

# Learning Local Neighboring Structure for Robust 3D Shape Representation

Zhongpai Gao<sup>1</sup>, Junchi Yan<sup>1</sup>, Guangtao Zhai<sup>1,2\*</sup>, Juyong Zhang<sup>3</sup>, Yiyang Yang<sup>1</sup>, Xiaokang Yang<sup>1</sup>

<sup>1</sup> MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University

<sup>2</sup> Institute of Image Communication and Network Engineering, Shanghai Jiao Tong University

<sup>3</sup> School of Mathematical Sciences, University of Science and Technology of China

{gaozhongpai, yanjunchi, zhaiguangtao}@sjtu.edu.cn, juyong@ustc.edu.cn, {jamson, xkyang}@sjtu.edu.cn

## Abstract

Mesh is a powerful data structure for 3D shapes. Representation learning for 3D meshes is important in many computer vision and graphics applications. The recent success of convolutional neural networks (CNNs) for structured data (e.g., images) suggests the value of adapting insight from CNN for 3D shapes. However, 3D shape data are irregular since each node's neighbors are unordered. Various graph neural networks for 3D shapes have been developed with isotropic filters or predefined local coordinate systems to overcome the node inconsistency on graphs. However, isotropic filters or predefined local coordinate systems limit the representation power. In this paper, we propose a local structure-aware anisotropic convolutional operation (LSA-Conv) that learns adaptive weighting matrices for each node according to the local neighboring structure and performs shared anisotropic filters. In fact, the learnable weighting matrix is similar to the attention matrix in the random synthesizer – a new Transformer model for natural language processing (NLP). Comprehensive experiments demonstrate that our model produces significant improvement in 3D shape reconstruction compared to state-of-the-art methods.

## Introduction

Representation learning for 3D meshes is crucial for many 3D tasks, e.g., reconstruction (Genova et al. 2018; Tran and Liu 2018; Gao et al. 2020), shape correspondence (Groueix et al. 2018), shape synthesis and modeling (Cao et al. 2014; Xie et al. 2018), face recognition (Liu et al. 2018) and shape segmentation (Donlic et al. 2017; Kalogerakis, Hertzmann, and Singh 2010), and graphics applications such as virtual avatar (Cao, Hou, and Zhou 2014). Inspired by the great success of convolutional neural networks (CNN) in the fields of natural language processing (1-D), image classification (2-D), and radiographic image analysis (3-D) where underlying data are Euclidean structured, deep neural networks on 3D meshes have recently driven significant interests. Directly applying CNN on 3D meshes is a challenge since they are non-Euclidean structured and are usually represented as graphs in which the number and orientation of each node's neighbors vary from one to another (node inconsistency). An

effective definition of convolutional operation analogous to that on Euclidean structured data is important for 3D shape representation learning.

Recently, many graph convolutional networks have been developed to handle irregular graph data and achieved promising results. Defferrard, Bresson, and Vandergheynst (2016) designed fast localized convolutional filters on graphs using Chebyshev expansion based on spectral graph theory, called *ChebNet*. The spectral filters are isotropic to overcome the irregularity of graph data. ChebNet is an efficient generalization of CNNs to graphs. Ranjan et al. (2018) built convolutional mesh autoencoder (COMA) for 3D meshes with fixed topology upon ChebNet and introduced mesh sampling operations that enable a hierarchical representation to capture non-linear variations of human faces. However, compared to CNN, isotropic filters used in ChebNet limit the representation power.

In order to introduce anisotropic filters on graph convolutions, Bouritsas et al. (2019) formulated a spiral convolution operator (SpiralNet) that defines an explicit order of the neighbors via a spiral scan for each vertex on 3D meshes with fixed topology. However, serializing the local neighbors of vertices by following a spiral cannot resolve the inconsistency between different nodes. Furthermore, this method requires manually assigning a starting point to determine the order of neighbors, which is difficult to make the local coordinate system consistent across meshes. The selection of the starting point may affect the performance of the spiral convolution operator. Explicitly defining the order of neighbors cannot efficiently exploit the irregular structure of graphs.

PointCNN (Li et al. 2018b) proposed to learn an  $\mathcal{X}$ -transformation from the input points to weight and permute each point's neighbors into a latent and potentially canonical order. KPConv (Thomas et al. 2019) presented a convolutional operation that weights each point's neighbors depending on the Euclidean distances to a set of predefined or deformable kernel points. Subsequently, these methods apply anisotropic filters on the resampled neighbors to extract features for point clouds. However, different from 3D meshes, point clouds do not have a fixed topology and the neighbors are obtained through a  $K$ -nearest neighbors (KNN) algorithm. The methods designed for point clouds do not consider the unique characteristics of 3D meshes, which limits the representation power for 3D meshes.

\*Corresponding author.

This paper proposes a novel local structure-aware anisotropic convolutional operation (LSA-Conv) for 3D meshes. Consider that 3D meshes are irregular and share the same topology of a template, e.g., 3D morphable models (3DMM) (Paysan et al. 2009; Cao et al. 2014; Loper et al. 2015; Romero, Tzionas, and Black 2017), we directly learn a weighting matrix for each vertex of the template to soft-permute the vertex’s neighbors. The weighting matrices are trained along with the whole network. The idea of learnable weighting matrix is by analogy with the random synthesizer (Tay et al. 2020), which is a new Transformer model for natural language processing (NLP). Then similar to CNNs, we apply shared anisotropic filters on the resampled neighbors to extract local features on 3D meshes. LSA-Conv is designed to adapt each vertex’s local structure without explicitly defining the order or any local pseudo-coordinate systems for each vertex.

LSA-Conv is easy to implement and integrate into existing deep learning models to improve their performance. In line with COMA (Ranjan et al. 2018), SpiralNet (Bouritsas et al. 2019), and SpiralNet++ (Gong et al. 2019) that are all designed for meshes with fixed topologies, we evaluate our approach on the reconstruction task which has been a fundamental testbed for further applications. Fixed topology is a common and practical setting for meshes in face, hand, and body related applications (Gao et al. 2020; Jiang et al. 2019). Note that, meshes with arbitrary topologies can directly be handled by point-cloud based methods (Verma, Boyer, and Verbeek 2018; Li et al. 2018b). The proposed method is complementary to these point-cloud works and makes its unique value for fixed-topology based applications. We use LSA-Conv to build convolutional mesh autoencoder and achieve state-of-the-art performance on two 3D shape datasets: human faces (COMA (Ranjan et al. 2018)) and human bodies (DFAUST (Bogo et al. 2017)). Comprehensive evaluation experiments show that the proposed method significantly outperforms existing models.

