Virtual Pose Coach: A Motion-Retargeting Approach for Pose Training

Tzu-Chun Chiu Ming-Han Lee Kun-Ru Wu Yu-Shuen Wang Yu-Chee Tseng

Department of Computer Science, National Yang Ming Chiao Tung University
Hsinchu, Taiwan
{zihjyun.cs11, mhlee.cs09, wufish, yushuen, yctseng}@nycu.edu.tw

Abstract

In sports training, effective learning necessitates that students accurately replicate their coach's movements. However, anatomical differences, such as variations in limb length and skeletal proportions, can make it difficult to achieve proper pose alignment and may diminish training effectiveness. To tackle this issue, we propose a virtual pose coach framework. This innovative approach uses motion retargeting to create personalized virtual poses tailored to each student's body structure, which enhances their ability to imitate the coach's movements accurately. Unlike current methods that adjust joint rotations between characters and often experience discrepancies in rotation distributions during training and testing – due to the countless possible transitions between two poses –our framework focuses on joint positions for retargeting. This strategy eliminates ambiguity and enables effective motion retargeting across different body structures, facilitating motion transfer between skeletons with varying anatomical designs. We illustrate the advantages of our system through a case study that shows how our retargeting method significantly enhances students' ability to replicate a coach's movements, indicating its potential to improve sports training outcomes.

1. Introduction

Mastering proper poses and motions is essential for beginners aiming to improve their skills in sports like golf and tennis. With advancements in communication and digital twin technologies, virtual coaching has become a feasible solution for "anytime, anywhere" learning [3, 20]. Various coaching systems have been developed to enhance training efficiency, leveraging sensor-based data collection [6, 23], AR Visualization [18], sports science theories [5, 7, 9], and imitation-based learning [10, 12, 16, 22]. However, a critical challenge remains: these systems often overlook the anatomical differences, such as variations in height and bone length, between beginners and professional athletes, which can hinder the effectiveness of training outcomes.

To address this challenge, this work focuses on motion retargeting - a technique for transferring motion from a coach's skeleton to a student's skeleton, before measuring the similarity between their motion sequences. Traditional approaches to motion retargeting [8] framed it as a spatiotemporal optimization problem, requiring the manual design of energy functions. With the advent of larger motion datasets, data-driven approaches [17, 21] have been proposed to achieve automated motion retargeting. Early methods, however, assumed that the source and target skeletons shared identical articulated structures. To overcome this limitation, recent approaches [2, 11] introduced the shared latent space for cross-structural retargeting. This advancement enables motion transfer between skeletons with different topologies, facilitating more robust motion adaptation across diverse body shapes and anatomical structures.

Early motion retargeting models use the joint positions of a source character as their input. However, recent techniques, such as PAN [11], first transform these joint positions into joint rotations before applying the animation to a target character. Generally, methods based on rotations produce higher quality results because they limit the degrees of freedom, ensuring that bone lengths remain consistent throughout the animation. However, calculating the joint rotations between two 3D poses is complex [19, 21], as there can be countless paths connecting the two poses. Additionally, during testing, the inputs to the model may fall outside the range of what it was trained on, leading to retargeting errors, which hinders the model's overall ability to generalize effectively.

To address the limitations of rotation-based methods, we present an enhanced version of the PAN model that uses joint positions as input while maintaining its ability to retarget motions effectively. Our method modifies the input format, data preprocessing techniques, and loss functions, which allows for clear and efficient motion retargeting. By taking joint positions as input, our model avoids the ambiguities often present in joint rotations.

Our approach improves coaching in sports poses by enabling students to replicate their coaches' movements effectively. We employ the dynamic time warping technique to compare the motion sequences of both the coach and student, regardless of differences in timing. This technique allows us to quantitatively evaluate how closely the student's poses match the coach's, accommodating individual anatomical variations and offering a tailored training experience. Furthermore, we include quantitative evaluations to demonstrate the effectiveness of our method for both intra-structural and cross-structural skeleton retargeting using the Mixamo dataset. Our contributions are outlined below:

- Revise the PAN model [11] by using joint positions instead of joint rotation for retargeting. Our approach also introduces redesigned input format, data preprocessing, and loss function, leading to better motion efficiency.
- Develop a system to analyze golf putting movements by comparing students' and coaches' virtual motions to provide posture suggestions and score. This demonstrates its potential to enhance the effectiveness of sports training in real-world applications.
- Extensive evaluations on the Mixamo dataset show that our method outperforms other position-based approaches on both intra-structure and cross-structure skeletons.

2. Related Work

The field of precision sports has made substantial progress, leveraging a variety of technologies to enhance training and performance assessment. Sensor-based systems have been widely explored, using pressure sensors to monitor center of pressure [6], body weight balance [23], and providing visual feedback [18]. Optical motion tracking [12] and virtual reality [10] have also been applied to improve posture visualization and interactive training experiences. Systems based on sports science theories focus on metrics such as center of gravity [9] and movement statistics [7] to guide users in optimizing their movements. Advanced pose estimation models and motion analysis techniques have been developed to identify and correct improper poses [5, 22], and analyze spatial and temporal differences between users and professionals [16]. These approaches aim to provide more personalized and effective training solutions, integrating advanced machine learning models and real-time feedback mechanisms for improved accuracy and usability.

Motion retargeting refers to the process of transferring motion from one character to another while maintaining the fidelity of the original poses and movements. Traditional methods primarily relied on constraint-based approaches, where motion adaptation was achieved through the use of spacetime constraints [8] or smooth motion transitions enabled by inverse kinematics and B-spline interpolation [15]. While effective, these methods required extensive manual tuning and lacked scalability.

To address these limitations, data-driven methods have been introduced to achieve end-to-end motion retargeting. These methods can be broadly categorized into position-based and rotation-based approaches. (1) Position-based methods focus on directly manipulating joint positions. Approaches like [1] decompose 2D pose sequences into motion, skeleton, and camera view, which are then reassembled to produce retargeted 2D motion. Similarly, [25] disentangled skeleton sequences into motion, structure, and view, enabling precise 2D-to-3D retargeting. For 3D motion, [21] employed RNNs with forward kinematics to retarget motion between 3D skeletons, while [17] enhanced adaptability by learning separate embeddings for pose and movement. (2) Rotation-based methods operate on joint rotations rather than positions, enabling better alignment of joint orientations. In [2], cross-structural retargeting is achieved by embedding motion from skeletons with different topologies into a shared latent space. Building on this, [11] divided the skeleton into body segments and employs attention mechanisms to dynamically adjust joint weights, further enhancing retargeting performance.

