Unsupervised 3D Shape Representation Learning using Normalizing Flow

Xiang Li✉[1*][0000−0002−9946−7000], Congcong Wen[2*][0000−0001−6448−003X], and
Hao Huang[2][0000−−0002−9131−5854]

[1] King Abdullah University of Science and Technology, Thuwal, Saudi Arabic
xiangli92@ieee.org
[2] New York University Abu Dhabi, Abu Dhabi, UAE
{cw3437,hh1811}@nyu.edu

Abstract. Learning robust and compact shape representation learning plays an important role in many 3D vision tasks. Existing supervised learning-based methods have achieved remarkable performance, meanwhile requiring large-scale human-annotated datasets for model training. Self-supervised/unsupervised methods provide an attractive solution to this issue that can learn shape representations without the need for ground truth labels. In this paper, we introduce a novel self-supervised method for shape representation learning using normalizing flows. Specifically, we build a model upon a variational normalizing flow framework where a sequence of normalizing flow layers are adopted to model exact posterior latent distribution and enhance the representation power of the learned latent code. To further encourage inter-shape separability and intra-shape compactness among a batch of shapes, we design a contrastive-center loss that performs metric learning on features on a hypersphere. We validate the representation learning ability of our model on downstream classification tasks. Experiments on ModelNet40/10, ScanobjectNN, and ScanNet datasets demonstrate the superior performance of our method compared with current state-of-the-art methods.

Keywords: Shape representation Learning · Normalizing Flow · Contrastive Learning.

1 Introduction

With recent advancements in range sensors (i.e. LiDAR and RGBD cameras) and imaging technologies (i.e. 3D MRI), the amount of available 3D geometric data has increased dramatically. It is therefore of great importance to develop methods that can take advantage of the ubiquity of 3D point cloud data for 3D scene understanding. One fundamental problem with 3D geometric data is learning representative and robust feature representations. To handle this problem, existing supervised-learning-based methods have achieved remarkable performance

[*] equal contribution. Corresponding author: Xiang Li.

with the help of large-scale human-annotated datasets. However, human annotations are usually labor-intensive and time-consuming and an inadequate dataset may lead to poor generalization ability of the learned models. Therefore, unsupervised representation learning stands out as an attractive alternative and drew huge research attention in the 3D vision community.

Several studies have been devoted to addressing this challenging problem [1–3]. To train the neural network models without ground truth labels, these methods formulate self-supervision signals from careful-designed generation or reconstruction tasks, including self-reconstruction [4, 1, 2], transformation equivariant [5–7], local-to-global reconstruction [3, 8, 9] and distribution approximation [10, 11]. Although these methods obtain ever-increasing downstream classification performance on several benchmark datasets, two challenging issues still exist and impede these methods to get better performance than state-of-the-art supervised methods. First, existing methods mostly focus on formulating self-supervision signals from latent representation while failing to regularize latent distribution, and thus the *learned latent representation cannot well characterize the structural distribution of input data.* Second, these methods usually overemphasize global representations while neglecting *semantic local structures and the relationship between local and global representations.*

For the first issue, a direct remedy is to use a simple Gaussian prior over shape representations, like the ones used in VAE models [12]. But it has been shown that a restricted prior tends to limit the performance of VAEs [13]. Inspired by the great success of normalizing flow-based models for unsupervised density modeling [14, 15]. In this paper, we introduce a variational normalizing flow-based module to encourage more flexible latent distribution which can potentially *better characterize the global structures of irregular 3D shapes by exact log-likelihood modeling.* To the best of our knowledge, we are the first to use normalizing flows for unsupervised shape representation learning.

For the second issue, we generate our solution based on the observation that local patterns of 3D shapes are highly related to global patterns. The human can recognize an object category from only part of the object and also identify whether a local patch can be a constructive part of a given object. A desirable shape representation model should take into account both local and global structures when designing the feature learning module. To this end, we aim to enhance global shape representations by incorporating a *self-supervised local-global semantic supervision.* Specifically, we formulate a contrastive-center loss on local and global embeddings to encourage inter-shape separability and intra-shape compactness of learned embeddings.

We validate the representation learning ability of our model on downstream classification tasks. Extensive experiments are conducted on three benchmark datasets and results show that the proposed unsupervised method obtains better performance than its supervised counterpart and exhibits robustness to sparse point sampling and input noise. The proposed method also reports new state-of-the-art performance on ModelNet40, ModelNet10, and ScanNet datasets, with a single view classification accuracy of 93.3%, 95.6%, and 90.8% respectively.

