Skeleton-Aware Motion Retargeting Using Masked Pose Modeling

Giulia Martinelli, Nicola Garau, Niccoló Bisagno, and Nicola Conci

Department of Information Engineering and Computer Science - DISI
University of Trento (Italy)
{giulia.martinelli-2,nicola.garau,niccolo.bisagno,nicola.conci}@unitn.it

Abstract. Motion retargeting aims at transferring a given motion from a source character to a target one. The task becomes increasingly challenging as the differences between the body shape and skeletal structure of input and target characters increase. We present a novel approach for motion retargeting between skeletons whose goal is to transfer the motion from a source skeleton to a target one in a different format. Our approach works when the two skeletons differ in scale, bone length, and number of joints. Surpassing previous approaches, our method can also retarget between skeletons that differ in hierarchy and topology, such as retargeting between animals and humans. We train our method as a transformer using a random masking strategy both in time and space, aiming at reconstructing the joints of the masked input skeleton to obtain a deep representation of the motion. At testing time, our proposal can retarget the input motion to different skeletons, reconstructing the disparities between the source and the target. Our method outperforms state-of-the-art results on the Mixamo dataset, which features a high variance between skeleton formats. Moreover, we show how our approach can effectively generalize to different domains by transferring between human motion and quadrupeds, and vice-versa. Our code is available at www.github.com/mmlab-cv/skeleton-aware-motion-retargeting.

1 Introduction

Retargeting the motion from an input subject to a target one can be applied to skeletons [1, 15, 23], meshes [13, 22] or directly on videos [6, 26]. Regardless of the chosen medium, the task becomes more difficult as the differences between source and target grow, and as the complexity of the animation to be retargeted increases.

In this work, we tackle the task of motion retargeting between 3D skeletons, which can be employed for multiple applications in motion analysis and computer animation, such as transferring the motion from a real-world character to a synthetic one. Such techniques would allow us to accurately animate digital representations in the metaverse, providing a personal set of animation patterns for each character and thus improving the realism and quality of the synthetic

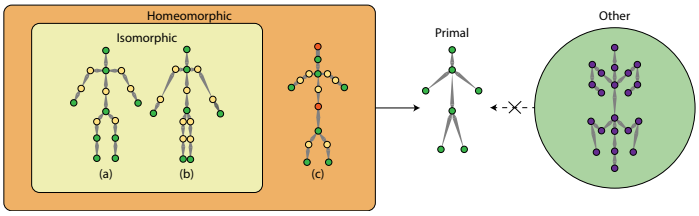

Fig. 1: Examples of isomorphic skeletons are (a) and (b), which have the same joints but varying bone lengths. Both (a) and (b) are homeomorphic to the skeleton (c), which has a different number of joints. They are defined as homeomorphic because they can all be reduced to a common primal skeleton (d). The skeleton of a quadruped (e) is neither homeomorphic nor isomorphic to the others because the topology is changed and thus (e) cannot be reduced to the common primal skeleton.

environment. In the case of motion retargeting between skeletons, source, and target can differ not only in scale and bone length but also in topology and number of joints.

Skeletons are defined as a graph with vertices and edges represented by bones and joints, with varying topologies. Similarly to graphs, and as shown in Fig. 1, two skeletons can be defined as *isomorphic* when they share the same number of joints (vertices) and bones (edges). If two skeletons are not isomorphic, they can be defined as *homeomorphic*, or topologically equivalent, if they can be reduced to a common primal skeleton [1], by removing the proper number of joints (vertices). While the approaches in the literature tend to confine the retargeting within homeomorphic and isomorphic skeletons [22, 23], in our work we deal with non-homeomorphic skeletons, such as retargeting between humans and quadrupeds that differ both in walking pattern and topology (quadrupeds' skeleton also include the tail).

The inability of the existing methods to deal with non-homeomorphic skeletons can be traced back to the use of graph-based algorithms [22, 23]. These algorithms usually represent the motion as spatial-temporal graphs and rely on complex encoder-decoder structures [13, 26], skeleton pooling and convolution operations [22], or GAN-based approaches [6]. Instead of representing the motion transfer between different skeletons using graphs, we represent each joint as a quaternion that describes the joint's rotation relative to the parent in the hierarchy. Each quaternion is then treated as an individual chunk of data to be processed by our architecture. Since the data chunks are both spatially related (they are joints belonging to the same skeleton) and temporally related (consecutive skeletons in time represent an animation), the task of our method is to model and learn this spatiotemporal relationship.

In computer vision, deep neural networks have exploited the continuous spatial and temporal relationship between pixels of videos and images to improve results on a variety of tasks [9]. However, convolutions and pooling operations do not achieve as good results when dealing with non-continuous data like skeletons, because they do not adequately model the sparsity of the information.

Differently, transformers [9] employ a token-based approach and rely on the positional encoder to model the spatial and hierarchical information between image tokens. When working with videos [10, 20], the positional encoding can be effectively extended to the temporal dimension. In this work, we show how the transformer architecture is more suited to deal with the 3D spatially distributed and hierarchically organized nature of pose skeleton data. A single simple transformer network can obtain state-of-the-art results without relying on complex encoder-decoder structures, graph convolutional networks, or GANs.