3. Methodology

Our objective is to develop a model that enables a student to replicate a coach's pose sequence in sports activities such as golf or tennis. Since the height and the bone lengths between a coach and a student are often different, the coach's pose sequence is then retargeted to the student's skeletal structure, producing a student's virtual pose sequence. Fig. 1 illustrates our pose retargeting framework, which takes three inputs: (1) the coach's 3D pose sequence, denoted as $\mathbf{P}^c \in \mathbb{R}^{\ell \times m \times 3}$, where ℓ is the sequence length and m represents the number of joints in the skeleton; (2) the student's reference T-pose, denoted as $\mathbf{T}^s \in \mathbb{R}^{m' \times 3}$, where m' is the number of joints; and (3) the student's 3D pose sequence, denoted as $\mathbf{P}^s \in \mathbb{R}^{\ell' \times m' \times 3}$. Notably, m and m' can differ due to variations in keypoint formats between the coach and student.

Once the coach's pose sequence is retargeted to the student's skeleton, the resulting virtual sequence is represented as $\mathbf{P}^{c \to s} \in \mathbb{R}^{\ell \times m' \times 3}$. To assess the student's mimicry performance, we compute the discrepancy between the virtual pose sequence $\mathbf{P}^{c \to s}$ and the student's observed pose sequence $\mathbf{P}^s$ using dynamic time warping. This approach provides a robust measure of the alignment between the student's movements and the coach's intended poses.

3.1. Countless Possible Transitions between Poses

A pose can be described in terms of either the positions of joints or the rotations of joints in relation to a standard T-pose. Recent studies [2, 11] that focus on joint rotations have shown remarkable success. However, this method has a significant limitation: many different rotations can con-

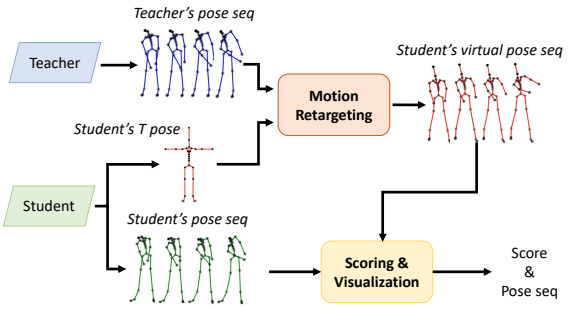

Figure 1. The coach's pose sequence is customized to match the student's body structure, resulting in a personalized virtual pose sequence. These two sequences are compared to produce a score and visual representation of the differences.

vert the T-pose into the same target pose [19]. This problem becomes particularly important when a model trained on certain joint rotations faces new, unseen rotations during real-world application. Such new rotations often fall outside the range of what the model was trained on, which can hinder the effectiveness of rotation-based methods.

To illustrate this problem, we conduct an experiment using the rotation-based method PAN model [11]. Let the source character's pose at frame i be denoted as $\mathbf{P}_i^{src} \in \mathbb{R}^{m\times 3}$, where m represents the number of joints. This pose $\mathbf{P}_i^{src}$ can be generated by applying a rotation representation $\mathbf{R}_i^{src}$ to the source T-pose $\mathbf{T}^{src}$ using forward kinematics (FK). Conversely, given the same source pose $\mathbf{P}_i^{src}$, we can compute a different rotation variant $\mathbf{R}_i^{src'}$ via inverse kinematics (IK) relative to $\mathbf{T}^{src}$. By applying this alternative rotation $\mathbf{R}_i^{src'}$ to the T-pose using forward kinematics, we obtain a new pose $\mathbf{P}_i^{src'}$. The entire process is mathematically formulated as:

$$\mathbf{P}_i^{src} = FK(\mathbf{T}^{src}, \mathbf{R}_i^{src}), \tag{1}$$

$$\mathbf{R}_i^{src'} = IK(\mathbf{T}^{src}, \mathbf{P}_i^{src}), \tag{2}$$

$$\mathbf{P}_i^{src'} = FK(\mathbf{T}^{src}, \mathbf{R}_i^{src'}). \tag{3}$$

We adapt the pose from the source skeleton to the target skeleton by applying the previously established joint rotations, $\mathbf{R}_i^{src}$ and $\mathbf{R}_i^{src'}$, starting from a common target T-pose $\mathbf{T}^{tar}$. Given that the skeletal structures of $\mathbf{T}^{src}$ and $\mathbf{T}^{tar}$ may vary, we refine the joint rotations using the PAN model. The resulting poses are defined as follows:

$$\mathbf{P}_i^{tar} = FK(\mathbf{T}^{tar}, PAN(\mathbf{T}^{tar}, \mathbf{R}_i^{src})), \tag{4}$$

$$\mathbf{P}_i^{tar'} = FK(\mathbf{T}^{tar}, PAN(\mathbf{T}^{tar}, \mathbf{R}_i^{src'})). \tag{5}$$

Fig. 2 provides a visual comparison of these poses. Subfigures (a) and (c) illustrate that $\mathbf{P}_i^{src}$ and $\mathbf{P}_i^{src'}$ are highly similar. This result is expected because $\mathbf{P}_i^{src'}$ is obtained by forward and inverse kinematics applied to $\mathbf{P}_i^{src}$. However, a different outcome is observed when the pose is retargeted. As shown in subfigures (b) and (d), the two retargeted poses $\mathbf{P}_i^{tar}$ and $\mathbf{P}_i^{tar'}$ exhibit significant differences, even though they originate from similar transformations of the same source pose $\mathbf{P}_i^{src}$.

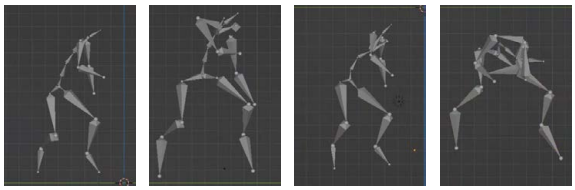

Figure 2. The poses shown in (a) and (c) are similar but represented with different joint rotations (i.e., $\mathbf{R}_i^{src}$ and $\mathbf{R}_i^{src'}$) in relation to the T-pose of the source character. However, when these poses are transferred to a target character, the outcomes differ significantly, likely because $\mathbf{R}_i^{src'}$ is not within the training set.

