Unsupervised 3D Pose Transfer with Cross Consistency and Dual Reconstruction

Chaoyue Song, Jiacheng Wei, Ruibo Li, Fayao Liu and Guosheng Lin

Abstract—The goal of 3D pose transfer is to transfer the pose from the source mesh to the target mesh while preserving the identity information (e.g., face, body shape) of the target mesh. Deep learning-based methods improved the efficiency and performance of 3D pose transfer. However, most of them are trained under the supervision of the ground truth, whose availability is limited in real-world scenarios. In this work, we present *X-DualNet*, a simple yet effective approach that enables unsupervised 3D pose transfer. In *X-DualNet*, we introduce a generator G which contains correspondence learning and pose transfer modules to achieve 3D pose transfer. We learn the shape correspondence by solving an optimal transport problem without any key point annotations and generate high-quality meshes with our elastic instance normalization (ElaIN) in the pose transfer module. With G as the basic component, we propose a cross consistency learning scheme and a dual reconstruction objective to learn the pose transfer without supervision. Besides that, we also adopt an as-rigid-as-possible deformer in the training process to fine-tune the body shape of the generated results. Extensive experiments on human and animal data demonstrate that our framework can successfully achieve comparable performance as the state-of-the-art supervised approaches.

Index Terms—3D Pose Transfer, Optimal Transport, Conditional Normalization Layer, Unsupervised Learning, Cross Consistency, As-rigid-as-possible Deformation.

1 Introduction

3D Pose Transfer is a crucial task in vision and computer graphics communities which has many applications in augmented reality, robotics, and video games. As shown in Figure 1, given a source mesh that provides the pose and a target mesh that provides the identity (e.g., face, body shape), 3D pose transfer aims to transfer the pose from the source to the target and preserve the identity information of the target mesh.

Traditional method [1] defines an optimization problem to address 3D pose transfer. They apply the affine transformations computed from the deformation of the source mesh to the target one, which are time-consuming and always need key points labeling to build the shape correspondence. With the huge success of neural networks, many methods [2], [3] were proposed to achieve 3D pose transfer in a deep learning manner. Wang *et al.* [2] re-purpose the latest style transfer methods for 3D pose transfer. The performance of their method is always unsatisfactory since they do not consider any correspondence between different human or animal meshes. Besides that, they also need the supervision of the ground truth label, which is hard to acquire in real-world scenarios.

In this work, we present *X-DualNet*, an effective approach to achieve unsupervised 3D pose transfer. With a generator G which contains correspondence learning and pose transfer modules as the basic component, we propose a cross consistency learning scheme and a dual reconstruction objective to learn the pose transfer without supervision. Specifically, G takes the vertex coordinates of the input meshes and extracts their features. Then we solve an optimal transport problem to build the shape correspondence in the correspondence learning module and achieve the 3D pose transfer using the proposed elastic instance normalization (ElaIN) in the pose transfer module. Similar to [2], [3], our approach can handle the training and test meshes with different numbers of vertices or triangles.

For the correspondence learning module, we treat shape correspondence learning as an optimal transport problem to learn the correspondence between meshes. Our network takes vertex coordinates of identity and pose meshes as inputs. We extract deep features at each vertex using point cloud convolutions and compute a matching cost between the vertex sets with the extracted features. Our goal is to minimize the matching cost to get an optimal matching matrix. With the optimal matching matrix, we warp the pose mesh and obtain the warped mesh which is actually a redistribution of the vertices of the pose mesh. We then transfer the pose from the warped mesh to the identity mesh with a set of elastic instance normalization residual blocks. The modulation parameters in the normalization layers are learned with our proposed *Elastic Instance Normalization (ElaIN)*. In order to generate smoother meshes with more details, we introduce a channel-wise weight in ElaIN to adaptively blend feature statistics of original features and the learned parameters from external data, which helps to keep the consistency and continuity of the original features.

In *X-DualNet*, with a main generator G_{AB}, we take two meshes M_A and M_B as inputs and generate the output

- *Chaoyue Song (Email: chaoyue.song@ntu.edu.sg), Ruibo Li and Guosheng Lin are with S-Lab for Advanced Intelligence, Nanyang Technological University, Singapore. Ruibo Li and Guosheng Lin are also with the School of Computer Science and Engineering, Nanyang Technological University, Singapore.*
- *Jiacheng Wei is with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore.*
- *Fayao Liu is with the Institute for Inforcomm Research, A*STAR, Singapore.*
- *Corresponding author: Guosheng Lin (Email: gslin@ntu.edu.sg)*

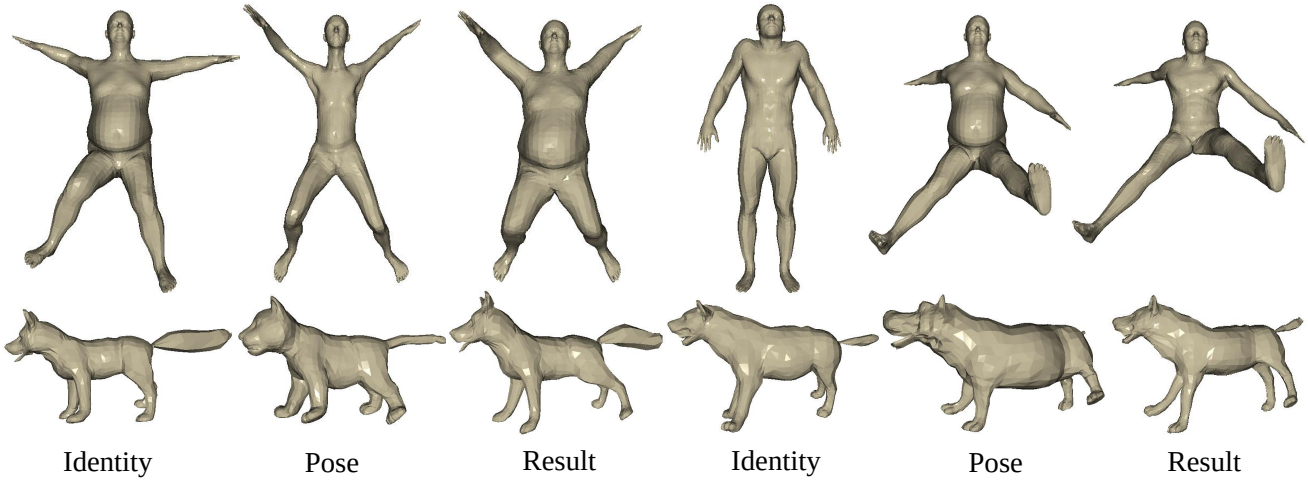


Fig. 1. With a target mesh and a source mesh, *X-DualNet* provides an unsupervised solution to transfer the pose from the source to the target and preserve the identity information of the target mesh. In the first row, the human meshes are from SMPL [4] and FAUST [5]. In the second row, the animal meshes are from SMAL [6]. X-DualNet can achieve the 3D pose transfer successfully on different human and animal datasets.

$M_{output} = G_{AB}(M_A, M_B)$. Here, we view M_A as the identity mesh and M_B as the pose mesh respectively. Besides that, we also add an ARAP deformer in the training process to fine-tune the body shape of the generated results and get the deformed output $\hat{M}_{output}$. Then we adopt an auxiliary generator G'_A to reconstruct the mesh M_A. Specifically, G'_A takes M_A as the pose mesh and $\hat{M}_{output}$ as the identity mesh respectively to generate $M'_A = G'_A(\hat{M}_{output}, M_A)$. Similarly, we can reconstruct M_B with another auxiliary generator G'_B to generate $M'_B = G'_B(M_B, \hat{M}_{output})$. Note that G_{AB}, G'_A and G'_B share the same network parameters. The pose transfer procedure is consistent across the main generator and two auxiliary generators. To achieve this dual reconstruction objective, we introduce a reconstruction loss that encourages $M'_A \approx M_A$ and $M'_B \approx M_B$. Based on this cross consistency learning scheme and the dual reconstruction objective, our *X-DualNet* can learn the 3D pose transfer without any supervision.

In summary, our main contributions are:

• We present *X-DualNet*, an unsupervised deep learning framework to solve the 3D pose transfer problem in an end-to-end fashion.

• In *X-DualNet*, we introduce a generator G that contains correspondence learning and pose transfer modules to achieve the 3D pose transfer. We learn the shape correspondence by solving an optimal transport problem without any key point annotations and generate high-quality meshes with our proposed elastic instance normalization in the pose transfer module. To the best of our knowledge, our method is the first to learn the correspondence between different meshes and transfer the poses jointly in the 3D pose transfer task.

• With the generator G as the basic component, we propose a cross consistency learning scheme and a dual reconstruction objective to learn the pose transfer without supervision. Besides that, we also adopt an as-rigid-as-possible deformer to fine-tune the body shape of the generated results.

