Shape embedding and retrieval in multi-flow deformation

Baiqiang Leng
Tsinghua University
Beijing, China
lbq19@mails.tsinghua.edu.cn

Jingwei Huang
Distributed and Parallel Software Lab, Huawei Technologies
Shenzhen, Guangdong, China
huangjingwei6@huawei.com

Guanlin Shen
Tsinghua University
Beijing, China
sgl17@mails.tsinghua.edu.cn

Bin Wang
Tsinghua University
Beijing, China
wangbins@tsinghua.edu.cn

Abstract

We propose a unified 3D flow framework for joint learning of shape embedding and deformation from different categories. Our goal is to recover shapes from imperfect point clouds by fitting the best shape template in a shape repository under deformation. Accordingly, we learn a shape embedding for template retrieval and a flow-based network for robust deformation. We identify that the deformation flow can be quite diverse for different shape categories. Therefore, we introduce a novel multi-hub module to learn multiple modes of deformation to incorporate such diversity. As a result, we obtain a unique network for handling universal objects from different categories. The shape embedding is trained to retrieve the best-fit template as the nearest neighbor in a latent space. We replace the standard fully connected layer with a tiny structure in the embedding that significantly reduces network complexity and further improves deformation quality. Experiments show superiority of our method over existing state-of-the-art methods according to qualitative and quantitative comparison. Finally, our method provides efficient and flexible deformation that can further be used for novel shape design.

1. Introduction

Recovering high-quality 3D shapes from imperfect point clouds is a fundamental problem in 3D vision and graphics. It provides ready-to-use 3D data for down-streamed tasks, including gaming, virtual reality, and augmented reality. The key challenge is to recover clean and accurate geometry with sharp features from ubiquitous noises in sparse point clouds, this complexity is commonly faced by traditional methods that directly triangulate points [28] or reconstruct the surface using volumetric representation [24]. Although deep learning-based methods alleviate the problem by learning priors from shape repositories [2] and recover shapes using auto-encoders [1, 9, 38], they generate over-smoothed shapes with missing sharp edges and corners. One recent promising direction is via template retrieval and deformation [48, 20], where the best template from a shape repository is determined and deformed to fit the input points. The reason is that deformation from template CAD models usually preserves geometry details, ensures completeness, and yields lightweight models. However, existing methods assume very limited settings where shapes are among the same category. Therefore, category information and multiple pretrained category-related networks are required to apply deformation and retrieval. In this work, we aim to design a unique network that handles joint retrieval and deformation for universal objects with unknown categories.

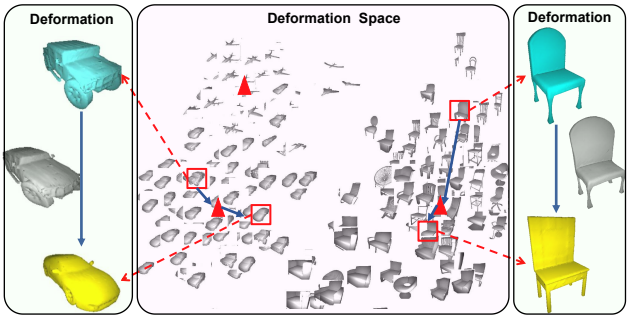

Figure 1. **Deformation latent space.** Left and right: Car and chair deformation through a hub. We learn a latent deformation space by jointly optimizing shape latent codes and their deformation hubs. Both the shape's latent codes and the multiple hubs in the deformation path are optimizable parameters.

Following [48], we address the retrieval problem by learning a shape embedding where the best template can

be retrieved according to the nearest neighbor in the embedded latent space. Mikaela et al. [48] adopt a fixed post-deformation step using as-rigid-as-possible [44], whereas we jointly train an end-to-end flow deformation model following [20] that generates a 3D flow field according to latent vectors of two shapes. Specifically, this field is used to advect the source to the target through a path in the latent space connecting shape vectors and a hub fixed at the origin. We identify that the core limitation of [20] is the assumption that learned embedding or deformation models are among the same category with similar deformation modes. As a result, the learned deformation cannot be extended to the real world with diverse object categories. Our insight is to learn multiple paths in the latent space, each of which incorporates a deformation mode. Therefore, we introduce additional degrees of freedom for the deformation paths in our network. First, we allow deformation through paths connecting multiple different hubs, each of which represents a different deformation mode. Second, Jiang et al. [20] fixes the hub at the origin, whereas we model hubs as mutable locations in the latent space and jointly train hubs with the embedding and the flow model. The joint learning helps automatically explore deformation modes in the latent space, which is shown in Figure 1. We learn a latent deformation space by jointly optimizing shape latent codes and their deformation hubs in the space. The shape's latent codes and the multiple hubs in deformation path are optimizable parameters.

We aim to learn the deformation to ensure that the nearest hub between shapes indicates the optimal deformation mode. Therefore, we model the training loss as the fitting loss according to deformation via the learned nearest hub. We propose a novel scheme that progressively identifies reasonable pairs of shapes during training. If we need deformation between different categories, then we select random pair of shapes from all categories of models during training. If we only want the deformation within the same category, we ensure that the source model and target are from same class. As a result, the trained embedding tends to group shapes into different clusters where intra-clusters are well deformed to each other through the same hub (shown in Figure 1) with a certain deformation mode. To capture the geometry features of multi-categories shapes more efficiently, we propose a novel backbone called DFF-Net for our flow model. It takes input features and passes them through multiple branches of IM-Net [6] and aggregates them. The multi-branch structure borrows the idea of InceptionNet [45] that improves feature diversity by different sizes of convolution kernels and a different number of channels, while each branch of IM-Net [6] helps to improve visual quality by concatenating point coordinates with shape features. This design improves retrieval and deformation performance but significantly reduces the number of parameters and time consumption.

We compare our method with existing state-of-the-art methods for shape reconstruction on ShapeNet [2], where our results are the best considering sharp features and realistic appearance for sparse point clouds as input. Different from neural cages [53] that preserves source shape structures, we provide more flexible deformations that better capture the target shape style. This feature leads to a byproduct application for novel shape design. Ablation studies highlight our contribution on the multi-hub module and the new backbone.

Our contributions are summarized as follows:

- A unique multi-hub network for learning retrieval and deformation of universal objects with unknown categories.
- A novel backbone for our framework, namely, DFF-Net, which achieves dramatic improvement in efficiency and deformation quality.
- A progressive training scheme to effectively learn universal object deformation.
- Improvements in applications including shape reconstruction and novel shape design.