The contributions of this paper are summarized in below:

- 1) Taking advantage of the readily available fixed-topology information of mesh data, we propose a local structure-aware anisotropic convolutional operation (LSA-Conv) for representation learning from 3D meshes. LSA-Conv learns a weighting matrix for each vertex to soft-permute its neighbors based on the local neighboring structure as derived from the object-level topology. The learnable weighting matrix is similar to the attention matrix in random synthesizer (Tay et al. 2020) for NLP. To our knowledge, this is the first work for learning local arrangement by utilizing the fixed-topology, in contrast to the non-adaptive methods either using isotropic filters (Ranjan et al. 2018) or predefined local coordinates (Bouritsas et al. 2019).

- 2) Unlike (Bouritsas et al. 2019) that needs to define the neighboring order explicitly, LSA-Conv needs no pre/post processing steps. Thus, LSA-Conv is orthogonal to other techniques for 3D meshes and can be readily integrated into existing pipelines (Ranjan et al. 2018; Gao et al. 2020) for 3D shape processing, by replacing the conv layer.

- 3) Extensive experiments show that our model significantly outperforms state-of-the-art methods for 3D

shape generations. Two applications demonstrates the effectiveness of the proposed method. The source code has been formally released at: <https://github.com/Gaozhongpai/PaiConvMesh>.

## Related Work

**Linear 3D morphable models** 3D morphable models (3DMM) are statistical models of 3D shapes, such as human faces, bodies, hands, etc., and are constructed by performing some form of dimensionality reduction on a training set that each mesh is in dense correspondence with each other (i.e., fixed topology). 3DMMs are powerful priors on 3D shape reconstruction or generation. Blanz and Vetter (1999a) proposed the first linear parametric 3DMM using principal component analysis (PCA) to model the shape and texture of 3D faces. The widely used 3DMM for faces (Zhu et al. 2015) was built by merging Basel Face Model (BFM) (Paysan et al. 2009) with 200 subjects in neutral expressions and FaceWarehouse (Cao et al. 2014) with 150 subjects in 20 different expressions. Skinned multi-person linear model (SMPL) (Loper et al. 2015) is the most well known body model as learned through PCA and represents a wide variety of body shapes in natural human poses. MANO (Romero, Tzionas, and Black 2017) is a hand model learned from around 1000 high-resolution 3D scans of human hands in a wide variety of hand poses. Those PCA-based models are commonly used for 3D faces, bodies, and hands reconstruction. In this paper, we introduce a non-linear 3DMM for 3D shapes with much higher representation power.

**Graph neural networks** The popularity of extending deep learning approaches for graph data has been rapidly growing in recent years. Convolutional graph neural networks fall into two categories: spectral-based and spatial-based. Spectral-based approaches define convolutional operation based on graph signal processing. Spectral CNN (Bruna et al. 2014) generalizes convolution to graphs via Laplacian eigenvectors. ChebNet (Defferrard, Bresson, and Vandergheynst 2016) and GCN (Kipf and Welling 2017) reduce the computation complexity of eigen-decomposition by using fast localized convolutional filters. AGCN (Li et al. 2018a) learns hidden structural relations unspecified by the graph adjacency matrix. Spatial-based approaches define graph convolutions based on a node’s spatial relations. GraphSage (Hamilton, Ying, and Leskovec 2017) samples a fixed number of neighbors and aggregates neighboring features for each node. GAT (Veličković et al. 2018) adopts attention mechanisms to learn the relative weights between two connected nodes. MoNet (Monti et al. 2017) introduces node pseudo-coordinates to determine the relative position between a node and its neighbors and assigns different weights to the neighbors. FeaStNet (Verma, Boyer, and Verbeek 2018) proposes a graph-convolution operator that learns a weighting matrix dynamically computed from features, which is similar to PointCNN (Li et al. 2018b).

**Point neural networks** Several point neural networks are related to our work. PointCNN (Li et al. 2018b) presents a method to learn an  $\mathcal{X}$ -transformation as a function of input points. RandLA-Net (Hu et al. 2020) uses an attention mechanism to learn local features instead of applying filters

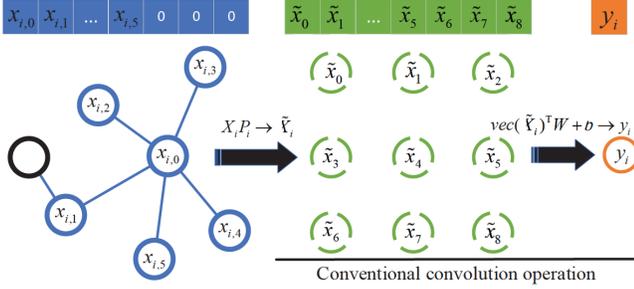


Figure 1: Local structure-aware anisotropic convolutional operation (LSA-Conv).  $\{\mathbf{x}_{i,0}, \mathbf{x}_{i,1}, \dots, \mathbf{x}_{i,5}\} \subseteq \mathcal{N}_i$  are the one-ring neighborhood of  $\mathbf{x}_i$  (including itself, i.e.,  $\mathbf{x}_{i,0}$ ). We construct  $\tilde{\mathbf{X}}_i = \{\mathbf{x}_{i,0}, \mathbf{x}_{i,1}, \dots, \mathbf{x}_{i,K-1}\} \in \mathbb{R}^{D_{in} \times K}$  where  $K$  is a predefined neighbor size. LSA-Conv contains: i) a learnable weighting matrix,  $\mathbf{P}_i \in \mathbb{R}^{K \times K}$  is used to soft-permute the node’s neighbors; ii) a conventional convolution operation with anisotropic filters,  $\mathbf{W} \in \mathbb{R}^{(D_{in} \cdot K) \times D_{out}}$ , is performed with bias  $\mathbf{b} \in \mathbb{R}^{D_{out}}$ .

for convolutional operation. KPConv (Thomas et al. 2019) proposes a convolution that takes radius neighbors as input and processes them with weights spatially located by a set of kernel points. Instead of calculating a weighting matrix as a function of inputs in FeaStNet (Verma, Boyer, and Verbeek 2018), PointCNN (Li et al. 2018b), and KPConv (Thomas et al. 2019), our LSA-Conv directly learns a weighting matrix for each vertex thanks to the fixed topology of meshes which otherwise is unavailable in cloud data.