3.2. Position-based Motion Retargeting

The current sports datasets [14, 24] provide 3D human poses, which are position-based data. Building on the previous analysis, we present a redesigned version of the PAN model [11] that facilitates position-based motion retargeting. While maintaining the original structure, the skeleton is divided into n key segments, such as the torso, head, and four limbs. Features of these segments are extracted and then organized for retargeting. To support position-based input, we adjust the tensor dimensions and modify the loss functions. The model produces joint rotations for the target character. We then use forward kinematics (FK) to convert the T-pose $\mathbf{T}^s$ into the student's virtual pose sequence $\mathbf{P}^{c\to s}$. This approach allows the position-based model to effectively manage motion retargeting tasks while ensuring high-quality results.

Data-Preprocessing. We create a normalized local pose sequence, denoted as $p^c \in \mathbb{R}^{\ell\times m\times 3}$, from $\mathbf{P}^c$, which is defined in the world coordinate system. In this context, we use the hip joint as the root joint, from which we establish five kinematic chains that extend to the endpoints of the head, left hand, right hand, left foot, and right foot. Each joint in p^c is represented by its relative position in relation to its parent joint according to these kinematic chains, with the root joint fixed at the coordinates (0, 0, 0). To account for the movement of the root joint, we represent its normalized trajectory in the world coordinate system as $v^c \in \mathbb{R}^{\ell\times 1\times 3}$. For clarity, we define the combined sequence as $\mathbf{X}^c = p^c \oplus v^c \in \mathbb{R}^{\ell\times(m+1)\times 3}$.

Network Architecture. Fig. 3 shows our motion retargeting module. It consists of three stages. The first stage is feature extraction, which consists of a skeletal feature ex-

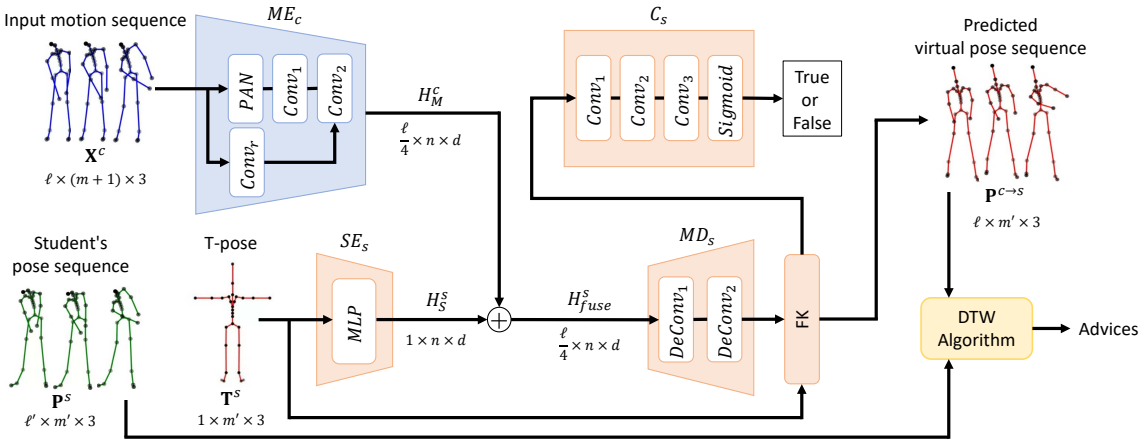

Figure 3. Our motion retargeting network consists of three main stages: (1) an extractor for skeleton features SE_s and an extractor for motion features ME_c; (2) the fusion of these features through addition (H_{fuse}^s); and (3) the generation of joint rotations using a decoder MD_s, followed by forward kinematics that transforms the joint rotations into a virtual pose sequence. During the training process, we implement a discriminator on C_s to assess the realism of the retargeted sequence from the coach. Finally, we compare the student's motion with the coach's retargeted motion using Dynamic Time Warping.

tractor (SE) and a motion feature extractor (ME):

$$H_S^s = SE_s(\mathbf{T}^s) \in \mathbb{R}^{1 \times n \times d} \tag{6}$$

$$H_M^c = ME_c(\mathbf{X}^c) \in \mathbb{R}^{\frac{\ell}{4} \times n \times d} \tag{7}$$

where d is the number of channels. The second stage fuses the above features by an addition: $H_{fuse}^s = H_S^s + H_M^c$. In the final stage, we feed H_{fuse}^s into a motion decoder (MD) and then apply the FK to generate the retargeted pose:

$$\mathbf{P}^{c \to s} = FK_s(\mathbf{T}^s, MD_s(H_{fuse}^s)) \tag{8}$$

Skeleton feature extractor (SE). Using the n body segments as the core components, both the Skeleton Encoder (SE) and the Motion Encoder (ME) extract features strictly within their respective segments. To avoid interference, a masking multi-layer perceptron (MLP) is implemented in the SE. In this masking process, we adjust the weight matrix $W \in \mathbb{R}^{d_{out} \times d_{in}}$, where d_{in} represents the size of the input channels and d_{out} denotes the size of the output channels. When d_{in} and d_{out} relate to different body segments, we set the relevant weights to zero. This allows us to create distinct skeleton features for each segment. Formally, the SE is formulated as $mlp_{out} = Relu(IW + b)$, where I represents the input matrix, and $W_{i,j} = 0$ if joints i and j pertain to different body segments. The output of the SE is denoted as $H_S^s \in \mathbb{R}^{1 \times n \times d}$.

Motion feature extractor (ME). For ME, the input sequence $\mathbf{X}^c$ will go through three modules. The first module is a Pose-aware Attention Network (PAN) for capturing the interrelationships among joints. The second module is masking 1D convolutions ($Conv_1$) for extracting temporal features of the joints. The third module is a layer of masking 1D convolutions ($Conv_2$) integrated with a residual connection ($Conv_r$) for maintaining the integrity of features. The result generated by ME is denoted as $H_M^c \in \mathbb{R}^{\frac{\ell}{4} \times n \times d}$.