• Through extensive experiments on human and animal meshes, we demonstrate *X-DualNet* achieves comparable performance as the state-of-the-art supervised approaches qualitatively and quantitatively and even outperforms some of them. In addition, our framework has better generalization capability than supervised methods.

This work is an extended version of 3D-CoreNet [3]. The code is available at https://github.com/ChaoyueSong/3d-corenet.

2 Related work

2.1 Deep networks for 3D representations

Recently, many different methods have been introduced to analyze 3D data. VoxNet [7] and 3DShapeNets [8] were introduced to learn on volumetric grids. However, the extra computation cost makes it difficult for them to apply to complex data. Due to the irregular format of point cloud, PointNet [9] processes each point independently followed by a symmetry function, but it cannot captures the local structure. To address this problem, SO-Net [10] and PointNet++ [11] propose to aggregate local structures with nearest neighbors. For 3D meshes, Meshnet [12] and mesh VAE [13] propose mesh neural networks to capture and learn features of polygon faces. However, their methods require many computing resources because of their fully-connected layers. There are also many works using graph convolutions with mesh sampling operators [14] to implement the hierarchical structure, such as CoMA [15], SpiralNet [16] and SpiralNet++ [17], etc. Their methods all require the mesh datasets registered to a template for training. In this work, we do not need the mesh template and our approach is able to handle non-registered meshes with different numbers of vertices or triangles.

2.2 3D pose transfer

3D pose transfer has been widely studied in vision and graphics communities. Many traditional methods require additional information to build the correspondence between different meshes or generate the final results. DT [1] and Yang *et al.* [18] need users to label the corresponding vertices.

Baran *et al.* [19] infer semantic correspondence between the shape spaces of two characters by providing examples of corresponding poses. Chu *et al.* [20] propose to use some examples that are similar to the sources in poses to generate results, their method has difficulties transferring pose automatically. Recently, deep neural networks have shown powerful learning and generalization capability to solve 3D pose transfer problems. VC-GAN [21] follows the idea of CycleGAN [22] to help 3D pose transfer, but their method relies heavily on the training data and cannot work on new shapes because of the visual similarity. Wang *et al.* [23] extend a traditional cage-based deformation method but need user labeling to handle the differences between meshes. Wang *et al.* [2] re-purpose the latest style transfer techniques for 3D pose transfer, the performance of their model is always not satisfied since they do not consider any correspondence between the identity and the pose meshes. To address this problem, we learn the shape correspondence of different human or animal meshes. The previous solutions either require additional information or ground truth to instruct the training process. In this work, we propose to achieve the 3D pose transfer in an unsupervised way with the cross consistency learning scheme and a dual reconstruction objective.

2.3 Correspondence learning

In CoCosNet [24], they introduced a correspondence network based on the correlation matrix between images without any constraints. To learn a better matching, we proposed to use optimal transport to learn the correspondence between meshes. Recently, optimal transport has received great attention in various computer vision tasks. Courty *et al.* [25] perform the alignment of the representations in the source and target domains by learning a transportation plan. Su *et al.* [26] compute the optimal transport map to deal with the surface registration and shape space problem. Other applications include generative model [27], [28], [29], [30], scene flow [31], semantic correspondence [32] and etc.

2.4 Conditional normalization layers

After normalizing the activation value, conditional normalization uses the modulation parameters calculated from the external data to denormalize it. Adaptive Instance Normalization (AdaIN) [33] aligns the mean and variance between content and style image which achieves arbitrary style transfer. Soft-AdaIN [34] introduces a channel-wise weight to blend feature statistics of content and style images to preserve more details for the results. Spatially-Adaptive Normalization (SPADE) [35] can better preserve semantic information by not washing away it when applied to segmentation masks. SPAdaIN [2] changes batch normalization [36] in SPADE to instance normalization [37] for 3D pose transfer. However, it will break the consistency and continuity of the feature map when doing the denormalization, which has a bad influence on the mesh smoothness and detail preservation. To address this problem, our ElaIN introduces an adaptive weight to implement the denormalization.

2.5 Unsupervised learning for 3D shapes

CorrNet3D [38] proposes to learn the shape correspondence without supervision using deformation-like reconstruction. NeuroMorph [39] solves the correspondence and interpolation problems with geometric priors for regularization. Zhou *et al.* [40] and Chen *et al.* [41] learn disentangled shape and pose without supervision. However, the method of Zhou *et al.* [40] can only train or test on registered data and the training process of Chen *et al.* [41] is very unstable and needs a great number of iterations to converge since the adoption of GAN [42]. In this work, our method can work on non-registered meshes (they can have different vertex numbers and orders) in an unsupervised manner and enable a more robust and faster convergence. The cross consistency constraint mentioned in Zhou *et al.* [40] is used on meshes with the same identity and is not sufficient to instruct the disentanglement. Different from it, our cross consistency learning scheme helps to achieve the unsupervised 3D pose transfer globally and effectively based on the main and auxiliary generators.

3 METHOD

With a source mesh and a target mesh, 3D pose transfer aims to transfer the pose from the source mesh to the target mesh while preserving the identity information (e.g., face, body shape) of the target mesh. In this section, we will introduce how to implement our *X-DualNet* for unsupervised 3D pose transfer.

To explain our method better, we define a 3D triangle mesh as $M(I, P, N, O)$. Here, I represents the identity of the mesh (e.g., face, body shape), P means the pose of the mesh, N and O denote the vertex number and order respectively.

3.1 Network architecture

As shown in Figure 2, we adapt a generator G as the basic component of our *X-DualNet*. Given the identity mesh $M_{id} = M(I_{id}, P_{id}, N_{id}, O_{id})$ and the pose mesh $M_{pose} = M(I_{pose}, P_{pose}, N_{pose}, O_{pose})$, G aims to transfer the pose from M_{pose} to M_{id} and generate the result $M'(I_{id}, P_{pose}, N_{id}, O_{id})$. The inputs of G are $V_{id} \in \mathbb{R}^{N_{id} \times 3}$ and $V_{pose} \in \mathbb{R}^{N_{pose} \times 3}$. Here, N_{id} and N_{pose} are the vertex number of M_{id} and M_{pose} respectively, V_{id} and V_{pose} are the corresponding vertex coordinates.

The basic component G contains two modules: correspondence learning and pose transfer. In the correspondence learning module, we learn an optimal matching matrix between the identity mesh and the pose mesh using the optimal transport. We discuss more details in Section 3.1.1. With the optimal matching matrix, the pose mesh is warped to get the warped mesh (as shown in Figure 2) which is actually a redistribution of the vertices of the pose mesh.

The pose transfer module aims to generate the output which can combine the pose of M_{pose} and the identity of M_{id}. This pose transfer procedure is achieved based on the proposed elastic instance normalization (ElaIN). We will introduce how to design ElaIN in Section 3.1.2.