## Approach

Aiming at entailing the GCNs with the anisotropic filtering ability like CNNs to improve its expressiveness, we propose a local structure-aware anisotropic convolutional operation (LSA-Conv) on graphs. Instead of directly applying anisotropic filters on each node’s neighbors, we soft-permute each node’s neighbors using a weighting matrix that is trained along with the deep neural networks.

### Local Structure-aware Anisotropic Convolution

Consider a 3D shape that is described as a mesh  $M = (\mathcal{V}, \mathcal{E})$ , where  $\mathcal{V} = \{1, \dots, N\}$  is a set of vertices and  $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$  is a set of edges. A graph may have node attributes  $\mathbf{X} \in \mathbb{R}^{D \times N}$ , where  $D$  and  $N$  represents the feature dimension and number of nodes, respectively. In the simplest setting of  $D = 3$ , each node contains 3D coordinates  $\mathbf{x}_i = [x_i, y_i, z_i]^\top$  in the Euclidean space. The node attributes can also include additional coordinates such as color and vertex normal. In a deep neural network, the output of each layer is as the input for the subsequent layer. Thus, generally,  $D$  represents the feature dimension of a given layer in the deep neural network.

We define LSA-Conv as follows. For each node,  $\mathcal{N}_i$  is a set of the one-ring neighborhood of  $\mathbf{x}_i$  (including itself, also as  $\mathbf{x}_{i,0}$ ), where  $\mathbf{x}_j \in \mathcal{N}_i$  and  $(\mathbf{x}_{i,0}, \mathbf{x}_{i,j}) \in \mathcal{E}$ . We denote  $\mathcal{N}_i = \{\mathbf{x}_{i,0}, \mathbf{x}_{i,1}, \dots, \mathbf{x}_{i,|\mathcal{N}_i|-1}\}$ , where  $|\mathcal{N}_i|$  is the number of node’s neighbors and it varies from one

node to another in a graph. In order to apply a shared anisotropic filter on each node, we define a constant neighbor size  $K$ , which corresponds to the kernel size in conventional convolution operation. For each node, we construct  $\tilde{\mathbf{X}}_i = \{\mathbf{x}_{i,0}, \mathbf{x}_{i,1}, \dots, \mathbf{x}_{i,K-1}\} \in \mathbb{R}^{D_{in} \times K}$ , where the first  $K$  neighbors are selected if  $K$  is smaller than or equal to  $|\mathcal{N}_i|$ ; otherwise zero-padding is applied, as shown in Figure 1. Note that, we put the node itself,  $\mathbf{x}_i$ , in the first place and the order of other neighbors  $\{\mathbf{x}_{i,1}, \dots, \mathbf{x}_{i,K-1}\}$  in  $\tilde{\mathbf{X}}_i$  is random and not specified.

Since the order and orientation of neighbors for each node vary from one to another, directly applying an anisotropic filter on unordered neighbors diminishes the representation power. While training, the anisotropic filter may struggle to adapt to the large variability of the unordered coordinate systems and tend to become an isotropic (rotational invariance) filter. In this paper, we introduce an adaptive weighting matrix to soft-permute each node’s neighbors, denoted as  $\mathbf{P}_i \in \mathbb{R}^{K \times K}$ . The resampled convolutional neighbors of each node can be obtained by

$$\tilde{\tilde{\mathbf{X}}}_i = \tilde{\mathbf{X}}_i \mathbf{P}_i, \quad (1)$$

where  $\tilde{\tilde{\mathbf{X}}}_i \in \mathbb{R}^{D_{in} \times K}$  and  $\mathbf{P}_i$  is a trainable parameter to be adaptive according to the geometric structure of the node’s neighbors.

The learned weighting matrix for each node can soft-permute the node’s neighbors into an implicit canonical order such that we are able to apply a shared anisotropic filter on each node of a graph. This operation is the same as the conventional convolution and can be expressed as

$$\mathbf{y}_i = \text{vec}(\tilde{\tilde{\mathbf{X}}}_i)^\top \mathbf{W} + \mathbf{b}, \quad (2)$$

where  $\mathbf{W} \in \mathbb{R}^{(D_{in} \cdot K) \times D_{out}}$  includes  $D_{out}$  anisotropic filters,  $\mathbf{b} \in \mathbb{R}^{D_{out}}$  is the bias,  $\mathbf{y}_i \in \mathbb{R}^{D_{out}}$  is the output feature node corresponding to the input node  $\mathbf{x}_i \in \mathbb{R}^{D_{in}}$ , and  $\text{vec}(\cdot)$  is a vectorization function which converts a matrix into a column vector. To introduce non-linearity, an activation function  $f(\cdot)$  such as ELU (Clevert, Unterthiner, and Hochreiter 2015) is introduced on Eq. (1) and (2). Thus, LSA-Conv is defined as

$$\mathbf{y}_i = f(\text{vec}(f(\tilde{\tilde{\mathbf{X}}}_i \mathbf{P}_i))^\top \mathbf{W} + \mathbf{b}). \quad (3)$$

Note, for meshes with fixed topology, each node corresponds to a weighting matrix and all nodes of the whole graph share the same anisotropic filter for each output channel.

### LSA-Conv based 3D Morphable Models

We propose a nonlinear 3D morphable model using our LSA-Conv as building blocks, called *LSA-3DMM*. The basic architecture of LSA-3DMM is the same as COMA (Ranjan et al. 2018) and Spiral (Bouritsas et al. 2019) for a fair comparison, as shown in Figure 2. The mesh sampling operations are adopted from (Ranjan et al. 2018). We simply replace the convolutional operations (i.e., ChebNet or SpiralNet) with LSA-Conv. LSA-3DMM is a deep convolutional mesh autoencoder with hierarchical mesh representations and is able to capture nonlinear variations in 3D shapes at multiple scales within the model.