2. Related works

3D shape deformation 3D shape deformation aims to generate new shapes by deforming existing shapes while retaining local geometry features. Earlier works model the deformation as an optimization problem that fit dense [43, 12, 59] or sparse key-point [57, 61] observations with rigid [17] or non-rigid [30] regularization. Deformation can be free-form [19, 27, 35] where vertices [52, 21, 33] are directly optimized, whereas shape templates or cages [14, 51, 23, 32] can serve as agents for deformation to preserve shape integrity.

With the development of deep learning, optimization can be replaced by efficient network prediction of vertex offsets under deformation [60, 15, 19, 52, 21]. Alternative solutions directly predict cage deformation [58] or model it as a continuous flow [37, 20]. We further develop the deformation network to handle universal objects with unknown categories by modeling multiple modes of deformation as learnable latent paths with hubs.

Cad-deform [18] is proposed to deal with deformation between scan and cad model. It can obtain more accurate CAD-to-scan fits by non-rigidly deforming retrieved CAD models. However, it does not jointly optimize the embedding and deformation process, and it mainly focuses on deformation optimization. Given an image input, the Deform-Net [27] obtains a nearest retrieved point cloud shape and deform a template to match the image. However, it can only

obtain coarse point cloud results, and not complete meshes, and this incapability leads to limited performance. Unlike these techniques, our method can obtain complete meshes of models with joint learning of embedding and shape deformation with flow models Neural ODE[4]. Flow models are useful in shape deformation. Neural ODE[4] is a method of continuous normalizing flow models that combine ordinary differential equation solver and neural networks.

As applications of neural flow models, ShapeFlow [20] can learn the geometry of different 3D models by using flowing models, and Pointflow [56] proposes a principled probabilistic framework to generate 3D point clouds by modeling them as the distribution of distributions which learn a two-level hierarchy of distributions. Occflow [37] is also a method which learns flow dynamics to reconstruct models that belong to 4D models. Apart from the neural flow models, autoregressive probability density estimation techniques, such as WaveNet [39], PixelRNN [50] and IAF [25], are used to learn joint probability density and transformed distribution. RealNVP [11] can be regarded as a particular case of bijective functions of IAF, and it mentions a novel distribution, which is batch regularized bijective function and can be used to stabilize the training process. Flow-based models are used combined with some generative models, such as auto-regressive models [25, 41, 16, 10] and VAEs [42, 49, 25, 13].

In our framework, the continuous deformation flow is learned by our proposed deformation flow network(DFF-Net) and it contributes to the natural deformation between shape pairs. Our model is inspired by the Neural ODE[4] method, we incorporate the advantages of flow models to get continuous deformation to achieve a better deformation, embedding, and reconstruction result of 3D models.

Shape embedding and retrieval Shape embedding and retrieval has become an important part of shape processing, and the technique has been used in object completion, shape reconstruction, and other applications. Among them, shape reconstruction has been motivated by shape embedding and retrieval techniques. Tatarchenko et al. [47] believe that, apart from traditional encoder and decoder reconstruction methods, shape retrieval also can be used for shape reconstruction.

The shape retrieval and shape embedding techniques often rely on each other, and they cannot be separated in most situations. Former retrieval technique [36] can be applied in indoor scene understanding. Multi-modal shape embedding [31, 46] is an important part of the embedding techniques. For example, Li et al. [31] propose the first deep learning technique for joint embeddings of shapes and images via CNN networks. Then Tabia et al. [46] provide a new technique for 3D shape retrieval using queries of different modalities, which include 3D models, sketches and images. Other techniques based on multiple modalities are also available, Wu et al.[55] designs a CNN architecture to jointly analyze shapes and images with few training data. Lee et al. [29] proposes a cross-domain image-based retrieval method which can learn joint embedding space for images and 3D shapes in an end-to-end manner. Some works [22, 3] are based on the embedding of 3D models and sketches. In addition to the multi-modal embedding learning between the CAD model and related images, embedding methods between 3D scan and CAD model [8] are available. Generative method [54] is also used to solve embedding problems by jointly learning geometry and structure for 3D shape structure modeling.

Apart from object embedding and retrieval for object reconstruction, indoor scene segmentation and retrieval techniques [34, 26] can also been used for scene reconstruction. Recent works [48] creatively combine embedding technique with deformation method, which can deform from a most similar source shape in the database to a target shape. Our framework can solve more generative problems and achieve SOTA results on the public datasets by extending the above mentioned ideas. Our joint embedding and deformation framework can automatically find the most suitable similar objects for deformation when inputting the sparse point cloud of some unknown objects.

3. Method

3.1. Background

We consider a set of 3D shapes $\mathcal{S} = \{S_1, S_2, \cdots, S_N\}$, where $S_i = \{V_i, E_i\}$ includes $V_i = \{v_1, v_2, \cdots, v_{n_i}\}$ as the ordered vertices set and $E_i = \{e_1, e_2, \cdots, e_{m_i}\}$ as a polygon set. e_i represents vertex indices for the i-th polygon.

We want to learn the deformation mapping from the source shape S_i to the target S_j for minimizing the geometry distance between deformed source and target:

$$\arg\min_{\theta} \mathcal{L}\left(\Phi_{\theta}^{i,j}\left(S_i\right), S_j\right), \tag{1}$$

$\Phi_{\theta}^{i,j}$ is a deformation function that moves a point from the source model to the deformed one depending on information of $\mathcal{S}_i$, $\mathcal{S}_j$, and a learned network with its parameters θ. $\mathcal{L}$ measures the squared Chamfer distance(CD) between two shapes. We view the deformation as a process of advecting 3D flows through a path connecting two input shapes in the shape latent space and define the deformation through a line in the latent space connecting $\boldsymbol{z}_i$ and $\boldsymbol{z}_j$ as

$$\mathcal{D}_{\theta}^{\boldsymbol{z}_i,\boldsymbol{z}_j}(\boldsymbol{p}) = \boldsymbol{p}(1), \boldsymbol{p}(T) = \boldsymbol{p} + \int_0^T \boldsymbol{f}_{\theta}^{\boldsymbol{z}_i,\boldsymbol{z}_j}(\boldsymbol{p}(t), t;)\,\mathrm{d}t, \tag{2}$$