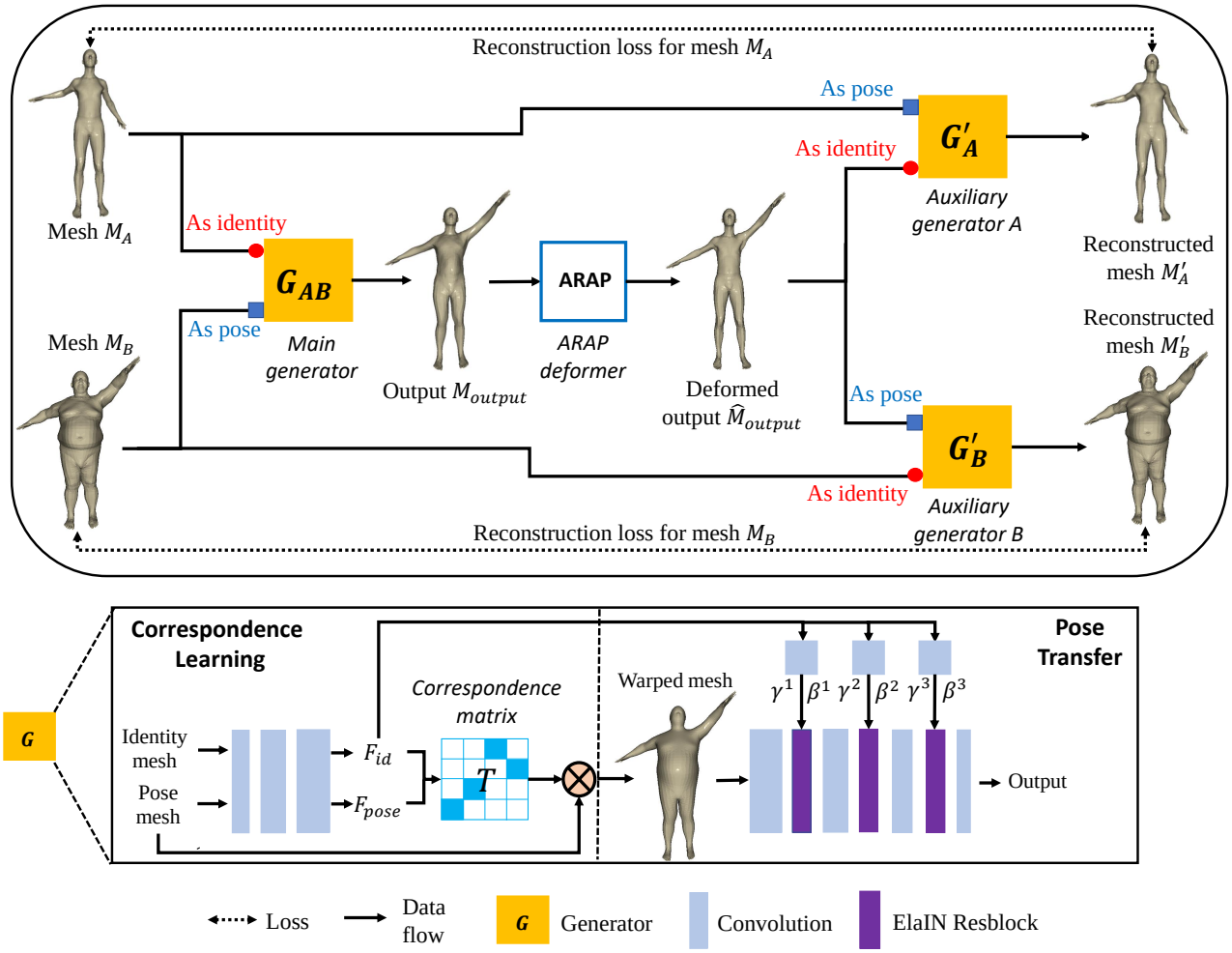


Fig. 2. **An overview of *X-DualNet*.** With a main generator G_{AB}, we take two meshes M_A and M_B as inputs and generate the output $M_{output} = G_{AB}(M_A, M_B)$. Here, we view M_A as the identity mesh and M_B as the pose mesh respectively. We also add an ARAP deformer in the training process to fine-tune the body shape of the generated results and get the deformed output $\hat{M}_{output}$. Then we adopt an auxiliary generator G'_A to reconstruct the mesh M_A. Specifically, G'_A takes M_A as the pose mesh and $\hat{M}_{output}$ as the identity mesh respectively to generate $M'_A = G'_A(\hat{M}_{output}, M_A)$. Similarly, we can reconstruct M_B with another auxiliary generator G'_B to generate $M'_B = G'_B(M_B, \hat{M}_{output})$. Note that G_{AB}, G'_A and G'_B share the same network parameters. The pose transfer procedure is consistent across the main generator and two auxiliary generators. We introduce a reconstruction loss to achieve the dual reconstruction objective $M'_A \approx M_A$ and $M'_B \approx M_B$. The generator G contains two modules. In the correspondence learning module, we solve an optimal transport problem to build the correspondence between input meshes with the extracted features F_{id} and F_{pose}. The pose transfer module contains several ElaIN residual blocks which help to achieve the 3D pose transfer.

3.1.1 *Correspondence learning*

Given an identity mesh and a pose mesh, our correspondence learning module calculates an optimal matching matrix, each element of the matching matrix represents the similarities between two vertices in the two meshes. The first step in our shape correspondence learning is to compute a correlation matrix with the extracted features, which is based on cosine similarity and denotes the matching similarities between any two positions from different meshes. However, the matching scores in the correlation matrix are calculated without any additional constraints. To learn a better matching, we solve this problem from a global perspective by modeling it as an optimal transport problem.

Correlation matrix. We first introduce our feature extractor which aims to extract features for the unordered input vertices. Different from NPT [2], we use pointnet++-like convolution layers to extract vertex features. Given the extracted vertex feature maps $f_{id} \in \mathbb{R}^{D \times N_{id}}$, $f_{pose} \in \mathbb{R}^{D \times N_{pose}}$ of identity and pose meshes (D is the channel-wise dimension), a popular method to compute correlation matrix is using the cosine similarity [24], [43]. Concretely, we compute the correlation matrix $\mathbf{C} \in \mathbb{R}^{N_{id} \times N_{pose}}$ as:

$$\mathbf{C}(i,j) = \frac{f_{id}(i)^\top f_{pose}(j)}{\|f_{id}(i)\| \, \|f_{pose}(j)\|}, \tag{1}$$

where $\mathbf{C}(i,j)$ denotes the individual matching score between $f_{id}(i)$ and $f_{pose}(j) \in \mathbb{R}^{\mathbb{D}}$, $f_{id}(i)$ and $f_{pose}(j)$ represent the channel-wise feature of f_{id} at position i and f_{pose} at j.

Optimal transport problem. To learn a better matching with additional constraints in this work, we model our shape correspondence learning as an optimal transport problem. We first define a matching matrix $\mathbf{T} \in \mathbb{R}_+^{N_{id} \times N_{pose}}$ between identity and pose meshes. Then we can get the total correlation as $\sum_{ij} \mathbf{C}(i,j)\mathbf{T}(i,j)$. The aim will be maximizing

the total correlation score to get an optimal matching matrix $\mathbf{T}_m$.

We treat the correspondence learning between identity and pose meshes as the transport of mass. A mass that is equal to N_{id}^{-1} will be assigned to each vertex in the identity mesh, and each vertex in the pose mesh will receive the mass N_{pose}^{-1} from identity mesh through the built correspondence between vertices. Then if we define $\mathbf{Z} = 1 - \mathbf{C}$ as the cost matrix, our goal can be formulated as a standard optimal transport problem by minimizing the total matching cost,

$$\begin{aligned} \mathbf{T}_m = & \underset{\mathbf{T} \in \mathbb{R}_+^{N_{id} \times N_{pose}}}{\arg\min} \sum_{ij} \mathbf{Z}(i,j)\mathbf{T}(i,j) \\ s.t. \quad & \mathbf{T}\mathbf{1}_{N_{pose}} = \mathbf{1}_{N_{id}} N_{id}^{-1}, \quad \mathbf{T}^\top \mathbf{1}_{N_{id}} = \mathbf{1}_{N_{pose}} N_{pose}^{-1}. \end{aligned} \tag{2}$$

where $\mathbf{1}_{N_{id}} \in \mathbb{R}^{N_{id}}$ and $\mathbf{1}_{N_{pose}} \in \mathbb{R}^{N_{pose}}$ are vectors whose elements are all 1. The first constraint in Equation 2 means that the mass of each vertex in M_{id} will be entirely transported to some of the vertices in M_{pose}. And each vertex in M_{pose} will receive a mass N_{pose}^{-1} from some of the vertices in M_{id} with the second constraint. This problem can be solved by the Sinkhorn-Knopp algorithm [44]. The details of the solving process will be given in the supplementary material.

With the matching matrix, we can warp the pose mesh and obtain the vertex coordinates $V_{warp} \in \mathbb{R}^{N_{id} \times 3}$ of the warped mesh,

$$V_{warp}(i) = \sum_j \mathbf{T}_m(i,j) V_{pose}(j), \tag{3}$$

the warped mesh M_{warp} inherits the number and order of vertex from identity mesh and can be reconstructed with the face information of identity mesh as shown in Figure 2.

3.1.2 Pose transfer

In this section, we introduce our pose transfer module which transfers the pose from the warped mesh to the identity mesh and generates the desired output.

Elastic instance normalization. Previous conditional normalization layers [2], [33], [35] used in different tasks always calculated their denormalization parameters only with the external data. We argue that it may break the consistency and continuity of the original features. Inspired by Soft-AdaIN [34], we propose *Elastic Instance Normalization (ElaIN)* which blends the statistics of original features and the learned parameters from external data adaptively and elastically.