More recently, masked image modeling [12, 24] demonstrated how transformers can learn data representations when trained as auto-encoders to reconstruct the masked portion of the input data. When dealing with videos [10, 20], the masked portion of the video data can increase up to 90% thanks to both spatial and temporal correlations. Following the same principle, we apply the reconstruction of masked data to skeleton motion modeling. At training time, we randomly mask skeleton joints in space and time, to obtain a rich representation of the input data. At test time, when retargeting a motion from one source topology to a target one, source and target skeletons usually share several common joints (e.g. knees, elbows, wrists). Thus, we selectively mask the missing joints in the target format that need to be retrieved.

Our proposed solution consists of a transformer-based [9, 21] architecture, where the encoding models the temporal and spatial relationships between joints, thus allowing for the retargeting of the motion of the skeletons between arbitrary topologies.

The novelties of our work can be summarised as follows:

- we propose a novel self-supervised pipeline to retarget the motion sequence between skeletons;
- we deal with motion retargeting between non-homeomorphic skeletons without paired motion;
- we apply the masking modeling strategy to spatial-temporal skeleton motion, training our architecture to reconstruct random joints of the input image after masking. At test time, the masking is specifically applied depending on the differences between the source and target topology;
- we obtain state-of-the-art results on motion retargeting on the Mixamo dataset [4];
- we retarget the motion between human and quadruped providing results on the quadruped dataset [27].

2 Related work

Human motion retargeting. Skeleton-based motion retargeting aims at transferring the motion from a skeleton topology to a different one [1, 15, 23]. The skeleton can vary in bone length, scale, and the number of joints.

Skeleton representations have been used to aid both video [6, 26] and mesh motion retargeting [13, 22].

Video motion retargeting [6, 26] aims at transferring a desired input motion to an input video of a person, giving as output a video of the same person performing the alternative input motion. These methods usually rely on separate encoding of video and motion parameters, which are combined to produce the output. To obtain these representations they usually rely on encoding-decoding structures with multiple loss functions.

When dealing with meshes and skeletons, [13] focuses on compact SMPL models, while [22] focuses on avoiding mesh interpenetration while doing retargeting of 3D motion, by detecting contacts between different parts of the mesh, as well as foot-ground contact detection.

Since both video and mesh approaches strongly rely on the skeleton before obtaining good results, we focus on refining the skeleton prediction in challenging scenarios, which can benefit both tasks.

Retargeting between isomorphic skeletons. A basic approach to performing motion retargeting between isomorphic skeletons is the so-called *copy rotation*; starting from a common pose (usually the T-pose), the rotations from one skeleton are copied to another one, without adapting to changes in scale or bone length, as described in [1].

Other works [3, 23, 29] rely on recurrent neural networks paired with forward kinematics to disentangle the motion from a given character to be transferred to a different one. The authors in [2] use separate encoders to encode motion from videos and feed them to a generator; the generator extracts the relevant motion components, with the same skeleton topology as the input one. On a similar note, GAN-based approaches synthesize novel motion sequences [15] using multi-scale hierarchical generative networks to produce new animations starting from a single motion sequence over multiple scales and time. These methods usually rely on a single input animation, and thus on a single isomorphic skeleton topology. Recent methods ([22, 28]) incorporate mesh data in the retargeting process to address and solve issues like mesh interpenetration. In [28] introduces a skeleton-aware and a shape-aware module. The shape-aware module is trained using an attractive/repulsive field mechanism to resolve collisions while maintaining adherence to the target motion. Despite these advancements and their ability to generalize to various body shapes, our approach achieves better results without relying on a shape-aware module to penalize mesh self-interpenetration.

Retargeting between homeomorphic skeletons. The algorithm presented by Aberman et al. in [1] can retarget the motion between homeomorphic and isomorphic skeletons using an encoder-decoder structure. It relies on a graph representation for skeletons since this data structure can be easily represented in terms of vertices and edges, achieving state-of-the-art results. However, being graph-based, the solution can not generalize to non-homeomorphic skeletons with different topologies, namely where the number of leaf joints changes. Dealing with non-homeomorphic skeletons is a research area that has been only partially investigated. Several works [19, 25] have tried to transfer the motion from humans to non-humanoid characters with non-homeomorphic skeletons. However, the aforementioned methods require either paired motions [25], mean-

ing a sample of the same motion performed by both the input and the target, or an explicit skeleton mapping [19] to be able to perform the motion transfer task.

To the best of our knowledge, we are the first to deal with skeleton-based motion retargeting between non-homeomorphic skeletons without paired motion.

Masked modeling for representation learning. Masked language modeling [8, 16] and masked image modeling [12, 24] have recently demonstrated how self-supervised approaches can achieve better performances than fully supervised ones, making them scalable representation learners for multiple tasks. In the video domain, where the data is not only spatially but also temporally correlated [10, 20], the relationship is strong enough to allow the network to reconstruct the input with up to 90% of the image masked. Transformers architectures [21] have further contributed to improving representation learning when combined with a masking strategy on randomly selected image parts [5, 7, 9]. Among others, SimMIM [24] has proven to be a simple yet effective strategy, masking image patches by replacing them with random token vectors of the same dimension. In [24], the goal is to train a network to obtain a strong image representation for the prediction task. We apply a similar strategy by randomly masking a subset of skeleton joints in space and time. We demonstrate the effectiveness of the self-supervised learned skeleton representation by using it in the motion retargeting task.