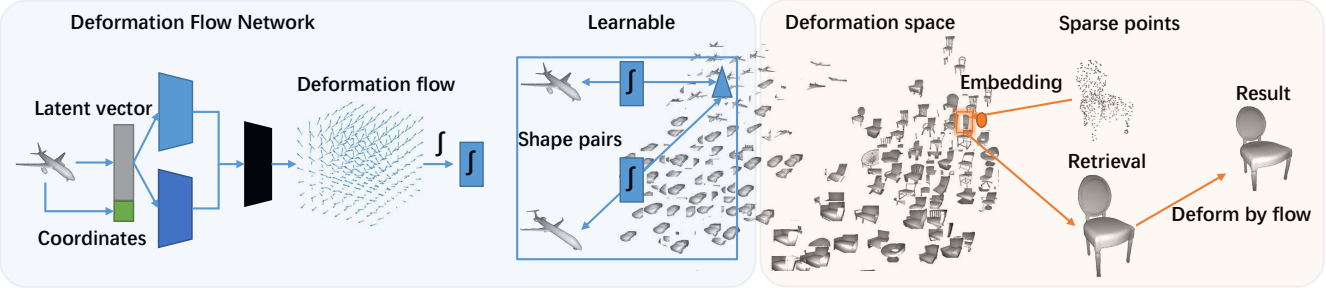

Figure 2. **Framework Overview.** It is a multi-hub flow network. During training, we jointly optimize latent hubs for input shapes and flow network parameter, and obtain nearest hubs between shapes, providing optimal deformation paths. At inference time, given a shape or a point cloud of unknown category, we will embed it to the latent space by minimizing the deformation loss from existing shapes. The closest existing shape will be retrieved and deformed to the input.

where each point p in the shape is deformed by integrating $\boldsymbol{f}$ through t from zero to one. $\boldsymbol{f}$ is a 3D flow model derived from a neural network:

$$\boldsymbol{f}_\theta^{\boldsymbol{z}_i,\boldsymbol{z}_j}(\boldsymbol{p},t) = \boldsymbol{h}_\theta\left(\boldsymbol{p}, \boldsymbol{z}_i + t\left(\boldsymbol{z}_j - \boldsymbol{z}_i\right)\right) \cdot \left\|\boldsymbol{z}_j - \boldsymbol{z}_i\right\|_2, \tag{3}$$

where $\boldsymbol{h}$ is a network that produces a 3D offset given a 3D position $\boldsymbol{p}$ and the location in the deformation path at time t. [20] deforms $\mathcal{S}_i$ to $\mathcal{S}_j$ through a path connecting their latent codes $\boldsymbol{z}_i$ and $\boldsymbol{z}_j$ through a latent hub at origin as

$$\Phi_\theta^{i,j} = \mathcal{D}_\theta^{\mathbf{0},\boldsymbol{z}_j} \circ \mathcal{D}_\theta^{\boldsymbol{z}_i,\mathbf{0}}, \tag{4}$$

and it jointly optimizes flow parameters θ with shape embeddings $\{\boldsymbol{z}_i\}$. To handle object deformations within unlimited categories, we explore learned and more flexible paths for multi-mode deformation discussed in the following section.

3.2. Multi-hub flow model

Figure 2 illustrates our framework. Our key novelty is a multi-hub flow network model framework that learns $\mathcal{D}$ together with flexible deformation paths. We redefine the deformation function as

$$\Phi_\theta^{i,j}(\boldsymbol{p};\boldsymbol{h}) = \mathcal{D}_\theta^{\boldsymbol{h},\boldsymbol{z}_j} \circ \mathcal{D}_\theta^{\boldsymbol{z}_i,\boldsymbol{h}}(\boldsymbol{p}), \tag{5}$$

to ensure that the deformation path connects $\boldsymbol{z}_i$ and $\boldsymbol{z}_j$ through a mutable latent hub $\boldsymbol{h}$. During training, we set $\boldsymbol{h}$ as learnable parameters and jointly optimize it with flow network parameters θ. As a result, we obtain an optimal hub location $\boldsymbol{h}$ in the latent space, which provides the optimal deformation mode among a set of shapes.

We identify diverse deformation modes among different categories of objects. For example, deformation of airplanes is usually via applying scale transform along horizontal plane-body axes, while chair deformation is usually along vertical axes. We aim to establish a network that handles deformations of universal objects with unknown and unlimited categories. Thus, we need to incorporate multiple deformation modes in our network. Fortunately, our design can be further extended to satisfy such a challenging requirement. We introduce multiple learnable hubs into our network, that is

$$\mathcal{H} = \{\boldsymbol{h}_1, \cdots, \boldsymbol{h}_M\}, \tag{6}$$

where $\boldsymbol{h}_i \in \mathbb{R}^d$ is a mutable hub location in a d-dimensional latent space, and is initialized using Gaussian distribution for training.

We aim to learn the deformation to ensure that the nearest hub between shapes indicates the optimal deformation mode. Accordingly, we define the final deformation function as

$$\Phi_\theta^{i,j}(\boldsymbol{p}) = \Phi_\theta^{i,j}(\boldsymbol{p};\boldsymbol{h}) \tag{7}$$

, where $\boldsymbol{h} = \underset{\boldsymbol{h}_k \in \mathcal{H}}{\arg\min} \|\boldsymbol{h}_k - \boldsymbol{z}_i\| + \|\boldsymbol{h}_k - \boldsymbol{z}_j\|$. Our training loss measures the geometry distance of two shapes taking their vertex sets V_1 and V_2 as input and measure the averaged squared distance in Equation 8, that is,

$$\begin{aligned}\mathcal{L}_{\mathrm{CD}}\left(V_1, V_2\right) = &\frac{1}{V_1}\sum_{x \in V_1} \min_{y \in V_2} \|x - y\|_2^2 \\ &+ \frac{1}{V_2}\sum_{y \in V_2} \min_{x \in V_1} \|y - x\|_2^2.\end{aligned} \tag{8}$$

We construct a latent space commonly shared by shapes and hubs by jointly optimizing latent codes of shapes in the repository with our deformation model. A shape with unknown category is given at inference time. We embed this shape into the the common latent space by solving its latent code $\boldsymbol{z}_p$ to minimize the deformation loss from existing shapes to it fixing other network parameters, that is,

$$\underset{\boldsymbol{z}_p}{\arg\min} \sum_{i \in \mathcal{S}} \mathcal{L}(\Phi_\theta^{i,p}(\mathcal{S}_i), \mathcal{S}_p) + \mathcal{L}(\Phi_\theta^{p,i}(\mathcal{S}_p), \mathcal{S}_i). \tag{9}$$

Notably, S_p can represent not only a CAD model but also a point cloud. In the latter case, we can use our network

