Computational Visual Media
https://doi.org/10.1007/s41095-023-0379-8

Research Article

Geometry-aware 3D pose transfer using transformer autoencoder

Shanghuan Liu[1], **Shaoyan Gai**[1] (🖂), **Feipeng Da**[1], **and Fazal Waris**[1]

Abstract 3D pose transfer over unorganized point clouds is a challenging generation task, which transfers a source's pose to a target shape and keeps the target's identity. Recent deep models have learned deformations and used the target's identity as a style to modulate the combined features of two shapes or the aligned vertices of the source shape. However, all operations in these models are point-wise and independent and ignore the geometric information on the surface and structure of the input shapes. This disadvantage severely limits the generation and generalization capabilities. In this study, we propose a geometry-aware method based on a novel transformer autoencoder to solve this problem. An efficient self-attention mechanism, that is, cross-covariance attention, was utilized across our framework to perceive the correlations between points at different distances. Specifically, the transformer encoder extracts the target shape's local geometry details for identity attributes and the source shape's global geometry structure for pose information. Our transformer decoder efficiently learns deformations and recovers identity properties by fusing and decoding the extracted features in a geometry attentional manner, which does not require corresponding information or modulation steps. The experiments demonstrated that the geometry-aware method achieved state-of-the-art performance in a 3D pose transfer task. The implementation code and data are available at `https://github.com/SEULSH/Geometry-Aware-3D-Pose-Transfer-Using-Transformer-Autoencoder`.

Keywords 3D pose transfer; geometry-aware; transformer autoencoder; cross-covariance attention

1 Introduction

Three-dimensional (3D) pose transfer has attracted increasing interest in the computer vision and graphics community for decades. This has enabled several applications such as generating new poses and animation sequences [1, 2] for making 3D movies and games. This is challenging when the given shapes differ significantly in intrinsic attributes, such as identity. For computing and transferring deformations, traditional methods have attempted to utilize mesh topology and build correspondences between the source and target meshes, such as key point annotations [3], skeleton poses [4], and auxiliary meshes [5]. Unfortunately, the information requires considerable effort and additional steps by users, limiting their applications to more practical 3D shape data. Recently, studies based on deep learning have been conducted for 3D pose transfer. Skeleton-Free [6] achieved pose transfer between different body proportions and topological structures by representing characters in an implicit unified articulation model and predicting the skinning weights and deformation transformations in a learning manner. However, the method requires part-level correspondence and an additional rest pose shape and cannot transfer poses to target shapes with arbitrary poses. The studies in Refs. [7, 8] disentangled or aggregated poses and identities in latent spaces for pose transfer between meshes. However, they still require a dense correspondence or a mesh topology and cannot be applied to shapes with arbitrary poses.

In this study, we focus on deep-learning-based works that deal with point cloud shapes without a consistent mesh topology and vertex order for the pose transfer task. We aim to solve some of the problems associated with the existing related methods. Inspired by the work [9–12] on style transfer

[1] Key Laboratory of Measurement and Control of Complex Systems of Engineering, Ministry of Education, Southeast University, Nanjing, 210096, China. E-mail: S. Liu, 230189545@seu.edu.cn; S. Gai, 101011375@seu.edu.cn (🖂); F. Da, dafp@seu.edu.cn; F. Waris, fazalwaris@seu.edu.cn.

Manuscript received: 2023-03-13; accepted: 2023-09-09

in 2D images, some models have achieved 3D pose transfer by considering the identity of the target shape as a style and modulating the pose of the source shape in terms of latent features. NPT [13] was the first neural framework proposed for learning and modulating the deformation of the target shape. It does not require a correspondence between the target and source shapes; however, this study generated degraded shapes when transferring poses. To improve the generation ability, the geometry-contrastive transformer (GCT) [14] follows the NPT and uses the self-attention mechanism in the decoder to perceive the geometry inconsistency information between the source and target shapes. For the 3D-CoreNet [15], the authors proposed a network architecture for both learning correspondences and deformation, in which the learned correspondences guide the deformation and modulation of the target shape. The work [16] presents an unsupervised version of 3D-CoreNet that transforms supervised training into self-supervised loss using cross-consistency and dual reconstruction. The method's key components and processes are identical to those of 3D-CoreNet. However, these studies ignored the geometric information of the input shapes across their models. In each layer of the encoder and decoder, the point-wise input features are mapped independently of the output point-wise features, which do not consider the correlations of different points on each shape. Their modulation operations use point-wise features of the target shape to modulate the input features in a point-by-point manner, where the point-wise features are obtained simply by mapping the coordinate values of the point to high-dimensional features. Therefore, the latent features in all layers of their models do not contain geometric information, such as the local surface details of the target shape and the global structure information of the source shape. Therefore, the loss of local geometry information in identity features can explain why they repeatedly utilize the target shape to modulate latent features in decoders, which leads to complex frameworks.

Transformers have been successfully used in natural language processing and are increasingly being adopted in several computer vision applications. Additional applications of attention mechanisms and transformers in computer vision can be found in surveys [17, 18]. The self-attention modules in transformers help capture correlations that are beneficial to their tasks. Motivated by these studies on transformers, we propose a transformer autoencoder to directly achieve a geometry-aware 3D pose transfer. Our transformer encoder extracts identity features with local geometric details and poses features using the global structural information. Our transformer decoder, also in a geometry-aware manner, fuses concatenated features of the identity and pose features and decodes them to learn the deformation and generate the deformed shape while maintaining the identity of the target shape and the pose of the source shape. Because the identity features from our encoder contain complete geometric information, our framework does not need to modulate the latent features in the decoder using the information of the target shape.

The contributions of this study can be summarized as follows:

(1) We present a transformer autoencoder for geometry-aware 3D pose transfer, which leverages the transformer's self-attention mechanism to effectively capture the geometric information of the input shapes and fuses them to learn the deformation of the target shape.

(2) Our framework has a simple and efficient structure with fewer parameters, which does not require a correspondence between the target and source shapes or multiple modulation steps using the style of the target shape.

(3) Our method outperforms other deep models and achieves state-of-the-art results in the evaluation experiments of generation and generalization abilities.

2 Related works

2.1 3D pose transfer

2.1.1 Traditional methods

Previous studies on 3D pose transfer have primarily focused on mesh shapes. They aimed to obtain a deformed mesh with the desired pose and identity, where the gestures and identities were from two given meshes with different identities and poses. Although traditional methods perform well in this task, they primarily rely on the mesh topology and sparse or dense correspondence between the target and source meshes, which limits their applications. DT [19] used manually specified landmarks to build face correspondences between the source and target meshes to transfer deformations in an optimized manner. Mesh topology information is also required to calculate the final deformed vertices. To compute and transfer deformation to mesh sequences, methods in Refs. [20–22] require correspondences of all points on the meshes. Instead of directly transferring deformation between meshes, Refs. [23–25] adopted different harmonic functions to represent spatial deformation, where they required user-selected correspondence points to generate the deformation. In addition to these techniques, Ref. [26] attempted to construct maps between the pose spaces of the source and target meshes and generated the projected deformation in the target shape. Unfortunately, a dense correspondence is still required to calculate and transfer the deformation. Reference and source meshes are also necessary to obtain the deformation in traditional methods, where the pose of the target mesh must be similar to that of the reference mesh. These disadvantages constrain the application of these methods from being applied to more general data, such as shapes with unordered points and arbitrary poses.

2.1.2 *Learning methods*

In recent years, deep-learning-based methods have been proposed for 3D pose transfer. Some studies have disentangled [27] or aggregated [28] the identity and pose features in learning methods; however, they still have extra constraints on the meshes. Chen et al. [29] attempted to decompose meshes into identity and pose latent representations using dual autoencoders; however, the training process requires dense correspondences and mesh topology information. Wang et al. [30] used dual-mesh autoencoders to predict shape cages for calculating and transferring deformations; however, this method cannot deal with shapes with arbitrary poses. Skeleton-free methods also need to acquire correspondence and cannot be applied to shapes with arbitrary poses. To overcome these limitations, some studies have proposed learning approaches based on a similar style transfer strategy for 2D images. NPT presented the first neural model for 3D pose transfer on 3D shapes with point clouds, where correspondences between the target and source shapes are unnecessary. However, this method only processes point-wise features independently in all model operations, discounting the geometric information of the input shapes. Consequently, the model exhibits distorted shapes. To improve the quality of pose transfer, GCT follows NPT and adds a self-attention module before each modulation layer to perceive the geometry-contrastive information between the source and target mesh. In addition, this model does not identify correlations between the points of each shape. Each self-attention layer applied to the current features of the target and source shapes was used only to find the geometric inconsistency information between the two shapes for learning the deformation. 3D-CoreNet proposes a new strategy for designing a pose-transfer framework for learning correspondences and deformation simultaneously, where the obtained correspondences help void artifacts in the deformation phase. However, the entire network and its unsupervised version learn and modulate the pointwise features independently. Although these methods have made significant breakthroughs in 3D pose transfer, they ignore the geometric information of each shape, such as the local geometry details of the target shape and global pose information of the source shape. The loss of local geometric features in their encoders also causes these methods to repeatedly modulate features using the target shape in their decoders. In this study, we focus on the geometric information of identity and pose shapes across the entire network. Based on our transformer's self-attention mechanism, we achieved geometry-aware 3D pose transfer in a simple and efficient manner without correspondences and without modulating the latent features by the target shape.