3 Method

Our goal is to train a neural network capable of transferring an input motion A from a source skeleton to a target skeleton with different topologies. The key idea is to use a masking strategy alongside a transformer network to reconstruct the masked portions of the input motion. By masking random parts of skeletons with varying numbers of joints and proportions, the network learns to reconstruct different sections, adapting to their different topologies and developing an awareness of the skeleton structures.

The mathematical notation for the motion representation and the detailed implementation of each step are further explained in the next paragraphs.

Motion representation. We represent the motion as a sequence of *armatures* (Fig. 2). Each armature is characterized by a dynamic part and a static part. The dynamic representation Q_K consists of N joints (nodes) and each joint j_n is expressed as a 4D quaternion, to describe the relative rotation with respect to the parent joint in the skeletal hierarchy. While the static representation S_k does not vary for a character during an animation, the dynamic representation Q_k is subject to changes to obtain the animation.

3.1 Skeleton-aware pose masking auto-encoder

In the following paragraphs we describe in detail the main components of our transformer-based auto-encoder architecture, illustrated in Fig. 3.

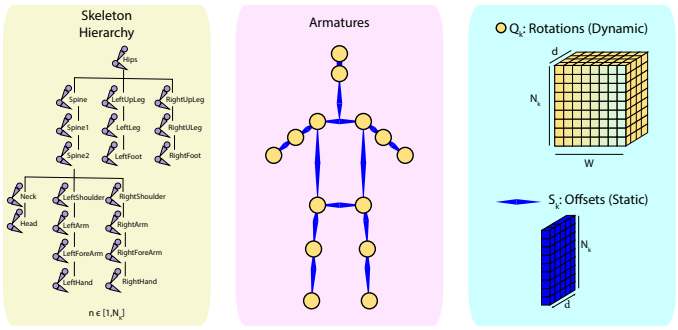

Fig. 2: Motion representation. Each skeleton can be represented as a hierarchical structure, where the parent-child relationship is defined by the kinematic chain. Moreover, each skeleton is described by a static representation S_k containing the offsets (bone lengths), and a dynamic representation Q_k. The length of the animation is a window of size $w \in [1, W]$ frames.

Objective. The motion of a generic character k is defined as $A((S_k, Q_k))$. The skeletal-only motion $A(S_k, Q_k)$ can be divided into the skeletal dynamic motion $A(Q_k)$ and the static representation, which is a learnable token S_k and does not vary across the motion. Given a skeletal motion $A(S_k, Q_k)$, our pose masking auto-encoder performs the retargeting to obtain the same motion $A(S_t, Q_t)$ in a target skeleton format S_T.

Tokenizer. Transformers process chunks of data (tokens) as it happens for words in NLP [21] or image patches in computer vision [9] in their multi-headed architecture. Since we are dealing with human motions, we define individual joint tokens to be embedded. To obtain a descriptor suitable for the motion retargeting task, we represent the joints' position in the 3D space as local rotations. Starting from the root joint (pelvis), each joint is expressed as a 4D quaternion $Q_k(j_n)$ which describes the rotation with respect to the parent joint in the hierarchy. The embedding $E_k(Q_k(j_n))$ is applied to each quaternion using a simple trainable linear layer followed by an activation function φ to obtain a vector of size d for each joint. Thus, each embedded motion A sequence can be expressed as $W \times N \times E_k(Q_k(j_n))$ with dimensions $W \times N \times d$.

Pose masking strategy. Similar to the techniques used in image-based masking methods ([10, 20]), our objective is to train an auto-encoder by using masked data as input and reconstructing the absent segments. This approach involves applying a masking strategy to the data, training the auto-encoder to take in incomplete data, and reconstructing the missing portions as output with the information of the unmasked part. Following the tokenization of the input data, we mask a subset of joints M from the embedded animation E_k, which has dimensions $W \times N_k \times d$, resulting in the masked embedded animation. Subsequently, we concatenate $N_C - N_k$ empty, learnable tokens to the masked embedded animation to form E_C^M. This process ensures that the latent representation E_C^M, now of size $W \times N_C \times d$, is sufficiently large to accommodate