Pose-aware attention network (PAN). Within the motion feature extractor (ME), there is a crucial component known as the pose-aware attention network (PAN), as depicted in Fig. 4. To begin, each joint in $\mathbf{X}^c$ is transformed into a higher-dimensional embedding space using an MLP. This transformation is enhanced by adding features derived from positional encoding, resulting in a feature representation $\dot{\mathbf{X}}^c \in \mathbb{R}^{\ell \times (m+1) \times d}$, where d denotes the channel size. The positional encoding enables the network to recognize the relationships between joints, maintaining awareness of the skeleton's structure. Subsequently, we utilize *body segment tokens* $O \in \mathbb{R}^{\ell \times n \times d}$ to encapsulate the motion features for each body segment. Specifically, we combine $\dot{\mathbf{X}}^c$ and O into $\ddot{\mathbf{X}}^c = O \oplus \dot{\mathbf{X}}^c$, $\ddot{\mathbf{X}}^c \in \mathbb{R}^{\ell \times (m+1+n) \times d}$. The tokens O are learnable parameters, with each initialized by sampling from a normal distribution. The random values are scaled down by multiplying by 0.1, similar to the PAN model [11]. Following this, we employ a self-attention mechanism to merge the features of the joints with their respective tokens in O. Specifically, we derive the query (Q), key (K), and value (V) matrices from $\ddot{\mathbf{X}}^c$. For effective motion retargeting across different skeleton structures, it's essential to prevent the exchange of information between different segments while calculating these attention values. Following [11], we incorporate a masking matrix U into this self-attention module and define the attention layer as follows:

$$Attention(Q, K, V, U) = softmax(\frac{QK^\top + U}{\sqrt{d_k}})V, \tag{9}$$

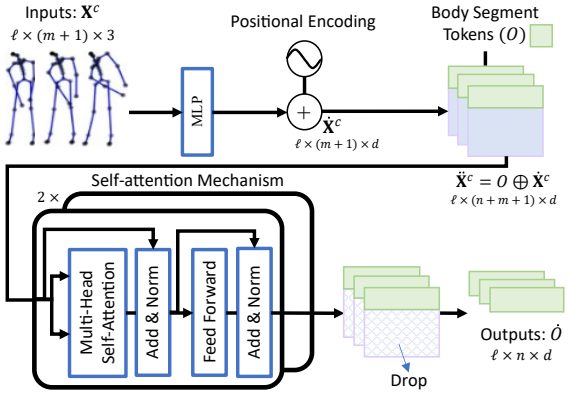

Figure 4. The pose-aware attention module processes input sequences with a multi-layer perceptron (MLP) that includes positional encoding, combined with tokens for body segments. Self-attention mechanisms aggregate joint features into these tokens, while joint features themselves are excluded to focus on body segment characteristics.

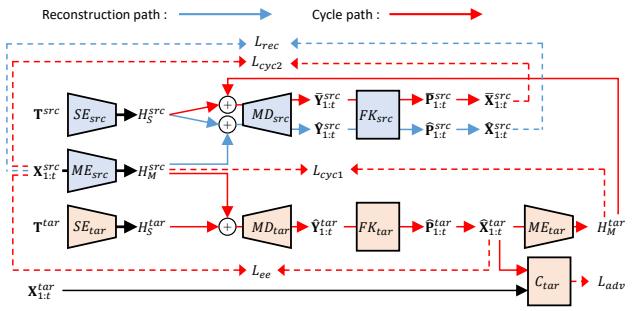

Figure 5. Motion retargeting architecture (training). Note that the two copies src and tar are different. The procedure is divided into a reconstruction path and a cycle path.

where $QK^\top$ represents the attention matrix, which indicates the correlation scores between different joint pairs. The value $U_{i,j}$ equals 0 if joints i and j are from the same body segment, and it equals $-\infty$ if they are not. This allows the softmax function to effectively suppress any attention values to 0 when $-\infty$ is present. The primary objective of the PAN is to learn motion features that are linked to specific body segments; therefore, we keep only the body segment tokens $\dot{O}$ in the outputs of this self-attention module.

Masking 1D Convolutions ($Conv_1$, $Conv_2$, and $Conv_r$). The tokens O from the previous PAN module are passed through two masking 1D conv to capture the motion's temporal dynamics. A masking matrix is created using body segments and joints; it starts as a zero matrix and updates specific elements to 1 based on the relationships between parts and joints. Additional zero columns are added, the diagonal is set to 1, and the last column is filled with 1 before truncating to a specified size. In each conv, the weight matrix $W \in \mathbb{R}^{d_{out} \times d_{in} \times l}$ is controlled through masking to maintain the independence of body segments, with d_{out} as the output channel size, d_{in} as the input channel size, and l as the kernel size. The output is $conv_{out} = Relu(IW + b)$, where I is input matrix, and $W_{d_{out},d_{in},l} = 0$ for different body segments. The diagonal structure preserves distinct features for each body part, ensuring anatomical consistency, and controlled joint influence. The stride of the masking 1D conv is set to 2, halving the temporal dimension after each layer. Finally, we obtain the independent motion feature $H_M^c \in \mathbb{R}^{\frac{\ell}{4} \times n \times d}$.

Feature fusion stage ($\oplus$). In the feature fusion stage, motion feature H_M^c and skeleton feature H_S^s are fused. Note that since the shape of H_M^c and H_S^s are dependent on n (the number of body segments), but not m and m', it is independent of skeleton structures. We duplicate the latter $\ell/4$ times and then perform element-wise addition. The fused feature is denoted by $H_{fuse}^c \in \mathbb{R}^{\frac{\ell}{4} \times n \times d}$.

Motion decoding stage (MD and FK). The fused feature H_{fuse}^s is fed into the motion decoder MD and then the FK procedure to generate the retargeted motion sequence. MD generates retargeted joint rotations and the global position of the root joint, denoted as $\mathbf{Y}^{c \to s} \in \mathbb{R}^{\ell \times ((m'-1) \times 4 + 3)}$, where m' is the number of joints in the target skeleton, with 4 for quaternions and 3 for the root's position. We make two remarks here. First, the output is joint rotations. However, in the loss function, positions will be used as a loss (refer to L_{rec}). Second, masking will be used for predicting joint rotations. However, the root's positions are treated as a global feature and no masking will be applied. Therefore, the deconvolution (deconv) consists of upsampling followed by a masking 1D conv. The upsampling uses a dilation factor of 2 and applies linear interpolation along the temporal dimension to generate a new tensor. The masking 1D conv is the same as the one in the ME, but with a stride of 1. As a result, after one layer of deconv, the temporal dimension doubles in length.