To be specific, we let $h^i \in \mathbb{R}^{S^i \times D^i \times N^i}$ as the activation value before the i-th normalization layer, where S^i is the batch size, D^i is the dimension of feature channel and N^i is the number of vertex. As shown in Figure 3, we normalize the feature maps of the warped mesh with instance normalization first and the mean and standard deviation are calculated across spatial dimension ($n \in N^i$) for each sample $s \in S^i$ and each channel $d \in D^i$,

$$\mu^i = \frac{1}{N^i} \sum_n h_{warp}^i, \tag{4}$$

$$\sigma^i = \sqrt{\frac{1}{N^i} \sum_n (h_{warp}^i - \mu^i)^2 + \epsilon}, \tag{5}$$

then the feature maps of the identity mesh are fed into a 1×1 convolution layer to get h_{id}^i, which shares the same size with h_{warp}^i. We calculate the mean of h_{warp}^i, h_{id}^i to make them $S^i \times D^i \times 1$ tensors. The tensors are then concatenated in channel dimension to get a $S^i \times (2D^i) \times 1$ tensor. A fully-connected layer is employed to compute an adaptive weight $w(h_{warp}^i, h_{id}^i) \in \mathbb{R}^{S^i \times D^i \times 1}$ with the concatenated tensor. With $w(h_{warp}^i, h_{id}^i)$, we can define the modulation parameters of our normalization layer.

$$\begin{aligned} \gamma'(h_{warp}^i, h_{id}^i) &= w(h_{warp}^i, h_{id}^i)\gamma^i + (1 - w(h_{warp}^i, h_{id}^i))\sigma^i, \\ \beta'(h_{warp}^i, h_{id}^i) &= w(h_{warp}^i, h_{id}^i)\beta^i + (1 - w(h_{warp}^i, h_{id}^i))\mu^i. \end{aligned} \tag{6}$$

where γ^i and β^i are learned from the identity feature h_{id}^i with two convolution layers. Finally, we can scale the normalized h_{warp}^i with γ' and shift it with β',

$$\begin{aligned} \text{ElaIN}(h_{warp}^i, h_{id}^i) = & \gamma'(h_{warp}^i, h_{id}^i)\left(\frac{h_{warp}^i - \mu^i}{\sigma^i}\right) \\ & + \beta'(h_{warp}^i, h_{id}^i). \end{aligned} \tag{7}$$

Pose transfer module. Our pose transfer module is designed to transfer the pose from the warped mesh to the identity mesh. Following NPT [2], CoCosNet [24], and SPADE [35], we design the ElaIN residual block with our ElaIN in the form of ResNet blocks [45]. As shown in Figure 2, our architecture contains 3 ElaIN residual blocks. Each of them consists of our proposed ElaIN followed by a simple convolution layer and LeakyReLU. With the ElaIN residual blocks, our network can generate high-quality results.

3.2 Cross consistency and dual reconstruction

In this section, we will introduce how to achieve unsupervised 3D pose transfer with our cross consistency learning scheme and dual reconstruction objective.

As shown in Figure 2, with a main generator G_{AB}, we take two meshes M_A and M_B as inputs and generate the output $M_{output} = G_{AB}(M_A, M_B)$. Here, we view M_A as the identity mesh and M_B as the pose mesh respectively. We also use an ARAP deformer that will be introduced in the following to get the deformed output $\hat{M}_{output}$. Then we adopt an auxiliary generator G'_A to reconstruct the mesh M_A. Specifically, G'_A takes M_A as the pose mesh and $\hat{M}_{output}$ as the identity mesh respectively to generate $M'_A = G'_A(\hat{M}_{output}, M_A)$. Similarly, we can reconstruct M_B with another auxiliary generator G'_B. For G'_B, we view M_B as the identity mesh and $\hat{M}_{output}$ as the pose mesh respectively to generate $M'_B = G'_B(M_B, \hat{M}_{output})$. Note that G_{AB}, G'_A and G'_B share the same network parameters. The pose transfer procedure is consistent across the main generator and two auxiliary generators.

To achieve the dual reconstruction objective, we introduce a reconstruction loss in the loss function section that encourages $M'_A \approx M_A$ and $M'_B \approx M_B$. Once the dual reconstruction objective is achieved, our pose transfer module will not only learn the identity information from the identity mesh but also inherit the pose from the pose mesh.

Based on this cross consistency learning scheme and the dual reconstruction objective, our *X-DualNet* can learn the

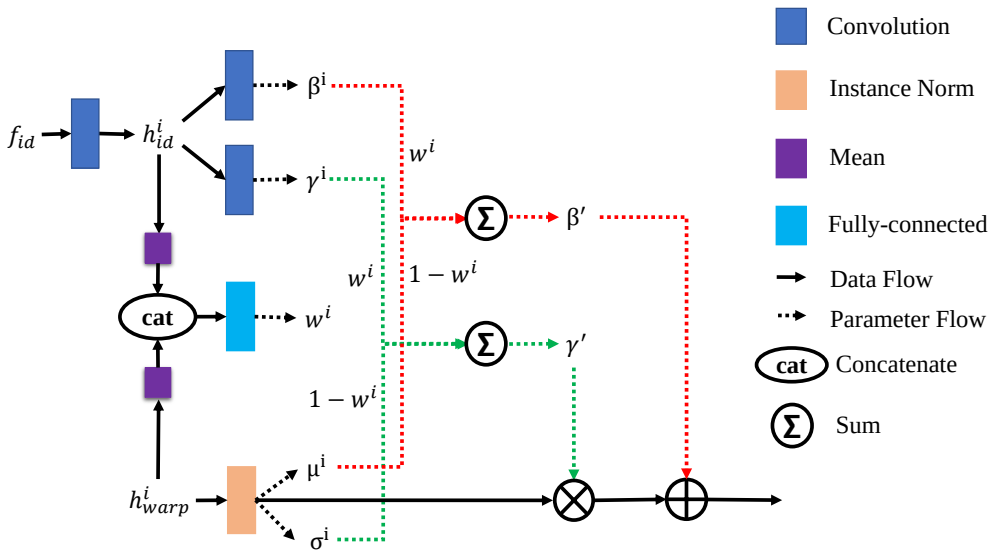


Fig. 3. **The detailed design of our elastic instance normalization**. Here, we normalize the features of the warped mesh h^i_{warp} with InstanceNorm and get the mean μ^i and standard deviation σ^i. Then, the features of the identity mesh are fed into a convolution layer to get h^i_{id}, which shares the same size with h^i_{warp}. We calculate the mean of h^i_{warp}, h^i_{id} and concatenate them in channel dimension. A fully-connected layer is employed to compute an adaptive weight w^i. We blend γ^i, σ^i and β^i, μ^i elastically with w^i to get γ' and β'. γ^i and β^i are learned from h^i_{id}. Finally, we scale the normalized h^i_{warp} with γ' and shift it with β'. The value on the parameter flow means the weight.

3D pose transfer without any supervision of the ground truth label.

3.3 ARAP deformer

To keep the original body shape of the generated results M_{output}, we adopt an as-rigid-as-possible (ARAP) deformer in the training process as shown in Figure 2,

$$\hat{M}_{output} = \text{ARAP}(M_A, M_{output}). \tag{8}$$

Here, M_A is the identity mesh for the generation of M_{output}. With the ARAP deformer, we deform M_A to match a few randomly fixed vertices of the generated mesh M_{output}. As shown in Figure 2, the deformed output $\hat{M}_{output}$ has better preservation of the body shape of the original identity mesh.

In the following, we introduce how ARAP works to preserve the body shape of the meshes. We define M as the original mesh and $\hat{M}$ as the deformed mesh. For the cell C_i (It includes vertex v_i and its one-ring neighbor $\mathcal{N}(i)$.) corresponding to vertex v_i and the deformed cell $\hat{C}_i$, if the deformation from C_i to $\hat{C}_i$ is rigid, then there is a rotation matrix $\mathbf{R}_i$ satisfies,

$$\hat{v}_i - \hat{v}_j = \mathbf{R}_i(v_i - v_j), \quad \forall j \in \mathcal{N}(i). \tag{9}$$

If the deformation between C_i and $\hat{C}_i$ is not rigid, we can still find the optimal rotation matrix that makes the deformation from C_i to $\hat{C}_i$ as rigid as possible, i.e., minimizes the energy

$$E(C_i, \hat{C}_i) = \sum_{j \in \mathcal{N}(i)} w_{ij} \left\| (\hat{v}_i - \hat{v}_j) - \mathbf{R}_i(v_i - v_j) \right\|^2, \tag{10}$$

where w_{ij} means the per-edge weight. In this work, we use the cotangent weighting for w_{ij}. ARAP uses an alternating minimization strategy to solve Equation 10. For a given set of vertex positions $\hat{v}$, it calculates the optimal rigid transformations $\{\mathbf{R}_i\}$ that minimize E. Then it updates the vertex positions $\hat{v}$ with the calculated $\{\mathbf{R}_i\}$ while minimizing E. These iterations continue until the local energy minimum is reached. For the detailed solving procedure please refer to [46].

In this work, we only add the ARAP deformer in the training loop to help keep the body shape of M_{output}. Different from Zhou *et al.* [40], we adopt ARAP on non-registered data. Under this circumstance, we have a trade-off between the training efficiency and the number of epochs that adds ARAP deformer. Through extensive experiments, we finally decided to add the ARAP deformer in the last 50 epochs for training.

3.4 Loss functions

3.4.1 *Reconstruction loss.*

Since our network is trained without the supervision of the ground truth label, we introduce a reconstruction loss to achieve the dual reconstruction objective in Section 3.2. The reconstruction loss is calculated using the $L2$ distance between the vertex coordinates of the reconstructed meshes and the original meshes. For the reconstruction of M_A,

$$\mathcal{L}_{A\_rec} = \|V'_A - V_A\|_2^2, \tag{11}$$

where V'_A and $V_A \in \mathbb{R}^{N_A \times 3}$ are the vertex coordinates of $M'_A = G'_A(\hat{M}_{output}, M_A)$ and M_A respectively. N_A is the vertex number. Similarly,

$$\mathcal{L}_{B\_rec} = \|V'_B - V_B\|_2^2, \tag{12}$$

where V'_B and $V_B \in \mathbb{R}^{N_B \times 3}$ are the vertex coordinates of $M'_B = G'_B(M_B, \hat{M}_{output})$ and M_B respectively. N_B is the vertex number.