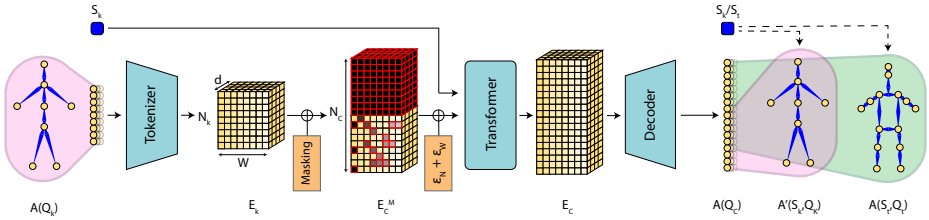

Fig. 3: Skeleton-aware pose masking auto-encoder. Starting with an initial animation $A(Q_k)$, the process begins with a **tokenizer**, which converts each joint of the input into a series of tokens E_k. We then proceed by **masking** (indicated by black squares) a random subset of E_k tokens. The remaining unmasked tokens are concatenated to include every possible topology, forming E_C^M. To capture the relationships between the joints in E_C^M, we add a **spatio-temporal positional embedding**, represented as $\varepsilon_N + \varepsilon_W$. This combined representation is augmented with a learnable token S_k that represents static information, and this forms the input to the **encoding transformer**. The transformer maps the masked input to a latent feature representation E_C, which contains the embedded motion with all masked joints predicted. Subsequently, a **decoder** extracts the super-skeleton motion $A(Q_C)$ from this latent space, using the learned token (S_k during training and S_t during testing). From this, we derive the reconstructed input motion $A'(S_k, Q_k)$ and the retargeted motion $A(S_t, Q_t)$. The auto-encoder is trained to predict the masked joints by minimizing the mean squared error (MSE) loss between the original input $A(S_k, Q_k)$ and the reconstructed output $A'(S_k, Q_k)$.

all possible input skeleton topologies, thereby enhancing the skeleton awareness of our network. Therefore, unlike image-based masking methods, our approach reconstructs not only the missing parts of the input skeleton Q_k but also the empty parts added during the masking process.

Spatial-temporal positional embedding. The purpose of the spatial-temporal positional embedding is to enable the network to learn the inter-dependencies among tokenized joints. Similarly to [10],e adopts separable positional embeddings, thus one for the space and the other for the time. The spatial embedding ε_N with dimensions $N_C \times d$ models the hierarchical graph representation shown in Fig. 2 for all the possible skeleton topologies, and it is repeated for each of the W frames. Its purpose is to learn the parent-child inter-dependencies between all the joints in the kinematic chain. The temporal embedding ε_W with dimensions $W \times d$ models the time relationship between frames, ensuring the smoothness of the final animation. It is repeated for each of the N_C joints. The total spatial-temporal embeddings are given by $\varepsilon_N + \varepsilon_W$, with dimensions $W \times N_C \times d$. Similarly to [9], the values for both ε_N and ε_W are learnable.

Encoding transformer. The encoding transformer takes as input the motion sequence combined with the spatial-temporal encoder, represented as $E_C^M + \varepsilon_N + \varepsilon_W$. Additionally, a learnable token vector S_k of size d is appended to model the static components of the skeleton for various skeleton topologies. This trans-

former, based on a modified ViT architecture ([9]), utilizes distinct activation functions and multiple attention heads to process the spatiotemporal joint data. The output is a latent space containing the embedded reconstructed motion E_C.

Decoder. In line with the approach from [24], our prediction head is replaced with a straightforward linear layer. The linear layer produces an animation $A(Q_{\mathbf{C}})$, which represents the animation of a super-skeleton $Q_{\mathbf{C}}$ encompassing all possible topologies in $\mathbf{C}$.

During training, we extract the motion $A'(Q_k)$ from $A(Q_{\mathbf{C}})$, corresponding to the input motion $A(Q_k)$ as reconstructed by the auto-encoder. The learned static token S_k serves as a selector for the joints related to a specific skeleton topology. The network is trained to reconstruct all input joints (both masked and unmasked) using Mean Square Error (MSE) loss for each frame of the animation. The training loss is defined as follows:

$$\mathcal{L}_{MSE}(A(Q_k), A'(Q_k)) = \frac{1}{W \times N_k} \sum_{w=1}^{W} \sum_{n=1}^{N_k} (FK(S_k, j_n) - FK(S_k, j'_n))^2 \quad (1)$$

where j_n and j'_n are the n-th joints of a frame w of $A(Q_k)$ and $A'(Q_k)$, respectively, and $FK(-)$ represents a Forward Kinematic layer ([1]) that allows to express the joints as 3D spatial positions computed from the quaternions.

At test time, given the reconstructed super-skeleton motion $A(Q_{\mathbf{C}})$ we obtain the motion $A(Q_t)$ for each of the possible skeletons Q_t by applying the same Forward Kinematic layer such that $A(S_t, Q_t) = FK(S_t, j'_n)$.

4 Experiments

To evaluate our framework, we perform the retargeting between isomorphic, homeomorphic, and non-homeomorphic skeletons.

Datasets. We evaluate our method on the Mixamo dataset [4], following the experimental setups of [28] and [23] for isomorphic skeletons. This involves collecting 7 characters for training and 11 for testing. For homeomorphic skeletons, we use the same protocol as [1], excluding 3 characters (Liam, Pearl, and Jasper) which are no longer available on Mixamo.

For extra-structural retargeting, we evaluate our method using two datasets: the Ubisoft La Forge Animation Dataset (LAFAN1) [11] and the quadruped dataset presented in [27]. The LAFAN1 dataset is a human motion dataset containing 5 subjects and 77 different sequences of motions. The quadruped dataset consists of 52 unique dog motion sequences, including idle, walk, run, sit, stand, and a few jumps.