2 Related Work

2.1 3D Point Cloud Representation Learning

Supervised-learning based methods. As a pioneering work, PointNet [16] introduce the first deep learning-based method that directly learns point features from unstructured raw point clouds. Although it provides a simple and efficient architecture for point cloud signature learning, it lacks the ability to capture the local structure information. PointNet++ [17] tries to address this issue using hierarchy point sampling and grouping techniques. Subsequent works try to improve the performance by designing new point convolution operations that can better capture local structural information. DGCNN [18] and its following works [19] regard point clouds as undirected graphs and formulate point feature learning a dynamic message passing process on graph data. PointCNN [20] learns an $\mathcal{X}$-transformation to reorder the input points into a canonical order. KPConv [21] build point convolution based on rigid or deformable kernel points. In light of the great success of vision transformers [22], recent works [23–25] develop point convolutions with self-attention networks. In this paper, we build our unsupervised shape representation learning framework using PointNet++ as the backbone network for point feature learning. Other PointNet++-like variants can be easily adapted into our pipeline.
Unsupervised-learning based methods. To learn feature representation for 3D point clouds without access to ground truth labels, previous methods have developed various types of self-supervision signals. The most intuitive self-supervised signal can be formulated in a self-reconstruction process where the global feature representations are first learned from the input point clouds and then a decoder network is used to reconstruct the inputs from the feature representations [1, 4]. Similarly, contrastive learning-based methods [26, 7] have also been explored for unsupervised pre-training for 3d representation learning. In the light of adversarial networks for various data generation tasks, researchers proposed to use generative adversarial networks (GANs) [27] to learn a probabilistic latent space of 3D objects [28]. Instead of using an explicit encoder network to learn 3D shape representations, recent works also explored auto-decoder networks for shape representation learning [29, 30]. Although these methods have obtained ever-increasing performance for unsupervised 3D shape representation learning, they usually fail to capture high-level semantic information thus the performance fall behind state-of-the-art supervised methods. To address this issue, recent works [3, 9] incorporated semantic knowledge by simultaneously exploiting local and global self-supervision in order to learn discriminative representations. In this paper, we aim to enhance the learned shape representation by exploiting the semantic relation between local and global structures by a newly designed contrastive-center loss.

2.2 Normalizing Flows

Normalizing Flows (NFs) are a family of generative models based on an invertible mapping between the data distribution and latent distribution. Pioneering work

[14] introduced the first flow-based deep learning framework for high-dimensional density estimation using change of variable theory. To enable the tractability of the Jacobian determinant, a coupling layer was proposed with efficient bijective transformation. Recent works have demonstrated the superior performance in many generation tasks, including image generation [13, 15], audio synthesis [31, 32], video generation [33], and machine translation [34]. Thanks to the attractive merits of exact log-likelihood modeling, normalizing flows have become a powerful technique for unsupervised density modeling.

Recent efforts have full-filled theoretical developments and applications of flow-based methods. In [13], the authors introduced a variational normalizing flow model that combines the merits of VAE and normalizing models in a unified framework where flow layers are used to transform latent variable from a simple diagonal Gaussian distribution to a highly flexible distribution that characterizes the true posterior. Glow [15] introduced a simple but effective generative flow using an invertible 1x1 convolution and demonstrated its effectiveness and efficiency for synthesizing realistic high-resolution natural images. A comprehensive review of normalizing flow can be found [35]. Recent works [11, 36–38] have developed numerous flow-based methods for for a wide range of 3D tasks, such as point cloud generation, single-view 3D reconstruction. In this paper, a variational normalizing flow module is designed to enhance latent representations by using normalizing flows to characterize the exact latent distribution.

3 Method

In this section, we introduce the normalizing Flow-based method for unsupervised 3D Shape representation learning, named SFlow. Our proposed method is built upon a self-reconstruction framework with normalizing flow modules to ensure the learned latent code can characterize the exact probability distribution of input data. A newly designed contrastive loss is further applied to the semantic embeddings and global representations to encourage the discrimination abilities of the learned features. Fig. 1 gives an overview of the proposed method. Our method includes three main components. The first component is **Self-Supervised Reconstruction** module. In this module, our model first leverages an encoder network Q_ϕ to learn global representation z from an input shape X, i.e., $z = Q_\phi(X)$, and then employs a decoder network D to decode the global representation into a reconstructed shape $\hat{X}$, i.e., $\hat{X} = D(z)$. The network architecture will be illustrated in section 3.1. The second component is **Variational Normalizing Flow** module, in which a reparametrization trick is leveraged to generate initial latent code z_0 from a Gaussian distribution $\mathcal{N}(\mu, \sigma)$. Then, we leverage a sequence of normalizing flow layers to learn the exact probability distribution z_K. The initial probability distribution after the encoder network 'flows' through the sequence of invertible mappings and is finally constrained by standard Gaussian prior, see section 3.2. For the third component, **Feature Contrastive** module, we formulate a contrastive-center loss to encourage intra-shape compactness and inter-shape separability of the

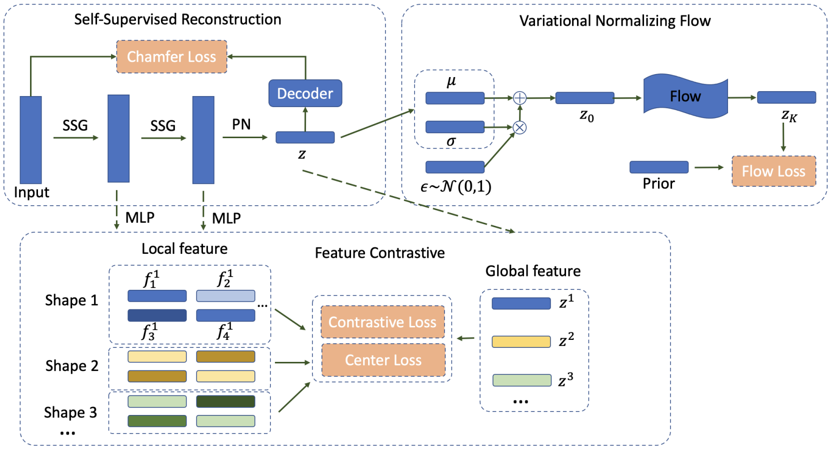

Fig. 1. Overview of the proposed method. Our SFlow model starts with a PointNet++ [16] backbone network to extract global feature z of the input shape. 'PN' denotes a unit PointNet [16], and 'SSG' denotes Single-Scale Grouping (SSG) proposed in PointNet++ [17]. A decoder network is leveraged to recover the input shape with Chamfer loss as a supervision signal. Then, a variational normalizing flow module is developed to transform the latent distribution into a standard Gaussian prior through a sequence of invertible mappings. A feature contrastive module with both contrastive loss and center loss is designed in the embedding space to encourage intra-shape compactness and inter-shape separability of the local and global embeddings.

local and global embeddings. Both a softmax-based contrastive loss and a center loss are defined to perform metric learning on features on a hypersphere, see section 3.3. A shared multi-layer perceptron (MLP) network is leveraged after each downsampling block to transform the local features into the same dimension as the global feature before calculating contrastive-center loss.