2.2 Transformers in 3D vision

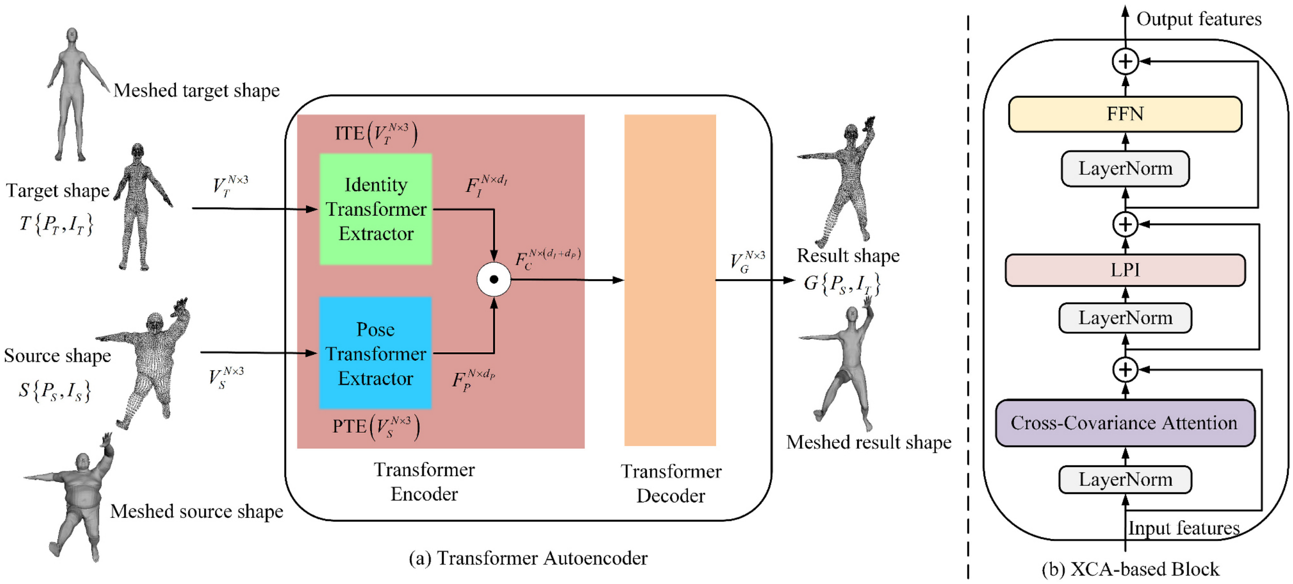
An original transformer model [31] was proposed for a language task, which showed a powerful inference ability for temporal sequences. The ideal structure of the transformer was later naturally adopted in vision tasks, such as image recognition [32], 3D mesh reconstruction [33, 34], and 3D object detection [35, 36]. However, the self-attention operation underlying transformers results in quadratic complexity in terms of time and memory, which hinders their application in long sequences and high-resolution images or point clouds. To solve this problem, Xcit [37] proposed a transposed version of self-attention called cross-covariance attention (XCA), which has a linear complexity with respect to the number of input tokens. This study demonstrated the strong ability of an image transformer to embed XCA. Based on the XCA mechanism and the original transformer architecture, Chandran et al. [38] constructed topology-independent 3D shape models for modeling 3D face and body shapes. In contrast to these studies using transformers, our transformer autoencoder model exploits the transformer's self-attention mechanism to capture the required geometric information of the input shapes and learn the deformation to achieve the 3D pose transfer task. The encoder and decoder of our model are based on transformers with XCA blocks, where we modify the XCA block in Xcit to deal with unorganized 3D point cloud data.

3 Method

This section introduces our transformer autoencoder network for achieving geometry-aware 3D pose transfer. Our model leverages the self-attention mechanism of the transformer in Xcit to effectively capture the nonlinear spatial correlation between two arbitrary points in each shape. This enables our encoder to extract features that are beneficial to our task and simplifies the process of the decoder to deform the target shape. Unlike existing deep models that learn point-wise features independently, our autoencoder focuses on the local surface details of the target shape and global structural features of the pose shape, thereby generating high-quality deformed shapes. Instead of adopting the style transfer method for 2D images, our model only requires the extraction and fusion of features that contain identity and pose information. Finally, deformation and identity are learned and decoded from the fused features. Therefore, the proposed method does not repeatedly use the shape of the input identity to modulate the deep features of the decoder.

3.1 Framework and XCA-based block

As shown in Fig. 1(a), our transformer autoencoder contains an identity transformer extractor (ITE), a pose transformer extractor (PTE), and a transformer decoder, where the modules ITE and PTE make up the transformer encoder. The input target shape $T\{P_T, I_T\}$ and the source shape $S\{P_S, I_S\}$ have different poses and identities, where P and I denote the pose and identity information, respectively. From the meshed figures, we can observe different identities. Each input shape in our framework is an unorganized point cloud with N disordered vertices $V^{N\times 3} \subseteq R^{N\times 3}$. We aimed to transfer the source shape's pose to the target shape and keep the target shape's identity. Thus, the generated shape, which is also a point cloud, can be denoted as $G\{P_S, I_T\}$ and has the same vertex order as the input target shape. The module ITE is responsible for extracting the target shape's identity features $F_I^{N\times d_I}$ and focuses on the



(a) Transformer Autoencoder

(b) XCA-based Block

Fig. 1 Framework of our transformer autoencoder (a) and the transformer unit block (b) based on the cross-covariance attention.

geometric information of the local surface details of the shape. The module PTE is in charge of obtaining the source shape's pose features $F_P^{N\times d_P}$ and attempts to remove its identity information and retain the pose component with the global structure of the shape. The variables d_I and d_P denote the corresponding feature channel sizes. After obtaining the identity and pose features, the transformer decoder processes the combined information $F_C^{N\times(d_I+d_P)}$ to deform the target shape and generate a new target shape that maintains its identity and has the pose of the source shape. The procedure for our framework can be described as Eq. (1):

$$\begin{aligned} G\left\{P_S, I_T\right\} &= Decoder\left(F_C^{N\times(d_I+d_P)}\right) \\ &= Decoder\left(\left[\mathrm{ITE}\left(V_T^{N\times 3}\right), \mathrm{PTE}\left(V_S^{N\times 3}\right)\right]\right) \end{aligned} \tag{1}$$

where [] denotes a concatenation operation.

The three submodules of our autoencoder are built from XCA-based blocks and other operations. The structure of the XCA-based block is shown in Fig. 1(b). We designed a unit block based on the transformer layer in Xcit by revising its original structure to suit our 3D shapes. Specifically, we modified the local patch interaction (LPI) and feedforward network (FFN) layers. The LPI layer was changed to one consisting of a sequence of operations {Conv1D, GeLU, BatchNorm1D, Conv1D}. The FFN layer includes the operations {Linear, GeLU, Linear}. The crucial part of an XCA-based block is the cross-covariance attention layer, which enables the capture of correlations on high-resolution images or point clouds. The previous self-attention mechanism computes the entire pairwise interaction between the input tokens, where the time and memory complexity increase quadratically with the number of input tokens. XCA substitutes it by designing self-attention among the features. The attention map is represented by a cross-covariance matrix computed over the key and query projections of the input-token features. The XCA function on the points can be defined as Eq. (2):

$$\begin{aligned} \text{XC-Attention}\left(Q, K, V\right) &= VA_{\mathrm{XC}}\left(K, Q\right) \\ &= V\mathrm{Softmax}\left(\hat{K}^{\mathrm{T}} \hat{Q} / \tau\right) \end{aligned} \tag{2}$$

where $Q \in R^{N\times d}$, $K \in R^{N\times d}$, and $V \in R^{N\times d}$ are the queries, keys, and values mapped from the input tokens, respectively. The attention weights A_{XC} are computed based on the cross-covariance matrix $\hat{K}^{\mathrm{T}} \hat{Q}$, where the query matrix Q and key matrix K are normalized to matrices $\hat{Q}$ and $\hat{K}$ by l_2-normalising, respectively. Therefore, the $d \times d$ cross-covariance matrix elements are within the range $[-1, 1]$. Moreover, the new self-attention mechanism introduces a learnable temperature parameter τ to scale the inner products before Softmax. This allows for a sharper or more uniform distribution of attention weights. XCA can also divide features into h groups instead of allowing all features to interact with each other, which further reduces the computational complexity and speeds up the optimization process. Further details regarding the XCA mechanism are available in Xcit.