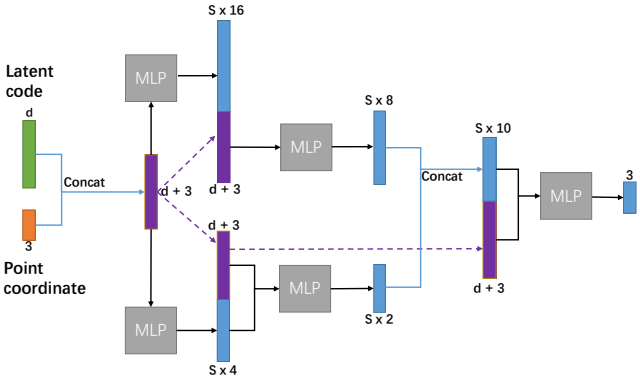

Figure 3. **The new decoder network structure of our framework.** Inputs are the d-dimensional latent code of each shape and the 3-dimensional point coordinate of each point in the shape, then we concatenate the two parts to get one input vector. The grey "MLP" means a fully-connected layer with an activation layer behind. The dimension of output in the blue rectangle of each network layer is different, and S is a number hyper-parameter about the width of network output, which can be adjusted.

to reconstruct the point cloud via retrieval and deformation. Specifically, we find the shape $\mathcal{S}_k$ in the repository that has a latent code $\boldsymbol{z}_k$ closest to $\boldsymbol{z}_p$, and reconstruct the point cloud by directly deforming the retrieved shape as $\Phi_\theta^{k,p}(\mathcal{S}_k)$.

3.3. Deformation flow backbone

Inspired by the structure of InceptionNet [45] and IM-Net [6], we propose a concise but efficient new decoder network for the deformation flow named DFF-Net. This network implements $\boldsymbol{h}_\theta$ in Equation 3, which takes inputs as a 3D point and a d-dimensional latent vector, and outputs 3D vector flow which can be integrated as a moving offset for this point. Inspired by the structure of InceptionNet [45] and IM-Net [6], we propose a concise but efficient new decoder network for the deformation flow named DFF-Net. This network implements $\boldsymbol{h}_\theta$ in Equation 3, which takes inputs as a 3 point and a d-dimensional latent vector, and outputs 3-dimensional vector flow which can be integrated as a moving offset for this point.

The network architecture is illustrated in Figure 3. Overall, we model it as a two-branch structure, each of which takes the concatenated point and latent code as input. For each branch, the input is passed through MLP layers twice where each output is concatenated with the original input. Such design borrows the idea of IM-Net [6] that concatenates point coordinates with shape features to improve deformation quality. Different from [6], our network is shallower but wider. As a result, it avoids the vanishing gradient problem for latent codes and thus generates more detail-preserving result. The difference between the two branches appears as different number of output feature dimensions after MLP layers, where the upper branch outputs features with 16 and 8 dimensions while the lower branch outputs features with 4 and 2 dimensions. This structure borrows the idea of InceptionNet [45] to improve feature diversity and capture shape features from different aspects.

Compared with IM-Net which is used in [6], our network reduces the parameters by 61.3%, the training time by 58.6% and the testing time by 41.5%. Apart from achieving significantly improved efficiency, we also improve the final deformation quality supported by ablation studies in Section 4.3.

4. Experiments

Our experiments are trained and evaluated on three categories of ShapeNet [2]: chair, car, and airplane. The dataset is split following its official guide. We train the model with batch size 64 on 2080Ti GPU for 200 epochs, and we set the number of hubs as 3 for all our experiments after try the number from 1 to 10. In this section, we first provide quantitative and qualitative results of our framework in terms of surface reconstruction from sparse point cloud, which is the most important application of our method. Then we provide experiment results to show that our method can improve the reconstruction results when trained with data of multiple classes and compared with ShapeFlow [20]. We also show the performances our method in shape embedding and retrieval, as well as shape deformation. We show the results of our ablation study to prove the effectiveness of our new backbone and our multi-hub method.

4.1. Shape reconstruction

We quantitatively and qualitatively compare the surface reconstruction results of our method against those of several classic and state-of-the-art baselines, including 3D-R2N2 [7], TMNet [40], BSP [5] and ShapeFlow [20]. We use a single class of ShapeNet data following its official data split guide to train R2N2, TMNet, and ShapeFlow. We directly use the pretrained model of BSP, which is trained on 13 categories of ShapeNet data, to obtain the result.

In terms of the quantitative results, we use the metrics of CD and Intersection over Union (IoU). For each shape, we randomly sample 512 points to calculate the CD. The results, which are demonstrated in Table 1, indicate that our model outperforms the others. Our method achieves the lowest CD in all the three categories, and has the highest IoU in the categories of car and airplane, as well as the mean IoU.

Figure 4 compares some of the reconstruction results between our method and the baseline methods. Among all the results, our method produces the most visually appealing, realistic results. As shown in the region with red frames, our model successfully captures the shape of the back of the chair(round back with thin connections to the mainframe), whereas other methods do not. Furthermore, the

	Ours		Shapeflow		TMNet		BSPNet		3D-R2N2	
category	CD ⇓	IoU ⇑	CD ⇓	IoU ⇑	CD ⇓	IoU ⇑	CD ⇓	IoU ⇑	CD ⇓	IoU ⇑
car	**0.02667**	**0.8431**	0.02923	0.7913	0.03542	0.6402	0.0715	0.5631	0.2001	0.7821
chair	**0.04132**	0.5479	0.05411	0.4611	0.04340	**0.5873**	0.1046	0.4969	0.2494	0.5120
airplane	**0.01837**	**0.7311**	0.06023	0.6513	0.02291	0.6760	0.1273	0.5083	0.2639	0.4185
mean	**0.02879**	**0.7074**	0.04786	0.6346	0.03391	0.6345	0.1011	0.5228	0.2378	0.5709

Table 1. **Quantitative 3D shape reconstruction evaluation results.** The best results are **boldfaced**. The second best results are underlined. (⇑ means a larger number, better performance, while ⇓ means smaller number, better performance.)

