Skinned Motion Retargeting with Preservation of Body Part Relationships

Jia-Qi Zhang, Miao Wang*, Fu-Cheng Zhang, Fang-Lue Zhang

Abstract—Motion retargeting is an active research area in computer graphics and animation, allowing for the transfer of motion from one character to another, thereby creating diverse animated character data. While this technology has numerous applications in animation, games, and movies, current methods often produce unnatural or semantically inconsistent motion when applied to characters with different shapes or joint counts. This is primarily due to a lack of consideration for the geometric and spatial relationships between the body parts of the source and target characters. To tackle this challenge, we introduce a novel spatially-preserving Skinned Motion Retargeting Network (SMRNet) capable of handling motion retargeting for characters with varying shapes and skeletal structures while maintaining semantic consistency. By learning a hybrid representation of the character's skeleton and shape in a rest pose, SMRNet transfers the rotation and root joint position of the source character's motion to the target character through embedded rest pose feature alignment. Additionally, it incorporates a differentiable loss function to further preserve the spatial consistency of body parts between the source and target. Comprehensive quantitative and qualitative evaluations demonstrate the superiority of our approach over existing alternatives, particularly in preserving spatial relationships more effectively.

Index Terms—Motion Retargeting, Spatial Relationship, Different Structure

1 Introduction

As animation technology has advanced, the integration of motion capture has become commonplace, enabling animators to capture realistic human or creature movements. However, the diversity in character designs, sizes, and skeletal structures often presents a significant challenge. Motion retargeting emerges as a crucial solution, allowing animators to seamlessly transfer motion capture data onto characters with different body proportions or skeletal structures [1], [2], [3], [4]. Additionally, motion retargeting enables the reuse of existing animations on characters with different builds without the need to recreate the entire animation from scratch, thereby reducing the production time and cost of character animation. Some other ongoing research explores fine-grained motion retargeting, such as human hand and facial motion [5], [6], expanding its applications in game development, visual effects, and the metaverse.

This paper addresses the challenge of motion retargeting for humanoid characters, which involves converting motion information from one skeletal structure to another while maintaining consistent spatial relationships between body parts. To effectively learn the disentanglement between motion and appearance of the characters, Aberman et al. [7] proposed training skeleton-aware networks (SAN) for processing skeletal motions of different character structures. Villegas et al. [8] additionally emphasized contact consistency between the character's body parts prior to and following the process of motion retargeting. However, these methods overlook the influence of the spatial relationship between the character's body part shape on the motion semantics. For example, the motion of a thin character holding a basketball with both hands might look like a defensive posture if directly transferred to a bigger character.

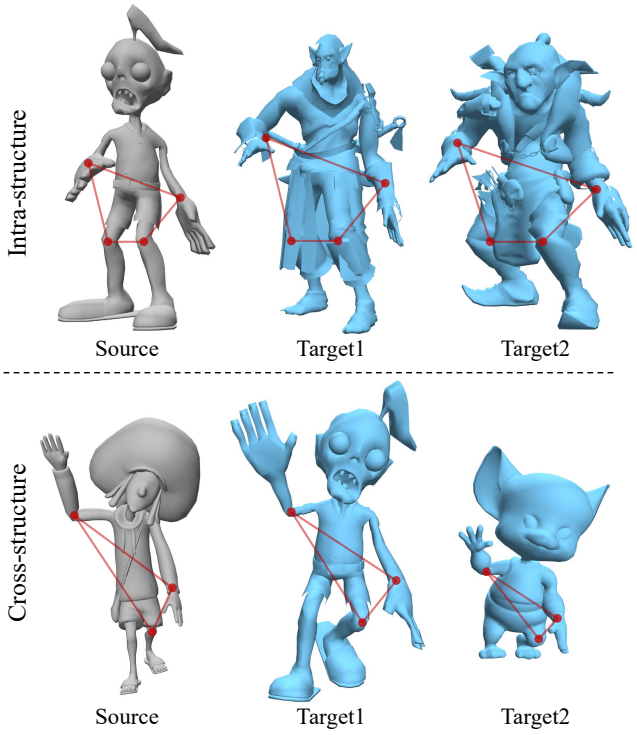


Fig. 1: Examples of our motion retargeting results between characters with identical (intra-) and different (cross-) structures. Spatial distances of typical body parts are visualized.

* *Corresponding Author*

- *M. Wang is with the State Key Laboratory of Virtual Reality Technology and Systems, School of Computer Science and Engineering, Beihang University, Beijing 100191, China, and also with Zhongguancun Laboratory, Beijing 100094, China. He is the corresponding author. E-mail: miaow@buaa.edu.cn.*
- *J.-Q. Zhang and F.-C. Zhang are with the State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, Beijing 100191, China. E-mail:{zhangjiaqi79, stringzfc}@buaa.edu.cn.*
- *F.-L. Zhang is with the School of Engineering and Computer Science, Victoria University of Wellington, New Zealand. E-mail: fanglue.zhang@ecs.vuw.ac.nz.*

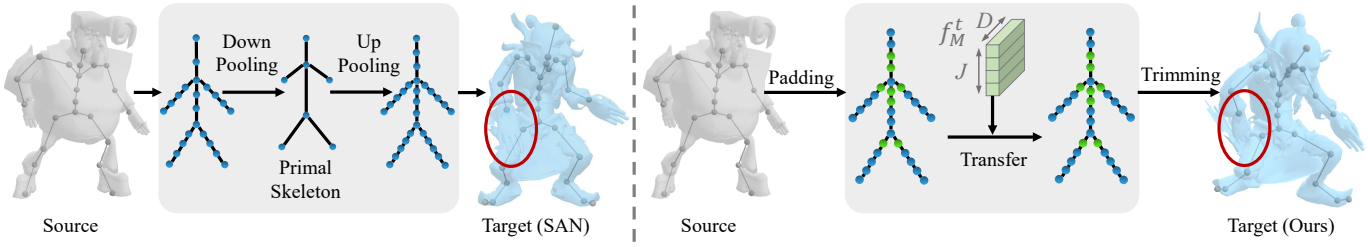


Fig. 2: Illustration of the motion retargeting process. Left: the motion retargeting process of the SAN [7]; Right: our motion retargeting process. The green points indicate virtual joints. f_M^t denotes the learned hybrid skeleton and mesh representation, J denotes the number of joints, D denotes the feature dimension associated with each joint.

We propose a solution to the challenge of preserving the spatial consistency of a character's surface mesh model while transferring skeletal motion between different structures. Our approach involves constructing a unified learning framework capable of accommodating motion retargeting for diverse humanoid character structures, eliminating the need to train separate networks for different characters [7]. According to SAN [7], motion retargeting is termed intra-structure for identical skeletons and cross-structure for different ones. Furthermore, our method addresses the limitations of previous approaches [9], which rely on predefined rules and struggle with the vertex-matching problem in complex-shaped characters. Additionally, our network considers both character shape and the spatial relationships among body parts, crucial factors for ensuring high-quality motion. Figure 1 showcases the motion retargeting results of our method under both identical and different skeleton structures.

Our network design is founded on two key concepts. Firstly, we employ an innovative joint padding strategy and learn a hybrid skeleton and mesh representation in rest pose to facilitate seamless motion retargeting across diverse skeletal structures and mesh envelopes. In contrast to the SAN method, which simplifies the source skeleton into a smaller primitive structure before augmenting joints to form the target character's skeleton (Figure 2, left), our approach learns a hybrid skeleton and mesh representation with a greater number of joints, preserving comprehensive information about diverse structural skeletons and shapes (Figure 2, right). Specifically, we first introduce virtual joints to the source skeleton and then transfer the motion to the target character using the hybrid skeleton and mesh representation. Subsequently, we remove unnecessary joints for precise motion retargeting across various structures. As illustrated, the target hand generated by our method is positioned behind the body, faithfully resembling the source pose. Secondly, by utilizing the distance matrix of vertices on the character's mesh rather than bone joints, the spatial relationships between different body parts are consistently maintained. It considers both shape characteristics and spatial relationships (referred to as identity information in the following sections), thereby enhancing motion semantics during retargeting.

We have developed a deep architecture that integrates the aforementioned concepts when learning to transfer motions between two characters. To ensure the successful disentanglement of identity information and the motion of both source and target characters, we have adopted a CycleGAN-based framework [10] to learn cycle-consistent bidirectional motion mappings. Our network has been trained and evaluated alongside the latest methods using the same dataset. Through quantitative and qualitative experimental comparisons, our method not only matches the performance of the latest techniques in the intra-structure motion retargeting task but also outperforms the current state-of-the-art in the cross-structure motion retargeting task. We summarize the main contributions of our work as follows:

- We propose a new motion retargeting method that harnesses the character's geometric features on both skeletons and meshes to achieve high-quality motion retargeting between diverse character structures.
- We introduce a unified deep learning framework with a novel hybrid skeleton and mesh representation learning scheme, capable of accommodating character motion retargeting across various skeletal structures. Our design includes new motion transfer and root position transfer modules to ensure effective learning.
- To preserve the semantic information of the motion, we introduce a new loss function that enforces spatial consistency among different body parts prior to and following the process of motion retargeting. This loss function simultaneously constrains spatial distances between character joints and mesh vertices.

2 Related Work

Our framework addresses the longstanding computer graphics problem of motion retargeting, a technology first proposed by Michael [11] as early as 1998, and since then, it has been the subject of extensive research. The methods of motion retargeting can be categorized into two main groups: motion retargeting between humans and non-human characters [2], [3], [12], and motion retargeting between humanoid characters [7], [8], [13], [14], [15]. Early methods relied on hand-designed kinematic constraints to solve spatiotemporal problems and map motions between different characters [16], [17]. However, these methods required significant human intervention and struggled with handling different skeleton structures and shape geometries. To overcome these limitations, data-driven methods have become more popular in recent years [13], [18]. These methods employ deep learning techniques to autonomously learn the mapping for motion retargeting from data, or use spatio-temporal graph convolutional network (STGCN) [19] to build hierarchical information for character motion, resulting in improved computational efficiency and accuracy. Based on the input and output data types, existing motion retargeting methods can be classified into two categories: skeleton-based methods and surface-based methods. The former handles low-dimensional data like skeleton pose or joint angle, while the latter handles high-dimensional data such as mesh vertices or skin weights. Additionally, we will review related works that maintain spatial consistency before and after the motion retargeting process, with a primary focus on avoiding collisions and contact failures during motion retargeting.