Additionally, we evaluate our method using the Carnegie Mellon University (CMU) Dataset, a real human motion capture dataset comprising 144 actors performing a wide range of motions. Unlike the Mixamo dataset, where retargeting between all characters is achieved through copy rotations, the CMU dataset has been collected using an Optical Motion Capture System. This presents a

more challenging scenario since it involves real actors, each exhibiting unique movement patterns and behaviors, characteristics that are not captured in the Mixamo dataset because of the copy rotation technique.

We cannot quantitatively evaluate the retargeting process between the CMU and Mixamo datasets since the motions in these datasets are not paired. Instead, we assess the network's ability to reconstruct parts of the input real motion. This simulates optical capture sessions where occlusions may cause markers to be undetected, resulting in non-smooth animations. By demonstrating our network's ability to reconstruct motion parts from noisy observations, we highlight its potential for enhancing motion capture data post-processing.

In all scenarios, we use an average of 1250 motions for each training character, 100 for validation, and 60 motions for each test character. The test set includes both seen and unseen characters, but only unseen motions. Due to inconsistencies in train/test splitting across different methods, we provide the full list of characters and animations used in the Supplementary Materials to establish a baseline for future research.

Implementation details. Our network is trained using PyTorch Lightning on a single NVIDIA GeForce RTX 3090. The token vector representing each joint has size $d = 192$, the transformer has 4 heads and the activation function is a Leaky ReLu both in the tokenizer and the transformer. At training time, we randomly mask a subset of $M = 10$ joints. At training time, we randomly mask a subset of $M = 10$ joints for the Mixamo dataset. During training, we set the following hyper-parameters: 25 epochs, learning rate $2e^{-4}$, exponential LR scheduler with $\gamma = 0.95$, batch size 64, and Adam optimizer without weight decay. The optimizer parameters are the same as [17] and $\lambda = 0.7$. As [1], we set the window length to $W = 8$.

Experimental setup and evaluation metrics. As in [1], depending on the variation between the input and target skeletons, the following experiments are defined:

1. retargeting between **isomorphic skeletons**, when the input and target topology have the same number of joints but different bone lengths.
2. retargeting between **homeomorphic skeletons**, when the input and target have different number of joints and bone lengths
3. retargeting between **non-homemorphic skeletons** when input and target have different bone length, number of joints N, and topology as in the human-quadruped retargeting.

To evaluate the performance of our approach we use the Mean Square Error (MSE) between joints for each frame of an animated skeleton. Similarly to [1,28], the MSE is calculated by aligning the root of the retargeted motion to the ground truth (GT), normalizing by each character height.

A crucial aspect of achieving high-quality animation retargeting is addressing mesh interpenetration issues. The Mixamo dataset, while widely used, does not consistently provide clean ground truth animations. In some cases, motions exhibit interpenetration, primarily because the copy-rotation techniques used

between characters with significantly different dimensions can lead to these issues.

To enhance the evaluation process comprehensively, we introduce the Face Interpenetration Error (FIE). This metric quantifies the percentage of faces that collide within a given animation. It serves as a direct indicator of the extent of mesh interpenetration present in the retargeted animations:

$$FIE = \frac{1}{W} \sum_{w=1}^{W} \frac{\#\Delta(S_t, Q_t)}{\#F(M_t)}\% \tag{2}$$

where W is the number of frames of the animation, $\#\Delta(S_t, Q_t)$ is the number of colliding faces and $\#F(M_t)$ is the number of total faces of a mesh.

We propose a comprehensive metric called Skeleton and Collisions Error (SCE), which combines both the skeleton-based metric (MSE) and the shape-based metric (FIE) to account for both collision and joint errors. The (SCE) is calculated as $SCE = MSE \times FIE$. This is the first time such a metric has been introduced.

Non-homeomorphic skeletons. As there is no dataset providing paired motion between two non-homeomorphic characters, it is not possible to compute quantitative results, as no ground truth of the retargeted motion is available.

4.1 Quantitative results

Isomorphic			
Methods	**MSE**	**FIE↓**	**SCE↓**
GT (Mixamo)	-	4.10	-
Copy	0.045	3.23	0.145
NKN*	0.575	-	-
PMnet*	0.281	-	-
SAN	0.141	**1.53**	0.216
R^2ET	**0.042**	3.96	0.166
Ours	0.043	3.13	**0.134**

Table 1: Results for the retargeting between isomorphic skeletons. * indicates values taken from [28].

Homeomorphic			
Methods	**MSE**	**FIE%↓**	**SCE↓**
SAN	0.108	**1.25**	0.135
Ours	**0.025**	3.6	**0.090**

Table 2: Results for the retargeting between homeomorphic skeletons.