3.2 Identity transformer extractor

In the 3D pose transfer task, we focus on the identity features of the target shape and eliminate the influence of its pose features. The identity information of a shape largely depends on local geometry information, such as the details and curvatures of its local surfaces. Therefore, the encoder module for 3D pose transfer should be able to capture the local geometry-aware representations of the input target shape. It is natural to consider deriving the identity information from the correlations of local points on the target shape. However, it is not easy to manually specify the relevant local 3D points for the complex geometric domains of human and animal bodies. Previous studies have used the coordinate values or deep features of each point on the target shape as identity information. The deep features are independently mapped from the coordinate values of each point by layers consisting of Conv1D operations. Consequently, the point-wise features in their encoders do not contain the local geometry information of the target shape, which severely affects the results of 3D pose transfer.

In contrast to previous models, we present an identity transformer extractor in our encoder to obtain identity features using local geometry information. The structure of the module ITE is shown in Fig. 2. ITE first processes the coordinate values of points on the input target shape using several layers consisting of Conv1D operations and ReLU activation functions. These layers map the

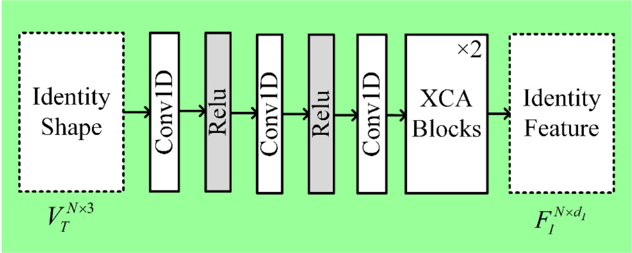


Fig. 2 Identity transformer extractor.

points onto high-dimensional latent positions. These operations enable our network to stretch and squeeze the distribution of the input shapes, which becomes an optimal distribution for our task. Therefore, using the initial layers before putting the tokens into XCA blocks makes our model more general, instead of being affected by the spatial distribution of 3D shapes. After obtaining the latent tokens, we use a sequence of two XCA blocks to extract the identity features of the target shape. In each transformer block, the XCA layer with its self-attention mechanism dynamically learns the correlations between points within arbitrary distance scopes without dictating the size of the local receptive field prior. The channel dimensions of the tokens after each XCA block are not changed. As network training proceeds, the identity transformer extractor gradually acquires a powerful ability to perceive the local geometry information of the target shape. It is an automatic and data-driven method for learning the identity information of a target shape.

3.3 Pose transformer extractor

In contrast to processing the target shape, the modules for handling the source shape in a 3D pose-transfer task should focus on the pose information of the shape and remove its identity attributes. The pose information of a shape relies primarily on its global features, such as its extension structure. Nevertheless, a global vector obtained by a max-pooling operation on each feature channel of all point-wise features cannot represent complete pose information. Accordingly, the pose information of the source shape can be extracted from the correlations between points at long distances. In encoders of previous methods, the authors used layers that are made up of Conv1D and Instance Norm operations to learn the pose features of the source shape. Instance Norm operations can eliminate identity properties. However, the Conv1D operations in their encoders still deal with point-wise features of the source shape independently, rendering them insufficient for extracting desirable pose information.

A pose transformer extractor was designed to learn the pose features of the source shape. The structure of the PTE module is shown in Fig. 3, which comprises two similar layers and an additional layer. Each of the two similar layers has a Conv1D operation, XCA block, instance norm operation, and ReLU activation function. The Conv1D operation maps the input features to higher-dimensional features. The XCA block, based on its self-attention, derives the global pose structure information of the source shape, which attempts to learn the correlations between long-distance points during training. To eliminate the identity features of the source shape that are reinforced again by each XCA block, we adopt an instance norm operation after the XCA block. Finally, the pose features are mapped to the latent features by an additional layer that includes a Conv1D operation, an instance norm operation, and a ReLU activation function. Throughout the processing of the three layers, our pose transformer extractor can efficiently learn the pose information using the global geometry information of the source shape.

3.4 Transformer decoder

Similar to previous methods, we concatenate the identity features $F_I^{N\times d_I}$ and pose features $F_P^{N\times d_P}$ to obtain the concatenated features $F_C^{N\times(d_I+d_P)}$ after the encoder. The difference in the concatenated features between our work and previous methods is that our point-wise features contain geometric information, which consists of local identity details of the target shape and global structure contents of the source shape. Owing to the loss of geometric features, NPT, GCT, and 3D-CoreNet must repeatedly modulate the latent features in their decoder layers. In their modulation operations, the input features of each layer are simply multiplied by the point-wise features of the target shape, which cannot

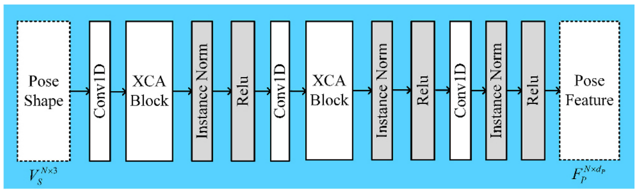


Fig. 3 Pose transformer extractor.

learn the complete identity information. Moreover, the Conv1D operations on point-wise features in their decoders remain independent, leading to inefficient deformation learning from point-wise features. Therefore, decoders are modules with complex structures that can easily result in target shapes with geometric distortion.

As shown in Fig. 4, the concatenated features with geometric information are passed into our transformer decoder, which is composed of Conv1D operations, XCA blocks, and activation functions. First, we used a Conv1D layer to merge the point's identity and pose information. The second and third Conv1D operations adjust the input features from higher to lower dimensions. After the first and second Conv1D layers, two XCA blocks exist for fusing and decoding the features between arbitrary points. Benefiting from the self-attention mechanism in each XCA block, our decoder recovers the identity of the target shape from fused features with local geometry-aware identity features. It also deforms the target shape under the guidance of fused features using global geometry-aware pose features. Our transformer decoder has a simple structure that does not require multiple modulations using the target shape.

3.5 Loss function

We trained our model by minimizing a combined loss function that includes a reconstruction loss, an edge loss, and a regularization term. Similar to previous studies, we also conducted training under the supervision of ground truth shapes, where each ground truth shape had the same vertex order as the target shape. Therefore, we can obtain the reconstruction loss by calculating the point-wise distance of the corresponding vertices between the deformed target shape and the ground truth shape as Eq. (3):

$$L_{\text{rec}} = \|V_{\text{gt}} - V_{\text{generate}}\|_2^2 \tag{3}$$

$V_{\text{gt}} \in R^{N\times 3}$ and $V_{\text{generate}} \in R^{N\times 3}$ are the corresponding vertices between the ground truth and generated shape, respectively. The reconstruction loss is a direct condition that causes a network to rapidly converge to the ground truth. We followed NPT and used edge loss to consider the connectivity of the mesh vertices. The edge loss can be defined as

$$L_{\text{edg}} = \sum_{v} \sum_{p\in N(v)} \|v - p\|_2^2 \tag{4}$$

where points $N(v)$ are the neighbors of point $v \in V_{\text{generate}}$. The neighbors of each vertex in the output shape are the same as those of the corresponding vertex in the target shape, which can be obtained using an indexical dictionary based on a random sequence used to disorder the original vertices. Therefore, we can easily calculate the edge loss to help restrain flying vertices and generate smoother surfaces. In addition, we introduce L2 regularization as a loss term to prevent overfitting in our model. The term is defined as

$$L_{\text{L2}} = \frac{1}{2n} \sum_{w} w^2 \tag{5}$$

where n is the total number of weight parameter w in the proposed model. Consequently, the final combined loss function is

$$L = \lambda_{\text{rec}} L_{\text{rec}} + \lambda_{\text{edg}} L_{\text{edg}} + \lambda_{\text{L2}} L_{\text{L2}} \tag{6}$$

λ_{rec}, λ_{edg}, and λ_{L2} denote the weights of the reconstruction, edge, and L2 regularization losses, respectively.