3.1 Self-Supervised Reconstruction

Self-supervised reconstruction, or point auto-encoding, is one of the first family of methods for unsupervised 3D shape representation learning [1, 2, 4]. This type of method starts by using an encoder network to learn global shape representation and then a decoder network to reconstruct input shapes. A self-reconstruction loss, e.g., Chamfer distance [39], can then be used to provide self-supervision signals for model training. In our method, we leverage a hierarchy point feature learning network proposed in PointNet++ [17] as the encoder. Given a 3D point set $X = \{x_1, x_2, ..., x_N\}$, where each point x_i is represented by a 3D coordinate and possibly attributes (e.g., surface normal), and N is the number of points. To directly learn feature representations from raw point sets, pioneering work PointNet [16] proposed to use a shared MLP network to learn per-point feature embeddings followed by a symmetry function, e.g. max-pooling, to get global shape representation. PointNet++ [17] enhances the method by introducing a set abstraction and feature interpolation layer to enable a hierarchy feature learning. Specifically, at each set abstraction layer, a smaller number of points are selected from the previous layer using farthest point sampling, and a unit PointNet is applied to the local neighborhood around each selected point. The global shape representation can be obtained by applying a smaller PointNet on the final abstraction layer.

To perform self-reconstruction, a folding-based [1] decoder network is adopted to transform the global shape representation into a set of 3D coordinates. Specifically, the global shape representation is concatenated with the coordinate of a canonical 2D grid and a multi-level MLP network is used to deform the 2D grid onto an underlying 3D object surface, i.e., $\hat{X} = D(z, \mathcal{G})$, where $\mathcal{G}$ is the coordinates of regular 2D grid. A self-supervised Chamfer loss is adopted to train the self-reconstruction network, defined as:

$$\mathcal{L}_{rec} = \sum_{x \in X} \min_{y \in \hat{X}} ||x - y||_2 + \sum_{x \in \hat{X}} \min_{y \in X} ||x - y||_2. \tag{1}$$

Optionally, a normal estimation network Ψ can be built upon the learned global representation to further encourage high-level semantic feature learning. Unlike previous methods [17] that use normal as additional inputs, our method uses normal information as auxiliary output supervision, thus relieving the need for normal information at the inference stage. Specifically, we concatenate the 3D coordinate of each input point x_i with the global feature vector z and feed it into a shared MLP network to predict the normal estimations. The cosine similarity loss is used to train the network:

$$\mathcal{L}_{nor} = -\frac{1}{N} \sum_{i} cos(\Psi(z, x_i), \mathbf{n}_{x_i}) \tag{2}$$

, where $\mathbf{n}_{x_i}$ denotes the ground truth normal for point x_i.

3.2 Variational Normalizing Flow

The above self-supervised point auto-encoding (AE) [40] model can be easily extended to a probabilistic form of variational auto-encoder (VAE) [12] by constraining the latent variable by some underlying probability distributions. Given input data X, a typical VAE model characterize the data distribution via latent variable z with a prior distribution $P_\psi(z)$, and captures the distribution of X given z using a decoder network $P_\theta(X|z)$. An encoder/inference network is typically used to generate the mean and variance of latent distribution $Q_\phi(z|X)$. During training, the parameters of the encoder and decoder networks are jointly optimized to maximize a lower bound on the log-likelihood of the input data,

$$\begin{aligned} \log P_\theta(X) &\geq \log P_\theta(X) - \mathcal{D}_{KL}(Q_\phi(z|X)||p_\theta(z|X)) \\ &= E_{Q_\phi(z|X)}[\log p_\theta(X|z)] - D_{KL}(Q_\phi(z|X)||p_\psi(z)) \\ &= -\mathcal{L}(X) \end{aligned} \tag{3}$$

, which is also called the evidence lower bound (ELBO). From the above equation, the ELBO jointly optimizes the negative reconstruction error (the first term) and a latent distribution regularizer (the second term), which is KL divergence between the approximate posterior and the prior distribution. In practice, $Q_\phi(z|x)$ is modeled by a diagonal Gaussian distribution $\mathcal{N}(\mu_\phi(X), \sigma_\phi(X))$ where the mean $\mu_\phi(X)$ and the standard-deviation $\sigma_\phi(X)$ are predicted by a deep neural network $Q_\phi(z|X)$.

One limitation of the VAE model lies in the available choices of posterior approximating families where the true posterior is unknown and is generally more complex than the assumption allows for. Choosing a highly flexible and computationally-feasible approximate posterior distribution stands as one of the bottlenecks of VAE models. To handle this issue, one feasible solution is to use normalizing flows to transform a simple distribution into a highly complex one as the posterior in VAE, which makes the model become variational normalizing flows [13].