2.1 Skeleton-based Motion Retargeting

Early-stage methods based on kinematic constraints used different techniques to adjust motion data for different characters. Michael [11] proposed a spatiotemporal optimization model for skeletons with similar structures but different lengths. Lee et al. [16] adjusted the joint poses in each frame to satisfy the constraints and interpolated the joint motion displacements using multi-layer B-spline curves to generate smooth motion. Choi et al. [20] proposed a real-time motion retargeting method based on inverse rate control, which computed the joint angle changes according to the end effector position changes. Feng et al. [21] proposed a set of heuristic methods to map the target skeleton to the canonical skeleton by finding bones with similar names. Unlike these methods that attempted to automatically derive motion semantic structure and constraints from example data, Hecker et al. [22] proposed the skeletal controller that allowed animators to edit motion in keyframes, preserving the skeletal structural relationship and the style of the animation.

The data-driven deep network method is currently a commonly used motion retargeting technology, capable of learning the mapping between different characters end-to-end, offering high computational efficiency and superior motion quality. Delhaisse et al. [23] utilized a shared Gaussian process latent variable model to learn a mapping function from a general latent representation to a target output. Jang et al. [18] designed a deep convolutional network based on U-Net [24] to learn motion retargeting at different bone length ratios. However, these methods require paired training data, which is often difficult to obtain in practical applications. To address this challenge, there are currently two mainstream ideas: one is to use unsupervised generative adversarial networks CycleGAN [10] for motion retargeting, such as Villegas et al. [13], who proposed a real-time motion retargeting generator based on a forward kinematics layer and a recurrent neural network; Aberman et al. [7] proposed a skeleton-aware framework specifically to skeletal data. The other idea is to disentangle character pose and movement, as demonstrated by Lim et al. [14], who proposed a pose-movement network (PMnet) capable of adjusting the overall motion while maintaining pose consistency; Zhang et al. [25] proposed a residual retargeting network (R2ET), which can reduce penetration and contact failure between target characters while preserving the source character motion semantics.

2.2 Surface-based Motion Retargeting

In practical applications, motion retargeting can encounter challenges when character models have different shapes but the same skeleton structure. This can lead to issues such as body parts overlapping or mismatched motion semantics. To address this, researchers have proposed surface-based motion retargeting methods. For example, Basset et al. [26] developed an optimization-based method for SMPL [27] format data that deforms the shape and pose of the source character while preserving the contact of the original reference character and preventing self-penetration of the target character. Villegas et al. [8] designed a contact-aware motion retargeting network that combines deep learning and energy optimization to ensure that the self-contact position of the target character is consistent with the reference character. Additionally, Musoni et al. [28] applied function mapping to learn the mapping of the reference character's skeleton and skin weight to the target character. However, these methods require that the reference and target characters have the same topological structure for their skeletons or shapes.

One area of research in motion retargeting involves transferring human motion capture data to non-humanoid characters, such as animals, robots, or everyday objects [29], [30], [31]. Yamane et al. [1] were the first to propose a method for this task, which learns the correspondence between key frame poses of humans and non-humanoids. Bharaj et al. [32] introduced a new motion retargeting method by computing the rigid skin weight and joint mapping relationship between different models. However, their method requires that the two skeletons have similar poses to effectively calculate the joint mapping. Seol et al. [3] designed a puppet system that can drive non-human characters in real-time, but this system needs to pre-compute the feature mapping relationship based on the original motion. Other methods use techniques such as matching bone chains [33], learning from a given matching example [4], using the Kinect device to capture 3D data and monitor human body posture [2], or introducing the concept of body part groups [34]. These methods typically involve a complex process, and a single instance of motion retargeting can be time-consuming.

2.3 Spatial Relationship Preserving

Many current motion retargeting methods concentrate on the spatial relationship between character joints and apply constraints on the distance matrix between joints before and after retargeting [17], [25]. Some methods also use markers on the character surface to preserve self-contact and interaction motions [35]. In addition, to preserve the spatial relationship between characters in two-person motion retargeting, Ho et al. [36] use an interaction mesh that captures how body parts relate to each other. Jin et al. [37] create an aura mesh for each character that fully encloses their shape. Unlike the interaction mesh method, the aura mesh only needs to be built once and does not require a different mesh for each interaction action.

However, the methods above do not fully consider the influence of the character's shape on the consistency of the spatial motion relationship. Liu et al. [9] proposed a context-aware motion retargeting framework. This method involves placing sampling points on the character surface and creating a context graph. Unnecessary edges are then removed based on specified rules. Ultimately, it ensures spatial relationship consistency by constraining the context graph of the target character to match the reference character. However, this method relies on the non-rigid iterative closest point (ICP) algorithm [38] to obtain the correspondence between the vertices of the character mesh, which cannot handle irregular and diverse 3D characters well. Li et al. [39] introduces a novel and effective unsupervised learning method (ACE) in the field of cross-morphological motion retargeting, leveraging adversarial learning to learn motion retargeting in a compact embedding space. However, for characters lacking explicit concepts of arms and legs, such as sharks or caterpillars, the effectiveness of this framework may be limited. CMI-SR [40] achieves physical simulation of complex multi-character interactions by encoding the spatial relationships between multiple characters based on an interaction graph, guiding the reinforcement learning process to focus on inter-character interactions. However, their training strategies may necessitate additional learning and adaptation to effectively generalize when applied to characters with diverse body configurations.

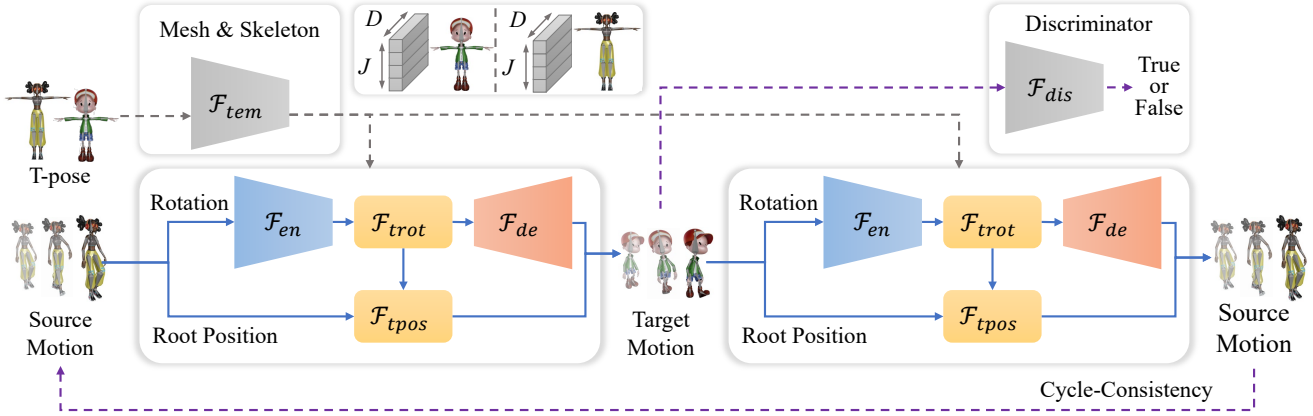


Fig. 3: Overview of our motion retargeting method, SMRNet, trained via a cycle-consistent scheme. Given the motion of the source character and the identity features of both the source and target characters, the network generates the retargeted motion.

In our work, we ensure the consistency of the spatial relationship before and after motion retargeting not only from the perspective of global features but also from the perspective of local mesh vertex motion. Furthermore, our method avoids the need to find complex correspondence between the vertices of the target and source characters.

3 Overview

We propose a unified learning framework, SMRNet, to perform skinned motion retargeting between characters with different structures (see Fig 3). Unlike SAN [7], which needs to train separate networks for each character, our framework can handle both intra-structure and cross-structure motion retargeting in the same framework. We employ a consistent character information representation and a motion transfer module. Initially, we automatically complete the joints for both characters, ensuring a consistent number of joints across the limbs and torso. Additionally, we supplement the skinning weights for corresponding characters. However, this does not eliminate the necessity to learn the mapping relationship between joints for different characters. Hence, we develop a motion transfer module that learns a hybrid skeleton and mesh representation in a rest pose. This module allows for motion retargeting across diverse structural characters by establishing the mapping between the embedded features of different structural characters in their rest poses and the hybrid skeleton and mesh representation.

Furthermore, to maintain semantic consistency between the retargeted result and the original input motion, we ensure the preservation of spatial relationships among the character's body parts prior to and following the process of motion retargeting. This preservation occurs on two levels: firstly, by limiting the spatial separation between the joints, and secondly, by maintaining the spatial proximity between the surface vertices of the character. These constraints play a crucial role in retaining motion semantic information throughout the motion retargeting process.

4 Motion Retargeting Framwork

A motion sequence of a character is primarily composed of the following information: joint rotations $Q \in \mathbb{R}^{T\times J\times 4}$, root joint positions $P \in \mathbb{R}^{T\times 3}$, character skeleton offsets $S \in \mathbb{R}^{J\times 3}$, mesh vertices $V \in \mathbb{R}^{N\times 3}$, and skinning weights $W \in \mathbb{R}^{N\times J}$. Here, N represents the number of vertices, J denotes the number of joints, and T is the length of the sequence. The quaternion representation is used for the rotations in Q. Moreover, the skeleton offsets, mesh vertices, and skinning weights collectively constitute the character's identity information, denoted as $(S, V, W) \in \mathcal{C}$. Let $(Q, P, \mathcal{C}) \in \mathcal{M}$ represent a complete motion sequence. Our goal is to transfer the motion $\mathcal{M}_A$ of the source character A to the target character B, resulting in the motion $\mathcal{M}_B$. Notably, the source and the target characters may have distinct skeletal structures, bone lengths and proportions, as well as body meshes. Therefore, we define the motion retargeting as a mapping $F^{A\to B}$:

$$\mathcal{F}^{A\to B}((Q_A, P_A, \mathcal{C}_A) \in \mathcal{M}_A, \mathcal{C}_B) \to (\hat{Q}_B, \hat{P}_B). \quad (1)$$

Figure 3 illustrates the architecture of our motion retargeting network designed to learn the mapping. The challenge of retargeting motion between two characters with different skeletons is akin to transferring style features between unpaired images with different scene layouts. Leveraging the capabilities of CycleGAN [10] in disentangling features and transferring style between two different domains, we have developed a cycle-consistent learning framework with an adversarial setup. The generator comprises five modules: the identity feature extraction module $\mathcal{F}_{tem}$, the motion information encoding module $\mathcal{F}_{en}$, the motion information transfer module $\mathcal{F}_{trot}$, the motion information decoding module $\mathcal{F}_{de}$, and the root position transfer module $\mathcal{F}_{tpos}$ (see also in Figure 4). The identity information of characters with different structures is extracted by $\mathcal{F}_{tem}$, which is then transmitted to the remaining four modules to predict the dynamic information of the character, including the rotation of all joints and the position of the root joint. Notably, to address the semantic inconsistency in motion retargeting caused by the difference in the characters' mesh envelope, the characters' bone offset and vertex information are simultaneously processed as character-independent information.