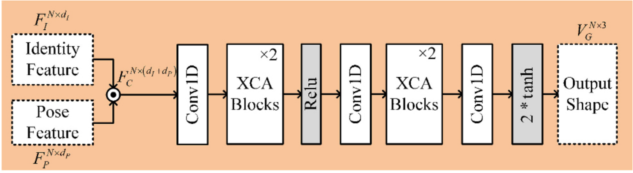


Fig. 4 Transformer decoder.

4 Experiments

In this section, we evaluate our method both quantitatively and qualitatively using human and animal datasets. Because the proposed deep-learning-based work deals with unorganized point clouds, our experiments follow the evaluations in previous studies. NPT was the first work to achieve 3D pose transfer on non-corresponding shapes in a deep learning manner. 3D-CoreNet is by far the state-of-the-art related method for 3D pose transfer tasks on disordered point clouds, surpassing the traditional DT and deep-learning-based methods, NPT and GCT. The main principle of the supervised version of 3D-CoreNet is the same as that of 3D-CoreNet. Therefore, we selected NPT and 3D-CoreNet as the main baseline models to demonstrate the effectiveness of our proposed approach. Even though the skeleton-free approach requires a reference pose and can only transfer the pose to the target shapes with a rest pose, we also compare our method with the method qualitatively. The experiments prove that our method outperforms these contrasting methods in terms of generation and generalization capabilities. In ablation studies, we verified the effectiveness of the proposed submodules based on XCA blocks and analyzed the influence on our model caused by the number of XCA blocks. We also analyzed the impact of different combined loss terms for training our network and proved the effectiveness of the L2 regularization loss term. Finally, we present a simple analysis of the limitations and robustness of the proposed method.

4.1 Experimental setup

4.1.1 Datasets

For the human body shapes, we used the mesh dataset in the NPT, which was generated using SMPL [39]. The dataset included 30 identities, each of which had 6890 vertices and 800 poses that shared the same topology. When training our network, we generated 6400 pairs from 16 identities with 400 poses and shuffled these pairs every epoch. Each pair had an identity (target) and pose (source) mesh. The ground-truth mesh was generated using the identity parameters of the target mesh and the pose parameters of the source mesh for each pair. To make the obtained model invariant to the vertex order, the mesh vertices were shuffled randomly before being fed into the network. Accordingly, the ground-truth mesh was shuffled to the same vertex order as the identity mesh for supervised training and evaluation. To evaluate our model, we randomly selected 400 pairs from the 14 unseen identities and 400 poses as the test dataset. To evaluate the generalization ability of our model further, we employed human shapes from FAUST [40] and MG [41]. For animal shapes, we used a dataset generated using SMAL models [42]. The dataset contained 41 identities and 60 poses, and each mesh contained 3889 vertices. The 41 identities included different animals, such as cats, dogs, deer, cows, and hippos. The training dataset contained 11,600 pairs comprising 29 identities and 400 poses. The test dataset comprised 400 randomly selected pairs of other identities and poses. For fair comparison with the baseline models, we used the same training and test datasets from the human and animal datasets. For all input shapes from the human and animal datasets, we shifted them to the center according to their bounding boxes.

4.1.2 Implementation details

In the pose transformer extractor, the output channels of the Conv1D filters from left to right are [64,128,256]. We set the number of groups in the two XCA blocks as 4 and 8, which split features, and kept each group's feature channel at 16. In our identity transformer extractor, the output sizes of the Conv1D filters were [64,128,256]. The number of groups in the XCA blocks was set to 16. In our transformer decoder, the output size of the Conv1D filters was [256,128,3]. The number of groups in the XCA blocks was set to [16,16,8,8]. In all XCA blocks of our network, the output sizes of the attention modules for the query, key, and values were the same as those of the input features. The output sizes of the Conv1D filters and the MLP in the LPI and FFN layers were the same as the channels of the input features. To train our network, we specify λ_{rec} and λ_{edg} of the loss functions (Eq. (6)) as 1000 and 0.5, respectively. To implement the L2 regularization loss, we chose the AdamW optimizer [43] and set the weight decay $\lambda_{\mathrm{L2}} = 0.0005$. Using PyTorch [44] on a single GTX 1080Ti GPU, we trained our models for 200 epochs with batch sizes of 8 for humans and 12 for animals. We started by adjusting our learning rate from 100 epochs using a fixed decay, where the decay value was 5×10^{-6}. The initial learning rate was set to 0.0005.

Following NPT, we utilized the point-wise mesh Euclidean distance (PMD) as an evaluation metric for our models. The PMD is the l_2 distance of the corresponding vertices between the output and ground-truth shapes. Similar to the 3D-CoreNet, we evaluated our models using the Chamfer distance (CD) proposed in Ref. [45].

4.2 Quantitative comparison

Based on the PMD and CD metrics, we compared our model with the baseline models NPT and 3D-CoreNet on test datasets of human and animal shapes. For the PMD and CD values, the lower is better. The units of PMD and CD are 10^{-3}. As shown in Table 1, the model's PMD and CD values on the two test datasets were the lowest, indicating that our method achieved state-of-the-art results on the 3D pose-transfer task. Our model reduced 85% of NPT and 43% of 3D-CoreNet on human shapes about PMD metric. In the CD metric, our model decreased 83% of NPT and 50% of 3D-CoreNet on the human dataset. Even though the evaluation results of our model and the baseline models on animal shapes are in an order of magnitude, the performances of our model are still better than those of other models. The PMD and

Table 1 PMD and CD evaluation of different models on human and animal shapes

Model		NPT	3D-CoreNet	Ours
SMPL	PMD	0.29	0.074	**0.042**
	CD	0.6	0.2	**0.1**
SMAL	PMD	2.4	2.3	**1.5**
	CD	4.4	4.4	**2.7**

CD values of the NPT and 3D-CoreNet models are close to each other in terms of animal shape, whereas our model outperformed them by approximately 34% (PMD) and 38% (CD), respectively.

The excellent performance of our model relies on the self-attention mechanism of the transformer across the autoencoder. This helps our model derive geometry-aware features of identity and pose information and enables it to learn and transfer deformation efficiently. Therefore, our model obtained high-quality deformed shapes with the best reconstruction losses. Moreover, our model is lighter than NPT and 3D-CoreNet. Table 2 lists the parameter numbers for the different models. As shown, our model has the minimum number of parameters. In particular, compared with 3D-CoreNet, the size of our model is one-sixth that of the 3D-CoreNet. Nonetheless, our method extends beyond baseline models.

Table 3 lists the testing time costs of the different models for calculating the pose transfer for a pair of human or animal shapes. As can be seen, our model requires less time than the baseline models, especially the baseline model, 3D-CoreNet. The main reason for the considerable time cost of 3D-CoreNet is that its encoder and decoder have complex network structures for extracting and modulating features. Another reason is that calculating the correspondence information between pose and identity shapes is an iterative and time-consuming process. Our transformer-based model has a more straightforward network structure and directly deforms the target shape; the process does not require corresponding information or multiple modulation steps. Although the testing time costs of NPT are close to ours, it performs poorly in the task and still requires multiple modulation steps, which leads to this model consuming much more time than our model, whereas our model outperforms NPT and 3D-CoreNet significantly.