A normalizing flow defines the transformation from an initial known distribution to a more complicated one using a sequence of invertible mappings. Let $f_1, ..., f_K$ denotes a sequence of invertible functions, where each $f : \mathbb{R}^d \rightarrow \mathbb{R}^d$ with inverse $f^{-1} = g$, s.t., $g \circ f(x) = x$. Given a latent variable z_0 ($z_0 = z$) with distribution $q(z_0)$, a variable z_K with more complex distribution can be generated by recursively apply the transformation, i.e., $z_K = f_K \circ f_{K-1} \circ f_1(z_0)$. The probability distribution of the resulting variable z_K can be generated by the change of variables formula:

$$\log q(z_K) = \log q(z_0) - \sum_{k=1}^{K} \log |\det \frac{\partial f_k}{\partial z_{k-1}}|. \tag{4}$$

Thanks to the invertible characteristic of each transformation function, z_0 can be computed from z_K using inverse flow: $z_0 = f_1^{-1} \circ f_2^{-1} \circ f_K^{-1}(z_K)$. In practice, $f_1, ..., f_n$ are implemented using neural networks with an architecture that ensures the determinant of the Jacobian det $\frac{\partial f_k}{\partial z_{k-1}}$ can be easily computed. In this paper, we use Glow-like 1x1 invertible convolutions for density transformation, interested readers can refer to [15] for details. After applying the above flow transformations, the marginal log-likelihood in eq. (3) can be reformulated as:

$$\begin{aligned}
-\mathcal{L}(X) &= \log P_\theta(X) - \mathcal{D}_{KL}(Q_\phi(z|X)||P_\theta(z|X)) \quad \textit{\%The first row of Eq. (4)} \\
&= \log P_\theta(X) - E_{Q_\phi(z|X)}(\log Q_\phi(z|X) - \log P_\theta(z|X)) \\
&= \log P_\theta(X) - E_{Q_\phi(z|X)}(\log Q_\phi(z|X) - \log P_\theta(z, X) + \log P_\theta(X)) \\
&= E_{Q_\phi(z|X)}[\log Q_\phi(z|X) - \log P_\theta(X, z)] \\
&= E_{q(z_0)}[\log q(z_K) - \log P_\theta(X, z_K)] \quad \textit{\%Replace } Q_\phi(z|X) \textit{ with } z_K \\
&= \mathcal{H}(q(z_0)) - E_{q(z_0)}[\sum_{k=1}^{K} \log |\det \frac{\partial f_k}{\partial z_{k-1}}|] \quad \textit{\%Replace } z_K \textit{ using Eq. (5)} \\
&- E_{q(z_0)}[\log p(X, z_K)]
\end{aligned} \tag{5}$$

, where $\mathcal{H}$ represents the entropy. The first term is the entropy of the approximated posterior, the second term is prior regularization. We denote the first two terms as $\mathcal{L}_{flow}$ in the following sections. The third term is the reconstruction log-likelihood of the input point set, calculated as eq. (1).

3.3 Contrastive-Center Loss

The above self-reconstruction process only characterizes input shapes from a global perspective. In this section, we aim to enhance shape representation by exploiting the relation between local structures and global shapes. Specifically, we first design a contrastive loss to encourage inter-shape separability by encouraging semantic embeddings of each point to be closer to the global representation of the same object than other objects. In the light of instance discrimination [41], our method treats the global representation of one object as the positive class and uses the global representation of other objects as the negative class, and formulates a classification loss to encourage separability. Given a bunch of input shapes $\{X^b\}_{b=1}^{B}$, $\mathbf{f}_i^b$ as the embedding of point x_i^b on shape X^b and the global representation z^b,

$$\mathcal{L}_{cont} = -\frac{1}{B*N}\sum_{b=1}^{B}\sum_{i=1}^{N}\log\frac{\exp(s\omega(\mathbf{f}_i^b)^T z^b)}{\sum_{j=1}^{B}\exp(s\omega(\mathbf{f}_i^b)^T z^j)}. \tag{6}$$

The above loss function will maximize the similarity of each point embeddings $\mathbf{f}_i^b$ with the global representation of the same shape z^b meanwhile minimizing the similarity of each point embeddings with the global representation of other shapes $z^j(j \neq b)$. ω is an MLP network that maps $\mathbf{f}$ to the same dimension as z. Similar to the metric losses used for face recognition [42, 43], we normalize all feature embeddings onto a hypersphere before computing similarities and use a constant value $\mathrm{s} = 64$ to re-scale the features. Note that eq. (6) is calculated on local embeddings at all downsampling levels.

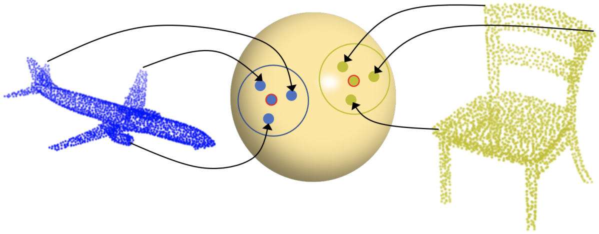

Fig. 2. Illustration of contrastive-center loss on the hypersphere of feature embeddings. The red circle indicates global shape representation, which is treated as the pseudo center point for center loss calculation.