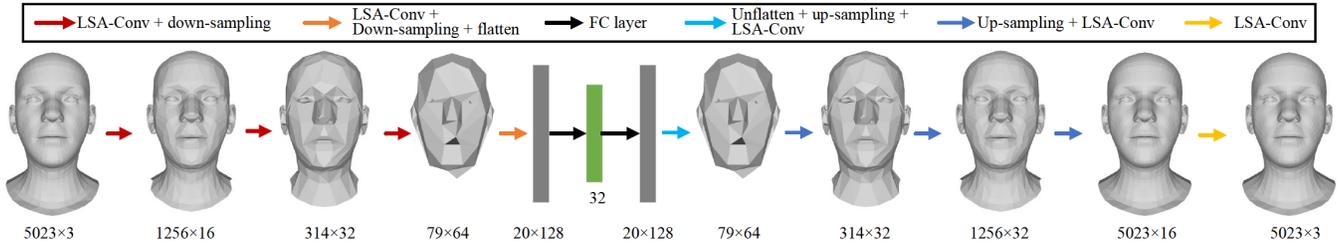


Figure 2: Architecture of our LSA-Conv based 3D morphable models (LSA-3DMM).

We denote as  $FC(\cdot)$  a fully connected layer,  $d$  the dimension of latent vector,  $l$  the number of vertices after the last down-sampling layer,  $PC(k, c)$  a LSA-Conv layer with neighbor size  $k$  and number of filters  $c$ ,  $DS(p)$  and  $US(p)$  are a down-sampling and a up-sampling layer by a factor of  $p$ , respectively. LSA-3DMM is listed as follows:

$$\begin{aligned}
 enc : & PC(9, 16) \rightarrow DS(4) \rightarrow PC(9, 32) \rightarrow DS(4) \rightarrow \\
 & PC(9, 64) \rightarrow DS(4) \rightarrow PC(9, 128) \rightarrow DS(4) \rightarrow FC(d), \\
 dec : & FC(l * 128) \rightarrow US(4) \rightarrow PC(9, 64) \rightarrow US(4) \rightarrow \\
 & PC(9, 32) \rightarrow US(4) \rightarrow PC(9, 32) \rightarrow US(4) \rightarrow \\
 & PC(9, 16) \rightarrow PC(9, 3).
 \end{aligned}$$

The model’s encoder effectively compresses a 3D shape into a low dimensional latent vector (e.g.,  $d = 32$  in Figure 2) and the decoder reconstructs the 3D shape from the latent vector. LSA-3DMM can be used in 3D shape recognition, reconstruction, and many other applications.

## Evaluation and Discussion

In this section, we first evaluate the proposed model on two different 3D shape datasets by comparing to state-of-the-art approaches. Then, a parameter reduction method is proposed for LSA-Conv. At last, ablation tests are conducted to demonstrate the effectiveness of LSA-Conv.

**Datasets** We evaluate our model on two datasets: COMA (Ranjan et al. 2018) and DFAUST (Bogo et al. 2017). COMA is a human facial dataset that consists of 12 classes of extreme expressions from 12 different subjects. The dataset contains 20,466 3D meshes that were registered to a common reference template with 5023 vertices. DFAUST is a human body dataset that collects over 40,000 real meshes, capturing 129 dynamic performances from 10 subjects. A mesh registration method that uses both 3D geometry and texture information to register all scans in a sequence to a common reference topology with 6890 vertices. The same as in (Ranjan et al. 2018), we split both two datasets into training and test set with a ratio of 9:1 and randomly select 100 samples from the training set for validation. We perform standardization on all the 3D shape meshes by subtracting the mean shape and dividing with each vertex’s standard deviation to improve the convergence speed of training.

**Training** We use Adam (Kingma and Ba 2014) optimizer with learning rate 0.001 and reduce the learning rate with decay rate 0.99 in every epoch. The batch size is 32 and total epoch number is 300. We initialize the weighting matrices with identity matrix,  $\mathbf{I} \in \mathbb{R}^{K \times K}$ , i.e., the network starts

without a predefined order for the node’s neighbors. Weight decay regularization is used for the network parameters except for the weighting matrices.

## Comparison to Existing Methods

We compare three existing methods: PCA (Bianz and Vetter 1999b), COMA (Ranjan et al. 2018), and Spiral (Bouritsas et al. 2019) on different dimensionalities of the latent space: 8, 16, 32, 64, and 128. The same architecture in Figure 2 is used in the methods of COMA, Spiral, and our LSA-3DMM for consistency and fair comparison. As shown in Figure 3, the proposed LSA-3DMM achieves the smallest reconstruction errors compared to COMA and Spiral on both COMA and DFAUST datasets by a large margin. LSA-3DMM consistently cuts the errors by around half for all the dimensionalities of latent space on both two datasets thanks to the proposed LSA-Conv operation. LSA-Conv significantly improves the expressive power on 3D shape representation learning compared to ChebNet and SpiralNet.

Compared to PCA, all the methods based on deep neural networks (DNN) have smaller reconstruction errors for a small latent size ( $d < 32$ ). This is because PCA-based linear 3DMMs can only capture global features and DNN-based nonlinear 3DMMs are able to capture local features using convolutional operations. The reconstruction accuracy of LSA-3DMM is comparable to PCA even when the latent size is 128. Note that, small latent size is favorable. Smaller latent size makes each latent feature more semantically meaningful. Furthermore, directly applying PCA needs a large amount of memory for a large dataset. At last, in real applications, the model is trained on batch. PCA parameters cannot be updated with additional data.

Figure 4 shows the validation reconstruction errors for the methods of COMA, Spiral, and our LSA-3DMM with the latent size of 32. The results of COMA and Spiral are rather close to each other: COMA performs slightly better on DFAUST dataset and Spiral performs slightly better on COMA dataset. LSA-3DMM converges much faster than COMA and Spiral. As shown in Figure 4b, after training for only 5 iterations, LSA-3DMM achieves smaller reconstruction error than COMA and Spiral training for 300 iterations.

In Figure 5, we qualitatively compare the reconstruction errors of some examples from the test sets of DFAUST and COMA datasets with the latent size  $d = 32$ . Our LSA-3DMM achieves smaller reconstruction errors for each case from the test sets. It is clearly visible that PCA has

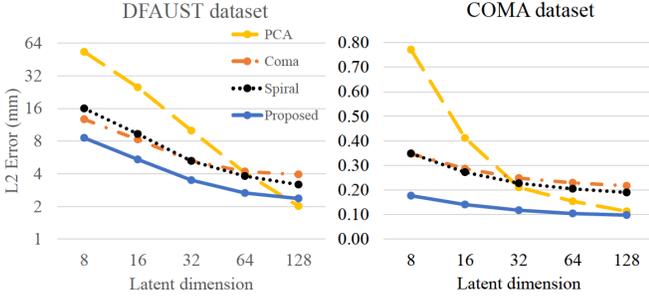


Figure 3: Evaluation of LSA-3DMM against peer methods: PCA, COMA, and Spiral on test sets.

the largest reconstruction errors on DFAUST dataset (Figure 5a) and Spiral has the largest reconstruction errors on COMA dataset (Figure 5b), which is consistent with the data shown in Figure 3. For Spiral, even though anisotropic filters are used compared to COMA that uses isotropic filters, the improvement is very limited since the manually designed neighboring order cannot capture the local structure very well. In contrast, our learnable weighting matrices can soft-permute each node’s neighbors that cooperate well with the shared anisotropic filters to extract local features of 3D shapes. As a result, LSA-3DMM performs better for 3D shape representation learning.