Table 2 Parameter values of different models

Model	NPT	3D-CoreNet	Ours
Parameter number (M)	6.05	24.46	**4.06**

Table 3 Testing time cost of different models for a pair of target and source shapes

Model	NPT	3D-CoreNet	Ours
Human time cost (s)	0.027	0.036	**0.024**
Animal time cost (s)	0.031	0.053	**0.025**

4.3 Qualitative comparison

As shown in Fig. 5 and Fig. 6, we transferred different poses to different identity shapes using our models and baseline models, where poses and identity shapes were not used during training. To demonstrate the advancement of our method, we show the deformed

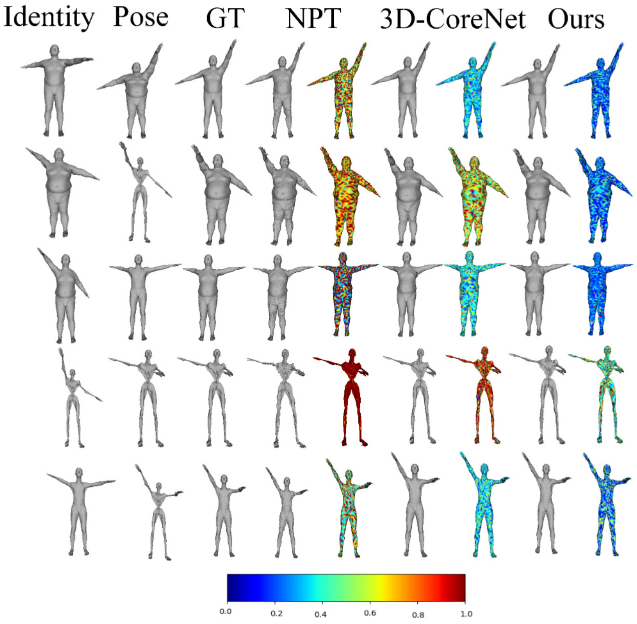


Fig. 5 Pose transfer results of different models on human shapes.

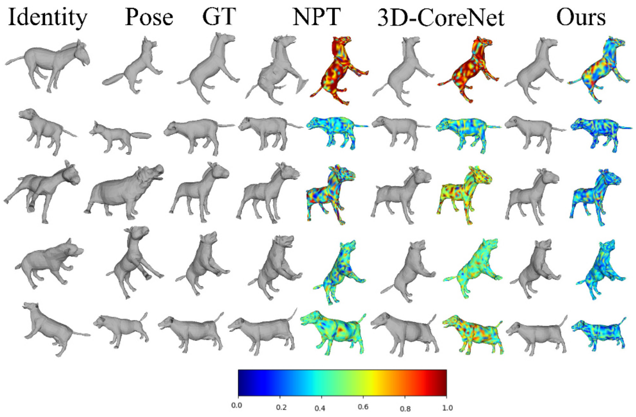


Fig. 6 Pose transfer results of different models on animal shapes.

shapes (the fourth, sixth, and eighth columns of Fig. 5 and Fig. 6) and hot maps (the fifth, seventh, and ninth columns of Fig. 5 and Fig. 6). Hot maps were drawn on ground truth meshes based on the normalized reconstruction errors between the generated and ground truth shapes. In the figures, the colors from left to right of the color bar indicate that the error increases. Blue indicates a small error and red indicates a large error. The deformed shapes and hot maps indicate that our model obtains results of the highest quality in the pose-transfer task. Owing to independent point-wise operations and no correspondence, which lead to the loss of geometric information on the identity and pose shapes, the NPT gets bad pose transfer results. For example, the surfaces of the generated human shapes (the fourth column in Fig. 5) were uneven and distorted. The generated animal shapes (the fourth column in Fig. 6) had spikes and artifacts on their bodies, heads, legs, and feet. The hot maps of NPT also show that this method generates deformed shapes with large reconstruction errors. With the assistance of the corresponding information obtained optimally and iteratively, the model 3D-CoreNet achieves better deformation than NPT. However, the operations of 3D-CoreNet are still point-wise and independent, which causes the identity and pose features to miss local and global geometric information. From the hot maps of 3D-CoreNet, we can see that this method also obtained results with significant gaps from the ground-truth shapes. Because the transformer modules in our network are geometry aware and interactive between points, our models can efficiently obtain the deformed shapes closest to the ground truth shapes. As seen in our models' deformed shapes and hot maps, the proposed method achieved results with more accurate geometric information and fewer errors than 3D-CoreNet.

To further prove the superiority of our method, we compared our method with the skeleton-free approach, which can transfer poses between shapes with different body proportions and topology structures. We used the skeleton-free pretrained model trained on the AMASS dataset [46], which is a large human motion dataset that fits SMPL to real-world human motion data. For a fair comparison, we selected shapes with identities and poses that did not exist in our training dataset and the AMASS dataset. As shown in Fig. 7, some poses are transferred to an identity shape with a rest pose using our model and are skeleton-free. We can see that our results are close to those of the GT and much better than those of the skeleton-free model. Although the skeleton-free model can obtain the correct poses, it produces distortions in the body areas enclosed by black boxes, such as the arm, knee, and belly regions. This is because the skeleton-free model focuses on part-wise latent features and transformations instead of considering geometric details. Of course, skeleton-free methods can achieve pose transfer between shapes with different categories, which our methods and the baseline methods NPT and 3D-CoreNet cannot achieve.

4.4 Generalization capability

The above human and animal shapes were generated from parametric models, which indicate that they cannot replace realistic shapes with true identities and wide poses. To demonstrate the generalization capability of our method, we transferred the shape poses from the realistic datasets FAUST and MG to

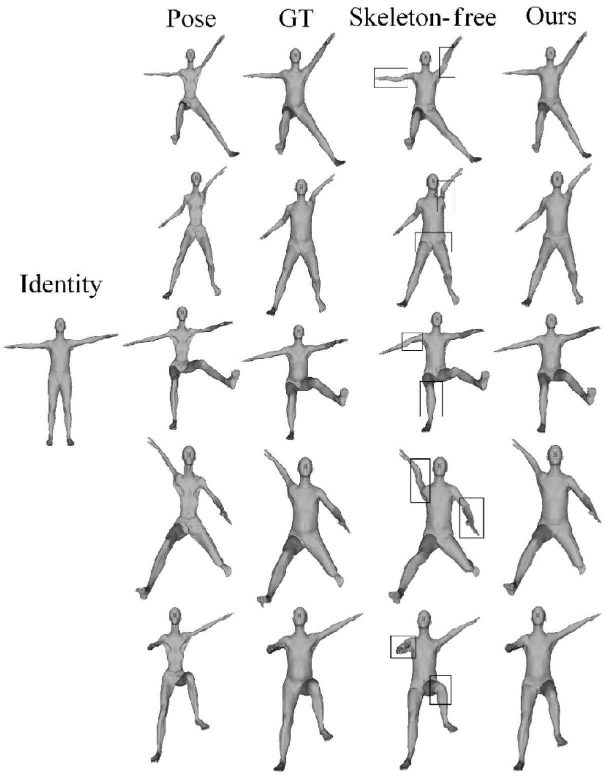


Fig. 7 Pose transfer comparison between the skeleton-free methods and our method.

the identity shapes in SMPL. As shown in Fig. 8 and Fig. 9, the wide poses of shapes in FAUST and the poses of clothed shapes in MG were transferred to the identity shapes in SMPL using different models. The shape in the MG dataset contained 27,554 vertices, which differed from the SMPL shape's vertex number. To transfer poses in the MG to identities in SMPL, we downsampled the vertices in the shapes of the MG and made the number of vertices equal to 6890. The model NPT cannot extract local and global geometric information regarding identities and poses, which results in uneven surfaces and inaccurate poses of deformed shapes. The model 3D-CoreNet also cannot obtain geometric information. This produces distorted shapes, where the arms, bodies, hands, and legs are unnatural because of incorrect correspondence information between the pose and identity shapes. Being far superior to baseline models, our model achieved the best results, where each deformed shape had the correct identity attributes and exact poses. Because of the powerful geometry perception ability of transformers with XCA in our entire network, our model has a strong generalization capability.

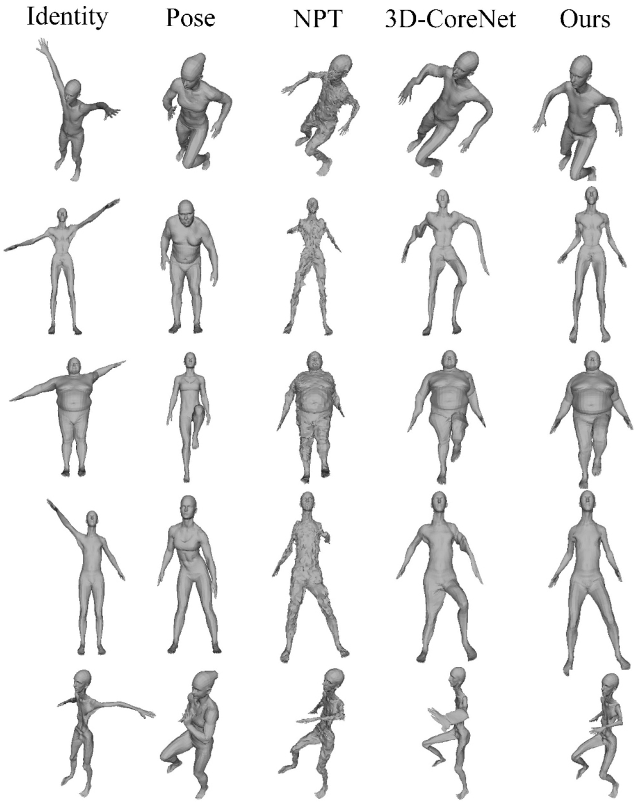


Fig. 8 Transfer wide poses of shapes in FAUST to target shapes in SMPL.