One may note that the above contrastive loss only encourages inter-class separability. Inspired by the studies on contrastive-center loss for face recognition

[44], we further introduce a center loss to enforce intra-shape compactness. Unlike [44] that dynamically updates class centers using gradient decent, we directly treat the global feature of each input shape as the "class center" and develop the center loss as,

$$\mathcal{L}_{center} = \frac{1}{B * N} \sum_{b=1}^{B} \sum_{i=1}^{N} ||\mathbf{f}_i^b - z^b||. \tag{7}$$

Fig. 2 gives an illustration of the proposed contrastive-center loss. The final metric loss used in our method is defined as:

$$\mathcal{L}_{met} = \mathcal{L}_{cont} + \mathcal{L}_{center}. \tag{8}$$

Combining self-reconstruction, normalizing flow loss, and contrastive-center loss, the overall training objective is defined as:

$$\mathcal{L} = \mathcal{L}_{rec} + \mathcal{L}_{flow} + \mathcal{L}_{met} + \mathcal{L}_{nor}. \tag{9}$$

Note that all loss terms in eq. (9) can be calculated in a purely unsupervised way, without using the ground truth labels.

4 Experiments and Results

4.1 Experimental Datasets

We evaluated the 3D shape representation learning performance of our model on ModelNet [45], ScanObjectNN [46] and ScanNet [47] datasets. The ModelNet40/10 dataset consists of 9832/3991 training shapes and 2468/908 test shapes from 40/10 object classes. All point sets are sampled from CAD models with surface normal information provided. ScanObjectNN is a real-world dataset that consists of 2902 3D objects from 15 categories. We used the "object-only" split in our experiments. ScanNet [47] is another real-world scan dataset with 17 object categories where we followed [20] to get points sets from instance segmentation labels. In all experiments, we randomly sampled 1024 points from each 3D object for model training and evaluation. We reported the performance using single view inputs without using the multi-view voting strategy for potential enhancement.

4.2 Implementation Details

Network architecture. In our experiments, we used the encoder part of PointNet++ (PN++) as the backbone network for feature learning. Three set abstraction layers were sequentially applied to reduce the number of points to 512, 128, and 64, with a radius of 0.23, 0.32, and 0.46 respectively, followed by a unit PointNet to get the global representations. In each set abstraction layer, we used Single-Scale Grouping (SSG) instead of Multi-Scale Grouping (MSG) [17] for feature aggregation to reduce model capacity. In a self-supervised reconstruction decoder, we adopted a two-level folding process to reconstruct 3D shapes.

Table 1. Classification accuracy (%) on ModelNet40 (MN40.) and ModelNet10 (MN10.) datasets. (L) denotes the model with a large PN++ backbone network. † indicates the mode is trained on the ShapeNet dataset.

Method	Input	Accuracy	
		MN40.	MN10.
TL Network [50]	voxel	74.40	-
VConv-DAE [51]	voxel	75.50	80.50
3DGAN [28]	voxel	83.30	91.00
VSL [52]	voxel	84.50	91.00
VIPGAN [53]	views	91.98	94.05
†LGAN [2]	points	85.70	95.30
LGAN [2]	points	87.27	92.18
†FoldingNet [1]	points	88.40	94.40
FoldingNet [1]	points	84.36	91.85
MAP-VAE [3]	points	90.15	94.82
GraphTER [7]	points	92.02	-
GLR [9]	points	92.22	94.82
GLR(L) [9]	points	93.02	95.53
SFlow	points	92.78	94.82
SFlow (L)	points	**93.31**	**95.60**

In the variational normalizing flow module, we used a Glow architecture with 16 flow layers, and each has 4 flow steps. The hidden dimension was set to 128 and divided into 8 groups. To evaluate the downstream classification performance, we trained a linear SVM [48] using the feature representations obtained from the training set and evaluated the classification performance on the test split.

Network optimization. Our model was optimized using the Adam optimizer. The initial learning rate was set to 1e-3 and decayed with a scale of 0.7 every 20 epochs. We used a momentum of 0.9 for Batch Normalization layers [49] and decayed with a rate of 0.5 every 20 epochs. Our model was trained for 300 epochs with a batch size of 32 and it took around 30 hours on a single Titan XP GPU.

4.3 Results on ModelNet

To demonstrate the effectiveness of our proposed method for unsupervised shape representation learning, we compared our method with state-of-the-art unsupervised methods in Table 1. We also included the results of our SFlow model with a larger backbone (4x channel width), similar to recent work GLR [9] which reports state-of-the-art performance on the ModelNet dataset. From Table 1, our method achieves new state-of-the-art performance, with a classification accuracy of 93.31% and 95.60% on ModelNet40 and ModelNet10 dataset respectively.

We further compared the performance of our SFlow model with its supervised counterpart. Specifically, we trained supervised PointNet++ models with the same backbone networks, followed by several fully connected layers and a softmax layer to generate the prediction labels. From Table 2, one can see that our SFlow model obtains better performance than its supervised counterpart using both small and large backbone networks. This demonstrates that our unsupervised SFlow model can potentially learn more discriminative representation than its supervised counterpart.

Table 2. Comparison with the supervised counterpart on ModelNet40 dataset.

Model	PN++	PN++ (L)
PN++ (Supervised)	91.69	92.01
SFlow (Unsupervised)	**92.78**	**93.31**

4.4 Cross-Dataset Evaluation

We conducted downstream classification experiments on the real-world ScanObjectNN and ScanNet datasets. Following the experimental settings of GLR [9], we trained our shape representation learning network on the ModelNet40 dataset and evaluated the downstream classification performance on the ScanObjectNN and ScanNet datasets. Note that we did not fine-tune our model on the target datasets. Even though ScanObjectNN and ScanNet datasets have different object categories with the ModelNet40 dataset, our SFlow model can still produce well-separable representations without training/finetuning on the target datasets, as shown in Table 3. This demonstrates that our SFlow method can successfully *learn generic representations from object structures without labels.* Moreover, our model achieves significant better performance than current STOA method GLR [9] on ScanNet dataset (90.8% vs. 89.2%), and a obtains a comparable performance with GLR [9] on ScanObjectNN dataset (87.0% vs. 87.2%). It should be noted that we only got a classification accuracy of 86.2% on the ScanObjectNN dataset using the official code of GLR [9].