Isomorphic skeletons. Tab. 1 presents the quantitative results for retargeting between isomorphic skeletons, with $N_k = 22$, $N_t = 22$, and $N_C = 25$. For fairness, we evaluated copy-rotation, SAN [1] and R^2ET [28] on our data for fairness. We also report the FIE for the ground truth (GT) data to indicate the number of collisions, highlighting the limitations of the Mixamo dataset. When examining the MSE values, both our method and R^2ET outperform the baseline copy-rotation approach, demonstrating better retargeting of skeletal animation between input and target characters. However, MSE alone is not a sufficient evaluation metric since low values can still result in numerous mesh collisions. By

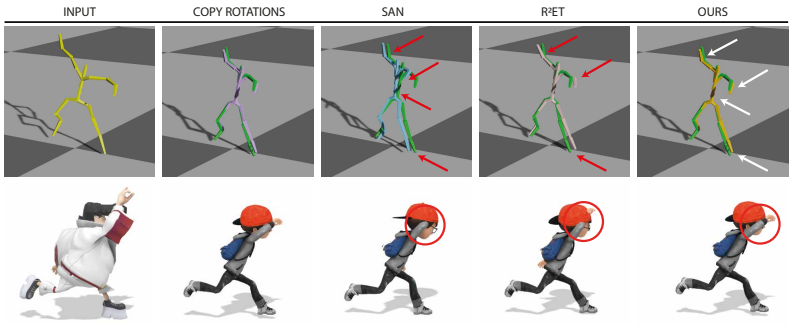

Fig. 4: Qualitative results for the retargeting between **isomorphic skeletons**. The outputs are overlaid with the ground truth, represented by the green skeleton. Our results demonstrate greater stability and alignment with the ground truth. In the second row, we present the mesh visualization. Notably, our method achieves superior qualitative results even without employing any mesh collision penalizer.

analyzing the FIE, we see that our approach, despite lacking any mesh collision avoidance implementation, achieves state-of-the-art performance. It delivers a similar MSE to other methods but with fewer collisions, outperforming all approaches except SAN. This indicates that our low MSE corresponds to effectively retargeted motion with fewer collisions. Furthermore, when considering the combined metric SCE, our approach achieves the lowest number of collisions while maintaining poses that closely resemble the original. Moreover, when evaluating the combined metric SCE, our approach achieves the lowest number of collisions while preserving poses that closely resemble the original. This highlights the effectiveness of our method in capturing the static properties of various skeleton types.

Homemorphic skeletons. Tab. 2 presents the quantitative results for retargeting between homeomorphic skeletons, with configurations of $N_k = 22$ and $N_t = 25$ or vice-versa $N_k = 25$ and $N_t = 22$, and $N_C = 25$. We compare our method to SAN, the only other approach capable of handling homeomorphic skeletons. Our method outperforms SAN in both MSE and SCE, while SAN is better able to limit collisions as shown by the FIE value. Our approach can achieve comparable results relying only on a simple network approach, that is lightweight and less computationally expensive. At inference time, our method can perform the retargeting at 23 fps compared to the 6 fps of SAN. Moreover, we can generalize to different T-poses, like the quadruped one, which is indicated as a failure case in SAN.

4.2 Qualitative results

We provide a demo video showcasing the results of motion sequence retargeting across various scenarios and characters in the Supplementary Material.

Isomorphic skeletons. In Fig. 4, we report the subset of the qualitative results for our approach, compared to copy-rotations, SAN, and R^2ET.

In the first row, we show how other methods such as copy-rotations and SAN cannot fully solve the collisions between the hands. R^2ET solves the collisions, but spreads the hands far apart from each other, losing the original dynamics of the motion. Our method can refine the retargeting using only the skeleton-aware module preserving the original poses and avoiding critical interpenetration between mesh parts.

In the second row, we provide further evidence that our face-based method can better solve the collisions with respect to other methods, being the only one able to avoid interpenetrations between the arm and the head.

Homeomorphic skeletons. In Fig. 5, we show how our method can avoid interpenetrations compared to the skeleton-aware SAN. It is worth noting that there is a large variation in both the skeleton and the shape, a challenging scenario that causes ambiguities in the position of the arm, also visible in the source taken from Mixamo.

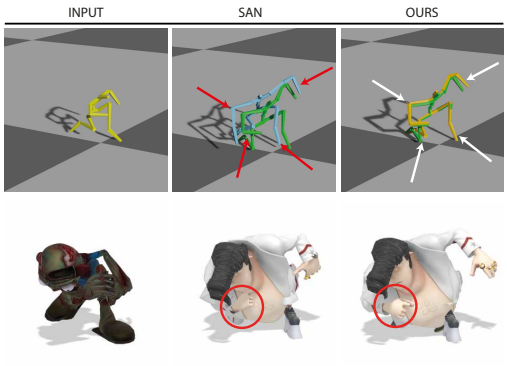

Fig. 5: Qualitative results for the retargeting between **homeomorphic skeletons**. In green the GT.

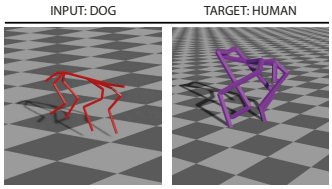

Fig. 6: Qualitative results for the retargeting between **non-homeomorphic skeletons**.

Non-homeomorphic skeletons. In Fig. 6, we demonstrate the retargeting process from a quadruped to a biped. The two skeletons have distinct joint configurations, as detailed in Fig. 1. The non-homeomorphic skeleton of the quadruped exhibits a significantly different kinematic chain and includes an additional end-effector due to the presence of a tail. As also shown in the Supplementary Material video, we are able to achieve good retargeting between such diverse skeletons and semantically diverse motions, surpassing what was indicated as a failure case in previous approaches [1].