After acquiring joint rotations, the FK rotates each joint from the root along the kinematic chains to determine its position. Since the model outputs are in quaternion format, they are converted into rotation matrices before any transformations take place. The position of joint k is computed using the equation $p^k = p^{\text{par}(k)} + R^k s^k$, where $p^{\text{par}(k)}$ represents the parent joint's position, R^k denotes its rotation matrix, and s^k is the offset vector in $\mathbf{T}^s$ between the parent joint and joint k. By assembling these global positions, we can derive the retargeted motion sequence $\mathbf{P}^{c \to s}$ for $\mathbf{T}^s$.

Discriminator (C). The overall framework can be regarded as a adversarial network with unsupervised learning, where the modules (SE, ME, and MD) function as a generator. A discriminator C is created using standard 1D convolutions (without masking) to assess the generated motion sequence and ultimately improve the generator.

3.3. Network Training

We adopt an unsupervised method to train our motion retargeting model because the desired retargeted motions are often nonexistent. As illustrated in Fig. 5, the training process comprises both a reconstruction path and a cycle path, during which we update the network parameters utilizing four different loss functions.

In the motion reconstruction process, the source motion sequence $\mathbf{X}_{1:t}^{src}$, which has a duration of t, is first transformed into a motion feature H_M^{src} by ME_{src}. After this transformation, H_M^{src} is combined with the source skeleton feature H_S^{src}. This unified representation is then passed to a motion decoder MD_{src} and a forward kinematic function FK_{src} to generate the reconstructed source motion $\hat{\mathbf{P}}_{1:t}^{src}$. To evaluate the accuracy of the reconstruction, we quantify the difference between the original motion sequence $\mathbf{X}_{1:t}^{src}$ and the reconstructed motion $\hat{\mathbf{P}}_{1:t}^{src}$. Specifically, the reconstruction loss L_{rec} is defined as follows:

$$\begin{aligned} L_{rec} = ||\mathbf{X}_{1:t}^{src} - \hat{\mathbf{X}}_{1:t}^{src}||^2 + 100||r_{1:t}^{src} - \hat{r}_{1:t}^{src}||^2/h_{src}^2 \\ + 200||u_{1:t}^{src} - \hat{u}_{1:t}^{src}||^2/h_{src}^2, \end{aligned} \tag{10}$$

where r denotes the local joint positions in relation to the root joint within a normalized space, while u refers to the trajectory of the root joint in the world coordinate system. The loss function is composed of three components. The first component measures the positional errors of each joint concerning its parent joint, whereas the second and third components assess the errors relative to the root joint. Moreover, we adjust the second and third components based on the character's height to minimize the effect of the character's size. The coefficients 100 and 200 are chosen based on empirical observations of their value ranges.

In the cycle path, the source trajectory $\mathbf{X}_{1:t}^{src}$ is transformed into the source feature H_M^{src} and combined with the target skeleton feature H_S^{tar}. This combined result is then processed by MD_{tar} and FK_{tar} to produce the target motion $\hat{\mathbf{P}}_{1:t}^{tar}$. Given the absence of ground truth for $\hat{\mathbf{P}}_{1:t}^{tar}$, we normalize it into $\hat{\mathbf{X}}_{1:t}^{tar}$ and input this into the discriminator C_{tar} to evaluate its authenticity. The discriminator C_{tar} is specifically trained to recognize the real input motions $\mathbf{X}_{1:t}^{tar}$ while categorizing the retargeted motions $\hat{\mathbf{X}}_{1:t}^{tar}$ as fake. The adversarial loss is constructed as follows:

$$L_{adv} = \left\|1 - C_{tar}(\mathbf{X}_{1:t}^{tar})\right\|^2 + \left\|C_{tar}(\hat{\mathbf{X}}_{1:t}^{tar})\right\|^2 \tag{11}$$

Although the source and target skeletons may be defined differently, their end-effectors remain consistent since these skeletons are all based on a humanoid form. We thus define an end-effector loss L_{ee} to ensure the consistency of end-effector velocities between $\mathbf{X}_{1:t}^{src}$ and $\hat{\mathbf{X}}_{1:t}^{tar}$:

$$L_{ee} = \sum_{i=1}^{E} \left\| \frac{V_{src}^i}{h_{src}} - \frac{\hat{V}_{tar}^i}{h_{tar}} \right\|^2, \tag{12}$$

where $E = 5$ represents the number of end-effector joints, V^i represents the movement speed of the ith end-effector (left hand, left foot, right hand, right foot, and head), and h represents height.

In this cycle path, we also retarget the target motions back to the source. Initially, $\hat{\mathbf{P}}_{1:t}^{tar}$ is translated into $\hat{\mathbf{X}}_{1:t}^{tar}$, which is then converted into feature H_M^{tar}. The result is added with H_S^{src} and then passed into MD_{src} and FK_{src} to synthesize the source motion, denoted as $\bar{\mathbf{P}}_{1:t}^{src}$, which is then normalized to $\bar{\mathbf{X}}_{1:t}^{src}$. We aim to encourage that the retargeted motion $\bar{\mathbf{X}}_{1:t}^{src}$ remains consistent with the original input motion $\mathbf{X}_{1:t}^{src}$. The cycle consistency loss can be formulated as:

$$L_{cyc1} = \|r_{1:t}^{src} - \bar{r}_{1:t}^{src}\|^2 / h_{src}^2, \tag{13}$$

where r represents the local joint positions relative to the $root$ positions. Furthermore, to achieve motion retargeting across different structures, we anticipate that the learned motion features can act as shared features among diverse skeleton structures. Since the target motion $\hat{\mathbf{X}}_{1:t}^{tar}$ should perform the same action as the source motion $\mathbf{X}_{1:t}^{src}$, we require that their corresponding motions, H_M^{src} and H_M^{tar}, be highly identical when being converted into features.

$$L_{cyc2} = \left\| H_M^{src} - H_M^{tar} \right\|_1 \tag{14}$$

Let $L_{cyc} = L_{cyc1} + L_{cyc2}$. The overall loss function is defined as follows:

$$L = \lambda_{rec} L_{rec} + \lambda_{cyc} L_{cyc} + \lambda_{ee} L_{ee} + \lambda_{adv} L_{adv}, \tag{15}$$

where $\lambda_{rec} = 1$, $\lambda_{cyc} = 2.5$, $\lambda_{ee} = 50$ and $\lambda_{adv} = 1$ are weighting factors.