4.5 Robustness Analysis

In this section, we investigate the robustness of our model under different numbers of sampled points and noise levels. To achieve this, we evaluate the downstream classification performance on the ModelNet40 dataset with sparser points of 1024, 512, 256, 128, and 64, while the backbone network is still trained on 1024 points. From Fig. 3(a), our SFlow model is a lot more robust than its supervised counterpart and maintains an accuracy higher than 86.1% with only 128 points. For the latter, we added Gaussian noise of $\mathcal{N}(0, \sigma_r)$ to input point sets and generated feature representation using the backbone trained on the clean dataset. We conducted experiments with σ_r choose from $[0, 0.01, 0.02, 0.03, 0.05]$. From Fig. 3(b), our SFlow shows a smaller performance drop than its supervised counterpart and maintains an accuracy higher than 86.5% with a noise level of 0.05.

Table 3. Transferring accuracy (%) on ScanObjectNN (SON.) and ScanNet (SN.) datasets. (L) denotes a large PN++ backbone network. 'Sup' denotes supervised methods. * denotes our reproduced results.

Method	Sup.	Accuracy	
		SON.	SN.
PointNet++[17]	✓	84.3	-
PointCNN [20]	✓	85.5	-
DGCNN [18]	✓	86.2	-
GLR(L) [9]	✗	**87.2**	89.2
*GLR(L) [9]	✗	86.2	89.2
SFlow (L)	✗	87.0	**90.8**

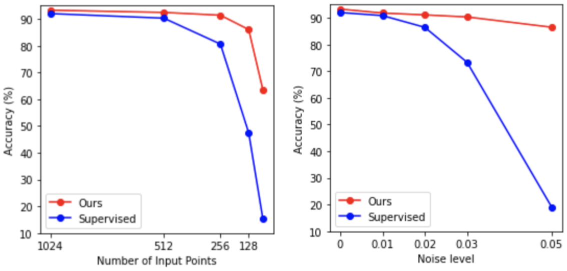

Fig. 3. Robustness test. The classification of our model with different numbers of sampled points and different noise levels.

4.6 Complexity Analysis

In Table 4, we report the model capacity and inference time of our SFlow model and its supervised counterpart. We calculated the total inference time on the whole test set of the ModelNet40 dataset with a batch size of 24 and a single Titan XP GPU. From 4, compared to the supervised PN++ model, our SFlow model only brings slightly more computation cost but can get significantly better performance. Moreover, our SFlow model with a small backbone shows a better trade-off in speed and accuracy compared to the one with a larger backbone.

Table 4. Model Complexity and inference time comparison.

Model	#Param	Time	Acc. (%)
PN++	**1.29M**	**2.99s**	91.69
PN++ (L)	12.11M	8.66s	92.01
SFlow	2.94M	3.29	92.78
SFlow (L)	14.99M	9.03	**93.31**

4.7 Ablation Analysis

In this section, we conduct a detailed ablation study to verify the effectiveness of our model design. We conduct our experiments on the ModelNet40 dataset using a small PointNet++ backbone. In Table 5, the baseline model (model A) is an variant of FoldingNet [1] and trained using reconstruction loss only. By introducing the proposed variational normalizing flow model, our model (model B) got a significant performance boost, from 86.77% to 88.65%. By comparing model A and model C, our newly proposed contrastive-center loss improves the performance by a large margin (+5.43%). This demonstrates our shape representation model benefits a lot by encouraging intra-shape compactness and inter-shape separability. Combining our normalizing flow module and feature contrastive module, our model (model D) gets a further performance boost, with a classification accuracy of 92.67%. We also witnessed a slight performance boost by introducing a normal estimation network.

Table 5. Ablation analysis of our method. We report the classification accuracy (%) on the ModelNet40 test set. ($\mathcal{L}_{rec}$: self-reconstruction loss, $\mathcal{L}_{flow}$: flow loss, $\mathcal{L}_{met}$: contrastive-center loss, $\mathcal{L}_{nor}$: normal estimation loss).

Model	$\mathcal{L}_{rec}$	$\mathcal{L}_{flow}$	$\mathcal{L}_{met}$	$\mathcal{L}_{nor}$	Acc.(%)
A	✓	✗	✗	✗	86.77
B	✓	✓	✗	✗	88.65
C	✓	✗	✓	✗	92.10
D	✓	✓	✓	✗	92.67
E	✓	✓	✓	✓	**92.78**

Self-reconstruction with normalizing flows. To better understand the effect of our normalizing flow module, we build a VAE model that directly constrains the latent representation by a Gaussian prior and trained the model by maximizing the lower bound defined in eq. 3. From Table 6, using a simple Gaussian prior leads to similar reconstruction performance as the baseline AE model, as indicated by Chamfer distance (0.062 vs. 0.063), but can slightly enhance the downstream classification accuracy with a more powerful representation (87.60% vs. 86.77%). In contrast, thanks to a more flexible latent distribution enabled by normalizing flow transformations, our model enhances the latent representation with a better reconstruction performance (0.059 vs. 0.063). The downstream classification performance in Table 6 also supports that our variational normalizing flow module contributes to a more powerful latent representation (88.65% vs. 86.77%). Moreover, we also investigated the effect of different numbers of flow blocks. From Table 6, our SFlow model gets the best performance with 16 flow blocks.