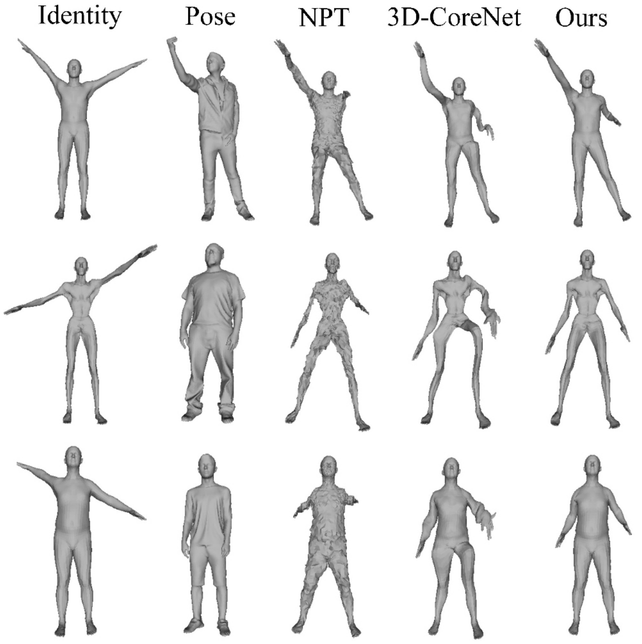


Fig. 9 Transfer poses of clothed shapes in MG to target shapes in SMPL.

We also transformed the SMPL poses into the identity shapes of DFAUST and MG. As shown in Fig. 10, we selected some unseen poses from the NPT dataset and transformed them into several identity shapes in DFAUST and MG using different models. To transfer poses in SMPL to identities in the MG, we upsampled the vertices of the SMPL shape repetitively to ensure that the shape has the same number of vertices as the identity shapes in the MG. From this figure, we can see that the proposed method obtains the best generative results. Our model considerably outperforms the NPT model. Our model also achieved more accurate shape details than the 3D-CoreNet model using its strong geometry-perceived ability, such as the regions of hands and legs in the body shapes. When processing shapes with stick parts, such as the identity shape in the last row of the figure, our method and the baseline methods failed to address this situation.

4.5 Ablation studies

In this section, we study the relations and effects of the different submodules proposed in this paper, such as ITE, PTE, and transformer decoder (TD), by replacing or using them in several networks for the 3D pose transfer task. We also studied the impact of various XCA blocks on the decoder. Subsequently, different combinations of loss terms were evaluated.

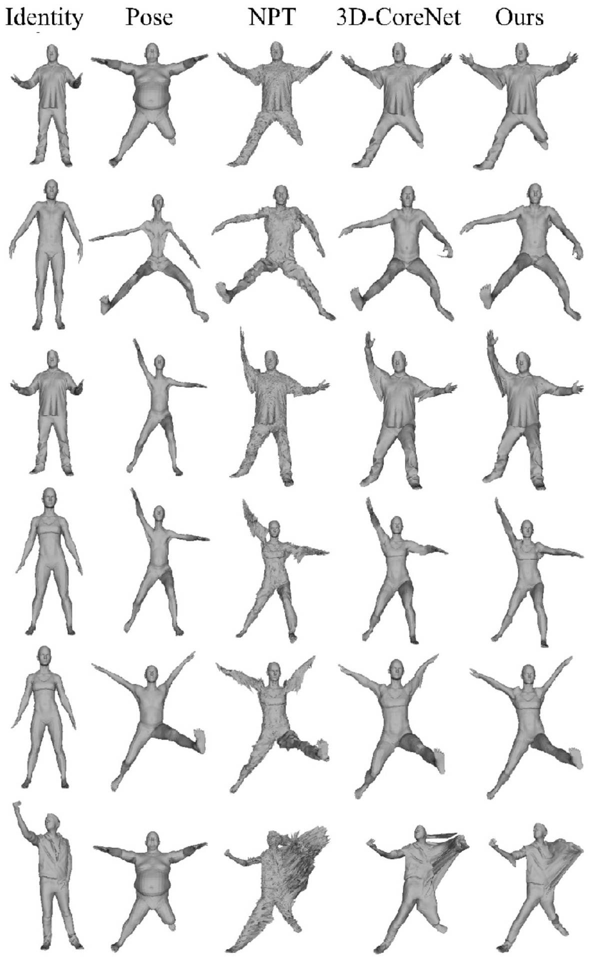


Fig. 10 Transfer SMPL poses into identity shapes in DFAUST and MG.

Finally, we analyzed the robustness and limitations of the proposed method.

4.5.1 Submodule analysis

Owing to the embeddable character of submodules, we replaced or used them to obtain many combined networks and studied them in a 3D pose transfer task. In the network proposed in this study, the ITE can be replaced with the identity feature extractor (IFE) of 3D-CoreNet. Our PTE can be replaced by the pose feature extractor (PFE) of NPT. We can remove PFE from NPT and use PTE for the NPT network. The PMD and CD evaluation results for the human shapes of different combined networks in the 3D pose transfer task are listed in Table 4. The PMD and CD had units of 10^{-3}. We also list the relative results of NPT, 3D-CoreNet, and our full network in this table for convenience of comparison.

As shown in Table 4, regardless of the replacement of PTE and ITE or both, the networks (from the third to fifth rows of the table) still performed better than NPT and 3D-CoreNet. The PMD values of the three networks were significantly lower than those of NPT and 3D-CoreNet. This implies that our transformer decoder based on XCA blocks plays a vital role in 3D pose-transfer tasks. Compared with our complete network (the last row of the table), the combined networks achieved deficient results, such as higher PMD and CD values. The PTE and ITE in our network also made contributions that worked together with our transformer decoder, leading to the best task performance. The combined network (PTE+NPT) that uses our PTE to replace PFE in NPT produces worse results than the original NPT results. This is because there is no process for decoding the relations between points in the decoder of NPT. Therefore, the PTE and ITE should be used with our transformer decoder to enhance their abilities.

4.5.2 Different numbers of XCA blocks

The impact of different numbers of XCA blocks in submodules on the 3D pose-transfer tasks must be studied. We set our transformer decoder with one, two, three, and four XCA blocks in two groups of XCA blocks. Considering the situation with zero XCA blocks in a decoder, we adopted the decoder structure of NPT, which uses point-wise convolutions to decode the features. Under the different settings, the PMD and CD evaluation results of the 3D pose transfer task on human shapes are shown in Table 5. The PMD and CD are in units of 10^{-3}. As can be seen from the table, there is a considerable difference between the zero XCA blocks and the other situations, proving that our

Table 4 Evaluation results of different combined networks trained on human shapes

Model	PMD	CD
NPT	0.29	0.6
3D-CoreNet	0.074	0.2
PFE+ITE+TD	0.050	0.2
PTE+IFE+TD	0.064	0.2
PFE+IFE+TD	0.061	0.2
PTE+NPT	0.30	0.6
Our complete network	0.042	0.1

Table 5 Evaluation results under decoder with different numbers of XCA blocks on human shapes

XCA block number	0	1	2	3	4
PMD	0.21	0.051	0.042	0.047	0.043
CD	0.5	0.2	0.1	0.1	0.1

decoder with XCA blocks has enormous advantages over the decoder with point-wise convolutions. The table also shows that one XCA block in each XCA group of the decoder is insufficient for the 3D pose transfer task. When the number of XCA blocks was two, our model achieved the best performance, with a PMD of 0.042 and a CD of 0.1. Other numbers of XCA blocks in the decoder have little impact on the performance of 3D pose transfer.

4.5.3 Loss analysis

To evaluate the influence of the different loss terms, different combinations of these loss terms were used to train the network. The corresponding PMD and CD results are listed in Table 6. Merely combining the reconstruction loss term with L2 regularization loss or edge loss can have negative consequences. In particular, combining the reconstruction loss term with the edge loss resulted in the worst PMD results. When using all loss terms, we can obtain the best performance, which indicates that the L2 loss term improves the combination of the reconstruction loss term with the edge loss.