Retargeting from real-world character. To demonstrate the robustness and the generalization ability of our method, we show some results from real-world complex motions taken from the Skills from Video (SFV) data ([18])

in Fig. 7. This unlocks the possibility of transferring to simulated characters virtually any human motion video processed by the common SMPL mesh ([14]).

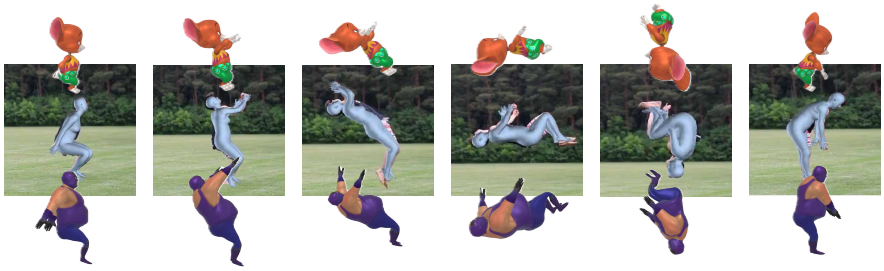

Fig. 7: Our approach can retarget from real-world characters, whose motion can be modeled by a common SMPL mesh, even with challenging and unseen motion such as backflip.

Results on CMU dataset. We evaluate our approach using the CMU dataset, as it consists of real motion capture data from a diverse set of actors performing a wide variety of motions. This allows us to validate our proposal on a realistic and varied dataset, unlike the Mixamo dataset, which is generated through copy rotation between characters. Since the motions in the CMU and Mixamo datasets are not paired, we test the network's capability to reconstruct the original data by masking different amounts of joints. This simulates a real capturing session where researchers might encounter issues such as occlusions or prediction errors. As shown in Table 4, our network is capable of reconstructing real data effectively, and the results are consistent with those obtained from the Mixamo dataset. Fig. 8 illustrates the corresponding qualitative results.

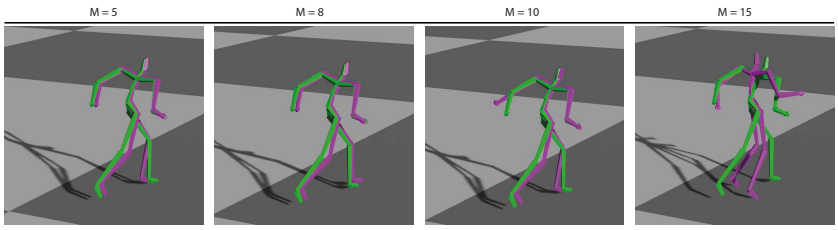

Fig. 8: Qualitative results on the CMU dataset with varying numbers of masked joints. In these visualizations, the ground truth (GT) is shown in green, while the predictions are depicted in purple. The network performs well when up to 10 joints are masked. However, as the number of masked joints exceeds 10, the prediction accuracy decreases significantly. This decline is primarily because masking more than half of the skeleton's joints introduces substantial challenges for the network, leading to increased errors.

M		Masking Strategy		Encoding	
5	0.095	**zero**	**0.043**	No encoding	0.524
10	**0.043**	random	0.095	ε_J	1.025
15	0.094	perturb	1.025	ε_W	1.450
				$\varepsilon_J + \varepsilon_W$	**0.043**

Table 3: Ablation studies on the MSE value for our method. Each column refers to a different experiment: masking of M joints (left), changing the masking strategy (center), ablating the embeddings (right). In bold our best configuration.

Masked Joints	MSE
5	0.047
8	0.035
10	0.081
15	0.15

Table 4: Results on CMU dataset with different number of masked joints.

4.3 Ablation studies

We conduct the ablation studies using the isomorphic skeletons setup.

In Tab. 3, we show the ablation studies for our network. Since it focuses only on the skeleton retargeting, we report the MSE values only. Each column reports a different type of experiment. We first evaluate the effect of the number of masked joints M. M plays a crucial role in the learning phase: in fact, if M is low it means that the network does not have many samples in the ground truth to learn from and to reconstruct the $N_C - N_k$ joints. On the other hand, if M is large, it leads the network to reconstruct too many missing tokens and struggles to learn the spatio-temporal relationships between joints, worsening the results. Next, we evaluate the masking strategy, with 0 masking providing a more consistent starting point for the process and leading to better results. Finally, we ablate the spatial and temporal embeddings. Used individually, each models the relationships only through either time or space, penalizing the other dimension. Without both embeddings, the network is neither penalized nor helped to learn the spatial and temporal relationships. We obtain the best results by combining both, meaning that our embedding can model the relationships across both space and time.

5 Conclusions

In this work, we presented a novel approach for skeleton-aware motion retargeting. To our knowledge, our method is the first method that enables fully automatic motion retargeting between isomorphic, homeomorphic, and non-homeomorphic character topologies. We obtain state-of-the-art results on the Mixamo dataset and a visually convincing motion retargeting between the different skeletal topologies. To achieve better results, we aim to develop a shape-aware module to improve collision avoidance between the final meshes. Moreover, we plan to collect a dataset with the paired motion of non-homeomorphic skeletons, to allow for a quantitative comparison.