Effect of different metric losses In the above section, we show the effectiveness of our proposed contrastive-center loss. In this section, we check the effect of each metric separately. Specifically, we conducted experiments using contrastive

Table 6. Performance comparison with different models and different numbers of flow layers. C.D. denotes Chamfer distance.

Model	C.D.	Acc.(%)
baseline (AE)	0.063	86.77
VAE	0.062	87.60
Ours (k=16)	**0.059**	**88.65**
Ours (k=12)	0.060	88.41
Ours (k=20)	0.058	88.43

loss, center loss, and both losses. From Table 7, both metrics can significantly enhance the performance, while softmax-based contrastive loss led to a larger performance boost than center loss (+5.04% vs. +1.40%). Our model got the best performance by using both metrics.

Table 7. Classification accuracy (%) on ModelNet40 test set with different metric losses.

Center loss	Contrastive loss	Acc.(%)
✗	✗	86.77
✓	✗	88.17
✗	✓	91.84
✓	✓	**92.10**

5 Conclusion

In this paper, we introduce an unsupervised method for 3D shape representation learning based on normalizing flow and a newly designed feature discrimination loss. By introducing a variational normalizing flow module to the self-reconstruction process, our model is able to model the exact log-likelihood of latent distribution thus enhancing the representation power of learned latent code. We further designed a feature discrimination loss that combines contrastive loss and center loss to encourage inter-shape separability and intra-shape compactness. We validate the representation learning ability of our model on downstream classification tasks. Experimental results demonstrated our unsupervised method could achieve better performance than its supervised counterpart and our SFlow model obtains new state-of-the-art performance on ModelNet40/10 and ScanNet datasets.

References

1. Yang, Y., Feng, C., Shen, Y., Tian, D.: Foldingnet: Point cloud auto-encoder via deep grid deformation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2018) 206–215

2. Achlioptas, P., Diamanti, O., Mitliagkas, I., Guibas, L.: Learning representations and generative models for 3d point clouds. In: International conference on machine learning, PMLR (2018) 40–49
3. Han, Z., Wang, X., Liu, Y.S., Zwicker, M.: 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: 2019 IEEE/CVF International Conference on Computer Vision (ICCV), IEEE (2019) 10441–10450
4. Deng, H., Birdal, T., Ilic, S.: Ppf-foldnet: Unsupervised learning of rotation invariant 3d local descriptors. In: Proceedings of the European Conference on Computer Vision (ECCV). (2018) 602–618
5. Zhang, L., Qi, G.J., Wang, L., Luo, J.: Aet vs. aed: Unsupervised representation learning by auto-encoding transformations rather than data. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2019) 2547–2555
6. Qi, G.J., Zhang, L., Chen, C.W., Tian, Q.: Avt: Unsupervised learning of transformation equivariant representations by autoencoding variational transformations. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. (2019) 8130–8139
7. Gao, X., Hu, W., Qi, G.J.: Graphter: Unsupervised learning of graph transformation equivariant representations via auto-encoding node-wise transformations. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2020) 7163–7172
8. Liu, X., Han, Z., Wen, X., Liu, Y.S., Zwicker, M.: L2g auto-encoder: Understanding point clouds by local-to-global reconstruction with hierarchical self-attention. In: Proceedings of the 27th ACM International Conference on Multimedia. (2019) 989–997
9. Rao, Y., Lu, J., Zhou, J.: Global-local bidirectional reasoning for unsupervised representation learning of 3d point clouds. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2020) 5376–5385
10. Li, C.L., Zaheer, M., Zhang, Y., Poczos, B., Salakhutdinov, R.: Point cloud gan. arXiv preprint arXiv:1810.05795 (2018)
11. Yang, G., Huang, X., Hao, Z., Liu, M.Y., Belongie, S., Hariharan, B.: Pointflow: 3d point cloud generation with continuous normalizing flows. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. (2019) 4541–4550
12. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. In: International Conference on Learning Representations. (2013)
13. Rezende, D., Mohamed, S.: Variational inference with normalizing flows. In: International conference on machine learning, PMLR (2015) 1530–1538
14. Dinh, L., Krueger, D., Bengio, Y.: Nice: Non-linear independent components estimation. arXiv preprint arXiv:1410.8516 (2014)
15. Kingma, D.P., Dhariwal, P.: Glow: generative flow with invertible 1× 1 convolutions. In: Proceedings of the 32nd International Conference on Neural Information Processing Systems. (2018) 10236–10245
16. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. Proc. Computer Vision and Pattern Recognition (CVPR), IEEE **1** (2017) 4
17. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In: Advances in Neural Information Processing Systems. (2017) 5099–5108