To further study the L2 loss, we also show the reconstruction loss values on the training and test datasets of animal shapes during the training processes of our models with or without the L2 regularization term in the loss function. The results of the comparison are shown in Fig. 11. The PMD value of our model without the loss term of L2 regularization on the training dataset decreased in the later stages. In contrast, the PMD value of the model for the test dataset increased significantly. This indicates that a model without L2 regularization can easily overfit during the training phase. After we added the L2 regularization term to the loss function, the PMD values of our model on the training and test datasets maintained a consistent trend in the later training stages. Therefore, the L2 regularization term in our loss function effectively relieves the network overfitting.

Table 6 Evaluation results of models trained by different combined losses on human shapes

Model under different loss	PMD	CD
$L_{\rm rec}$	0.044	0.1
$L_{\rm rec} + L_{\rm L2}$	0.046	0.1
$L_{\rm rec} + L_{\rm edg}$	0.047	0.1
$L_{\rm rec} + L_{\rm L2} + L_{\rm edg}$	0.042	0.1

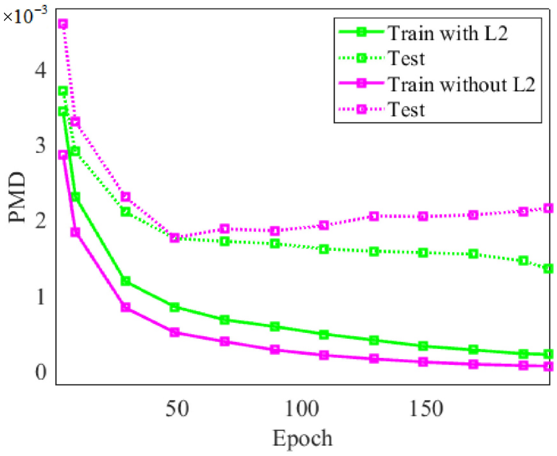


Fig. 11 Training and evaluation of animal shapes.

4.5.4 Robustness and limitations

We also analyze the robustness and limitations of the proposed method. As shown in Fig. 12, we evaluated our approach using shapes with more adhesive regions and different visual angles. In the first and second rows, our method failed to transfer poses to identity shapes with many sticky areas. The results for the moving postures were also poor when the identity and pose shapes had different visual angles, as shown in the third and fourth rows of Fig. 12. Although the generated bodies of our model were slightly

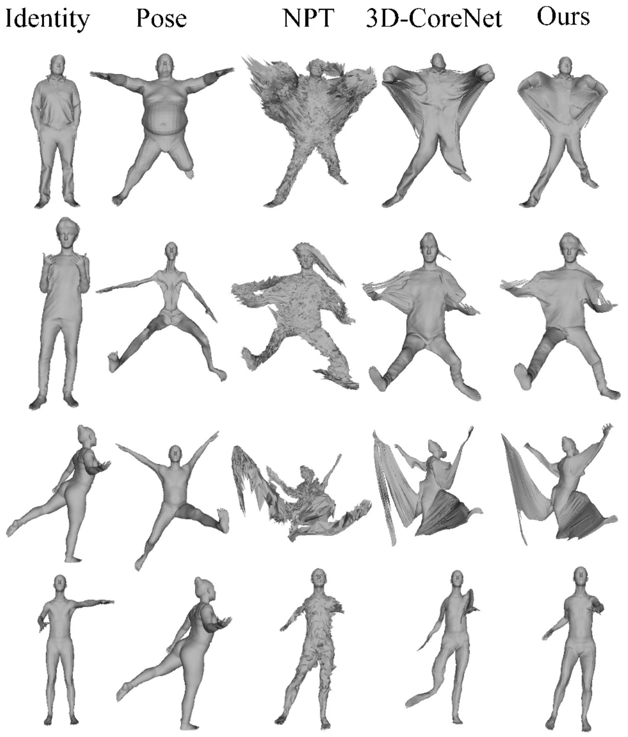


Fig. 12 Failed generation shapes.

better than those of the previous methods in the two situations, the results were much worse than the expected shapes. Therefore, our approach is not robust in these two situations.

5 Conclusions

This study proposes a novel transformer autoencoder for achieving geometry-aware 3D pose transfer. A self-attention mechanism, cross-covariance attention, is embedded in the entire network to perceive the local geometry details of the identity information and global properties of the pose structure. These geometric features are extracted and fused to learn the deformation of the target shape dynamically while maintaining its identity attributes. Our method does not require correspondence information, and is conducted in a simple and data-driven manner without multiple modulation steps based on the guidance of the target shape. Compared with other models, our model with fewer parameters efficiently executes 3D pose transfer and achieves state-of-the-art results in terms of generation and generalization capabilities. In future work, we will attempt to address limitations, such as the same vertex number, supervised training, and pose transfer between shapes with adhesive regions and different visual angles.

Acknowledgements

This work was supported by the Special Project on Basic Research of Frontier Leading Technology of Jiangsu Province, China (Grant No. BK20192004C).

Declaration of competing interest

The authors have no competing interests to declare that are relevant to the content of this article.

References

[1] Ye, Y. P.; Song, Z.; Zhao, J. High-fidelity 3D real-time facial animation using infrared structured light sensing system. *Computers & Graphics* Vol. 104, 46–58, 2022.

[2] Roberts, R. A.; dos Anjos, R. K.; Maejima, A.; Anjyo, K. Deformation transfer survey. *Computers & Graphics* Vol. 94, 52–61, 2021.

[3] Ben-Chen, M.; Weber, O.; Gotsman, C. Spatial deformation transfer. In: Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation, 67–74, 2009.

[4] Chu, H. K.; Lin, C. H. Example-based deformation transfer for 3D polygon models. *Journal of Information Science and Engineering* Vol. 26, No. 2, 379–391, 2010.

[5] Zhang, Y. Z.; Zheng, J. M.; Cai, Y. Y. Proxy-driven free-form deformation by topology-adjustable control lattice. *Computers & Graphics* Vol. 89, 167–177, 2020.

[6] Liao, Z.; Yang, J. M.; Saito, J.; Pons-Moll, G.; Zhou, Y. Skeleton-free pose transfer for stylized 3D characters. In: *Computer Vision – ECCV 2022. Lecture Notes in Computer Science, Vol. 13662.* Avidan, S.; Brostow, G.; Cissé, M.; Farinella, G. M.; Hassner, T. Eds. Springer Cham, 640–656, 2022.

[7] Zhou, K. Y.; Bhatnagar, B. L.; Pons-Moll, G. Unsupervised shape and pose disentanglement for 3D meshes. In: *Computer Vision – ECCV 2020. Lecture Notes in Computer Science, Vol. 12367.* Vedaldi, A.; Bischof, H.; Brox, T.; Frahm, J. M. Eds. Springer Cham, 341–357, 2020.

[8] Cosmo, L.; Norelli, A.; Halimi, O.; Kimmel, R.; Rodolà, E. LIMP: Learning latent shape representations with metric preservation priors. In: *Computer Vision – ECCV 2020. Lecture Notes in Computer Science, Vol. 12348.* Vedaldi, A.; Bischof, H.; Brox, T.; Frahm, J. M. Eds. Springer Cham, 19–35, 2020.

[9] Huang, X.; Belongie, S. Arbitrary style transfer in real-time with adaptive instance normalization. In: Proceedings of the IEEE International Conference on Computer Vision, 1510–1519, 2017.

[10] Park, T.; Liu, M. Y.; Wang, T. C.; Zhu, J. Y. Semantic image synthesis with spatially-adaptive normalization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2332–2341, 2019.

[11] Ulyanov, D.; Vedaldi, A.; Lempitsky, V. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint* arXiv:1607.08022, 2016.

[12] Chen, Y. G.; Chen, M. C.; Song, C. Y.; Ni, B. B. CartoonRenderer: An instance-based multi-style cartoon image translator. In: *MultiMedia Modeling. Lecture Notes in Computer Science, Vol. 11961.* Ro, Y., et al. Eds. Springer Cham, 176–187, 2020.

[13] Wang, J. S.; Wen, C.; Fu, Y. W.; Lin, H. T.; Zou, T. Y.; Xue, X. Y.; Zhang, Y. D. Neural pose transfer by spatially adaptive instance normalization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 5830–5838, 2020.

[14] Chen, H. Y.; Tang, H.; Yu, Z. T.; Sebe, N.; Zhao, G. Y. Geometry-contrastive transformer for generalized 3D pose transfer. *Proceedings of the AAAI Conference on Artificial Intelligence* Vol. 36, No. 1, 258–266, 2022.