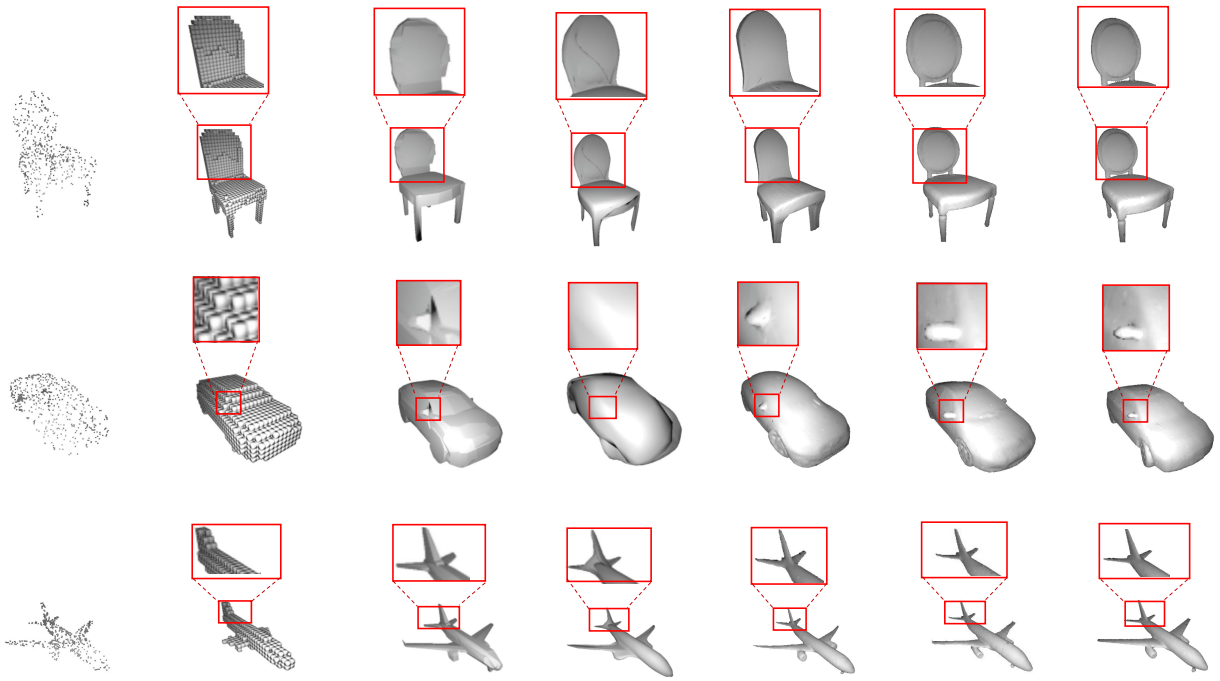

Figure 4. **Visualization of shape reconstruction from sparse point cloud.** We reconstruct from sparse point clouds and compare our methods with other approaches, including 3D-R2N2 [7], TMNet [40], BSPNet [5], Shapeflow [20].

comparison of the cars shows that our method can reconstruct the front of the cars(including the window, the bonnet, and the rearview) more realistically and with higher quality, whereas other methods cannot. So we get more competitive results in these shapes as a result of our proposed method.

Multi-class shape reconstruction We also conduct experiments to compare the performance of our method in surface reconstruction against that of ShapeFlow [20] when trained with data of multiple classes altogether. Multi-class shape reconstruction from sparse point cloud is very useful when the category of the input point cloud is unknown. We use data of all the three kinds as input, but at the inference time, we assume the input shape category is unknown. Our model will correctly embed the shape into its category and retrieve the best existing shape to deform.

Table 2 shows that, when trained in multiple categories altogether, our method outperforms Shapeflow both in CD and IoU measures in all tests. Despite the performance of our multi-class trained model cannot reach that of the single-class trained one, the gap between multi-class training and other single-class methods is small so we get a meaningful result.

4.2. Shape retrieval and shape deformation

We compare our method with some baselines on shape retrieval on chairs of ShapeNet, including AutoEncoder [1] and DAR [48]. As shown in Figure 5, the retrieval results of our method are closer to the ground truth than that of others given a complete shape as input. The result shows that our method can learn a reasonable latent space to enable retrieval of a shape that closely resembles the input.

	Ours(multi)		Shapeflow(multi)	
category	CD ⇓	IoU ⇑	CD ⇓	IoU ⇑
car	**0.03265**	**0.7466**	0.03650	0.6686
chair	**0.05123**	**0.4483**	0.06800	0.3581
airplane	**0.02458**	**0.5929**	0.03001	0.5272
mean	**0.03615**	**0.5959**	0.04484	0.5179

Table 2. **Quantitative multi-class shape reconstruction evaluation results.**

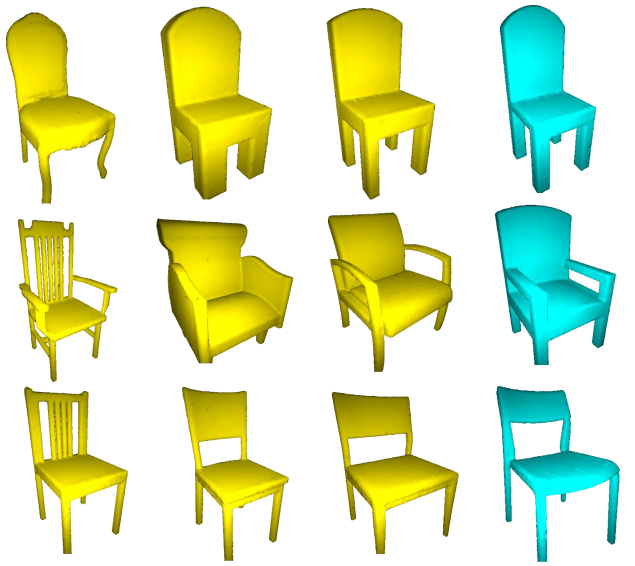

Figure 5. **Retrieval results of AutoEncoder [1], DAR [48] and our method.** The last column is the ground truth.

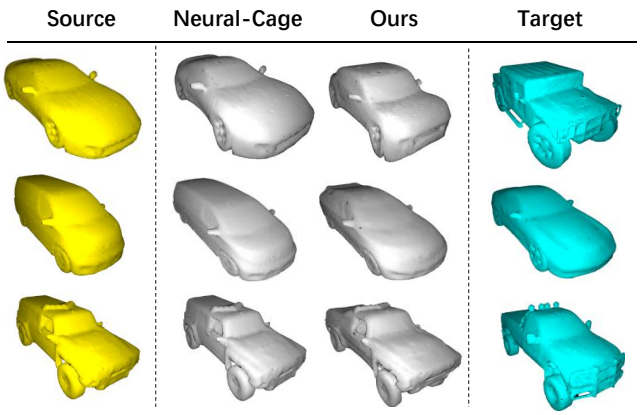

Figure 6. **Deformation comparison between Neural-Cage [58] and ours.**

We also test our model on the task of shape deformation, including both intra-class deformation and inter-class deformation. In terms of intra-class shape deformation, we compare the shape deformation performance of our model against those of Neural-Cage [58], which is the state-of-the-art learning-based shape deformation method. After our models and Neural-Cage are trained with the car data of ShapeNet, we input the source and the target shapes, which are different cars from ShapeNet. Then, we compare the deformation results. The results are illustrated in Figure 6. On the one hand, our method preserves the shape style and the detailed features of the source shapes well compared with Neural-Cage. On the other hand, our method can deform the global geometry shape of the source to that of the target, whereas Neural-Cage can only change their scales such as the length or width.