3.4. Pose Scoring and Visualization

Once we adjust the coach's pose sequence for the student, we calculate a performance score by considering both the spatial and temporal aspects. To assess the differences between the student's pose $\mathbf{P}^s$ and the adjusted coach's pose sequence $\mathbf{P}^{c\to s}$, we utilize the Dynamic Time Warping (DTW) algorithm [4]. This algorithm helps align the two sequences by minimizing the joint position errors, with the root joint fixed at the origin. As illustrated in Fig. 6, the DTW algorithm successfully matches the poses in both sequences and generates an optimal alignment path, indicated as $DTW(\mathbf{P}^s, \mathbf{P}^{c\to s})$. The summed distance along this path is used as the performance score. By following the optimal alignment path, we can correlate each frame of the student's pose sequence with the corresponding frame from the adjusted virtual pose sequence.

3.5. Case Study

Fig. 6 shows a retargeting example of our coaching system by a golf putting motion. There are four types of advices that our system can offer to a student: (i) pose alignment advice: Using DTW to analyze the correspondence of

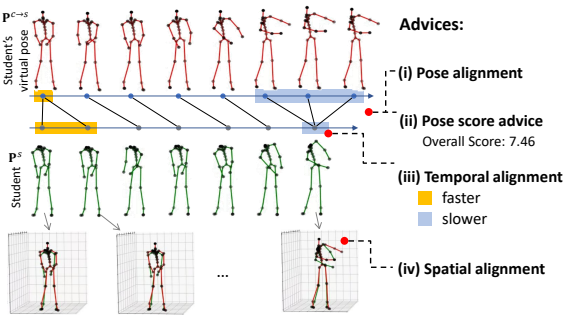

Figure 6. This example demonstrates virtual pose coaching for golf putting. Red skeletons show the coach's retargeted motion, while green skeletons represent the student's motion. Using DTW, the system aligns these two movements to provide four types of feedback to students.

poses between a student's original sequence and the virtual sequence on the timeline, helping the student understand when to display the corresponding pose. (ii) pose score advice: The joint position error is used to score student's pose, with lower scores indicating closer similarity between the student's original pose and virtual pose. This score s serves as a quantitative indicator of sport performance. (iii) temporal alignment advice: The execution rate of the student's motions is compared with the student's virtual motions. If certain actions are too fast or too slow, the system provides specific timing adjustment suggestions to help the student achieve the correct rhythm. (iv) spatial alignment advice: By overlaying the student's skeleton onto the student's virtual skeleton, the student can directly see the spatial differences and correct their poses accordingly.

4. Evaluations

4.1. Setups and Quality Assessments

We used a 3D motion dataset from Mixamo [13], created by SAN [2], featuring 29 unique humanoid characters performing over 2,000 distinct motion sequences. We followed [2] for training and testing splits, dividing the characters into two groups by skeletal structure. Group A had 24 characters, with 20 for training, while Group B had 5 characters, 4 of which were also for training. Group A includes six more joints than Group B, located in the limbs, torso, and head. Each motion sequence was randomly assigned to a character to prevent duplication. We conducted intra-structural experiments with four characters from Group A serving as both source and target, and cross-structural experiments with four Group A characters as targets and one Group B character as the source.

We assess the joint position errors to evaluate the quality of the retargeting results. Given the variations in body structure among characters, we adjust this error based on each character's height as follows:

$$e = \frac{1}{I|N_s||N_t|K} \sum_{i=1}^{I} \sum^{N_s} \sum^{N_t} \sum_{k=1}^{K} ||\hat{\mathbf{P}}_{i,k}^{tar} - \mathbf{P}_{i,k}^{gt}||^2 / h_{i,k}^2. \tag{16}$$

Here, I represents the total number of motion sequences, N_s indicates the number of source characters, and N_t signifies the number of target characters. The symbol $\hat{\mathbf{P}}_{i,k}^{tar}$ denotes the predicted position of joint k in the target skeleton while $\mathbf{P}_{i,k}^{gt}$ refers to the corresponding ground truth position. To minimize the influence of body size, we divide the error by the character's height $h_{i,k}$.

Table 1. We compare baseline methods on the Mixamo dataset, reporting joint position errors across all motions and clips. (**Bold** indicates the best, underline the second, and *italics* the third.)

Method		Intra-Stru	Cross-Stru
baseline	Direct-Copy	8.86 e-3	-
position-based	NKN/NKN* [21]	5.84 e-3	7.36 e-3
	PMnet/PMnet* [17]	4.93 e-3	6.88 e-3
	Ours	1.76 e-3	*6.77 e-3*
rotation-based	SAN [2]	*2.76 e-3*	2.25 e-3
	PAN [11]	**0.5 e-3**	**1.62 e-3**

4.2. Comparisons to Existing Models

We evaluated our models against several frameworks: NKN [21], PMnet [17], SAN [2], and PAN [11], as well as a baseline method called "direct-copy," which applies joint rotations directly to the target character. To reduce the impact of skeleton differences, we adjusted the root movement speed based on the height ratios of the characters.

Tab. 1 shows our experimental results. NKN and PMnet use joint positions as input, while SAN and PAN use joint rotations. To enable cross-structural comparison, the modified NKN and PMnet methods, based on [11], are referred to as NKN* and PMnet*. Our approach excels in intra- and cross-structural evaluations among position-based methods. Compared to rotation-based methods, it surpasses SAN in intra-structural evaluations but does not perform as well in cross-structural evaluations. It's important to note that joint rotations during inference may differ from those in training, leading to unexpected outcomes, as shown in Fig. 2. This issue may seem minor due to the similarity of joint rotations between the training and test sets of the Mixamo. We also provide visual examples to compare our model's results with those of the PAN and the ground truth, as shown in Fig. 2 of the supplementary material. Although our method may not completely outperform rotation-based methods, it can still be effectively applied to position-based scenarios, as most real-world datasets, such as sports-related datasets like [14, 24], are recorded in a position-based format.