### Parameter Reduction for LSA-3DMM

Our model needs to learn a weighting matrix for each node of the template. When the latent size is  $d = 32$ , the parameter numbers of our model and PCA model are 1,867K and 482K, respectively. However, our model achieves much better reconstruction accuracy compared to PCA:  $0.117mm$  vs.  $0.21mm$ . The proposed model is the choice for situations that require low reconstruction errors and do not strictly limit the model size. For example, in the application of monocular 3D face reconstruction (Gao et al. 2020) that uses a template with 37,202 vertices, the number of extra parameters (around 3M) of the proposed method is still relatively small compared to the backbone ResNet50 (around 23M).

We also provide a method to reduce the number of our model’s parameters for situations where the model size should be limited. When the number of nodes is large, it is not necessary to learn the weighting matrix for each node since the geometric shapes of many nodes are similar to each other. We assume that the weighting matrices for all the nodes fall into a small subspace. We apply a matrix factorization technique in LSA-Conv. For meshes with  $N$  nodes, we denote  $\mathbf{P} \in \mathbb{R}^{N \times K \times K}$  for all the weighting matrices that can be factorized as follows,

$$\mathbf{P} = \mathbf{V}\mathbf{P}_b, \quad (4)$$

where  $\mathbf{P}_b \in \mathbb{R}^{B \times K \times K}$  is the  $B$ -dimensional subspace’s bases of weighting matrices,  $\mathbf{V} \in \mathbb{R}^{N \times B}$  is the weights correspond to the  $N$  nodes ( $B \ll N$  e.g.  $N = 5023, B = 8$ ). Instead of learning  $N$  weighting matrices directly, we learn a small number of weighting matrix bases and each node’s

	DFAUST		COMA	
	L2(mm)	parm #	L2(mm)	parm #
PCA	9.977	661K	0.210	482K
same architecture				
COMA	5.238	361K	0.248	303K
Spiral	5.258	446K	0.227	414K
<b>LSA-3DMM</b>	<b>3.492</b>	<b>2,478K</b>	<b>0.117</b>	<b>1,867K</b>
around same # of parm				
COMA (v2)	5.110	658K	0.198	532K
Spiral (v2)	4.667	647K	0.193	533K
<b>LSA (small)</b>	<b>4.544</b>	<b>644K</b>	<b>0.179</b>	<b>532K</b>

Table 1: Comparison of reconstruction errors for the models of PCA, COMA, Spiral, and our LSA-3DMM with latent size  $d = 32$ . *LSA (small)* is the model of LSA-3DMM with parameter reduction, where the dimension of weighting matrix subspace  $B = 8$ . COMA (v2) and Spiral (v2) are the models with increasing channel size to have around the same parameter number with *LSA (small)*.

corresponding weight. This can largely reduce the number of parameters decided by the choice of  $B$ .

Table 1 shows the comparison of reconstruction errors for models with different parameter numbers on DFAUST and COMA datasets. When using the same architecture (Figure 2), LSA-3DMM has the smallest reconstruction errors. We use the technique of matrix factorization to reduce the parameter number of our LSA-3DMM, denoted as *LSA (small)* where the subspace dimension is set to  $B = 8$ . For a fair comparison, we increase the channel sizes of COMA and Spiral so that the models have around the same number of parameters, denoted as *COMA (v2)* and *Spiral (v2)*. The channel sizes in *COMA (v2)* are [64, 96, 112, 128, 128, 112, 96, 96, 64]; the channel sizes in *Spiral (v2)* are [32, 64, 64, 128, 128, 110, 64, 64, 32]; while the channel sizes in *LSA (small)* are [16, 32, 64, 128, 128, 64, 32, 32, 16]. In Table 1, *LSA (small)* achieves the best results even we use only 8 basis weighting matrices, much smaller channel size, and smaller overall parameter number. Importantly, we can balance the tradeoff between the model’s reconstruction accuracy and model size by adjusting the subspace dimension.

### Ablation Study

The initial order of node’s neighbors, i.e.,  $\{\mathbf{x}_{i,0}, \mathbf{x}_{i,1}, \dots, \mathbf{x}_{i,|\mathcal{N}_i|-1}\}$ , is random. In order to evaluate the robustness and effectiveness of LSA-Conv, we reshuffle the order of each node’s neighbors by creating a randomized index list from 0 to  $|\mathcal{N}_i| - 1$ . Then we re-train the model and evaluate on the reshuffled neighbors. As shown in Table 2, in both DFAUST and COMA datasets, the reconstruction errors produced by the model with neighboring reshuffle are very close to the baseline. Thus, the learnable weighting matrix in our LSA-Conv for each node is robust to the initial neighboring order. LSA-Conv is easy to be implemented without involving into any manual design for the local coordinate systems and archives remarkable results on 3D shapes.