4.1 Graph-Based Convolutional Networks

Considering the superior proficiency of the Spatio-Temporal Graph Convolutional Network (STGCN) [19] in handling complex time-series and spatial hierarchical data, we have chosen to build our network based on STGCNs to improve the comprehension and learning of character motion. Let $G = (V, E)$ denote a spatial-temporal graph. $V = \{v_{t,j} | 1 \le t \le T; 1 \le j \le J\}$ represents all joints in the motion sequence. E consists of two types of edges: $E_S = \{\langle v_{t,j}, v_{t,k}\rangle \ | 1 \le t \le T; 1 \le j, k \le J\}$ represents the connection relationship between the joints in the same frame, and

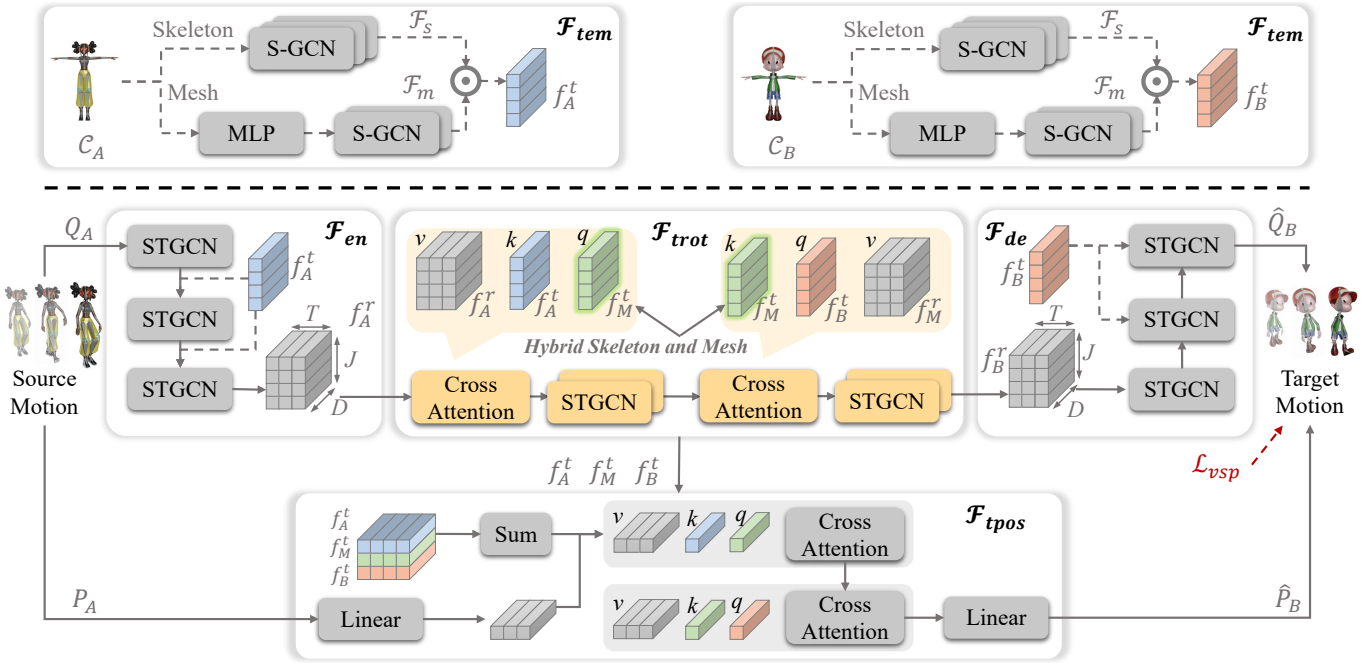


Fig. 4: Illustration of the generator of our motion retargeting framework. Given the source character's joint rotations, root joint position, identity information, and the additional identity information for the target character $((Q_A, P_A, \mathcal{C}_A), \mathcal{C}_B)$, the generator outputs joint rotations and the root joint position for the target character. $(\hat{Q}_B, \hat{P}_B)$.

$E_T = \{\langle v_{t,j}, v_{t+1,j}\rangle | 1 \le t \le T; 1 \le j \le J\}$ represents the edges of the same joint across adjacent frames. We assume that the number of joints of the skeleton remains constant throughout the network, ensuring that the adjacency relationships within any STGCN layer remain unaltered. Meanwhile, each STGCN layer has independent learnable edge importance weights. Figure 4 shows the details of the generator.

Identity feature extractor. $\mathcal{F}_{tem}$ consists of two main branches that process the character's skeleton offset and the mesh vertex information, respectively. The skeleton branch $\mathcal{F}_s$ comprises three layers of spatial graph convolutional networks (S-GCN) that take the skeleton offset S under the character's rest pose as input. The mesh branch $\mathcal{F}_m$ consists of a multilayer perceptron (MLP) and two layers of S-GCN. This approach accommodates varying vertex counts among different characters. Consequently, an MLP is employed initially to extract features by inputting V, followed by collapsing the vertex features into a set of J deep offsets using the relative skinning weight W. Given the input $(S, V, W) \in \mathcal{C}$, the identity feature extractor produces the identity feature f^t of the character following:

$$f^t = \mathcal{F}_{tem}((S, V, W) \in \mathcal{C}) = \mathcal{F}_m(V, W) \odot \mathcal{F}_s(S), \tag{2}$$

where $\odot$ denotes the feature concatenation operator.

Motion encoder and decoder. Given the local rotation Q_A of each joint of the source character's motion and the identification feature f_A^t of the source character, the motion encoder employs a three-layer STGCN to extract the motion feature f_A^r. The encoding process can be expressed as $f_A^r = \mathcal{F}_{en}(Q_A, f_A^t)$. Likewise, given the motion feature f_B^r and the identification feature f_B^t of the target character, the motion decoder utilizes a three-layer STGCN to generate its local rotation of each joint. The entire decoding process can be formulated as $\hat{Q}_B = \mathcal{F}_{de}(f_B^r, f_B^t)$. The network also outputs the global positions of the target character's joints and vertices $(\hat{P}_B^J, \hat{P}_B^V) = \textbf{LBS}(\hat{Q}_B, \hat{P}_B, \mathcal{C}_B)$.

Motion transfer module. The motion transfer module, denoted as $\mathcal{F}_{trot}$, is designed to transfer motions between two characters with different structures. The key idea of this module is to employ a network to learn the hybrid skeleton and mesh representation f_M^t of a rest pose. Subsequently, it determines the joint mapping relationship between any two characters with distinct structures by comparing the identity features of the source character, the target character, and the learned hybrid skeleton and mesh representation. The motion transfer module $\mathcal{F}_{trot}$ takes the identity features of the source and target characters, denoted as f_A^t and f_B^t, as input, along with the motion feature f_A^r of the source character. It then produces the motion feature f_B^r of the target character. The motion transfer process can be formulated as:

$$f_B^r = \mathcal{F}_c^{m\to t}(f_M^t, f_B^t, \mathcal{F}_c^{s\to m}(f_A^t, f_M^t, f_A^r)), \tag{3}$$

where $\mathcal{F}_c^{s\to m}$ represents the process of transferring the motion feature from the source character to the hybrid skeleton and mesh representation, and $\mathcal{F}_c^{m\to t}$ represents the process of transferring the motion feature from the hybrid skeleton and mesh representation to the target character.

Each part $\mathcal{F}_c$ of the motion transfer module $\mathcal{F}_{trot}$ contains a cross-attention layer and two layers of STGCN. The cross-attention layer consists of a multi-head cross-attention layer, two normalization layers and a multi-layer perceptron layer, which is a standard transformer encoder layer. The whole motion transfer process of $\mathcal{F}_c^{s\to m}$ is conducted as:

$$\begin{aligned} f_M^r &= \textbf{Norm}(f_A^r + \textbf{MultiHeadAtt}(f_A^t, f_M^t, f_A^r)), \\ f_M^r &= \textbf{Norm}(f_M^r + \textbf{MLP}(f_M^r)), \\ f_M^r &= \mathcal{F}_c^{s\to m}(f_A^t, f_M^t, f_A^r) = \textbf{STGCN}(f_M^r), \end{aligned} \tag{4}$$

where f_M^r represents the motion feature of the learned hybrid skeleton and mesh representation. Essentially, the transfer module applies two transformations to the input source motion. The first transformation produces the motion features f_M^r corresponding

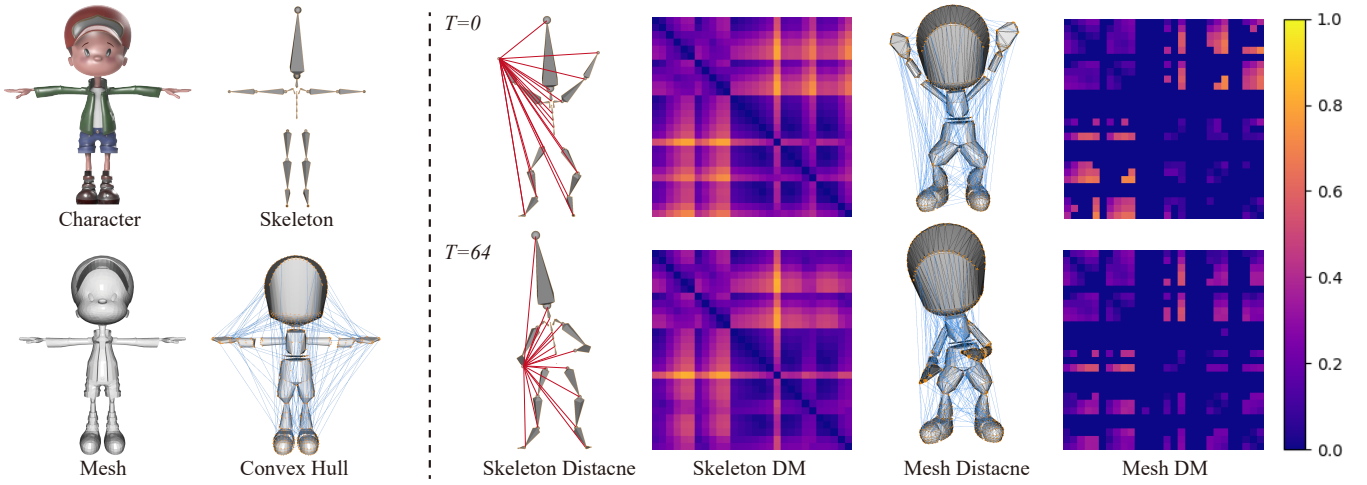


Fig. 5: Comparison of the distance matrix between joints and the distance matrix between mesh vertices. Left: the character model, skeleton, mesh, and the edges extracted after identifying the convex hull of the mesh (depicted by the blue line), all in a rest pose. Right: sampled edges (10%) from the character skeleton and convex hull at frames 0 and 64. "Skeleton DM" refers to the Euclidean distance matrix between joints, while "Mesh DM" refers to the Euclidean distance matrix between different parts of the body.

to the hybrid representation, while the second transformation generates the motion features f_B^r of the target character.