Table 2. Ablation study on input motion formats, reconstruction loss, and cycle consistency loss.

(a) global root position errors ($\times 10^{-3}$).

	Intra-structural			Cross-structural		
Method	G↔Mo	Mr↔V	Overall	B→G	B→V	Overall
parent-based	0.245	0.282	0.412	**7.423**	**9.233**	**6.645**
root-based	0.315	0.592	0.604	59.672	78.224	53.750
use L'_{rec}	**0.213**	**0.119**	**0.307**	120.349	155.341	107.288
use L'_{cyc}	0.219	0.147	0.329	72.061	94.448	65.190

(b) local joint position errors ($\times 10^{-3}$).

	Intra-structural			Cross-structural		
Method	G↔Mo	Mr↔V	Overall	B→G	B→V	Overall
parent-based	**1.737**	1.375	**1.614**	**1.629**	**1.431**	**1.380**
root-based	1.915	1.584	1.763	2.641	2.339	1.794
use L'_{rec}	1.953	**1.364**	1.718	3.730	4.014	3.323
use L'_{cyc}	1.739	1.431	1.636	2.568	2.073	1.644

4.3. Ablation Study

We conducted an ablation study examining factors like input motion formats, reconstruction loss, and cycle consistency loss. We report both global and local joint position errors. Global errors, assessed in the world coordinate system, can be substantial if two visually similar poses have different root positions. In contrast, the local method resets the root position to the origin, focusing on pose similarity alone. This study involves five characters: Mousey (Mo), Goblin (G), Mremireh (Mr), Vampire (V), and BigVegas (B). In intra-structural retargeting, the process is bidirectional ($\leftrightarrow$), while in cross-structural retargeting, it is unidirectional ($\rightarrow$), as shown in Tab. 2. All results are located in Tab. 2 and Tab. 1 in Sec. 2.3 of the supplementary material.

Joint Position Representations. Previous studies [17, 21] defined a joint's position relative to the root joint, which implied connections to parent and child joints rather than making them explicit. This ambiguity can hinder accurate inferences. In our research, we define a joint's position relative to its parent joint, labeling these approaches as "root-based" and "parent-based." Results in Tab. 2 show that the parent-based representation outperforms the root-based approach in both intra- and cross-structural retargeting scenarios, demonstrating the effectiveness of our design.

Design of reconstruction loss. In the reconstruction loss L_{rec}, both $\mathbf{X}$ and r provide local position information, differing only in their reference points: $\mathbf{X}$ is relative to the parent joint, and r is relative to the root joint. To eliminate redundancy, we remove the $\mathbf{X}$ component from L_{rec} and redefine the loss as

$$L'_{rec} = 100||r^{src}_{1:t} - \hat{r}^{src}_{1:t}||^2/h^2_{src} + 200||u^{src}_{1:t} - \hat{u}^{src}_{1:t}||^2/h^2_{src}. \quad (17)$$

Compared to L_{rec}, L'_{rec} lacks $||\mathbf{X}^{src}_{1:t} - \hat{\mathbf{X}}^{src}_{1:t}||^2$. The results are labeled as "use L'_{rec}" in Tab. 2. For intra-structural retargeting, the source and target skeletons share the same structure but vary in bone proportions, making our parent-based design less critical. This leads to mingled global and local joint position errors, as seen in Tab. 2. However, in cross-structural retargeting, our parent-based method clearly outperforms the L'_{rec} approach, as indicated in Tab. 2. This emphasizes the value of including the term $||\mathbf{X}^{src}_{1:t} - \hat{\mathbf{X}}^{src}_{1:t}||^2$ for accurate motion reconstruction and preservation of key motion features. In summary, supervising with both $\mathbf{X}$ and r enhances our model. The relative position to the parent joint tracks fine details, while the position to the root joint ensures overall motion coherence, enabling the generation of precise and cohesive actions.

Design of Cycle Consistency Loss. The cycle consistency loss L_{cyc} ensures that the retargeted motion from the target character to the source character is consistent with the original input motion. We compare our cycle consistency loss L_{cyc} with that of PAN:

$$L'_{cyc} = ||H^{src}_M - H^{tar}_M||_1 + ||r^{src}_{1:t} - \bar{r}^{src}_{1:t}||^2/h^2_{src} + ||u^{src}_{1:t} - \bar{u}^{src}_{1:t}||^2/h^2_{src}. \quad (18)$$

Compared to L_{cyc}, L'_{cyc} adds an additional term $||u^{src}_{1:t} - \bar{u}^{src}_{1:t}||^2/h^2_{src}$, referenced as "use L'_{cyc}" in Tab. 2. In intra-structural retargeting, using L'_{cyc} slightly outperforms our parent-based method in global joint position error but falls slightly short in local joint position error, making the overall impact unclear. However, for cross-structural retargeting tasks, our design of L_{cyc} shows clear advantages. When dealing with different source and target structures, enforcing root position consistency may force the model into unnatural positions, overlooking target features. Thus, removing the additional term in L'_{cyc} benefits our model.

5. Conclusions

In this study, we tackled the difficulties associated with motion retargeting caused by the numerous joint rotations between two poses. Although methods using joint rotations as inputs generally produce higher quality results than those that depend on joint positions, this rotation-based representation can occasionally result in failures. To address this, we have restructured existing rotation-based models to work with joint positions instead. Experiments conducted using the Mixamo dataset demonstrate promising outcomes for both intra- and cross-structural retargeting.

Our case study primarily focused on golf, but the framework we developed can be applied to various other sports as well. Furthermore, our motion retargeting model takes joint positions as input, making it ideal for digital twin applications. This allows for realistic interactions between real people and virtual characters in gaming environments.