With the reconstruction loss, we can achieve the 3D pose transfer without any supervision and the meshes generated by our model are very close to the ground truth.

3.4.2 Backward correspondence loss.

To learn better shape correspondence in the correspondence learning module, we also introduce a backward correspondence loss inspired by CorrNet3D [38]. As discussed in Section 3.1.1, we define the optimal matching matrix as $\mathbf{T}_m \in \mathbb{R}_+^{N_{id} \times N_{pose}}$ between the identity and pose meshes. With $\mathbf{T}_m$, we can warp the pose mesh and obtain the warped pose mesh M_{warp}.

Then we can reconstruct the pose mesh based on the transpose of the correspondence matrix and the warped mesh,

$$V'_{pose}(i) = \sum_j \mathbf{T}_m^\top(i,j) V_{warp}(j). \tag{13}$$

Here, $i \in [0, N_{pose})$ and $j \in [0, N_{id})$. Finally, the backward correspondence loss can be defined as,

$$\begin{aligned} \mathcal{L}_{corr} &= \left\| V'_{pose} - V_{pose} \right\|_2^2 \\ &= \left\| \mathbf{T}_m^\top \mathbf{T}_m V_{pose} - V_{pose} \right\|_2^2, \end{aligned} \tag{14}$$

we can also rewrite the backward correspondence loss as,

$$\mathcal{L}_{corr} = \left\| \mathbf{T}_m^\top \mathbf{T}_m - \mathbf{I}_{N_{pose}} \right\|_2^2, \tag{15}$$

where $\mathbf{I}_{N_{pose}}$ is the identity matrix whose size is $N_{pose} \times N_{pose}$. Minimizing Equation 14 is equivalent to minimizing Equation 15.

3.4.3 Edge loss.

Following several 3D mesh generation works [2], [47], [48], we adopt edge loss which can penalize flying vertices and make the connection between vertices smoother. Since the reconstruction loss does not consider the vertex connection, the predicted mesh can be affected by flying vertices, which usually leads to excessively long edges. Edge loss can help alleviate this issue. We can define the edge loss as,

$$\mathcal{L}_{edge} = \sum_v \sum_{v_\mathcal{N} \in \mathcal{N}(v)} \|v - v_\mathcal{N}\|_2^2, \tag{16}$$

where $v \in V_{output}$ is the vertex of the output mesh and $\mathcal{N}(v)$ is the neighbor of v.

To sum up, the total loss $\mathcal{L}$ for unsupervised 3D pose transfer is,

$$\mathcal{L} = \lambda_{rec}(\mathcal{L}_{A\_rec} + \mathcal{L}_{B\_rec}) + \lambda_{corr}\mathcal{L}_{corr} + \mathcal{L}_{edge}, \tag{17}$$

where λ_{rec}, λ_{corr} denote the weight of reconstruction loss and backward correspondence loss respectively.

4 Experiments

Datasets. We use the same human mesh dataset generated by SMPL [4] as NPT [2] and 3D-CoreNet [3]. This dataset consists of 30 identities with 800 poses. Each mesh in the dataset has 6890 vertices. For the training set, we randomly choose 4000 mesh pairs (identity and pose meshes) and shuffle them every epoch. All vertices of the meshes will be shuffled randomly before input to be close to the real-world problem. And we also shift the meshes to the center according to their bounding boxes. When doing the test, we randomly choose 400 unseen mesh pairs that are different from 3D-CoreNet [3] for evaluation. They are pre-processed in the same manner as the training data. To further verify the generalization capability of our approach, we also test it on FAUST [5] and MGN [49] in the experiments.

We use the animal mesh dataset generated by SMAL model [6] in this work. It includes 21 Felidae animals, 5 Canidae animals, 8 Equidae animals, 4 Bovidae animals, and 3 Hippopotamidae animals. Each mesh has 3889 vertices. For the training set, we randomly choose 8400 mesh pairs from 21 identities (10 Felidae animals, 3 Canidae animals, 4 Equidae animals, 2 Bovidae animals, and 2 Hippopotamidae animals) with 400 poses. And we randomly choose 400 unseen pairs for testing. The choice of testing data is also different from 3D-CoreNet [3]. The animal meshes are pre-processed in the same way as we do in the human dataset.

Evaluation metrics. To compare different methods quantitatively, we use Pointwise Mesh Euclidean Distance (PMD), Chamfer Distance (CD), and Earth Mover's Distance (EMD) as our evaluation metrics. For all of them, the lower is better. We compare the averages of these three metrics obtained on the test datasets for quantitative evaluation.

Implementation details. The values of hyperparameters in the loss function are $\lambda_{rec} = 2000$, $\lambda_{corr} = 200$. Our model is trained for 200 epochs with the Adam optimizer [50] on two RTX 3090 GPUs, the learning rate is initialized as $1e$-4 in the first 100 epochs and decays $1e$-6 each epoch from the 100th epoch. For more implementation details of the network, please refer to our supplemental material.

4.1 Baselines

Neural pose transfer. Neural pose transfer(NPT) [2] transfers the pose with spatially adaptive instance normalization (SPAdaIN). It does not consider any correspondence between different meshes. Therefore, the performance of the model will be degraded although their method is convenient. We train NPT using the implementations provided by the authors.

3D-CoreNet. We can achieve a supervised 3D pose transfer method, which is named 3D-CoreNet following [3]. 3D-CoreNet contains a single generator G and is trained with the supervision of the ground truth mesh M_{gt}. We first process the ground truth mesh to have the same vertex order as the identity mesh. Then we define a new reconstruction loss by calculating the point-wise $L2$ distance between the vertices of M_{output} and M_{gt},

$$\mathcal{L}_{rec} = \|V_{output} - V_{gt}\|_2^2 \tag{18}$$

where V_{output} and $V_{gt} \in \mathbb{R}^{N_{id} \times 3}$ are the vertices of M_{output} and M_{gt} respectively. Notice that they all share the same size and order with the vertices of the identity mesh. With the reconstruction loss, the mesh predicted by 3D-CoreNet will be closer to the ground truth. The inference of 3D-CoreNet and X-DualNet only needs a single generator.

Unsupervised baseline. We design the unsupervised baseline into two modules: cycle reconstruction and self reconstruction. It contains five generators. All five generators share the same network parameters. And we adopt three reconstruction losses to instruct the training. Compared with *X-DualNet*, the unsupervised baseline needs more time for training. Please refer to the appendices for the detailed structure of the proposed unsupervised baseline.

TABLE 1
Quantitative comparison. We compare X-DualNet with NPT, 3D-CoreNet, and the proposed unsupervised baseline on human and animal meshes. For the evaluation metrics, the lower is better. 3D-CoreNet has the best performance. X-DualNet achieves comparable performances as 3D-CoreNet and even outperforms NPT which is a fully supervised method.

Dataset	Type	Method	PMD ($\times 10^{-3}$)	CD ($\times 10^{-3}$)	EMD ($\times 10^{-2}$)
SMPL	*Sup.*	NPT	2.77	5.91	7.88
		3D-CoreNet	**0.49**	**0.90**	**3.61**
	Unsup.	Baseline	2.23	4.05	7.05
		Ours	1.67	2.71	5.96
SMAL	*Sup.*	NPT	8.01	17.76	12.57
		3D-CoreNet	**3.55**	**6.49**	**9.09**
	Unsup.	Baseline	16.50	37.54	23.62
		Ours	4.48	9.05	10.19