Root position transfer module. In addition to the rotation of joints, the changes in the position of the root joint are also essential for proper motion retargeting. The root joint position transfer module $\mathcal{F}_{tpos}$ adopts a structure similar to module $\mathcal{F}_{trot}$. The hybrid skeleton and mesh representation f_M^t, learned by module $\mathcal{F}_{trot}$, is used, along with the identification features of the source character f_A^t and the target character f_B^t, to achieve the position transformation of the root joint from the source character P_A to the target character $\hat{P}_B$. Specifically, the identification features (f_A^t, f_B^t, and f_M^t) are first summed along the joint number dimension, and then are fed into two cross-attention layers, $\mathcal{F}_p^{s\to m}$ and $\mathcal{F}_p^{m\to t}$, along with the encoded root joint position features. Therefore, the entire position process can be expressed following:

$$\hat{P}_B = \mathcal{F}_{tpos}(\mathcal{F}_p^{m\to t}(f_M^t, f_B^t, \mathcal{F}_p^{s\to m}(f_A^t, f_M^t, P_A))). \quad (5)$$

4.2 Spatial Relationship Preservation

The spatial relationship between different body parts is crucial for interpreting movement in character animation. Therefore, maintaining the relative positions of the body parts after motion transfer is a key concern of motion retargeting methods. Many previous methods, such as the method in [25], maintain semantic consistency by ensuring similarity in skeleton distance matrices before and after the process of motion retargeting. Figure 5 shows an example of a skeleton Distance Matrix (DM) in the center. However, these approaches neglect the impact of character shape. Therefore, we suggest adopting distances between surface vertices of the character instead of inter-joint distances to more accurately capture the spatial relationships among the body parts. Specifically, considering distances between surface vertices provides a more comprehensive understanding of the character's shape and posture. For example, the right side of Figure 5 illustrates a mesh Distance Matrix (DM), depicting distances among multiple vertices across different body parts.

Vertex spatial relationship loss. Most vertices of the character surface have little influence on the semantic information of the motion. However, the vertex connections between different body parts during movement, especially those that do not cross the body, are crucial for preserving the motion semantics. Based on that consideration, we employ a novel local spatial relationship constraint loss to supervise the consistency of the spatial relationship between mesh vertices after the motion retargeting process. Our approach includes filtering out the vertex pairs involved in the motion, the computation of their spatial relationships, and subsequently enforcing the spatial relationship consistency for mesh vertices after motion retargeting.

We filter out edges $E_H = (P_i, P_j)$ on the character mesh based on two criteria: 1. The vertices at both ends of the edge belong to different joints, ensuring the edge traverses the body; 2. The edge does not intersect with any triangle of the mesh, ensuring geometric integrity. However, identifying connections that do not pass through the body of the character mesh directly is impractical, as characters may consist of multiple disjointed or overlapping meshes. To tackle this challenge, we first obtain the convex hull H corresponding to the mesh V, and then select edges that meet the criteria on the convex hull H. Subsequently, we locate the vertices in the original mesh corresponding to the vertices of the convex hull, thereby identifying the edges that meet the criteria. We accelarate this process with CUDA. The left side of Figure 5 illustrates the original model, skeleton, mesh, and convex hull of a character, respectively.

After the filtered edges are acquired, the mesh distance matrix linked to the motion is calculated. Based on skinning weights, each unpenetrated body edge is linked to a joint and a pairwise joint distance is calculated by tallying the lengths of all edges under different joints, creating a distance matrix $D^V \in \mathbb{R}^{T\times J\times J}$. This matrix represents the spatial distance between the different body parts of the character. For more details of D^V, please refer to the supplementary materials.

It is important to note that the source and target characters may have different structures, leading to a mapping between the distance matrices of the source and target motions, denoted as $T_{A\to B}$. As mentioned in Section 4.1, the mapping relationship between motion features of the source and target can be learned by sub-modules $\mathcal{F}_c^{s\to m}$ and $\mathcal{F}_c^{m\to t}$. Here, we assume that this mapping relationship can also reflect the changes in spatial dis-

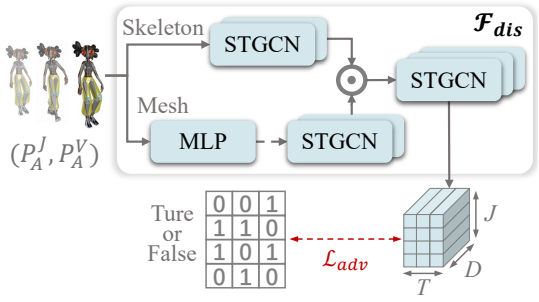


Fig. 6: Our discriminator architecture. The discriminator comprises a multi-layer STGCN framework with two branches: one processes the global joint positions, and the other processes the global vertex positions on the mesh surface.

tance between the body parts of the source and target characters with different structures. $T_{A\to B}$ is derived from the product of matrices $T_{s\to m}$ and $T_{m\to t}$, which represents the attention weights determined by the cross-attention layer within the sub-modules $\mathcal{F}_c^{s\to m}$ and $\mathcal{F}_c^{m\to t}$. In practical terms, a weight coefficient matrix $W \in \mathbb{R}^{J\times J}$ is incorporated into the distance matrix. This weight matrix prioritizes the influence of nearby body parts in the computation, and it is formulated as follows:

$$w_{ij} = \frac{exp(-\alpha\|P_{ij}\|)}{\sum_{ij} exp(-\alpha\|P_{ij}\|)} \times \beta, \tag{6}$$

where $P_{ij} = P_i - P_j$, $\alpha = 10$, $\beta = 10000$.

Finally, we add spatial relationship constraints on the source motion and the generated target motion. The entire loss is defined as follows:

$$\mathcal{L}_{vsp}(\mathcal{M}_A, \hat{\mathcal{M}}_B) = \|w_{ij}T_{A\to B}(D_A^V) - w_{ij}D_B^V\|_2^2, \tag{7}$$

where $(Q_A, P_A, \mathcal{C}_A) \in \mathcal{M}_A$, $(\hat{Q}_A, \hat{P}_A, \mathcal{C}_B) \in \hat{\mathcal{M}}_B$, D_A^V and D_B^V indicate the spatial distance between the body parts of the character motion, for the source and target characters, respectively.

4.3 Training

After receiving a source motion (Q_A, P_A) and the identity information of both the source and target characters $(\mathcal{C}_A, \mathcal{C}_B)$, the entire network is trained in an end-to-end manner using the cycle adversarial learning framework, incorporating unpaired data and the following loss terms.

Reconstruction loss: To enhance the network training, reconstruction loss terms are integrated for the global rotation of joints, global positions of joints, and global positions of character mesh vertices that generate motion. Specifically, the global rotation of each joint is calculated by multiplying the local rotations along the skeleton topology, while the global position of each joint and character mesh vertex is derived from the vertex set using linear blend skinning (LBS) to deform the character mesh. The complete loss term is defined as follows.

$$\begin{aligned}\mathcal{L}_{rec}(Q, P, \hat{Q}, \hat{P}, \mathcal{C}) &= \|f_G(Q, \gamma) - f_G(\hat{Q}, \gamma)\|_2^2 \\ &+\|\mathbf{LBS}(Q, P, \mathcal{C}) - \mathbf{LBS}(\hat{Q}, \hat{P}, \mathcal{C})\|_2^2\end{aligned} \tag{8}$$

where $(\hat{Q}, \hat{P})$ represents output motions, γ represents the skeleton topology, and f_G represents the process of calculating the global rotation based on the local rotation of the motion and the skeleton topology.

Cycle consistency loss. When provided with the input source motion $(Q_A, P_A, \mathcal{C}_A) \in \mathcal{M}_A$ and the target character identity information $\mathcal{C}_B$, the network generates the motion $(\hat{Q}_B, \hat{P}_B)$ for the target character. Subsequently, $(\hat{Q}_B, \hat{P}_B, \mathcal{C}_B) \in \mathcal{M}_B$ is fed back into the network to obtain the reconstructed motion $(\tilde{Q}_A, \tilde{P}_A)$ for the original source character. Finally, we enforce cycle consistency constraints between the original source input motion $(Q_A, P_A, \mathcal{C}_A)$ and the reconstructed motion $(\tilde{Q}_A, \tilde{P}_A, \mathcal{C}_A)$. Similar to the reconstruction loss, this constraint also applies to the global rotation and position of the joints, as well as the global position of the character mesh vertices that produce the motion.

$$\begin{aligned}\mathcal{L}_{cyc}(Q_A, P_A, \tilde{Q}_A, \tilde{P}_A, \mathcal{C}_A) &= \|f_G(Q_A, \gamma) - f_G(\tilde{Q}_A, \gamma)\|_2^2 \\ &+\|\mathbf{LBS}(Q_A, P_A, \mathcal{C}_A) - \mathbf{LBS}(\tilde{Q}_A, \tilde{P}_A, \mathcal{C}_A)\|_2^2\end{aligned} \tag{9}$$

Adversarial loss. To train the network with unpaired data and evaluate the realism of the generated results, we introduce a discriminator $\mathcal{F}_{dis}$ that takes the global positions of the mesh vertices and the skeleton joints as input, as depicted in Figure 6. The discriminator then produces a binary score for each feature vector created by the network, indicating its authenticity. The adversarial loss is defined as follows:

$$\begin{aligned}\mathcal{L}_{adv}(P_A^J, P_A^V, \hat{P}_B^J, \hat{P}_B^V) &= \mathbb{E}_{\mathcal{M}_A\sim p_{\text{real}}}[\|\mathcal{F}_{dis}(P_A^J, P_A^V)\|_2^2] \\ &+\mathbb{E}_{\mathcal{M}_B\sim p_{\text{fake}}}[\|1 - \mathcal{F}_{dis}(\hat{P}_B^J, \hat{P}_B^V)\|_2^2]\end{aligned} \tag{10}$$

where $p(\cdot)$ represents the distribution of motions. Our overall loss function is formulated as:

$$\mathcal{L} = \lambda_a\mathcal{L}_{adv} + \lambda_r\mathcal{L}_{rec} + \lambda_c\mathcal{L}_{cyc} + \lambda_v\mathcal{L}_{vsp} \tag{11}$$

where $\lambda_a = 1$, $\lambda_r = 100$, $\lambda_c = 100$, $\lambda_v = 10$ are parameters.