Table 2 also shows the impact of initialization methods

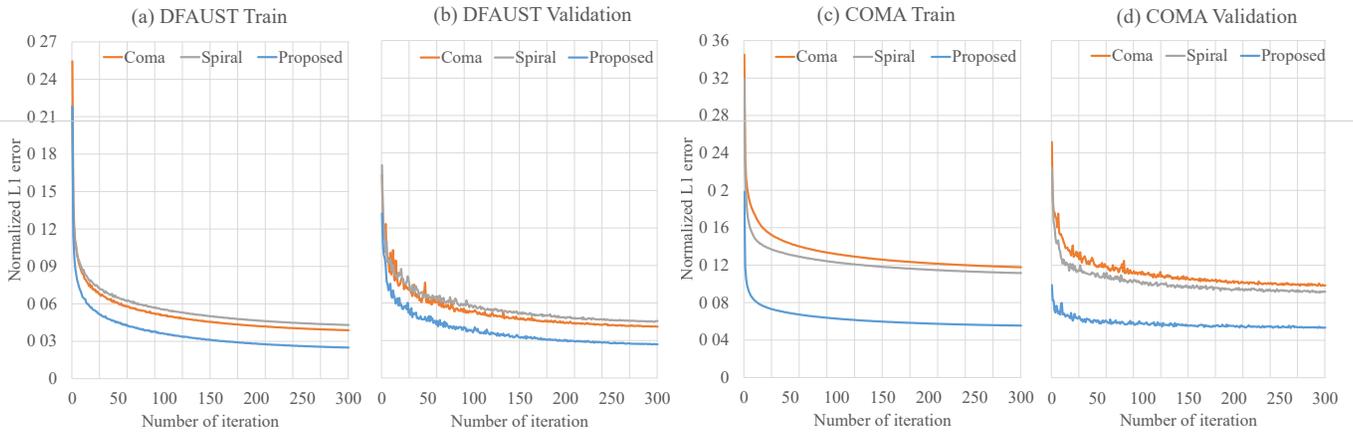


Figure 4: Reconstruction errors on COMA and DFAUST datasets when the latent size  $d = 32$ .

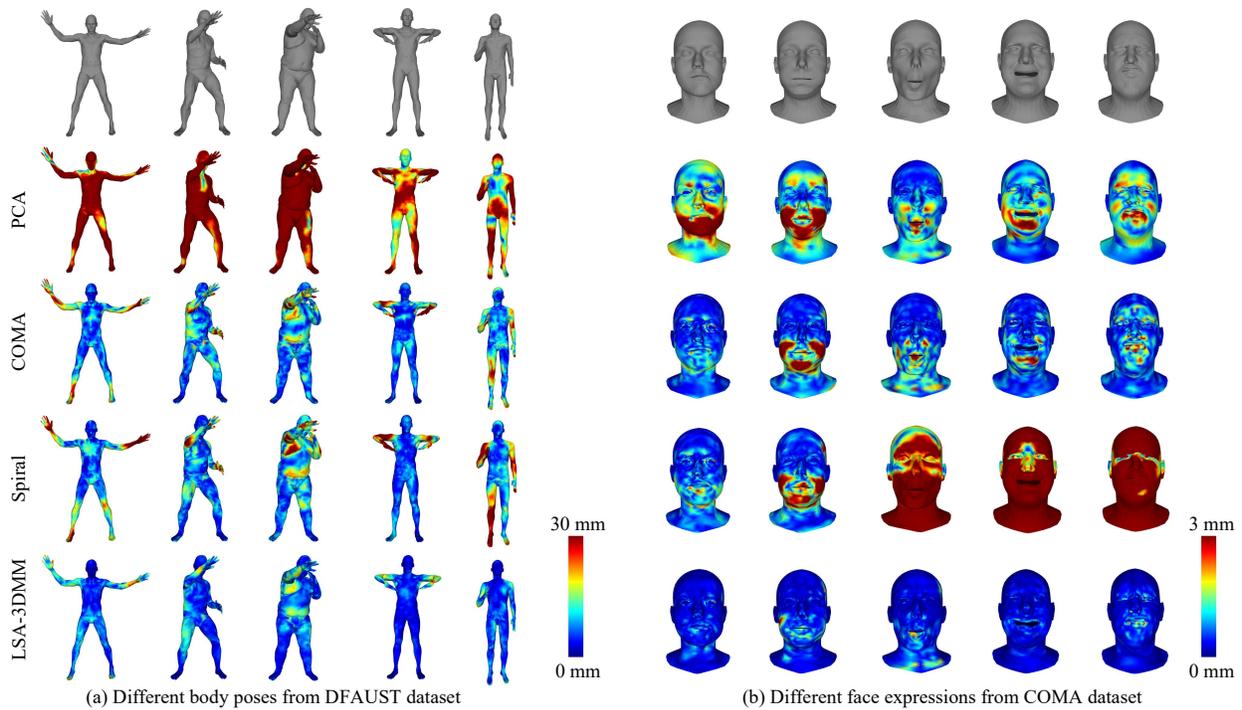


Figure 5: Colormaps of per vertex Euclidean error of the reconstructions produced by PCA, COMA, Spiral, and LSA-3DMM. Top row is the ground truth meshes from test sets. Latent size  $d = 32$ .

for the weighting matrices. In baseline and “*Random init*”, we initialize the weighting matrices with identity matrix and randomly from uniform distribution, respectively. Random initialization for the weighting matrices degrades the model’s performance. This is because random initialization neutralizes each node’s neighbors at the beginning of training and make each node’s neighbors indistinguishable, resulting in difficulty of extracting local features of 3D shapes.

We further evaluate the importance of the local structure-aware weighting matrices. Without weighting matrix, reconstruction errors of the two datasets increase. For COMA dataset, when without weighing matrix, the reconstruction

error (0.212) is smaller than Spiral (0.227), meaning the pre-defined spiral order in SpiralNet may not be useful.

In our experiments, we set the neighbor size  $K = 9$ . The effect of the neighbor size on LSA-3DMM is shown in Figure 6. When we increase the neighbor size, the reconstruction error decreases and model size increases. Compared to Table 1, even when the neighbor size  $K = 5$ , the reconstruction error (3.869) are still better than PCA (9.977), COMA (5.238), and Spiral (5.258). In practice, we can choose a neighbor size to balance the tradeoff between the reconstruction accuracy and model size.

	DFAUST (mm)	COMA (mm)
<b>Our baseline</b>	<b>3.492</b>	<b>0.117</b>
Reshuffle neighbors	3.527	<b>0.117</b>
Random init	4.488	0.137
w/o Weighting matrix	5.511	0.212

Table 2: Ablation tests of LSA-3DMM with the latent size  $d = 32$ . “Reshuffle neighbors” means we reshuffle the order of each node’s neighbors randomly. “Random init” means we initialize the weighting matrices randomly from uniform distribution.

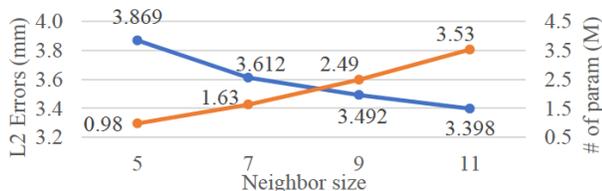


Figure 6: Comparison of different neighbor sizes of LSA-3DMM on DFAUST for latent size  $d = 32$ . Blue line denotes reconstruction error and orange curve represents model’s parameter size.