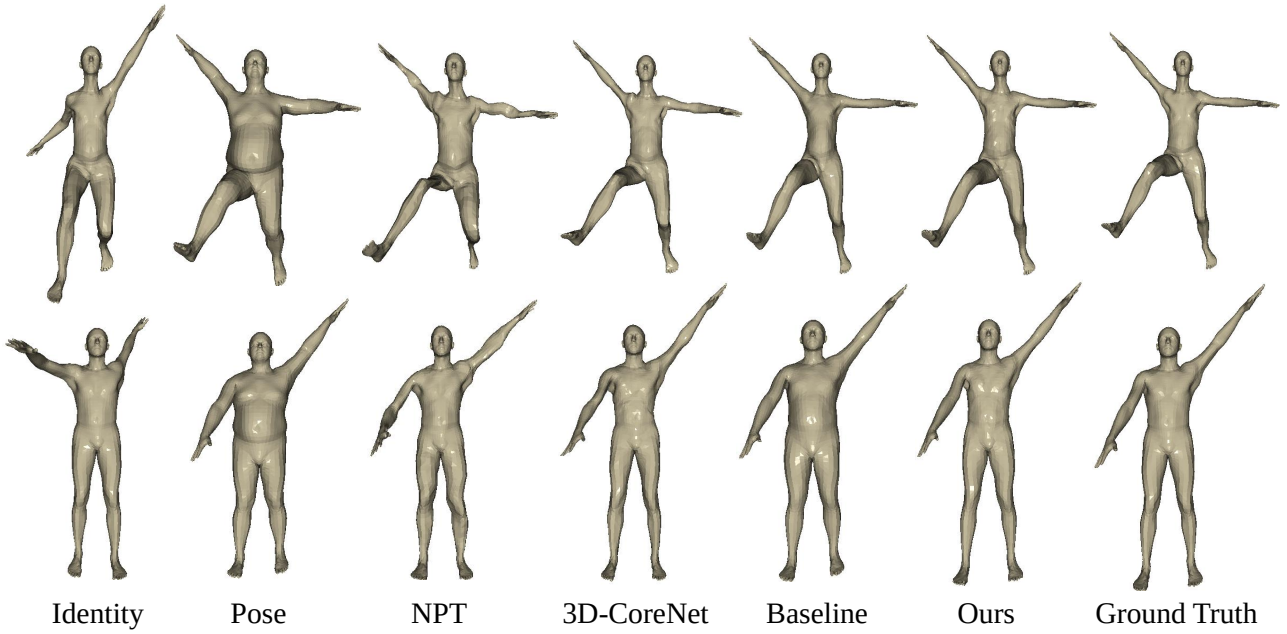


Fig. 4. **Qualitative comparison on human data.** Our method and 3D-CoreNet can generate better results than NPT and the proposed unsupervised baseline. The surfaces of meshes generated by NPT are not always smooth. And the proposed unsupervised baseline cannot preserve the body shapes very well. The human meshes are from SMPL [4].

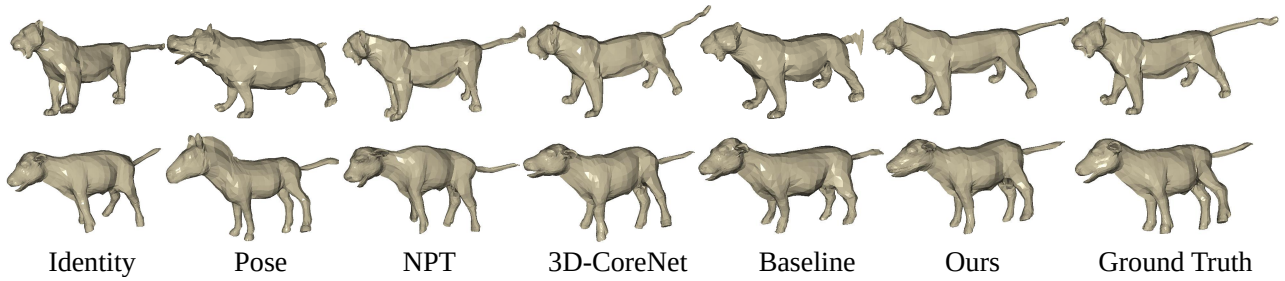


Fig. 5. **Qualitative comparison on animal data.** Our method and 3D-CoreNet produce satisfactory results on the animal data. NPT produces many artifacts and cannot transfer the pose successfully. The proposed baseline sometimes cannot preserve the shape identity (e.g., the tail) well. The animal meshes are from SMAL [6].

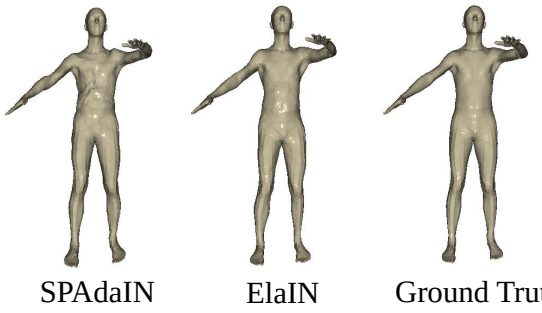


Fig. 6. **Comparison of different conditional normalization layers.** We compare our ElaIN with SPAdaIN used in NPT [2] on SMPL [4]. The surface of the mesh has clear artifacts and is not smooth when we replace ElaIN with SPAdaIN. Experiments are carried out on 3D-CoreNet.

4.2 Comparison with the state-of-the-art approaches

We compare *X-DualNet* with NPT [2], 3D-CoreNet, and our proposed unsupervised baseline. NPT and 3D-CoreNet are supervised with the ground truth.

Figure 4 provides the qualitative results tested on SMPL [4]. It can be clearly seen that our method and 3D-CoreNet can produce better results with the least artifacts. The surfaces of meshes generated by NPT are not smooth, especially on the legs and arms because NPT does not learn the shape correspondence. The proposed unsupervised baseline cannot preserve the body shapes well although it transfers the poses successfully. To be clear, we will discuss the reason in supplemental material following the detailed structure of the unsupervised baseline. The qualitative results tested on animal data are shown in Figure 5, NPT always fails to transfer the pose and produces many artifacts. The proposed baseline sometimes cannot keep the identity information (e.g., the tail). In comparison, our method and 3D-CoreNet still produce successful results.

To compare different methods quantitatively, we use PMD, CD and EMD as the evaluation metrics which are calculated from the ground truth and the generated meshes. We show the comparison results in Table 1. 3D-CoreNet has the best performance. Our *X-DualNet* outperforms NPT and the proposed unsupervised baseline in all metrics over two datasets although NPT is trained with the ground truth.

TABLE 2
Comparison of different conditional normalization layers. We compare our ElaIN with SPAdaIN used in NPT [2]. The lower is better. Experiments are carried out on 3D-CoreNet.

Dataset		SPAdaIN	ElaIN
SMPL	PMD	1.01	**0.49**
	CD	2.13	**0.90**
	EMD	4.66	**3.61**

4.3 Ablation studies

In this section, we study the effectiveness of several components in our model on human data.

ElaIN. We compare our ElaIN with SPAdaIN proposed in NPT [2] for 3D pose transfer to verify the effectiveness of ElaIN. To make the comparisons clearer, we conduct experiments on 3D-CoreNet, which requires only one generator since ElaIN works in the pose transfer module. When we replace our ElaIN with SPAdaIN, the surface of the mesh has clear artifacts and is not smooth as shown in Figure 6. The metrics are also worse than using ElaIN as shown in Table 2. We can know that ElaIN is very helpful in generating high-quality results.

ARAP deformer. We evaluate our *X-DualNet* without the as-rigid-as-possible (ARAP) deformer. The qualitative ablation study results are shown in Figure 7. When we do not add the ARAP deformer in the training loop, the generated results do not preserve the body shape well which can be shown in the green bounding boxes. In the third column, the body shape is sunken from the left and right sides. The quantitative ablation study analysis is shown in Table 3. As we can see, the quantitative performance of our method degrades when removing the ARAP deformer.

Backward correspondence loss. We test the effectiveness of the backward correspondence loss $\mathcal{L}_{corr}$ in *X-DualNet*. When

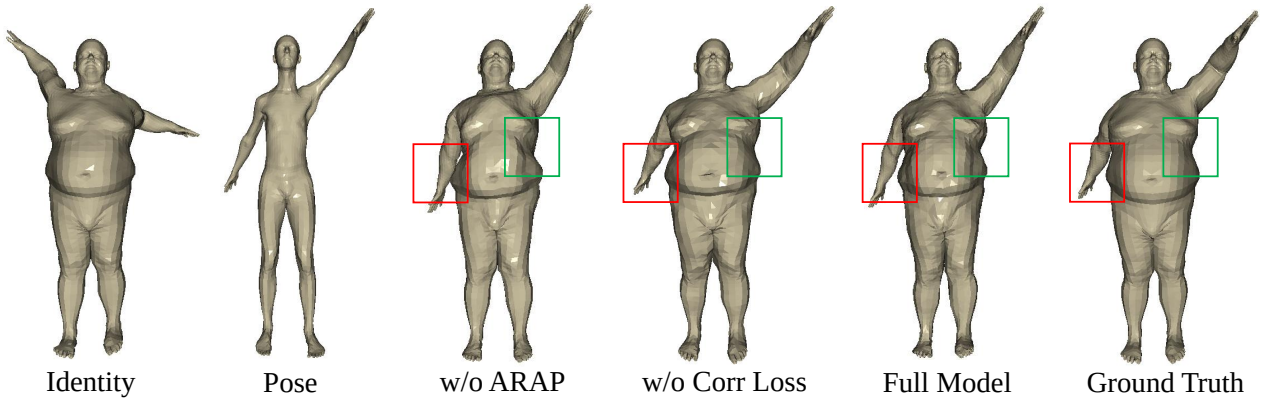


Fig. 7. **Qualitative ablation studies of two components in X-DualNet.** When we do not add the ARAP deformer in the training loop, the generated results do not preserve the body shape well which can be shown in the green bounding boxes. In the third column, the body shape is sunken from the left and right sides. When we remove the backward correspondence loss $\mathcal{L}_{corr}$ in the training, the pose transfer results are not accurate which can be shown in the red bounding boxes. In the fourth column, the right arm is farther from the body than others. And the reason why the right arm of the mesh in the third column looks so close to the body is that the body shape is not well maintained without ARAP deformer. The input human meshes are from SMPL [4]. w/o means we remove this component.