5 Experiments

In this section, we will first introduce the dataset used and outline our experimental setup. Following that, we will analyze the experimental results in three aspects: results and evaluation, ablation study, and user perception study. The findings demonstrate that our proposed method adeptly manages character motion retargeting across diverse skeleton structures, consistently outperforming existing and alternative methods.

5.1 Datasets and Implementation details

To ensure a comprehensive and fair evaluation of our method, we trained and evaluated it on two distinct publicly available datasets. The first dataset, Mixamo-SAN, compiled by SAN [7] from Mixamo, facilitates evaluation of both intra-structure and cross-structure motion retargeting tasks. The second dataset, Mixamo-R2ET, also sourced from Mixamo, was curated by R2ET [25] and features motion data with a consistent skeletal topology. Further details about these datasets can be found in the supplementary materials.

Our motion retargeting network underwent training and testing on a PC equipped with an Intel i9-10900X/3.70GHz CPU (32 GB RAM) and an NVIDIA GeForce RTX 3090 GPU. The framework was implemented using PyTorch [41], and we utilized the AdamW [42] optimizer with a learning rate of 0.0001, $\beta_1 = 0.5$, and $\beta_2 = 0.99$. The complete training process of the model took roughly 24 hours. When considering solely the model prediction time, disregarding data loading, the proposed model can attain an inference speed of around 450 frames per second (FPS).

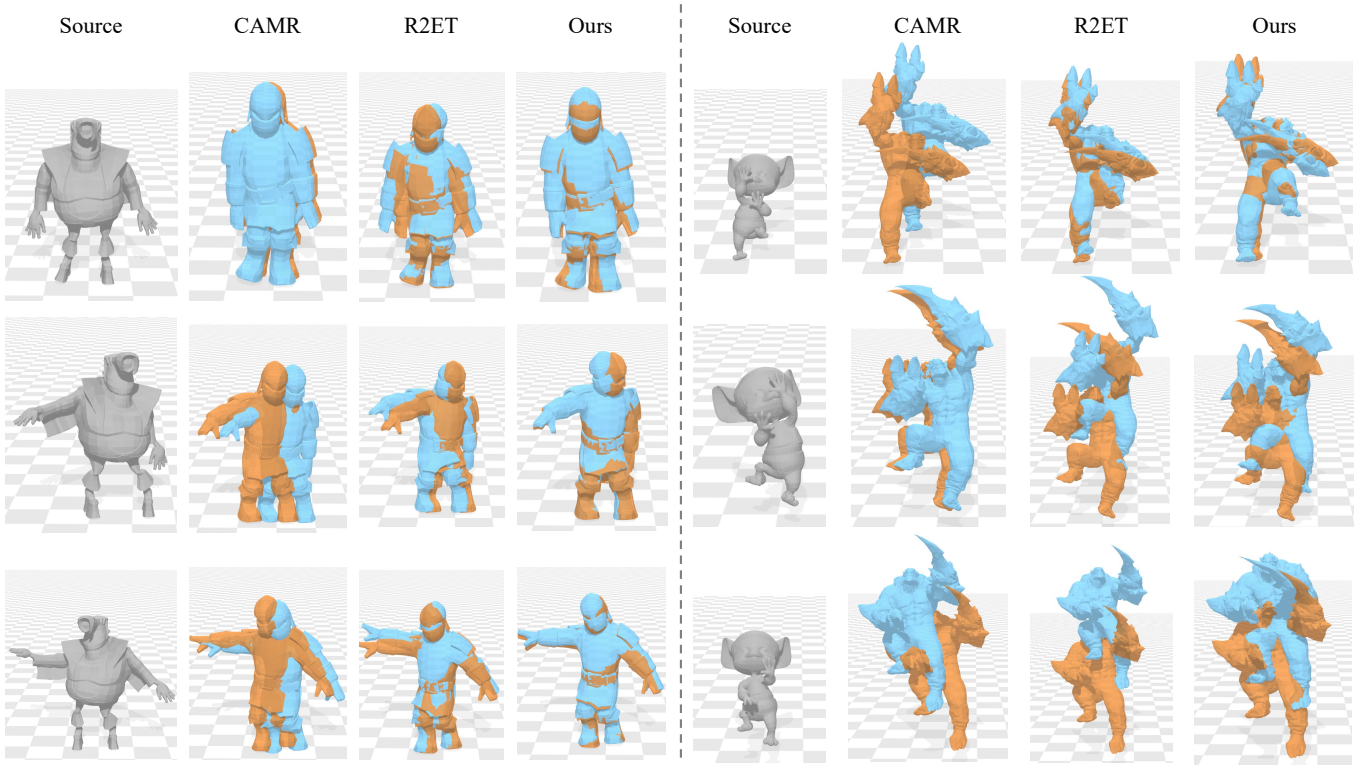


Fig. 7: Qualitative evaluation. We carried out comparative experiments on intra-structure motion retargeting using the CAMR [8] and R2ET [25] within the MixamoR2ET dataset. The input motion is represented in gray, while the motions generated by the CAMR and R2ET methods are depicted in blue. The ground-truth motion, serving as a reference, is illustrated in orange.

5.2 Evaluation metrics

In line with the SAN test suite setup, we partitioned the complete test set into two subsets: Group A and Group B. Group A comprises four characters that share the same skeletal framework. To evaluate the effectiveness of motion retargeting within identical structures, we retargeted the movements of each character in Group A to the others in the group. Conversely, Group B includes a character with a skeletal framework distinct from those in Group A. We transferred the motion from the Group B character to all members of Group A, thereby assessing the motion retargeting capability across diverse structures.

In assessing the accuracy and naturalness of motion retargeting, we utilized three metrics: Mean Square Error (MSE) of global vertex positions, MSE of global joint positions, and spatial relationship distance. These metrics showcase the efficacy of motion retargeting in preserving the positions of vertices and joints, as well as the relative spatial relationships of body parts for each character. All evaluation metrics are normalized by the character's height.

To obtain a more accurate spatial distance metric between body parts, we initially extract the character mesh vertices in the motion sequence and remove the vertex connections that pass through the body or belong to the same joint. This process resembles the computation of vertex spatial relationship loss $\mathcal{L}_{vsp}$. We then tally the lengths of all edges linking various joints to create the distance matrix D, which initially contains only zeros. Furthermore, in the context of cross-structure motion retargeting, we have formulated a skeletal mapping T_{gt} between the source character and the target character. Our spatial distance metric between joints is defined as:

$$D_{sp} = \frac{1}{N}\sum_{n=0}^{N}\left(\frac{D_{src}}{h_{src}} - \frac{T_{gt}(\hat{D}_{tar})}{h_{tar}}\right), \tag{12}$$

where N represents the length of the motion sequence, h denotes the height of the character, D_{src} stands for the spatial distance matrix of the source motion, and $\hat{D}_{tar}$ stands for the spatial distance matrix of the generated target motion. Additionally, despite the test set including paired motion data, it lacks the character's shape information, leading to self-interpenetration post-skeleton binding.

5.3 Results and Evaluations

In this section, we provide quantitative and qualitative comparisons of our approach with the CAMR and R2ET methods for intra-structure motion retargeting. Additionally, we evaluate our method against SAN, considering both intra-structure and cross-structure retargeting.

Intra-structure Retargeting on Mixamo-R2ET. We first compare our method with prior works, CAMR [8] and R2ET [25], for the task of intra-structure motion retargeting on the Mixamo-R2ET dataset. Table 1 summarizes the quantitative comparison results between our method and the aforementioned studies, while Figure 7 offers a visual comparison of these methods. The results demonstrate that our method achieves the best performance in terms of joint positions and mesh vertex locations, while R2ET maintains semantic relationships between different body parts at a level comparable to our approach. In the qualitative comparison, we overlay the given reference ground-truth motion (in orange) onto the generated motion (in blue). It can be observed that

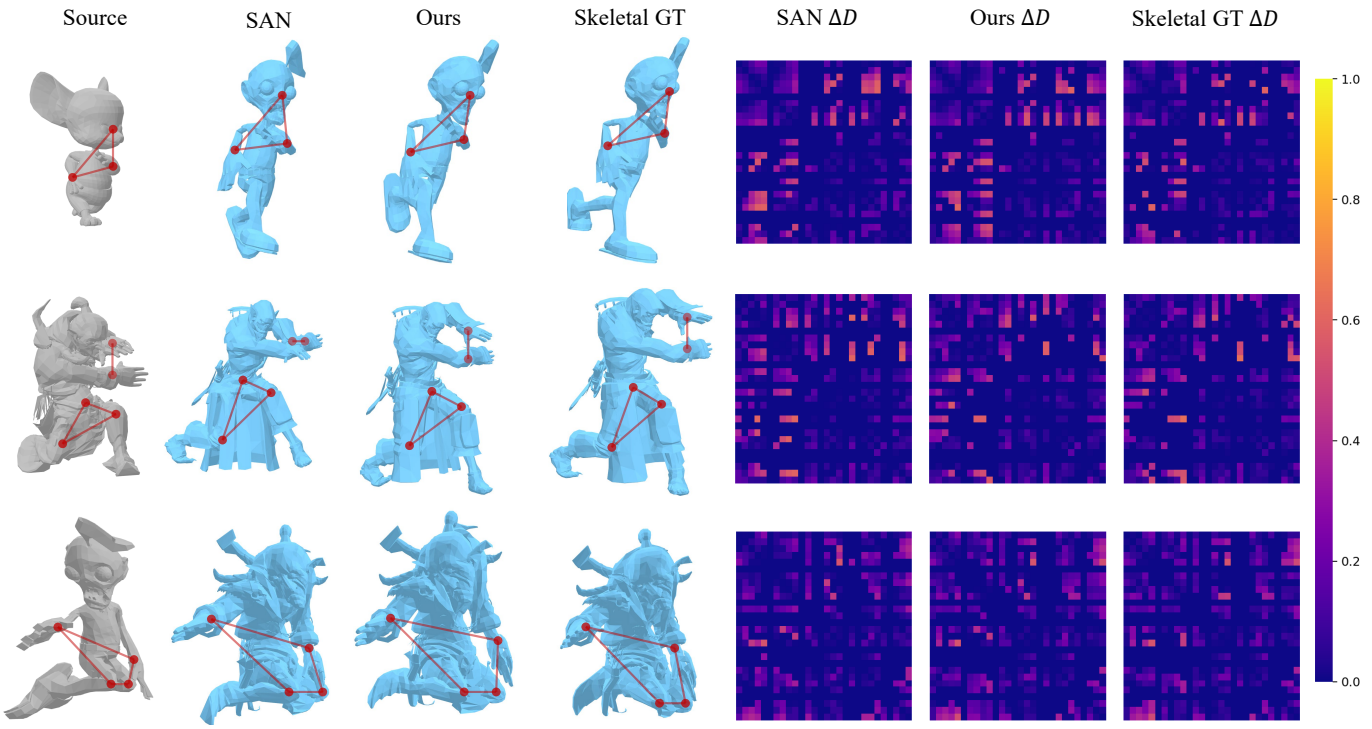


Fig. 8: Visual comparisions between our approach and the SAN [7] method in instra-structure motion retargeting. ΔD denotes the difference in spatial distances between corresponding body parts in the generated target motion and the source motion.