18. Wang, Y., Sun, Y., Liu, Z., Sarma, S.E., Bronstein, M.M., Solomon, J.M.: Dynamic graph cnn for learning on point clouds. Acm Transactions On Graphics (tog) **38** (2019) 1–12
19. Wang, L., Huang, Y., Hou, Y., Zhang, S., Shan, J.: Graph attention convolution for point cloud semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2019) 10296–10305
20. Li, Y., Bu, R., Sun, M., Wu, W., Di, X., Chen, B.: Pointcnn: Convolution on x-transformed points. Advances in neural information processing systems **31** (2018) 820–830
21. Thomas, H., Qi, C.R., Deschaud, J.E., Marcotegui, B., Goulette, F., Guibas, L.J.: Kpconv: Flexible and deformable convolution for point clouds. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. (2019) 6411–6420
22. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in neural information processing systems. (2017) 5998–6008
23. Zhao, H., Jiang, L., Jia, J., Torr, P.H., Koltun, V.: Point transformer. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. (2021) 16259–16268
24. Guo, M.H., Cai, J.X., Liu, Z.N., Mu, T.J., Martin, R.R., Hu, S.M.: Pct: Point cloud transformer. Computational Visual Media **7** (2021) 187–199
25. Engel, N., Belagiannis, V., Dietmayer, K.: Point transformer. IEEE Access **9** (2021) 134826–134840
26. Xie, S., Gu, J., Guo, D., Qi, C.R., Guibas, L., Litany, O.: Pointcontrast: Unsupervised pre-training for 3d point cloud understanding. In: European Conference on Computer Vision, Springer (2020) 574–591
27. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. Advances in neural information processing systems **27** (2014)
28. Wu, J., Zhang, C., Xue, T., Freeman, W.T., Tenenbaum, J.B.: 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. (2016) 82–90
29. Park, J.J., Florence, P., Straub, J., Newcombe, R., Lovegrove, S.: Deepsdf: Learning continuous signed distance functions for shape representation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2019) 165–174
30. Duan, Y., Zhu, H., Wang, H., Yi, L., Nevatia, R., Guibas, L.J.: Curriculum deepsdf. In: European Conference on Computer Vision, Springer (2020) 51–67
31. Kim, S., Lee, S.G., Song, J., Kim, J., Yoon, S.: Flowavenet: A generative flow for raw audio. In: International Conference on Machine Learning, PMLR (2019) 3370–3378
32. Prenger, R., Valle, R., Catanzaro, B.: Waveglow: A flow-based generative network for speech synthesis. In: ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE (2019) 3617–3621
33. Kumar, M., Babaeizadeh, M., Erhan, D., Finn, C., Levine, S., Dinh, L., Kingma, D.: Videoflow: A flow-based generative model for video. arXiv preprint arXiv:1903.01434 **2** (2019)
34. Ma, X., Zhou, C., Li, X., Neubig, G., Hovy, E.: Flowseq: Non-autoregressive conditional sequence generation with generative flow. In: Proceedings of the 2019

Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). (2019) 4282–4292

35. Kobyzev, I., Prince, S., Brubaker, M.: Normalizing flows: An introduction and review of current methods. IEEE Transactions on Pattern Analysis and Machine Intelligence (2020)
36. Klokov, R., Boyer, E., Verbeek, J.: Discrete point flow networks for efficient point cloud generation. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIII 16, Springer (2020) 694–710
37. Kim, H., Lee, H., Kang, W.H., Lee, J.Y., Kim, N.S.: Softflow: Probabilistic framework for normalizing flow on manifolds. Advances in Neural Information Processing Systems **33** (2020)
38. Pumarola, A., Popov, S., Moreno-Noguer, F., Ferrari, V.: C-flow: Conditional generative flow models for images and 3d point clouds. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2020) 7949–7958
39. Fan, H., Su, H., Guibas, L.J.: A point set generation network for 3d object reconstruction from a single image. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2017) 605–613
40. Hinton, G.E., Zemel, R.S.: Autoencoders, minimum description length, and helmholtz free energy. Advances in neural information processing systems **6** (1994) 3–10
41. Wu, Z., Xiong, Y., Yu, S.X., Lin, D.: Unsupervised feature learning via non-parametric instance discrimination. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2018) 3733–3742
42. Liu, W., Wen, Y., Yu, Z., Li, M., Raj, B., Song, L.: Sphereface: Deep hypersphere embedding for face recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2017) 212–220
43. Deng, J., Guo, J., Xue, N., Zafeiriou, S.: Arcface: Additive angular margin loss for deep face recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2019) 4690–4699
44. Qi, C., Su, F.: Contrastive-center loss for deep neural networks. In: 2017 IEEE International Conference on Image Processing (ICIP), IEEE (2017) 2851–2855
45. Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., Xiao, J.: 3d shapenets: A deep representation for volumetric shapes. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2015) 1912–1920
46. Uy, M.A., Pham, Q.H., Hua, B.S., Nguyen, T., Yeung, S.K.: Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. (2019) 1588–1597
47. Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M.: Scannet: Richly-annotated 3d reconstructions of indoor scenes. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2017) 5828–5839
48. Cortes, C., Vapnik, V.: Support-vector networks. Machine learning **20** (1995) 273–297
49. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: International conference on machine learning, PMLR (2015) 448–456

50. Girdhar, R., Fouhey, D.F., Rodriguez, M., Gupta, A.: Learning a predictable and generative vector representation for objects. In: European Conference on Computer Vision, Springer (2016) 484–499
51. Sharma, A., Grau, O., Fritz, M.: Vconv-dae: Deep volumetric shape learning without object labels. In: European Conference on Computer Vision, Springer (2016) 236–250
52. Liu, S., Giles, L., Ororbia, A.: Learning a hierarchical latent-variable model of 3d shapes. In: 2018 International Conference on 3D Vision (3DV), IEEE (2018) 542–551
53. Han, Z., Shang, M., Liu, Y.S., Zwicker, M.: View inter-prediction gan: Unsupervised representation learning for 3d shapes by learning global shape memories to support local view predictions. In: Proceedings of the AAAI Conference on Artificial Intelligence. Volume 33. (2019) 8376–8384