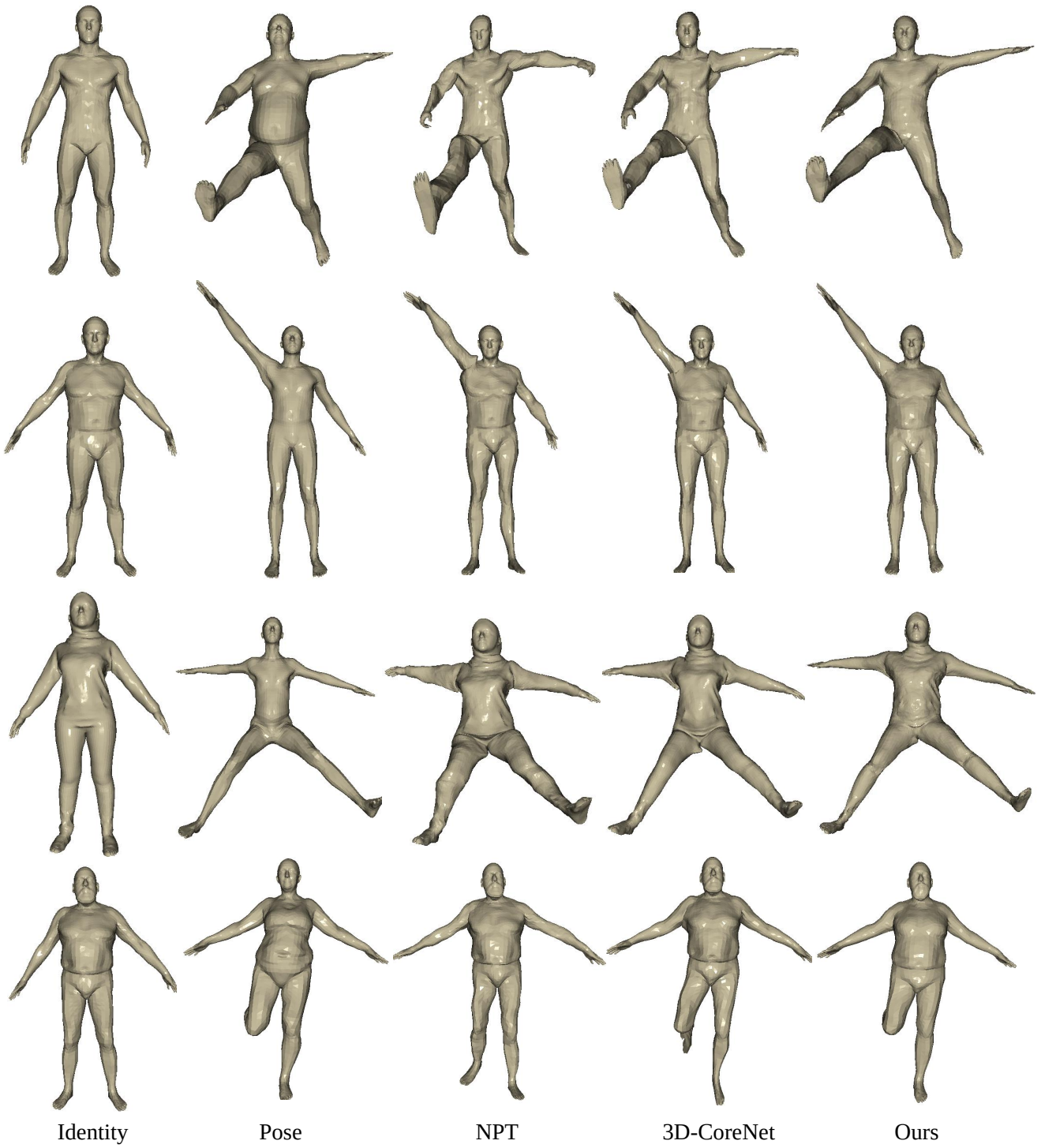


Fig. 8. **Pose transfer results on FAUST [5] and MGN [49].** We choose four pose meshes and transfer the poses to the unseen meshes in FAUST and MGN. To evaluate the generalization ability of X-DualNet, we compare it with NPT [2] and 3D-CoreNet. As shown in the first three rows, the results generated by NPT and 3D-CoreNet always have many artifacts on the mesh surfaces. Our results are smoother than theirs. In the last row, NPT does not transfer the pose successfully since they do not learn the shape correspondence. 3D-CoreNet also fails to transfer the pose accurately. Our method has the best performance when meeting unseen poses. More generated results will be given in the supplemental material.

TABLE 3
Quantitative ablation studies of two components in X-DualNet. We evaluate the importance of ARAP deformer and the backward correspondence loss. For PMD, CD, and EMD, the lower is better. w/o means we remove the component.

Dataset		PMD	CD	EMD
SMPL	w/o ARAP	1.98	3.37	6.60
	w/o $\mathcal{L}_{corr}$	1.84	3.20	6.35
	Full model	**1.67**	**2.71**	**5.96**

we remove $\mathcal{L}_{corr}$ in the training, the pose transfer results will not be accurate. As we can see from the red bounding boxes in Figure 7, the poses of the generated results that add the backward correspondence loss are more similar to the ground truth. In the fourth column, the right arm is farther from the body than others. And the reason why the right arm of the mesh in the third column looks so close to the body is that the body shape is not well maintained without ARAP deformer. The quantitative results are also worse after removing the backward correspondence loss. We will visualize the built shape correspondence between human or animal meshes in the supplementary materials.

4.4 Generalization capability

We evaluate *X-DualNet* on unseen datasets FAUST [5] and MGN [49] to verify the generalization capability of it. In FAUST, the human meshes have more unseen identities which share the same number of vertices with SMPL [4]. For human meshes in MGN, they are dressed in clothes and they all have 27,554 vertices.

As shown in Figure 8, we choose four pose meshes and transfer the poses to the human meshes in FAUST and MGN. Our method still produces high-quality results when testing on unseen datasets although the number of vertices is different. To further verify the generalization ability of *X-DualNet*, we compare it with NPT [2] and 3D-CoreNet. As shown in the first three rows, the results generated by NPT and 3D-CoreNet always have many artifacts on the mesh surfaces. Our results are smoother than both of them. In the last row, the input identity mesh and the pose mesh are both from FAUST. NPT does not transfer the pose successfully since they do not consider the correspondence between the identity and pose meshes. 3D-CoreNet also fails to transfer the pose accurately. Our unsupervised method has better performance when meeting unseen poses.

5 Conclusion

In this work, we introduce *X-DualNet*, a simple yet effective approach that enables unsupervised 3D pose transfer. In *X-DualNet*, we propose a generator G which contains correspondence learning and pose transfer modules to achieve the 3D pose transfer. We learn the shape correspondence by solving an optimal transport problem without any key point annotations and generate high-quality final meshes with our proposed elastic instance normalization (ElaIN) in the pose transfer module. To the best of our knowledge, our method is the first to learn the correspondence between different meshes and transfer the poses jointly in the 3D pose transfer task.

With the generator G as the basic component, we propose a cross consistency learning scheme and a dual reconstruction objective to learn the pose transfer without any supervision of the ground truth label. Besides that, we also adopt an as-rigid-as-possible (ARAP) deformer to help preserve the body shape of the generated results. Extensive experiments on human and animal meshes demonstrate that our unsupervised solution achieves comparable performance as the state-of-the-art supervised approaches qualitatively and quantitatively and even outperforms some of them. In addition, our method has better generalization capability than supervised methods. In the future, we will try to build a system to connect the pose transfer problems between 2D images and 3D representations.

Acknowledgments

This research is supported under the RIE2020 Industry Alignment Fund – Industry Collaboration Projects (IAF-ICP) Funding Initiative, as well as cash and in-kind contribution from the industry partner(s). This research is also supported by the National Research Foundation, Singapore under its AI Singapore Programme (AISG Award No: AISG-RP-2018-003), the Ministry of Education, Singapore, under its Academic Research Fund Tier 2 (MOE-T2EP20220-0007) and Tier 1 (RG95/20). This research is also supported by the Agency for Science, Technology and Research (A*STAR), Singapore under its MTC Young Individual Research Grant (Grant No. M21K3c0130).

References

[1] R. W. Sumner and J. Popović, "Deformation transfer for triangle meshes," *ACM Transactions on graphics (TOG)*, vol. 23, no. 3, pp. 399–405, 2004.

[2] J. Wang, C. Wen, Y. Fu, H. Lin, T. Zou, X. Xue, and Y. Zhang, "Neural pose transfer by spatially adaptive instance normalization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 5831–5839.