CAMR shows larger positional deviations. Over time, the error accumulation in the root position of the R2ET method becomes more apparent compared to our method. It is important to note that the CAMR results presented here are based on a model different from their official one, so their performance may not match the level reported in their work.

Intra-structure Retargeting on Mixamo-SAN. Figure 8 displays the outcomes of our method and SAN for intra-structure motion retargeting. Various techniques can generate poses for the target character that resemble those of the source character, while preserving consistent spatial distances between body parts. Additionally, the figure illustrates the disparities in the spatial distance matrices of different body parts among our method, SAN, and the provided ground truth of target and source motions. The calculation procedure for this spatial distance matrix is detailed in Equation 12, where smaller matrix values indicate a closer alignment of the generated target motion with the motion semantics of the source motion.

Upon reviewing the examples depicted in Figure 8, it becomes apparent that our method performs on par with SAN, exhibiting only a slightly superior distinction in the spatial distance matrix. In further quantitative comparisons, our method aligns with SAN's performance in three core metrics: global vertex position mean squared error ranging from 0.47 to 0.84, global joint position mean squared error ranging from 0.53 to 0.99, and spatial relationship distance ranging from 48.71 to 48.81. For detailed comparative results, please refer to Table 1.

Cross-structure Retargeting on Mixamo-SAN. The comparative results of our approach and SAN in cross-structure motion retargeting are depicted in Figure 9. When comparing the blue character poses with the gray character poses, it is evident that our method accurately reproduces the ground truth motion. Furthermore, the comparison of the distances between the red lines demonstrates that our method maintains superior spatial consistency across different body parts. Moreover, the spatial distance matrices (ΔD), which illustrate the variations in distance between the target and source motions across different body parts, further substantiate the advantages of our method.

TABLE 1: Quantitative comparison using global joint position MSE (J), global vertex position MSE (V), and spatial relationship distance (D) metrics. All metrics are normalized by the character's height and scaled by 10^3 for readability. The ($*$) indicates that the method is implemented by ourselves.

Methods	Intra-Structure			Cross-Structure		
	$J\downarrow$	$V\downarrow$	$D\downarrow$	$J\downarrow$	$V\downarrow$	$D\downarrow$
CAMR$*$	13.89	19.03	114.15	-	-	-
R2ET	8.24	9.04	101.69	-	-	-
Ours	**1.36**	**2.05**	**101.43**	-	-	-
SAN	**0.47**	**0.53**	**48.71**	2.25	3.09	223.24
Ours	0.84	0.99	48.81	**2.12**	**2.01**	**119.87**
Skeletal GT	-	-	43.95	-	-	115.51
Ours - Skeleton	0.89	1.18	50.58	**1.75**	1.85	121.17
Ours - Mesh	0.98	1.24	49.99	1.83	**1.85**	120.15
Ours - $\mathcal{L}_{jsp}$	**0.83**	**0.98**	**48.68**	2.42	2.18	121.56
Ours - $\mathcal{L}_{mix}$	0.99	1.17	49.32	2.15	1.99	122.28
Ours - $\mathcal{L}_{vsp}$	0.84	0.99	48.81	2.12	2.01	**119.87**

Table 1 presents quantitative comparisons, further demonstrating the superiority of our method in handling motion retargeting tasks involving different skeletal structures while preserving semantic continuity. Specifically, when training the network solely using skeleton data without the spatial relationship constraint loss, our method (indicated in the "Ours - Skeleton" row of Table 1) achieves a reduction in global joint position mean squared error from 2.25 to 1.75, global vertex position MSE from 3.09 to 1.85, and a significant decrease in spatial relationship distance from 223.24 to 121.17. These results reinforce the effectiveness of our approach in accurately retargeting motions across different body structures, highlighting its potential for practical applications.

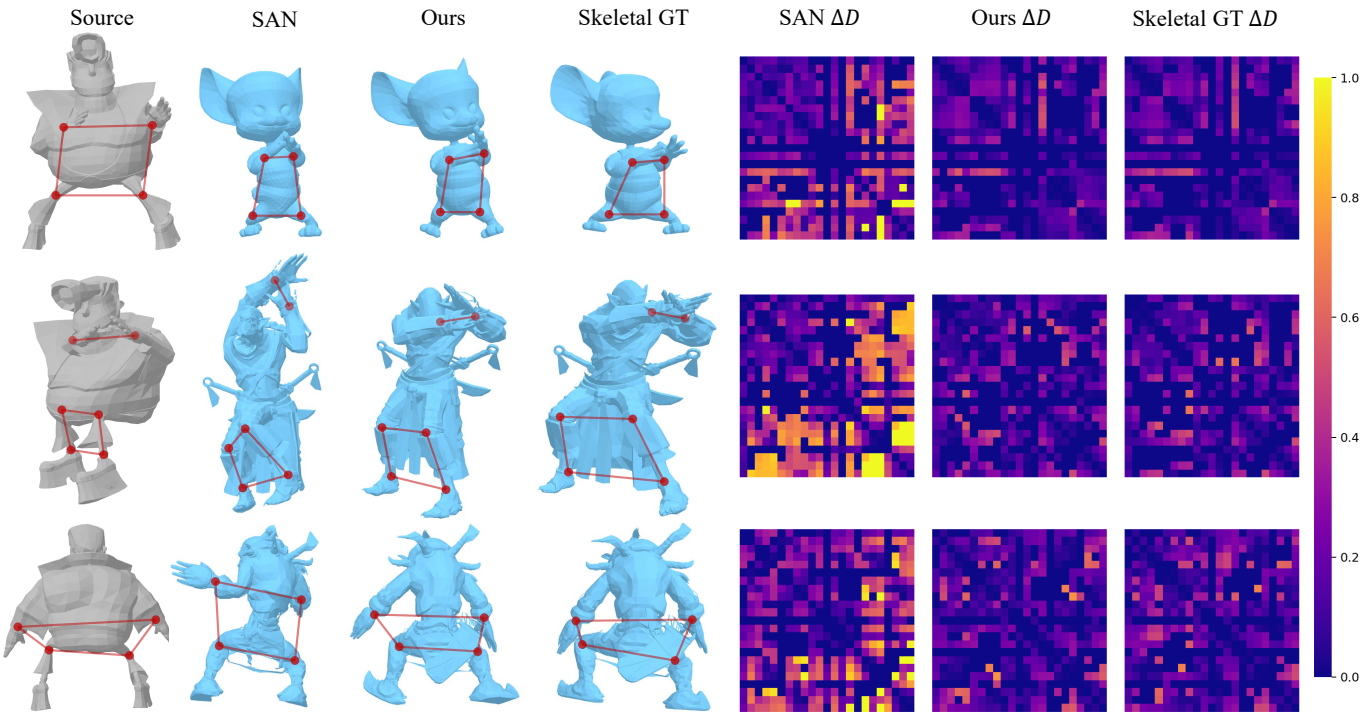


Fig. 9: Visual comparisions between our approach and the SAN [7] method in cross-structure motion retargeting. The depicted ΔD represents the disparity in spatial distances of corresponding body parts between the generated target motion and the source motion.

To further explore the performance in maintaining spatial relationship consistency, we also calculated the spatial relationship distance metric between the target skeletal ground-truth (GT) motion and the input source motion and observed the metric differences between our method and skeletal GT. The results in Table 1 show that our approach achieves a spatial relationship distance that closely matches the provided skeletal ground-truth data, with differences of 4.73 and 4.36 for the intra-structure and cross-structure retargeting tasks, respectively.

5.4 Ablation Study

In this section, we analyze the impact of key components of SMRNet on motion retargeting by comparing the outcomes of training with solely skeletal data against training with both skeletal and mesh data, aiming to evaluate the effect of mesh data on the final retargeting results. Furthermore, we examine the effects of three loss functions, $\mathcal{L}_{jsp}$, $\mathcal{L}_{vsp}$, and $\mathcal{L}_{mix}$. $\mathcal{L}_{jsp}$ is a spatial relationship constraint loss function widely adopted in earlier research, based on the spatial distances among joints. $\mathcal{L}_{vsp}$, presented in this study, is a similar type of loss function, but it is based on the distances between mesh vertices. Meanwhile, $\mathcal{L}_{mix}$, which combines $\mathcal{L}_{jsp}$ and $\mathcal{L}_{vsp}$, aims to investigate the interaction between these two loss functions. Specifically, it explores whether they synergistically enhance training or present conflicts. For detailed definitions of $\mathcal{L}_{jsp}$ and $\mathcal{L}_{mix}$, please refer to the supplementary material.

Figure 10 illustrates the visual comparisons, and Table 1 presents the quantitative metrics. The results indicate that networks trained with mesh data tend to generate poses with greater fidelity to the original motion input when compared to those trained solely on skeletal data. Specifically, the poses produced by the mesh-trained networks exhibit higher accuracy in capturing fine details of the input motion. As indicated by the red circle in the first row of Figure 10, the target pose generated from skeleton data exhibits overlap between the hand and leg, whereas the pose generated from mesh data keeps the hand and leg seperated.

Moreover, models trained with $\mathcal{L}_{vsp}$ demonstrate superior detail representation compared to those trained with $\mathcal{L}_{jsp}$ or $\mathcal{L}_{mix}$, as evidenced by the hand-leg distance and the bow-legged posture of characters highlighted in Figure 10. Experimental results also reveal that networks trained with the $\mathcal{L}_{mix}$ hybrid loss slightly underperform in intra-structure motion retargeting tasks compared to those trained solely with $\mathcal{L}_{vsp}$. In tasks of cross-structure motion retargeting, $\mathcal{L}_{mix}$ may even occasionally generate incorrect poses, potentially due to the network's inability to simultaneously satisfy the spatial constraints between joints and mesh vertices.

