Zhaoqun Li, Qiang Qiu, Biao Leng, and Jingfei Jiang. 2019. Enhancing 2D Representation via Adjacent Views for 3D Shape Retrieval. In *The IEEE International Conference on Computer Vision*.

[76] Xinchen Yan, Jimei Yang, Ersin Yumer, Yijie Guo, and Honglak Lee. 2016. Perspective Transformer Nets: Learning Single-View 3D Object Reconstruction without 3D Supervision. In *Advances in Neural Information Processing Systems*. 1696–1704.

[77] Yaoqing Yang, Chen Feng, Yiru Shen, and Dong Tian. 2018. FoldingNet: Point Cloud Auto-encoder via Deep Grid Deformation. In *CVPR*.

[78] L. Zhang and Z. Zhu. 2019. Unsupervised Feature Learning for Point Cloud Understanding by Contrasting and Clustering Using Graph Convolutional Neural Networks. In *3DV*. 395–404.

[79] Yongheng Zhao, Tolga Birdal, Haowen Deng, and Federico Tombari. 2018. 3D Point-Capsule Networks. *CoRR* abs/1812.10775 (2018).

[80] Tinghui Zhou, Shubham Tulsiani, Weilun Sun, Jitendra Malik, and Alexei A. Efros. 2016. View Synthesis by Appearance Flow. In *European Conference on Computer Vision*, Vol. 9908. 286–301.