## Further Applications

To further verify the effectiveness of our LSA-Conv, we test it on two applications: 3D shape correspondences and monocular 3D face reconstruction, shown in Figure 7.

For 3D shape correspondences, we follow the pipeline of Groueix et al. (2018) where PointNet (Qi et al. 2017) is used as the encoder. We only replace the shape deformation decoder with our LSA-Conv based decoder, which has channel sizes of [256, 128, 64, 64, 32, 3]. We train the network on the same synthetic data created by Groueix et al. (2018), which has  $2.3 \cdot 10^5$  human meshes with a large variety of realistic poses and body shapes. We evaluate our method on FAUST (Bogo et al. 2014) that has online challenges<sup>1</sup>. The error is the average Euclidean distance between the estimated projection and the ground-truth projection. For the FAUST-inter dataset, we achieve 2.501cm: **an improvement of 15% over** Groueix et al. (2018): 2.878cm. Two examples of 3D shape correspondences are presented in Figure 8.

For monocular 3D face reconstruction, we follow the pipeline of Gao et al. (2020). The whole network architecture is presented in Figure 2 of Gao et al. (2020). ResNet-50 (He et al. 2016) is used as the encoder. We only replace the COMA based decoder with our LSA-Conv based decoder that has channel sizes of [256, 128, 64, 32, 16, 3]. The number of vertices is 37202. We train our model on the same datasets with Gao et al. (2020): hybrid batches of unlabeled face images from CelebA dataset (Liu et al. 2015) and labeled face images from 300W-LP dataset (Zhu et al. 2016). We evaluate our model qualitatively on the AFLW2000-3D dataset (Zhu et al. 2016). as shown in Figure 9.

<sup>1</sup>[http://faust.is.tue.mpg.de/challenge/Inter-subject\\_challenge](http://faust.is.tue.mpg.de/challenge/Inter-subject_challenge)

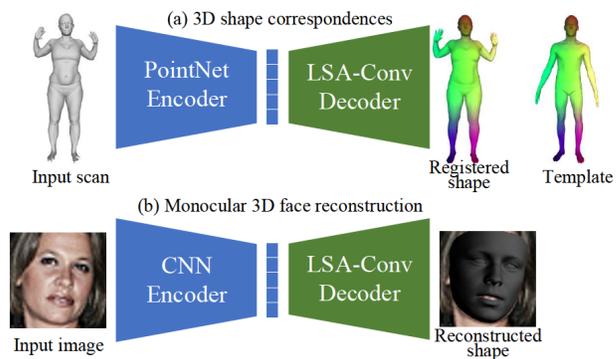


Figure 7: Applications of LSA-Conv. (a) 3D shape correspondences. (b) Monocular 3D face reconstruction. Both apply encoder-decoder architecture and use LSA-Conv as a basic building block in the decoders.

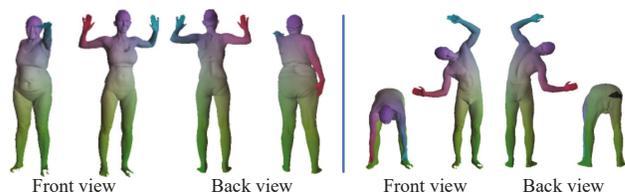


Figure 8: Examples of 3D shape correspondences.



Figure 9: Examples of monocular 3D face reconstruction.

## Conclusion

In this work, we propose a convolutional operation for 3D mesh representation learning and demonstrate its performance on 3D shape generation tasks, including two applications: 3D shape correspondences and monocular 3D face reconstruction. We use learnable weighting matrices to soft-permute each node’s neighbors and apply shared anisotropic filters across all the nodes. Compared to previous methods that either use isotropic filters (e.g., ChebNet and GCN) or use anisotropic filters with predefined local coordinate systems (e.g., SpiralNet and MoNet), LSA-Conv is able to extract local features depending on the geometric shape of each node and has much higher representation power. Using the same architecture of convolutional mesh autoencoder, our model achieves significant improvement in 3D shape reconstruction accuracy compared to state-of-the-art methods.

## Acknowledgements

This work was supported by China Major State Research Development Program (2020AAA0107600), the National Natural Science Foundation of China (U19B2035, 61831105, 61901259, 61972250), Shanghai Municipal Science and Technology Major Project (2021SHZDZX0102) and China Postdoctoral Science Foundation (BX2019208).

## Ethics Statement

This paper aims to advance the technology of 3D information processing, which has wide applications in virtual reality, entertainment, sports, architecture, etc. The more accurate and efficient 3D shape representation can help people more conveniently and cost-effectively record the physical world while in the mean time, the privacy of individuals may be put at risk. Hence we shall take additional measures to protect privacy along the development of such technology.