In the quantitative analysis presented in Table 1, a decrease in the spatial relationship distance metric corresponds to a slight increase in both the global vertex position MSE and global joint position MSE. This outcome is anticipated, as the network's adjustments to the target pose to maintain spatial consistency do not entirely align with the provided ground truth. To offer a more comprehensive evaluation of the generated results and the preservation of spatial relationships, we introduced user perception evaluations. For an in-depth discussion on this, please refer to the supplementary materials.

6 Discussion and Conclusion

This work introduces a novel approach, the Spatially-Preserving Skinned Motion Retargeting Network (SMRNet), which is tailored for transferring motion between characters with different shapes and skeletal structures. Initially, we enhance the source character with virtual joints to create a unified skeletal representation. Subsequently, we learn a hybrid skeleton and mesh representation and use it with our proposed motion transfer and root position transfer modules to enable motion mapping across characters with

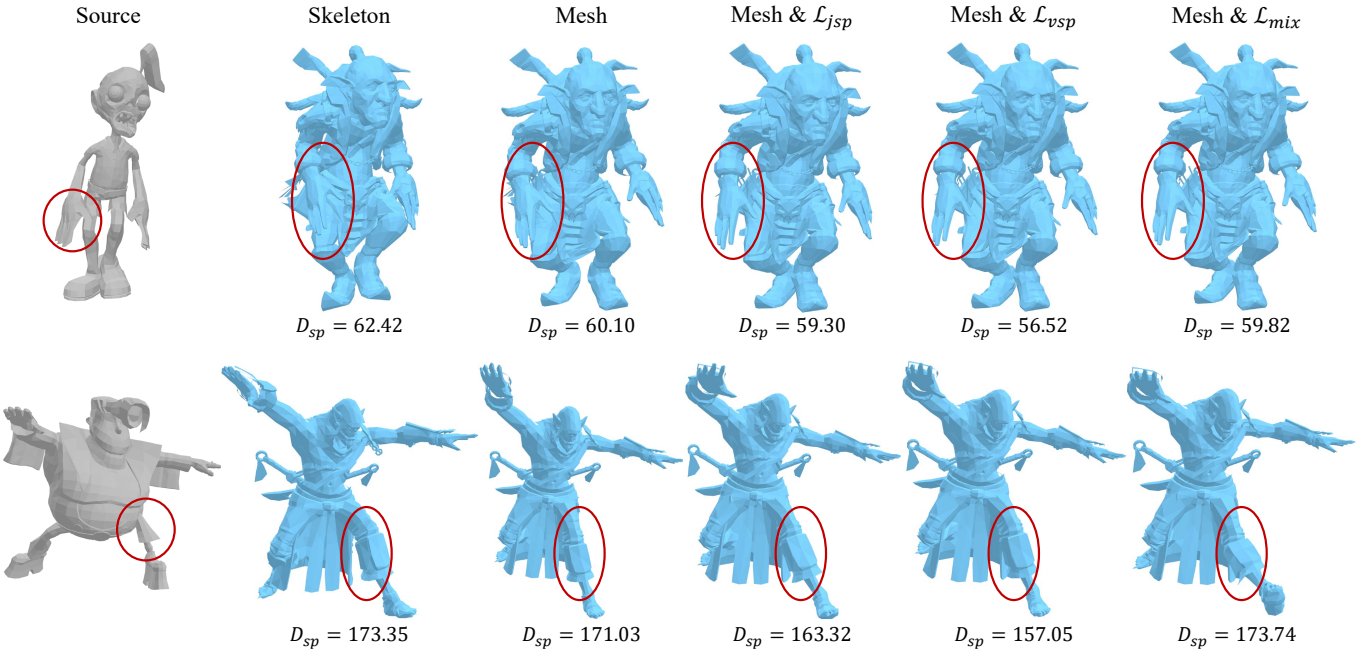


Fig. 10: Results of our method using various configurations, including training with skeletal data and mesh data, along with the results obtained by incorporating three distinct loss functions: $\mathcal{L}_{jsp}$, $\mathcal{L}_{vsp}$, and $\mathcal{L}_{mix}$. Top: intra-structure motion retargeting; Bottom: cross-structure motion retargeting. The variable D_{sp} represents the spatial relationship distance, which is utilized to measure the semantic consistency between the target and source motion.

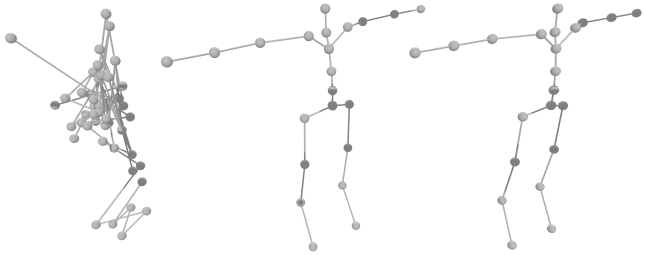


Fig. 11: Visualization of feature decoding. (a) shows the result obtained by decoding the learned hybrid skeleton representation for rest pose; (b) shows the decoding identity feature for rest pose; (c) shows the ground truth rest pose of the character in (b).

varied skeletal structures. Furthermore, we propose a new spatial relationship constraint loss. By constraining the distances between mesh vertices, we ensure the preservation of spatial relationships between character body parts prior to and following the process of motion retargeting. On the same dataset, our method demonstrates comparable performance with state-of-the-art techniques in intra-structure motion retargeting tasks. It also achieves superior results in cross-structure motion retargeting tasks, particularly excelling in maintaining spatial relationships.

A central question often receives attention: can we directly learn the joint motion mapping between characters with different structures using only their identity information, and what specific content has been learned by the hybrid representation f_M^t. In initial experiments, we tried to use the identity information of two characters to learn the joint motion mapping between them, but the results were significantly worse than the benchmark SAN method. We also established a decoding network to decode the extracted character identity features back into the original rest pose and used this network to observe what the hybrid representation

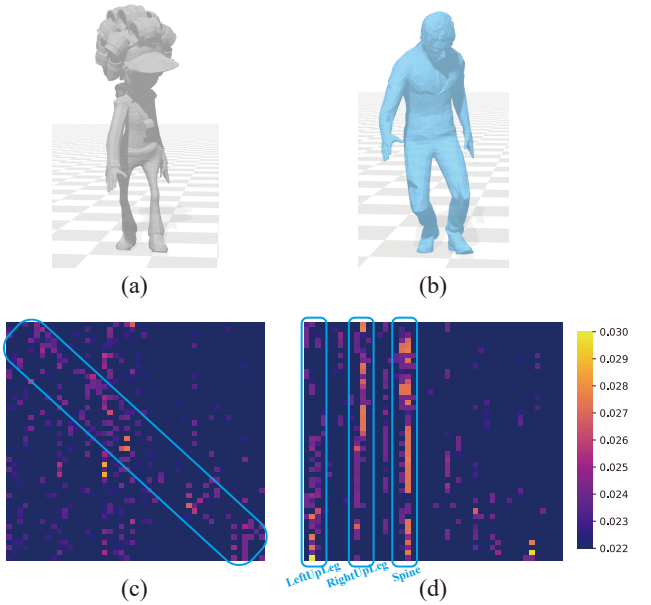


Fig. 12: Visualization of joint motion transfer. (a) Input pose of the source character; (b) Retargeted pose of the target character; (c) Heatmap displaying the joint motion mapping from the source character to the learned hybrid representation; (d) Heatmap showing the joint motion mapping from the learned hybrid representation to the target character.

f_M^t learned. The results showed that the poses decoded from the hybrid representation f_M^t were disordered and not standard rest poses. In Figure 11, we demonstrate the outcome of decoding the skeletal feature from the learned hybrid representation f_M^t, the results of the network decoding the character identity features, and the corresponding ground-truth rest pose of the character. Therefore, we concluded that relying solely on character identity

information for motion retargeting between characters with different structures is ineffective. The hybrid representation f_M^t did not simply learn an average pose of all character identity information, but instead, it learned a unified encoding that combines skeleton and mesh information based on the identity information of characters with different structures. This encoding serves as a bridge that can quickly calculate the joint motion mapping between characters with different skeletal structures by comparing different structural character identity features with the hybrid representation f_M^t.

Another frequent question pertains to the content of the motion mapping relationships computed in the motion transfer module. We demonstrate an example in Figure 12, showing the collective motion mapping relationships generated by the motion transfer module during the motion retargeting process. This mapping is represented by two heatmaps. Upon inspection, it is clear that the values in both heatmaps mainly fall between 0.02 and 0.03, indicating the transfer of motion features from each joint of the source character to all joints of the target character. Further analysis of the heatmaps in Figure 12 shows distinct areas, suggesting that aside from taking into account the motion features of each joint of the source character, the motion retargeting process specifically highlights the motion features of the corresponding joints of the target character. Notably, in the areas marked with blue boxes in Figure 12, we can see that in heatmap (c), higher values are mainly along the diagonal, while in heatmap (d), the distribution of joint motion features is more concentrated on the "LeftUpLeg", "RightUpLeg", and "Spine". This suggests that when retargeting motion to a target character, the network prioritizes the motion of these three parent joints, which may be more important for generating accurate motion in the extremity joints.

Limitations and Future Work. Our method mainly focuses on cross-structure motion retargeting for humanoid characters, leaving a significant gap in achieving motion retargeting between quadruped animals and humanoid characters. Additionally, our approach to calculating distances between vertices of character parts relies on a simple averaging method. While more complex calculation methods, like using weights based on vertex skinning, could improve results, they also require much more computation time. Moreover, the proposed spatial relationship constraint loss, which is based on mesh vertices, faces challenges in effectively accommodating the spatial relationships among all body parts simultaneously during network training. Consequently, the model tends to prioritize preserving the distances between specific body parts. It is more likely to maintain these proximities for parts that are naturally close to each other, even when minor movements occur. Future research should concentrate on creating efficient and accurate methods for calculating distances between body parts, possibly with a user-friendly approach for fine-tuning motion retargeting outcomes. For example, investigating the use of large language models to naturally adjust retargeted motion through natural language input could be a promising way to enhance the interactivity when applying the method.

ACKNOWLEDGMENTS

This work was supported by the National Natural Science Foundation of China (Project Number: 62372025 and 61932003) and the Fundamental Research Funds for the Central Universities.

REFERENCES

[1] Katsu Yamane, Yuka Ariki, and Jessica Hodgins. Animating non-humanoid characters with human motion data. In *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 169–178, 2010.

[2] Jiawen Chen, Shahram Izadi, and Andrew Fitzgibbon. Kinêtre: animating the world with the human body. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*, pages 435–444, 2012.