Our method has reached state-of-the-art performance not only at intra-class shape deformation itself, but also at the entire pipeline of shape embedding, retrieval and deformation. We compare the shape embedding, retrieval and deformation performance of our method against that of ShapeFlow [20], which is the state-of-the-art shape embedding, retrieval and deformation method. We use a single class of ShapeNet data to train both our model and ShapeFlow model. Then we input sparse point clouds from the same class, retrieve their most appropriate shapes, and deform them to the point cloud. We then measure the CD between

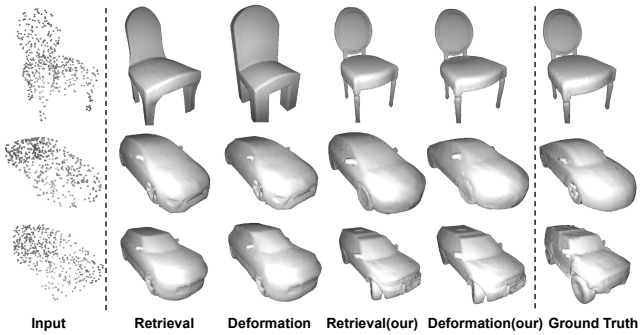

Figure 7. **Visualization of retrieval and deformation results.** Middle: The second and third coloums are retrieval and deformation results of method [20], and the last two column models before the ground truth column are retrieval and deformation results of ours.

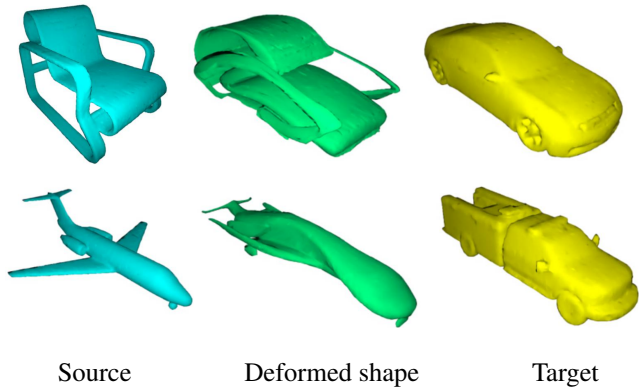

Figure 8. **Deformation between different classes in our latent space.**

the deformed results and the target shapes. The quantitative results are shown in Table 3, where our method reaches lower CD in every category. The qualitative results are further illustrated in Figure 7. The chair retrieved using our method highly resembles the ground truth, while that retrieved using ShapeFlow does not. With regard to the jeep, the 3D shape retrieved using our method is also a jeep, whereas that retrieved using ShapeFlow is a car. Although the cars retrieved using our method and ShapeFlow look alike, the deformed result of our method resembles the target car more than that of ShapeFlow.

	Ours	Shapeflow
car	**6.30**	8.10
chair	**12.66**	14.02
airplane	**2.19**	3.79

Table 3. **Quantitative comparison in shape deformation performance.** CD should be multiplied by 10^{-4}.

Furthermore, our method is also capable of deforming shapes between different model classes although the deformed shape is often strange. The shape may be strange because it preserves the shape style and detail features of the source model and maintain the global geometry shape of the target model. The results are shown in Figure 8. But it also shows that the limits of our deformation method, as we preserve the vertices' connection relationship while deforming a source to a target, we cannot endure huge topological changes.

4.3. Ablation study

We compare four scenarios in this part to analyze the effect of our two contributions: the multi-hub deformation flow method and our new backbone DFF-Net. The result is shown in Table 4. (1)"Multihub+DFF-Net" is our method, which uses multi-hub and DFF-Net as the backbone. (2)"Multihub+IM-Net" is the version that uses multi-hub and IM-Net [6] as the backbone. (3)"Zerohub+DFF-Net" is the version that uses zerohub, which is used in ShapeFlow [20], and DFF-Net as the backbone. (4)"Zerohub+IM-Net" is the version that uses zerohub and IM-Net as the backbone. The four frameworks are trained in the car category of ShapeNet [2] data. Then they are used in the retrieval and deformation task, in which we use CD as the metric. The reason we only compare these two networks is that they are the only networks that can be used for our unique flow model. The result shows that our original method in scenario(4) obtains the lowest CD. This result indicates that both our multi-hub method and BIM-Net, which is our new backbone, contribute to the performance of our method.

We compare the numbers of parameters of DFF-Net and IM-Net and the time of training and testing the models to further quantitatively prove the advantages of our proposed new network structure DFF-Net compared with the previous network method. The comparison result in Figure 9 shows that DFF-Net is more effective and lightweight than previous proposed network IM-Net. DFF-Net reduces the number of model parameters by 61.3%, the length of training time by 58.6% and the testing time by 41.5%, which are notable improvements compared with former useful network structure IM-Net.

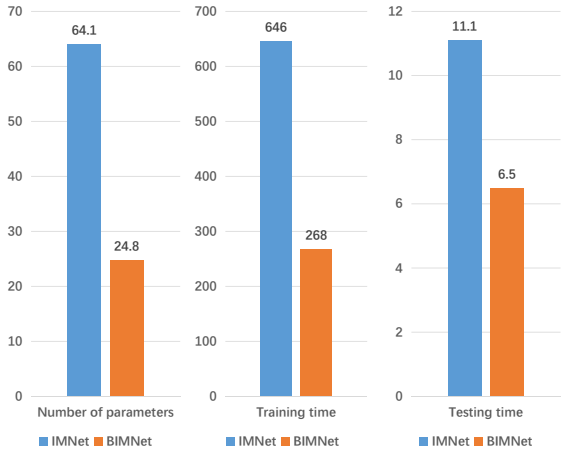

Figure 9. **Comparison between IM-Net and our proposed network DFF-Net.** It includes number of network parameters, training time and testing time. The unit of the number of network model parameters is ten thousand, and training and test time are measured in seconds.