References

[1] Kfir Aberman, Rundi Wu, Dani Lischinski, Baoquan Chen, and Daniel Cohen-Or. Learning Character-Agnostic Motion for Motion Retargeting in 2D. *ACM Transactions on Graphics (TOG)*, 38(4):1–14, 2019. 2

[2] Kfir Aberman, Peizhuo Li, Dani Lischinski, Olga Sorkine-Hornung, Daniel Cohen-Or, and Baoquan Chen. Skeleton-Aware Networks for Deep Motion Retargeting. *ACM Transactions on Graphics (TOG)*, 39(4):62–1, 2020. 1, 2, 7

[3] Barbara Rita Barricelli, Elena Casiraghi, Jessica Gliozzo, Alessandro Petrini, and Stefano Valtolina. Human Digital Twin for Fitness Management. *IEEE Access*, 8:26637–26664, 2020. 1

[4] Donald J Berndt and James Clifford. Using Dynamic Time Warping to Find Patterns in Time Series. In *Proc. International Conference on Knowledge Discovery and Data Mining*, pages 359–370, 1994. 6

[5] Bhat Dittakavi, Divyagna Bavikadi, Sai Vikas Desai, Soumi Chakraborty, Nishant Reddy, Vineeth N Balasubramanian, Bharathi Callepalli, and Ayon Sharma. Pose Tutor: An Explainable System for Pose Correction in the Wild. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3540–3549, 2022. 1, 2

[6] Don Samitha Elvitigala, Denys JC Matthies, Löic David, Chamod Weerasinghe, and Suranga Nanayakkara. GymSoles: Improving Squats and Dead-lifts by Visualizing the User's Center of Pressure. In *Proc. CHI Conference on Human Factors in Computing Systems*, pages 1–12, 2019. 1, 2

[7] Mihai Fieraru, Mihai Zanfir, Silviu Cristian Pirlea, Vlad Olaru, and Cristian Sminchisescu. AIFit: Automatic 3D Human-Interpretable Feedback Models for Fitness Training. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9919–9928, 2021. 1, 2

[8] Michael Gleicher. Retargetting Motion to New Characters. In *Proc. the 25th annual conference on Computer graphics and interactive techniques*, pages 33–42, 1998. 1, 2

[9] Shoichi Hasegawa, Seiichiro Ishijima, Fumihiro Kato, Hironori Mitake, and Makoto Sato. Realtime Sonification of the Center of Gravity for Skiing. In *Proc. Augmented Human International Conference*, pages 1–4, 2012. 1, 2

[10] Jana Hoffard, Takuto Nakamura, Erwin Wu, and Hideki Koike. PushToSki - An Indoor Ski Training System Using Haptic Feedback. In *Proc. ACM SIGGRAPH Posters*, pages 1–2, 2021. 1, 2

[11] Lei Hu, Zihao Zhang, Chongyang Zhong, Boyuan Jiang, and Shihong Xia. Pose-aware Attention Network for Flexible Motion Retargeting by Body Part. *IEEE Transactions on Visualization and Computer Graphics*, 30(8):4792–4808, 2024. 1, 2, 3, 4, 7

[12] Atsuki Ikeda, Yuka Tanaka, Dong-Hyun Hwang, Homare Kon, and Hideki Koike. Golf Training System using Sonification and Virtual Shadow. In *Proc. ACM SIGGRAPH 2019 Emerging Technologies*, pages 1–2, 2019. 1, 2

[13] Adobe Systems Inc. Mixamo, 2018. Accessed: 2024-02-20. 7, 1

[14] Christian Keilstrup Ingwersen, Christian Mikkelstrup, Janus Nørtoft Jensen, Morten Rieger Hannemose, and Anders Bjorholm Dahl. Sportspose: A dynamic 3d sports pose dataset. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2023. 3, 7

[15] Jehee Lee and Sung Yong Shin. A Hierarchical Approach to Interactive Motion Editing for Human-like Figures. In *Proc. the 26th annual conference on Computer graphics and interactive techniques*, pages 39–48, 1999. 2

[16] Chen-Chieh Liao, Dong-Hyun Hwang, Erwin Wu, and Hideki Koike. AI Coach: A Motor Skill Training System using Motion Discrepancy Detection. In *Proc. Augmented Humans International Conference*, pages 179–189, 2023. 1, 2

[17] Jongin Lim, Hyung Jin Chang, and Jin Young Choi. PMnet: Learning of Disentangled Pose and Movement for Unsupervised Motion Retargeting. In *Proc. British Machine Vision Conference (BMVC)*, page 7, 2019. 1, 2, 7, 8

[18] Tica Lin, Rishi Singh, Yalong Yang, Carolina Nobre, Johanna Beyer, Maurice A Smith, and Hanspeter Pfister. Towards An Understanding of Situated AR Visualization for Basketball Free-throw Training. In *Proc. CHI Conference on Human Factors in Computing Systems*, pages 1–13, 2021. 1, 2

[19] Jingyuan Liu, Mingyi Shi, Qifeng Chen, Hongbo Fu, and Chiew-Lan Tai. Normalized Human Pose Features for Human Action Video Alignment. In *Proc. IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11521–11531, 2021. 1, 3

[20] Luka Lukač, Iztok Fister Jr, and Iztok Fister. Digital Twin in Sport: From an Idea to Realization. *Applied Sciences*, 12 (24):12741, 2022. 1

[21] Ruben Villegas, Jimei Yang, Duygu Ceylan, and Honglak Lee. Neural Kinematic Networks for Unsupervised Motion Retargetting. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8639–8648, 2018. 1, 2, 7, 8

[22] Jianbo Wang, Kai Qiu, Houwen Peng, Jianlong Fu, and Jianke Zhu. AI Coach: Deep Human Pose Estimation and Analysis for Personalized Athletic Training Assistance. In *Proc. ACM international conference on multimedia*, pages 374–382, 2019. 1, 2

[23] Mikołaj P Woźniak, Julia Dominiak, Michał Pieprzowski, Piotr Ładoński, Krzysztof Grudzień, Lars Lischke, Andrzej Romanowski, and Paweł W Woźniak. Subtletee: Augmenting Posture Awareness for Beginner Golfers. *Proc. ACM on Human-Computer Interaction*, 4(ISS):1–24, 2020. 1, 2

[24] Calvin Yeung, Tomohiro Suzuki, Ryota Tanaka, Zhuoer Yin, and Keisuke Fujii. Athletepose3d: A benchmark dataset for 3d human pose estimation and kinematic validation in athletic movements, 2025. 3, 7

[25] Wentao Zhu, Zhuoqian Yang, Ziang Di, Wayne Wu, Yizhou Wang, and Chen Change Loy. MoCaNet: Motion Retargeting in-the-wild via Canonicalization Networks. In *Proc. AAAI Conference on Artificial Intelligence*, pages 3617–3625, 2022. 2