[3] Yeongho Seol, Carol O'Sullivan, and Jehee Lee. Creature features: online motion puppetry for non-human characters. In *Proceedings of the 12th ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 213–221, 2013.

[4] Ufuk Celikcan, Ilker O Yaz, and Tolga Capin. Example-based retargeting of human motion to arbitrary mesh models. In *Computer Graphics Forum*, volume 34, pages 216–227. Wiley Online Library, 2015.

[5] Dafni Antotsiou, Guillermo Garcia-Hernando, and Tae-Kyun Kim. Task-oriented hand motion retargeting for dexterous manipulation imitation. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0, 2018.

[6] Hyewon Pyun, Yejin Kim, Wonseok Chae, Hyung Woo Kang, and Sung Yong Shin. An example-based approach for facial expression cloning. In *ACM SIGGRAPH 2006 Courses*, pages 23–es. 2006.

[7] Kfir Aberman, Peizhuo Li, Dani Lischinski, Olga Sorkine-Hornung, Daniel Cohen-Or, and Baoquan Chen. Skeleton-aware networks for deep motion retargeting. *ACM Trans. Graph.*, 39(4), aug 2020.

[8] Ruben Villegas, Duygu Ceylan, Aaron Hertzmann, Jimei Yang, and Jun Saito. Contact-aware retargeting of skinned motion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9720–9729, 2021.

[9] Zhiguang Liu, Antonio Mucherino, Ludovic Hoyet, and Franck Multon. Surface based motion retargeting by preserving spatial relationship. In *Proceedings of the 11th ACM SIGGRAPH Conference on Motion, Interaction and Games*, pages 1–11, 2018.

[10] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.

[11] Michael Gleicher. Retargetting motion to new characters. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '98, page 33–42, New York, NY, USA, 1998. Association for Computing Machinery.

[12] Peizhuo Li, Kfir Aberman, Zihan Zhang, Rana Hanocka, and Olga Sorkine-Hornung. Ganimator: Neural motion synthesis from a single sequence. *ACM Transactions on Graphics (TOG)*, 41(4):1–12, 2022.

[13] Ruben Villegas, Jimei Yang, Duygu Ceylan, and Honglak Lee. Neural kinematic networks for unsupervised motion retargetting. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[14] Jongin Lim, Hyung Jin Chang, and Jin Young Choi. Pmnet: Learning of disentangled pose and movement for unsupervised motion retargeting. In *30th British Machine Vision Conference 2019, BMVC 2019, Cardiff, UK, September 9-12, 2019*, page 136. BMVA Press, 2019.

[15] Zhouyingcheng Liao, Jimei Yang, Jun Saito, Gerard Pons-Moll, and Yang Zhou. Skeleton-free pose transfer for stylized 3d characters. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part II*, pages 640–656. Springer, 2022.

[16] Jehee Lee and Sung Yong Shin. A hierarchical approach to interactive motion editing for human-like figures. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 39–48, 1999.

[17] Antonin Bernardin, Ludovic Hoyet, Antonio Mucherino, Douglas Gonçalves, and Franck Multon. Normalized euclidean distance matrices for human motion retargeting. In *Proceedings of the 10th International Conference on Motion in Games*, pages 1–6, 2017.

[18] Hanyoung Jang, Byungjun Kwon, Moonwon Yu, Seong Uk Kim, and Jongmin Kim. A variational u-net for motion retargeting. In Nafees Bin Zafar and Kun Zhou, editors, *SIGGRAPH Asia 2018 Posters, Tokyo, Japan, December 04-07, 2018*, pages 1:1–1:2. ACM, 2018.

[19] Bing Yu, Haoteng Yin, and Zhanxing Zhu. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, pages 3634–3640. ijcai.org, 2018.

[20] Kwang-Jin Choi and Hyeong-Seok Ko. Online motion retargetting. *The Journal of Visualization and Computer Animation*, 11(5):223–235, 2000.

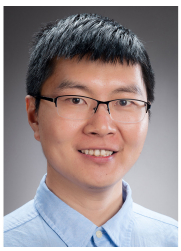
[21] Andrew Feng, Yazhou Huang, Yuyu Xu, and Ari Shapiro. Automating the transfer of a generic set of behaviors onto a virtual character. In *Motion in Games: 5th International Conference, MIG 2012, Rennes, France, November 15-17, 2012. Proceedings 5*, pages 134–145. Springer, 2012.

[22] Chris Hecker, Bernd Raabe, Ryan W Enslow, John DeWeese, Jordan Maynard, and Kees van Prooijen. Real-time motion retargeting to highly varied user-created morphologies. *ACM Transactions on Graphics (TOG)*, 27(3):1–11, 2008.
[23] Brian Delhaisse, Domingo Esteban, Leonel Rozo, and Darwin Caldwell. Transfer learning of shared latent spaces between robots with similar kinematic structure. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 4142–4149. IEEE, 2017.
[24] Tejas D Kulkarni, William F Whitney, Pushmeet Kohli, and Josh Tenenbaum. Deep convolutional inverse graphics network. *Advances in neural information processing systems*, 28, 2015.
[25] Jiaxu Zhang, Junwu Weng, Di Kang, Fang Zhao, Shaoli Huang, Xuefei Zhe, Linchao Bao, Ying Shan, Jue Wang, and Zhigang Tu. Skinned motion retargeting with residual perception of motion semantics & geometry. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13864–13872, 2023.
[26] Jean Basset, Stefanie Wuhrer, Edmond Boyer, and Franck Multon. Contact preserving shape transfer for rigging-free motion retargeting. In *Proceedings of the 12th ACM SIGGRAPH Conference on Motion, Interaction and Games*, pages 1–10, 2019.
[27] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, 34(6):248:1–248:16, October 2015.
[28] Pietro Musoni, Riccardo Marin, Simone Melzi, and Umberto Castellani. Reposing and retargeting unrigged characters with intrinsic-extrinsic transfer. *Smart Tools and Applications in Graphics*, 2021.
[29] Sen Wang, Xinxin Zuo, Runxiao Wang, Fuhua Cheng, and Ruigang Yang. A generative human-robot motion retargeting approach using a single depth sensor. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5369–5376. IEEE, 2017.
[30] Sunwoo Kim, Maks Sorokin, Jehee Lee, and Sehoon Ha. Humanconquad: Human motion control of quadrupedal robots using deep reinforcement learning. In *SIGGRAPH Asia 2022 Emerging Technologies*, pages 1–2. 2022.
[31] George Fletcher, Yiguo Qiao, Rebecca Fribourg, Jake Deane, Rachel McDonnell, and Darren Cosker. Exploring the perception of quadruped motion retargeting. In *14th ACM SIGGRAPH Conference on Motion, Interaction and Games*, 2021.
[32] Gaurav Bharaj, Thorsten Thormählen, Hans-Peter Seidel, and Christian Theobalt. Automatically rigging multi-component characters. In *Computer Graphics Forum*, volume 31, pages 755–764. Wiley Online Library, 2012.
[33] Andrea Sanna, Fabrizio Lamberti, Gianluca Paravati, Gilles Carlevaris, and Paolo Montuschi. Virtual character animations from human body motion by automatic direct and inverse kinematics-based mapping. *EAI Endorsed Transactions on Creative Technologies*, 2(2):e6–e6, 2015.
[34] Michel Abdul-Massih, Innfarn Yoo, and Bedrich Benes. Motion style retargeting to characters with different morphologies. In *Computer Graphics Forum*, volume 36, pages 86–99. Wiley Online Library, 2017.
[35] Eray Molla, Henrique Galvan Debarba, and Ronan Boulic. Egocentric mapping of body surface constraints. *IEEE transactions on visualization and computer graphics*, 24(7):2089–2102, 2017.
[36] Edmond SL Ho, Taku Komura, and Chiew-Lan Tai. Spatial relationship preserving character motion adaptation. In *ACM SIGGRAPH 2010 papers*, pages 1–8. 2010.
[37] Taeil Jin, Meekyoung Kim, and Sung-Hee Lee. Aura mesh: Motion retargeting to preserve the spatial relationships between skinned characters. In *Computer Graphics Forum*, volume 37, pages 311–320. Wiley Online Library, 2018.
[38] Hao Li, Robert W Sumner, and Mark Pauly. Global correspondence optimization for non-rigid registration of depth scans. In *Computer graphics forum*, volume 27, pages 1421–1430. Wiley Online Library, 2008.
[39] Tianyu Li, Jungdam Won, Alexander Clegg, Jeonghwan Kim, Akshara Rai, and Sehoon Ha. Ace: Adversarial correspondence embedding for cross morphology motion retargeting from human to nonhuman characters. In *SIGGRAPH Asia 2023 Conference Papers*, pages 1–11, 2023.
[40] Yunbo Zhang, Deepak Gopinath, Yuting Ye, Jessica Hodgins, Greg Turk, and Jungdam Won. Simulation and retargeting of complex multi-character interactions. In *ACM SIGGRAPH 2023 Conference Proceedings*, pages 1–11, 2023.
[41] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.
[42] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Jia-Qi Zhang is currently a student with the State Key Laboratory of Virtual Reality Technology and Systems, Beihang University. He obtained the bachelor degree and master degree in software engineering from North China Electric Power University, Beijing, China, in 2016 and 2020, respectively. His research interests include computer vision, computer graphics and virtual reality.

Miao Wang is an Associate Professor with the State Key Laboratory of Virtual Reality Technology and Systems, School of Computer Science and Engineering, Beihang University, China. His research interests include Virtual Reality, Computer Graphics and Visual Computing. During 2016-2018, he did postdoc research in Visual Computing at Tsinghua University. He received his Ph.D. degree from Tsinghua University in 2016 and the bachelor degree from Xidian University in 2011. He serves a program committee member of IEEE VR and ISMAR conferences. He is a member of IEEE, ACM and AsiaGraphics.

Fu-Cheng Zhang is currently a student with the State Key Laboratory of Virtual Reality Technology and Systems, Beihang University. He obtained the bachelor degree in computer science and technology from Central South University, Changsha, China, in 2022. His research interests include computer vision, computer graphics and virtual reality.



Fang-Lue Zhang is currently a senior lecturer in computer graphics with Victoria University of Wellington, New Zealand. He received the Doctoral degree from Tsinghua University, Beijing, China, in 2015, and the Bachelor's degree from Zhejiang University, Hangzhou, China, in 2009. His research interests include image and video editing, computer vision, and computer graphics. He received Victoria Early-Career Research Excellence Award in 2019. He served as program chair of Pacific Graphics 2020 & 2021 and Computational Visual Media 2024. He is on the editorial board of Computer & Graphics. He is a committee member of IEEE Central New Zealand Sector.