Method	CD ⇓
Zerohub+IM-Net	8.10
Zerohub+DFF-Net	6.68
Multihub+IM-Net	7.38
Multihub+DFF-Net	**6.30**

Table 4. **Ablation study about hubs and DFF-Net on car deformation.** CD should be multiplied by 10^{-4}.

5. Conclusion

We propose a versatile multi-hub flow deformation framework with a new backbone to learn a multi-class shape deformation space for better embedding, retrieval and deformation. Our new backbone DFF-Net can capture more diverse shape features and contribute to better qualitative or quantitative results in shape deformation or reconstruction experiments. Our method can restore the full mesh model from an unknown category of sparse point cloud input. We demonstrate different application scenarios of our framework such as shape reconstruction, shape retrieval and shape deformation.

References

[1] P. Achlioptas, O. Diamanti, I. Mitliagkas, and L. Guibas. Learning representations and generative models for 3d point clouds. In *International conference on machine learning*, pages 40–49. PMLR, 2018. 1, 6, 7

[2] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 1, 2, 5, 8

[3] M. Chen, C. Wang, and L. Liu. Cross-domain retrieving sketch and shape using cycle cnns. *Computers & Graphics*, 89:50–58, 2020. 3

[4] R. T. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud. Neural ordinary differential equations. In *Advances in neural information processing systems*, pages 6571–6583, 2018. 3

[5] Z. Chen, A. Tagliasacchi, and H. Zhang. Bsp-net: Generating compact meshes via binary space partitioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 45–54, 2020. 5, 6

[6] Z. Chen and H. Zhang. Learning implicit fields for generative shape modeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5939–5948, 2019. 2, 5, 8

[7] C. B. Choy, D. Xu, J. Gwak, K. Chen, and S. Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *European conference on computer vision*, pages 628–644. Springer, 2016. 5, 6

[8] M. Dahnert, A. Dai, L. J. Guibas, and M. Niessner. Joint embedding of 3d scan and cad objects. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8749–8758, 2019. 3

[9] A. Dai, C. Ruizhongtai Qi, and M. Nießner. Shape completion using 3d-encoder-predictor cnns and shape synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5868–5877, 2017. 1

[10] N. De Cao, W. Aziz, and I. Titov. Block neural autoregressive flow. In *Uncertainty in Artificial Intelligence*, pages 1263–1273. PMLR, 2020. 3

[11] L. Dinh, J. Sohl-Dickstein, and S. Bengio. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016. 3

[12] L. Gao, J. Yang, T. Wu, Y.-J. Yuan, H. Fu, Y.-K. Lai, and H. Zhang. Sdm-net: Deep generative network for structured deformable mesh. *ACM Transactions on Graphics (TOG)*, 38(6):1–15, 2019. 2

[13] W. Grathwohl, R. T. Chen, J. Bettencourt, I. Sutskever, and D. Duvenaud. Ffjord: Free-form continuous dynamics for scalable reversible generative models. In *International Conference on Learning Representations*, 2018. 3

[14] T. Groueix, M. Fisher, V. G. Kim, B. C. Russell, and M. Aubry. 3d-coded: 3d correspondences by deep deformation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 230–246, 2018. 2

[15] R. Hanocka, N. Fish, Z. Wang, R. Giryes, S. Fleishman, and D. Cohen-Or. Alignet: Partial-shape agnostic alignment via unsupervised learning. *ACM Transactions on Graphics (TOG)*, 38(1):1–14, 2018. 2

[16] C.-W. Huang, D. Krueger, A. Lacoste, and A. Courville. Neural autoregressive flows. In *International Conference on Machine Learning*, pages 2078–2087, 2018. 3

[17] T. Igarashi, T. Moscovich, and J. F. Hughes. As-rigid-as-possible shape manipulation. *ACM transactions on Graphics (TOG)*, 24(3):1134–1141, 2005. 2

[18] V. Ishimtsev, A. Bokhovkin, A. Artemov, S. Ignatyev, M. Niessner, D. Zorin, and E. Burnaev. Cad-deform: Deformable fitting of cad models to 3d scans. *arXiv preprint arXiv:2007.11965*, 2020. 2

[19] D. Jack, J. K. Pontes, S. Sridharan, C. Fookes, S. Shirazi, F. Maire, and A. Eriksson. Learning free-form deformations for 3d object reconstruction. In *Asian Conference on Computer Vision*, pages 317–333. Springer, 2018. 2

[20] C. Jiang, J. Huang, A. Tagliasacchi, and L. J. Guibas. Shapeflow: Learnable deformation flows among 3d shapes. *Advances in Neural Information Processing Systems*, 33, 2020. 1, 2, 3, 4, 5, 6, 7, 8

[21] Z.-H. Jiang, Q. Wu, K. Chen, and J. Zhang. Disentangled representation learning for 3d face shape. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11957–11966, 2019. 2

[22] A. Jin, Q. Fu, and Z. Deng. Contour-based 3d modeling through joint embedding of shapes and contours. In *Symposium on Interactive 3D Graphics and Games*, pages 1–10, 2020. 3

[23] P. Joshi, M. Meyer, T. DeRose, B. Green, and T. Sanocki. Harmonic coordinates for character articulation. *ACM Transactions on Graphics (TOG)*, 26(3):71–es, 2007. 2

[24] M. Kazhdan and H. Hoppe. Screened poisson surface reconstruction. *ACM Transactions on Graphics (ToG)*, 32(3):1–13, 2013. 1

[25] D. P. Kingma, T. Salimans, R. Jozefowicz, X. Chen, I. Sutskever, and M. Welling. Improved variational inference with inverse autoregressive flow. In *Advances in neural information processing systems*, pages 4743–4751, 2016. 3

[26] W. Kuo, A. Angelova, T.-Y. Lin, and A. Dai. Mask2cad: 3d shape prediction by learning to segment and retrieve. In *European Conference on Computer Vision (ECCV)*, volume 1, page 3. Springer, 2020. 3

[27] A. Kurenkov, J. Ji, A. Garg, V. Mehta, J. Gwak, C. Choy, and S. Savarese. Deformnet: Free-form deformation network for 3d shape reconstruction from a single image. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 858–866. IEEE, 2018. 2

[28] D.-T. Lee and B. J. Schachter. Two algorithms for constructing a delaunay triangulation. *International Journal of Computer & Information Sciences*, 9(3):219–242, 1980. 1

[29] T. Lee, Y.-L. Lin, H. Chiang, M.-W. Chiu, W. Hsu, and P. Huang. Cross-domain image-based 3d shape retrieval by view sequence learning. In *2018 International Conference on 3D Vision (3DV)*, pages 258–266. IEEE, 2018. 3