[3] C. Song, J. Wei, R. Li, F. Liu, and G. Lin, "3d pose transfer with correspondence learning and mesh refinement," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[4] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black, "Smpl: A skinned multi-person linear model," *ACM transactions on graphics (TOG)*, vol. 34, no. 6, pp. 1–16, 2015.

[5] F. Bogo, J. Romero, M. Loper, and M. J. Black, "Faust: Dataset and evaluation for 3d mesh registration," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 3794–3801.

[6] S. Zuffi, A. Kanazawa, D. W. Jacobs, and M. J. Black, "3d menagerie: Modeling the 3d shape and pose of animals," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 6365–6373.

[7] D. Maturana and S. Scherer, "Voxnet: A 3d convolutional neural network for real-time object recognition," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 922–928.

[8] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3d shapenets: A deep representation for volumetric shapes," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1912–1920.

[9] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 652–660.

[10] J. Li, B. M. Chen, and G. H. Lee, "So-net: Self-organizing network for point cloud analysis," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 9397–9406.

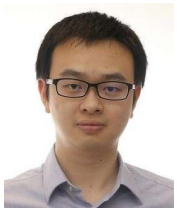
[11] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++ deep hierarchical feature learning on point sets in a metric space," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017, pp. 5105–5114.
[12] Y. Feng, Y. Feng, H. You, X. Zhao, and Y. Gao, "Meshnet: Mesh neural network for 3d shape representation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 8279–8286.
[13] Q. Tan, L. Gao, Y.-K. Lai, and S. Xia, "Variational autoencoders for deforming 3d mesh models," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 5841–5850.
[14] M. Garland and P. S. Heckbert, "Surface simplification using quadric error metrics," in *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, 1997, pp. 209–216.
[15] A. Ranjan, T. Bolkart, S. Sanyal, and M. J. Black, "Generating 3d faces using convolutional mesh autoencoders," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 704–720.
[16] I. Lim, A. Dielen, M. Campen, and L. Kobbelt, "A simple approach to intrinsic correspondence learning on unstructured 3d meshes," in *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, 2018, pp. 0–0.
[17] S. Gong, L. Chen, M. Bronstein, and S. Zafeiriou, "Spiralnet++: A fast and highly efficient mesh convolution operator," in *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, 2019, pp. 0–0.
[18] J. Yang, L. Gao, Y.-K. Lai, P. L. Rosin, and S. Xia, "Biharmonic deformation transfer with automatic key point selection," *Graphical Models*, vol. 98, pp. 1–13, 2018.
[19] I. Baran, D. Vlasic, E. Grinspun, and J. Popović, "Semantic deformation transfer," in *ACM SIGGRAPH 2009 papers*, 2009, pp. 1–6.
[20] H.-K. Chu and C.-H. Lin, "Example-based deformation transfer for 3d polygon models." *J. Inf. Sci. Eng.*, vol. 26, no. 2, pp. 379–391, 2010.
[21] L. Gao, J. Yang, Y.-L. Qiao, Y.-K. Lai, P. L. Rosin, W. Xu, and S. Xia, "Automatic unpaired shape deformation transfer," *ACM Transactions on Graphics (TOG)*, vol. 37, no. 6, pp. 1–15, 2018.
[22] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.
[23] 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*, 2020, pp. 75–83.
[24] P. Zhang, B. Zhang, D. Chen, L. Yuan, and F. Wen, "Cross-domain correspondence learning for exemplar-based image translation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 5143–5153.
[25] N. Courty, R. Flamary, D. Tuia, and A. Rakotomamonjy, "Optimal transport for domain adaptation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 9, pp. 1853–1865, 2016.
[26] Z. Su, Y. Wang, R. Shi, W. Zeng, J. Sun, F. Luo, and X. Gu, "Optimal mass transport for shape matching and comparison," *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 11, pp. 2246–2259, 2015.
[27] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *International conference on machine learning*. PMLR, 2017, pp. 214–223.
[28] C. Bunne, D. Alvarez-Melis, A. Krause, and S. Jegelka, "Learning generative models across incomparable spaces," in *International Conference on Machine Learning*. PMLR, 2019, pp. 851–861.
[29] I. Deshpande, Y.-T. Hu, R. Sun, A. Pyrros, N. Siddiqui, S. Koyejo, Z. Zhao, D. Forsyth, and A. G. Schwing, "Max-sliced wasserstein distance and its use for gans," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10 648–10 656.
[30] J. Wu, Z. Huang, D. Acharya, W. Li, J. Thoma, D. P. Paudel, and L. V. Gool, "Sliced wasserstein generative models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3713–3722.
[31] G. Puy, A. Boulch, and R. Marlet, "Flot: Scene flow on point clouds guided by optimal transport," *arXiv preprint arXiv:2007.11142*, 2020.
[32] Y. Liu, L. Zhu, M. Yamada, and Y. Yang, "Semantic correspondence as an optimal transport problem," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 4463–4472.
[33] X. Huang and S. Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1501–1510.
[34] Y. Chen, M. Chen, C. Song, and B. Ni, "Cartoonrenderer: An instance-based multi-style cartoon image translator," in *International Conference on Multimedia Modeling*. Springer, 2020, pp. 176–187.
[35] T. Park, M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu, "Semantic image synthesis with spatially-adaptive normalization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2337–2346.
[36] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*. PMLR, 2015, pp. 448–456.
[37] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Instance normalization: The missing ingredient for fast stylization," *arXiv preprint arXiv:1607.08022*, 2016.
[38] Y. Zeng, Y. Qian, Z. Zhu, J. Hou, H. Yuan, and Y. He, "Corrnet3d: Unsupervised end-to-end learning of dense correspondence for 3d point clouds," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 6052–6061.
[39] M. Eisenberger, D. Novotny, G. Kerchenbaum, P. Labatut, N. Neverova, D. Cremers, and A. Vedaldi, "Neuromorph: Unsupervised shape interpolation and correspondence in one go," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 7473–7483.
[40] K. Zhou, B. L. Bhatnagar, and G. Pons-Moll, "Unsupervised shape and pose disentanglement for 3d meshes," in *European Conference on Computer Vision*. Springer, 2020, pp. 341–357.
[41] H. Chen, H. Tang, H. Shi, W. Peng, N. Sebe, and G. Zhao, "Intrinsic-extrinsic preserved gans for unsupervised 3d pose transfer," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 8630–8639.
[42] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2020.
[43] B. Zhang, M. He, J. Liao, P. V. Sander, L. Yuan, A. Bermak, and D. Chen, "Deep exemplar-based video colorization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8052–8061.
[44] R. Sinkhorn, "Diagonal equivalence to matrices with prescribed row and column sums," *The American Mathematical Monthly*, vol. 74, no. 4, pp. 402–405, 1967.
[45] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *European conference on computer vision*. Springer, 2016, pp. 630–645.
[46] O. Sorkine and M. Alexa, "As-rigid-as-possible surface modeling," in *Symposium on Geometry processing*, vol. 4, 2007, pp. 109–116.
[47] 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/CVF International Conference on Computer Vision*, 2019, pp. 9964–9973.
[48] 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)*, 2018, pp. 52–67.
[49] B. L. Bhatnagar, G. Tiwari, C. Theobalt, and G. Pons-Moll, "Multi-garment net: Learning to dress 3d people from images," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 5420–5430.
[50] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

Chaoyue Song Chaoyue Song received the B.Eng. degree from Shanghai Jiao Tong University, China. He is currently a research engineer at Nanyang Technological University. His research interests include computer vision and machine learning, with a current focus on 3D vision and generative models.

Jiacheng Wei Jiacheng Wei received his B.Eng. in Electronic Information Engineering from University of Electronic Science and Technology of China and B.Eng. in Electronics and Electrical Engineering with Honours of the First Class from University of Glasgow in 2017. He is currently working towards the Ph.D. degree in School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore. His research interests include computer vision and artificial intelligence.

Ruibo Li Ruibo Li is a PhD candidate with the School of Computer Science and Engineering, Nanyang Technological University, Singapore. He received a B.Eng. degree and a M.S. degree from Huazhong University of Science and Technology, Wuhan, China in 2016 and 2019 respectively. His research interests are in computer vision and machine learning.

Fayao Liu Fayao Liu is a scientist at Institute for Infocomm Research (I2R), A*STAR, Singapore. She received her PhD in computer science from University of Adelaide, Australia in Dec. 2015. She mainly works on machine learning and computer vision problems, with particular interests in self-supervised learning, few-shot learning, and generative learning.



Guosheng Lin Guosheng Lin received the bachelor's and master's degrees from the South China University of Technology, in 2007 and 2010, respectively, and the PhD degree from the University of Adelaide, in 2014. He is an assistant professor with the School of Computer Science and Engineering, Nanyang Technological University, Singapore. His research interests include computer vision and machine learning.