[15] Song, C.; Wei, J.; Li, R.; Liu, F.; Lin, G. 3D pose transfer with correspondence learning and mesh

refinement. In: Proceedings of the Advances in Neural Information Processing Systems, Vol. 34, 2021.

[16] Song, C. Y.; Wei, J. C.; Li, R. B.; Liu, F. Y.; Lin, G. S. Unsupervised 3D pose transfer with cross consistency and dual reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 45, No. 8, 10488–10499, 2023.

[17] Guo, M. H.; Xu, T. X.; Liu, J. J.; Liu, Z. N.; Jiang, P. T.; Mu, T. J.; Zhang, S. H.; Martin, R. R.; Cheng, M. M.; Hu, S. M. Attention mechanisms in computer vision: A survey. *Computational Visual Media* Vol. 8, No. 3, 331–368, 2022.

[18] Xu, Y. F.; Wei, H. P.; Lin, M. X.; Deng, Y. Y.; Sheng, K. K.; Zhang, M. D.; Tang, F.; Dong, W. M.; Huang, F. Y.; Xu, C. S. Transformers in computational visual media: A survey. *Computational Visual Media* Vol. 8, No. 1, 33–62, 2022.

[19] Sumner, R. W.; Popović J. Deformation transfer for triangle meshes. In: Proceedings of the ACM SIGGRAPH Papers, 399–405, 2004.

[20] Xu, W. W.; Zhou, K.; Yu, Y. Z.; Tan, Q. F.; Peng, Q. S.; Guo, B. N. Gradient domain editing of deforming mesh sequences. *ACM Transactions on Graphics* Vol. 26, No. 3, 84–es, 2007.

[21] Domadiya, P. M.; Shah, D. P.; Mitra, S. Guided deformation transfer. In: Proceedings of the 16th ACM SIGGRAPH European Conference on Visual Media Production, Article No. 7, 2019.

[22] Basset, J.; Wuhrer, S.; Boyer, E.; Multon, F. Contact preserving shape transfer: Retargeting motion from one shape to another. *Computers & Graphics* Vol. 89, 11–23, 2020.

[23] Yang, J.; Gao, L.; Lai, Y. K.; Rosin, P. L.; Xia, S. H. Biharmonic deformation transfer with automatic key point selection. *Graphical Models* Vol. 98, 1–13, 2018.

[24] Ben-Chen, M.; Weber, O.; Gotsman, C. Variational harmonic maps for space deformation. *ACM Transactions on Graphics* Vol. 28, No. 3, Article No. 34, 2009.

[25] Jacobson, A.; Baran, I.; Popović J.; Sorkine, O. Bounded biharmonic weights for real-time deformation. *ACM Transactions on Graphics* Vol. 30, No. 4, Article No. 78, 2011.

[26] Baran, I.; Vlasic, D.; Grinspun, E.; Popović J. Semantic deformation transfer. *ACM Transactions on Graphics* Vol. 28, No. 3, Article No. 36, 2009.

[27] Chen, H.; Tang, H.; Sebe, N.; Zhao, G. AniFormer: Datadriven 3D animation with transformer. In: Proceedings of the British Machine Vision Conference, 2021.

[28] Gao, L.; Yang, J.; Qiao, Y. L.; Lai, Y. K.; Rosin, P. L.; Xu, W. W.; Xia, S. H. Automatic unpaired shape deformation transfer. *ACM Transactions on Graphics* Vol. 37, No. 6, Article No. 237, 2018.

[29] Chen, H. Y.; Tang, H.; Shi, H. L.; Peng, W.; Sebe, N.; Zhao, G. Y. Intrinsic-extrinsic preserved GANs for unsupervised 3D pose transfer. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 8610–8619, 2021.

[30] Wang, Y. F.; Aigerman, N.; Kim, V. G.; Chaudhuri, S.; Sorkine-Hornung, O. Neural cages for detail-preserving 3D deformations. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 72–80, 2020.

[31] Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In: Proceedings of the 31st International Conference on Neural Information Processing Systems, 6000–6010, 2017.

[32] Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X. H.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint* arXiv:2010.11929, 2020.

[33] Lin, K.; Wang, L. J.; Liu, Z. C. End-to-end human pose and mesh reconstruction with transformers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 1954–1963, 2021.

[34] Lin, K.; Wang, L. J.; Liu, Z. C. Mesh graphormer. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 12919–12928, 2021.

[35] Misra, I.; Girdhar, R.; Joulin, A. An end-to-end transformer model for 3D object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2886–2897, 2021.

[36] Mao, J. G.; Xue, Y. J.; Niu, M. Z.; Bai, H. Y.; Feng, J. S.; Liang, X. D.; Xu, H.; Xu, C. J. Voxel transformer for 3D object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 3144–3153, 2021.

[37] Ali, A.; Touvron, H.; Caron, M.; Bojanowski, P.; Douze, M.; Joulin, A.; Laptev, I.; Neverova, N.; Synnaeve, G.; Verbeek, J.; et al. Xcit: Cross-covariance image transformers. In: Proceedings of the Advances in Neural Information Processing Systems, Vol. 34, 20014–20027, 2021.

[38] Chandran, P.; Zoss, G.; Gross, M.; Gotardo, P.; Bradley, D. Shape transformers: Topology-independent 3D shape models using transformers. *Computer Graphics Forum* Vol. 41, No. 2, 195–207, 2022.

[39] Loper, M.; Mahmood, N.; Romero, J.; Pons-Moll, G.; Black, M. J. SMPL: A skinned multi-person linear model. *Seminal Graphics Papers: Pushing the Boundaries* Vol. 2, Article No. 88, 851–866, 2023.

[40] Bogo, F.; Romero, J.; Loper, M.; Black, M. J. FAUST: Dataset and evaluation for 3D mesh registration.

In:Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 3794–3801, 2014.

[41] Bhatnagar, B.; Tiwari, G.; Theobalt, C.; Pons-Moll, G. Multi-garment net: Learning to dress 3D people from images. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 5419–5429, 2019.

[42] Zuffi, S.; Kanazawa, A.; Jacobs, D. W.; Black, M. J. 3D menagerie: Modeling the 3D shape and pose of animals. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 5524–5532, 2017.

[43] Loshchilov, I.; Hutter, F. Decoupled weight decay regularization. *arXiv preprint* arXiv:1711.05101, 2017.

[44] Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. PyTorch: An imperative style, high-performance deep learning library. In: Proceedings of the 33rd International Conference on Neural Information Processing Systems, Article No. 721, 8026–8037, 2019.

[45] Fan, H. Q.; Su, H.; Guibas, L. A point set generation network for 3D object reconstruction from a single image. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2463–2471, 2017.

[46] Mahmood, N.; Ghorbani, N.; Troje, N. F.; Pons-Moll, G.; Black, M. AMASS: Archive of motion capture as surface shapes. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 5441–5450, 2019.

Shanghuan Liu received his B.E. and M.E. degrees from the College of Internet of Things Engineering, Hohai University, Nanjing, China, in 2015 and 2018, respectively. He is currently pursuing the Ph.D. degree in Southeast University, Nanjing, China, with a focus on 3D sensing and deep learning in computer vision.

Shaoyan Gai received his Ph.D. degree from Southeast University in 2008. He is currently an associate professor and a Ph.D. advisor at Southeast University. His main research interests include 3D measurement and 3D face recognition.

Feipeng Da received his Ph.D. degree from the School of Automation, Southeast University, in 1998. He is currently a professor with the School of Automation, Southeast University. He has published an academic monograph and authored or coauthored over 150 high quality articles, of which are retrieved by SCI, EI, and ISTP more than 100 times. He has 40 authorized invention patents, one authorized patent for utility models, four software copyrights, and three international invention patents (PCT applied). He also serves as a reviewer for the journals from different areas, such as *Optics Express*, *Optics Letters*, *Optical and Lasers in Engineering*, *IEEE TRANSACTIONS ON NEURAL NETWORKS*, *IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS-I: REGULAR PAPERS*, *IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS-II: EXPRESS BRIEFS*, *PHYSICS LETTER A*, *Neural Networks*, and *Pattern Recognition*.

Fazal Waris received his B.Sc. degree from NWFP University of Engineering and Technology, Peshawar and M.S. degree from University of Lahore, Pakistan. He is currently a Ph.D. candidate at Southeast University, Nanjing, China. His research interests include machine learning, computer vision, and deep learning.