## References

- Blanz, V.; and Vetter, T. 1999a. A Morphable Model for the Synthesis of 3D Faces. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '99*, 187–194. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co. ISBN 0-201-48560-5. doi:10.1145/311535.311556.
- Blanz, V.; and Vetter, T. 1999b. A Morphable Model for the Synthesis of 3D Faces. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 99*, 187194. USA: ACM Press/Addison-Wesley Publishing Co. ISBN 0201485605. doi:10.1145/311535.311556.
- Bogo, F.; Romero, J.; Loper, M.; and Black, M. J. 2014. FAUST: Dataset and evaluation for 3D mesh registration. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. Piscataway, NJ, USA: IEEE.
- Bogo, F.; Romero, J.; Pons-Moll, G.; and Black, M. J. 2017. Dynamic FAUST: Registering Human Bodies in Motion. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Bouritsas, G.; Bokhnyak, S.; Ploumpis, S.; Bronstein, M.; and Zafeiriou, S. 2019. Neural 3D Morphable Models: Spiral Convolutional Networks for 3D Shape Representation Learning and Generation. In *The IEEE International Conference on Computer Vision (ICCV)*.
- Bruna, J.; Zaremba, W.; Szlam, A.; and LeCun, Y. 2014. Spectral Networks and Locally Connected Networks on Graphs. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.
- Cao, C.; Hou, Q.; and Zhou, K. 2014. Displaced Dynamic Expression Regression for Real-time Facial Tracking and Animation. *ACM Trans. Graph.* 33(4): 43:1–43:10. ISSN 0730-0301.
- Cao, C.; Weng, Y.; Zhou, S.; Tong, Y.; and Zhou, K. 2014. FaceWarehouse: A 3D Facial Expression Database for Visual Computing. *IEEE Transactions on Visualization and Computer Graphics* 20(3): 413–425. ISSN 1077-2626. doi:10.1109/TVCG.2013.249.
- Clevert, D.-A.; Unterthiner, T.; and Hochreiter, S. 2015. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*.
- Defferrard, M.; Bresson, X.; and Vandergheynst, P. 2016. Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering. In Lee, D. D.; Sugiyama, M.; Luxburg, U. V.; Guyon, I.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 29*, 3844–3852. Curran Associates, Inc.
- Donlic, M.; Petkovic, T.; Peharec, S.; Berryman, F.; and Pribanic, T. 2017. On the Segmentation of Scanned 3D Human Body Models. In *8th International Scientific Conference on Kinesiology*.
- Gao, Z.; Zhang, J.; Guo, Y.; Ma, C.; Zhai, G.; and Yang, X. 2020. Semi-supervised 3D Face Representation Learning from Unconstrained Photo Collections. In *IEEE Computer Vision and Pattern Recognition Workshops*.
- Genova, K.; Cole, F.; Maschinot, A.; Sarna, A.; Vlastic, D.; and Freeman, W. T. 2018. Unsupervised Training for 3D Morphable Model Regression. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Gong, S.; Chen, L.; Bronstein, M.; and Zafeiriou, S. 2019. SpiralNet++: A Fast and Highly Efficient Mesh Convolution Operator. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 0–0.
- Groueix, T.; Fisher, M.; Kim, V. G.; Russell, B. C.; and Aubry, M. 2018. 3D-CODED: 3D Correspondences by Deep Deformation. In *The European Conference on Computer Vision (ECCV)*.
- Hamilton, W.; Ying, Z.; and Leskovec, J. 2017. Inductive Representation Learning on Large Graphs. In Guyon, I.; Luxburg, U. V.; Bengio, S.; Wallach, H.; Fergus, R.; Vishwanathan, S.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 30*, 1024–1034. Curran Associates, Inc.
- He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Hu, Q.; Yang, B.; Xie, L.; Rosa, S.; Guo, Y.; Wang, Z.; Trigoni, N.; and Markham, A. 2020. RandLA-Net: Efficient Semantic Segmentation of Large-Scale Point Clouds. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
- Jiang, Z.-H.; Wu, Q.; Chen, K.; and Zhang, J. 2019. Disentangled Representation Learning for 3D Face Shape. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Kalogerakis, E.; Hertzmann, A.; and Singh, K. 2010. Learning 3D Mesh Segmentation and Labeling. New York, NY, USA: Association for Computing Machinery. ISBN 9781450302104.

- Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Kipf, T. N.; and Welling, M. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations (ICLR)*.
- Li, R.; Wang, S.; Zhu, F.; and Huang, J. 2018a. Adaptive graph convolutional neural networks. In *Thirty-Second AAAI Conference on Artificial Intelligence*.
- Li, Y.; Bu, R.; Sun, M.; Wu, W.; Di, X.; and Chen, B. 2018b. PointCNN: Convolution On  $\mathcal{X}$ -Transformed Points. In Bengio, S.; Wallach, H.; Larochelle, H.; Grauman, K.; Cesa-Bianchi, N.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 31*, 820–830. Curran Associates, Inc. URL <http://papers.nips.cc/paper/7362-pointcnn-convolution-on-x-transformed-points.pdf>.
- Liu, F.; Zhu, R.; Zeng, D.; Zhao, Q.; and Liu, X. 2018. Disentangling Features in 3D Face Shapes for Joint Face Reconstruction and Recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Liu, Z.; Luo, P.; Wang, X.; and Tang, X. 2015. Deep Learning Face Attributes in the Wild. In *Proceedings of International Conference on Computer Vision (ICCV)*.
- Loper, M.; Mahmood, N.; Romero, J.; Pons-Moll, G.; and Black, M. J. 2015. SMPL: A Skinned Multi-Person Linear Model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)* 34(6): 248:1–248:16.
- Monti, F.; Boscaini, D.; Masci, J.; Rodola, E.; Svoboda, J.; and Bronstein, M. M. 2017. Geometric Deep Learning on Graphs and Manifolds Using Mixture Model CNNs. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Paysan, P.; Knothe, R.; Amberg, B.; Romdhani, S.; and Vetter, T. 2009. A 3D Face Model for Pose and Illumination Invariant Face Recognition. In *2009 Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance*, 296–301. doi:10.1109/AVSS.2009.58.
- Qi, C. R.; Su, H.; Mo, K.; and Guibas, L. J. 2017. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Ranjan, A.; Bolkart, T.; Sanyal, S.; and Black, M. J. 2018. Generating 3D Faces using Convolutional Mesh Autoencoders. In *The European Conference on Computer Vision (ECCV)*.
- Romero, J.; Tzionas, D.; and Black, M. J. 2017. Embodied Hands: Modeling and Capturing Hands and Bodies Together. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)* 36(6).
- Tay, Y.; Bahri, D.; Metzler, D.; Juan, D.-C.; Zhao, Z.; and Zheng, C. 2020. Synthesizer: Rethinking self-attention in transformer models. *arXiv preprint arXiv:2005.00743*.
- Thomas, H.; Qi, C. R.; Deschard, J.-E.; Marcotegui, B.; Goulette, F.; and Guibas, L. J. 2019. KPConv: Flexible and Deformable Convolution for Point Clouds. In *The IEEE International Conference on Computer Vision (ICCV)*.
- Tran, L.; and Liu, X. 2018. Nonlinear 3D Face Morphable Model. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; and Bengio, Y. 2018. Graph Attention Networks. *International Conference on Learning Representations* Accepted as poster.
- Verma, N.; Boyer, E.; and Verbeek, J. 2018. FeaStNet: Feature-Steered Graph Convolutions for 3D Shape Analysis. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Xie, J.; Zheng, Z.; Gao, R.; Wang, W.; Zhu, S.-C.; and Nian Wu, Y. 2018. Learning Descriptor Networks for 3D Shape Synthesis and Analysis. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Zhu, X.; Lei, Z.; Liu, X.; Shi, H.; and Li, S. Z. 2016. Face Alignment Across Large Poses: A 3D Solution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Zhu, X.; Lei, Z.; Yan, J.; Yi, D.; and Li, S. Z. 2015. High-fidelity Pose and Expression Normalization for face recognition in the wild. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 787–796. ISSN 1063-6919. doi:10.1109/CVPR.2015.7298679.