[30] H. Li, R. W. Sumner, and M. Pauly. Global correspondence optimization for non-rigid registration of depth scans. In *Computer graphics forum*, volume 27, pages 1421–1430. Wiley Online Library, 2008. 2

[31] Y. Li, H. Su, C. R. Qi, N. Fish, D. Cohen-Or, and L. J. Guibas. Joint embeddings of shapes and images via cnn image purification. *ACM transactions on graphics (TOG)*, 34(6):1–12, 2015. 3

[32] Y. Lipman, D. Levin, and D. Cohen-Or. Green coordinates. *ACM Transactions on Graphics (TOG)*, 27(3):1–10, 2008. 2

[33] Y. Lipman, O. Sorkine, D. Cohen-Or, D. Levin, C. Rossi, and H.-P. Seidel. Differential coordinates for interactive mesh editing. In *Proceedings Shape Modeling Applications, 2004.*, pages 181–190. IEEE, 2004. 2

[34] M. Liu, K. Zhang, J. Zhu, J. Wang, J. Guo, and Y. Guo. Data-driven indoor scene modeling from a single color image with iterative object segmentation and model retrieval. *IEEE transactions on visualization and computer graphics*, 26(4):1702–1715, 2018. 3

[35] E. Mehr, A. Jourdan, N. Thome, M. Cord, and V. Guitteny. Disconet: Shapes learning on disconnected manifolds for 3d editing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3474–3483, 2019. 2

[36] L. Nan, K. Xie, and A. Sharf. A search-classify approach for cluttered indoor scene understanding. *ACM Transactions on Graphics (TOG)*, 31(6):1–10, 2012. 3

[37] M. Niemeyer, L. Mescheder, M. Oechsle, and A. Geiger. Occupancy flow: 4d reconstruction by learning particle dynamics. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5379–5389, 2019. 2, 3

[38] C. Niu, J. Li, and K. Xu. Im2struct: Recovering 3d shape structure from a single rgb image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4521–4529, 2018. 1

[39] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016. 3

[40] J. Pan, X. Han, W. Chen, J. Tang, and K. Jia. Deep mesh reconstruction from single rgb images via topology modification networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9964–9973, 2019. 5, 6

[41] G. Papamakarios, T. Pavlakou, and I. Murray. Masked autoregressive flow for density estimation. In *Advances in Neural Information Processing Systems*, pages 2338–2347, 2017. 3

[42] D. J. Rezende and S. Mohamed. Variational inference with normalizing flows. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning-Volume 37*, pages 1530–1538, 2015. 3

[43] Y. Sahillioğlu and Y. Yemez. Coarse-to-fine combinatorial matching for dense isometric shape correspondence. In *Computer Graphics Forum*, volume 30, pages 1461–1470. Wiley Online Library, 2011. 2

[44] O. Sorkine and M. Alexa. As-rigid-as-possible surface modeling. In *Symposium on Geometry processing*, volume 4, pages 109–116, 2007. 2

[45] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015. 2, 5

[46] H. Tabia and H. Laga. Learning shape retrieval from different modalities. *Neurocomputing*, 253(C):24–33, 2017. 3

[47] M. Tatarchenko, S. R. Richter, R. Ranftl, Z. Li, V. Koltun, and T. Brox. What do single-view 3d reconstruction networks learn? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3405–3414, 2019. 3

[48] M. A. Uy, J. Huang, M. Sung, T. Birdal, and L. Guibas. Deformation-aware 3d model embedding and retrieval. In *European Conference on Computer Vision*, pages 397–413. Springer, 2020. 1, 2, 3, 6, 7

[49] R. Van Den Berg, L. Hasenclever, J. M. Tomczak, and M. Welling. Sylvester normalizing flows for variational inference. In *34th Conference on Uncertainty in Artificial Intelligence 2018, UAI 2018*, pages 393–402. Association For Uncertainty in Artificial Intelligence (AUAI), 2018. 3

[50] A. Van den Oord, N. Kalchbrenner, L. Espeholt, O. Vinyals, A. Graves, et al. Conditional image generation with pixelcnn decoders. In *Advances in neural information processing systems*, pages 4790–4798, 2016. 3

[51] N. Wang, Y. Zhang, Z. Li, Y. Fu, W. Liu, and Y.-G. Jiang. Pixel2mesh: Generating 3d mesh models from single rgb images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 52–67, 2018. 2

[52] W. Wang, D. Ceylan, R. Mech, and U. Neumann. 3dn: 3d deformation network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1038–1046, 2019. 2

[53] Y. Wang, N. Aigerman, V. G. Kim, S. Chaudhuri, and O. Sorkine-Hornung. Neural cages for detail-preserving 3d deformations. In *CVPR*, 2020. 2

[54] Z. Wu, X. Wang, D. Lin, D. Lischinski, D. Cohen-Or, and H. Huang. Sagnet: Structure-aware generative network for 3d-shape modeling. *ACM Transactions on Graphics (TOG)*, 38(4):1–14, 2019. 3

[55] Z. Wu, Y. Zhang, M. Zeng, F. Qin, and Y. Wang. Joint analysis of shapes and images via deep domain adaptation. *Computers & Graphics*, 70:140–147, 2018. 3

[56] G. Yang, X. Huang, Z. Hao, M.-Y. Liu, S. Belongie, and B. Hariharan. Pointflow: 3d point cloud generation with continuous normalizing flows. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4541–4550, 2019. 3

[57] J. Yang, L. Gao, Y.-K. Lai, P. L. Rosin, and S. Xia. Biharmonic deformation transfer with automatic key point selection. *Graphical Models*, 98:1–13, 2018. 2

[58] W. Yifan, N. Aigerman, V. G. Kim, S. Chaudhuri, and O. Sorkine-Hornung. Neural cages for detail-preserving 3d deformations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 75–83, 2020. 2, 7

[59] K. Yin, Z. Chen, H. Huang, D. Cohen-Or, and H. Zhang. Logan: Unpaired shape transform in latent overcomplete space. *ACM Transactions on Graphics (TOG)*, 38(6):1–13, 2019. 2

[60] M. E. Yumer and N. J. Mitra. Learning semantic deformation flows with 3d convolutional networks. In *European Conference on Computer Vision*, pages 294–311. Springer, 2016. 2

[61] K. Zhou, W. Xu, Y. Tong, and M. Desbrun. Deformation transfer to multi-component objects. In *Computer Graphics Forum*, volume 29, pages 319–325. Wiley Online Library, 2010. 2