Funding Funded by the European Union under NextGenerationEU. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or The European Research Executive Agency. Neither the European Union nor the granting authority can be held responsible for them.

References

1. Aberman, K., Li, P., Lischinski, D., Sorkine-Hornung, O., Cohen-Or, D., Chen, B.: Skeleton-aware networks for deep motion retargeting. ACM Transactions on Graphics (TOG) **39**(4), 62–1 (2020) 1, 2, 3, 4, 8, 9, 10, 12
2. Aberman, K., Weng, Y., Lischinski, D., Cohen-Or, D., Chen, B.: Unpaired motion style transfer from video to animation. ACM Transactions on Graphics (TOG) **39**(4), 64–1 (2020) 4
3. Aberman, K., Wu, R., Lischinski, D., Chen, B., Cohen-Or, D.: Learning character-agnostic motion for motion retargeting in 2d. arXiv preprint arXiv:1905.01680 (2019) 4
4. Adobe: Mixamo (2020) 3, 8
5. Bao, H., Dong, L., Wei, F.: Beit: Bert pre-training of image transformers. arXiv preprint arXiv:2106.08254 (2021) 5
6. Chan, C., Ginosar, S., Zhou, T., Efros, A.A.: Everybody dance now. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 5933–5942 (2019) 1, 2, 3, 4
7. Chen, M., Radford, A., Child, R., Wu, J., Jun, H., Luan, D., Sutskever, I.: Generative pretraining from pixels. In: International conference on machine learning. pp. 1691–1703. PMLR (2020) 5
8. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018) 5
9. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020) 2, 3, 5, 6, 7, 8
10. Feichtenhofer, C., Fan, H., Li, Y., He, K.: Masked autoencoders as spatiotemporal learners. arXiv preprint arXiv:2205.09113 (2022) 3, 5, 6, 7
11. Harvey, F.G., Yurick, M., Nowrouzezahrai, D., Pal, C.: Robust motion in-betweening. ACM Transactions on Graphics (TOG) **39**(4), 60–1 (2020) 8
12. He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R.: Masked autoencoders are scalable vision learners. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16000–16009 (2022) 3, 5
13. Jiang, B., Zhang, Y., Wei, X., Xue, X., Fu, Y.: H4d: Human 4d modeling by learning neural compositional representation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 19355–19365 (2022) 1, 2, 3, 4
14. Kocabas, M., Athanasiou, N., Black, M.J.: Vibe: Video inference for human body pose and shape estimation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 5253–5263 (2020) 13
15. Li, P., Aberman, K., Zhang, Z., Hanocka, R., Sorkine-Hornung, O.: Ganimator: Neural motion synthesis from a single sequence. ACM Transactions on Graphics (TOG) **41**(4), 1–12 (2022) 1, 3, 4
16. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692 (2019) 5
17. Pavlakos, G., Choutas, V., Ghorbani, N., Bolkart, T., Osman, A.A.A., Tzionas, D., Black, M.J.: Expressive body capture: 3D hands, face, and body from a single image. In: Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). pp. 10975–10985 (2019) 9

18. Peng, X.B., Kanazawa, A., Malik, J., Abbeel, P., Levine, S.: Sfv: Reinforcement learning of physical skills from videos. ACM Transactions On Graphics (TOG) **37**(6), 1–14 (2018) 12
19. Seol, Y., O'Sullivan, C., Lee, J.: Creature features: online motion puppetry for non-human characters. In: Proceedings of the 12th ACM SIGGRAPH Symposium on Computer Animation. pp. 213–221 (2013) 4, 5
20. Tong, Z., Song, Y., Wang, J., Wang, L.: Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. arXiv preprint arXiv:2203.12602 (2022) 3, 5, 6
21. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017) 3, 5, 6
22. Villegas, R., Ceylan, D., Hertzmann, A., Yang, J., Saito, J.: Contact-aware retargeting of skinned motion. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9720–9729 (2021) 1, 2, 3, 4
23. Villegas, R., Yang, J., Ceylan, D., Lee, H.: Neural kinematic networks for unsupervised motion retargetting. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 8639–8648 (2018) 1, 2, 3, 4, 8
24. Xie, Z., Zhang, Z., Cao, Y., Lin, Y., Bao, J., Yao, Z., Dai, Q., Hu, H.: Simmim: A simple framework for masked image modeling. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9653–9663 (2022) 3, 5, 8
25. Yamane, K., Ariki, Y., Hodgins, J.: Animating non-humanoid characters with human motion data. In: Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation. pp. 169–178 (2010) 4
26. Yang, Z., Zhu, W., Wu, W., Qian, C., Zhou, Q., Zhou, B., Loy, C.C.: Transmomo: Invariance-driven unsupervised video motion retargeting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5306–5315 (2020) 1, 2, 3, 4
27. Zhang, H., Starke, S., Komura, T., Saito, J.: Mode-adaptive neural networks for quadruped motion control. ACM Transactions on Graphics (TOG) **37**(4), 1–11 (2018) 3, 8
28. Zhang, J., Weng, J., Kang, D., Zhao, F., Huang, S., Zhe, X., Bao, L., Shan, Y., Wang, J., Tu, Z.: Skinned motion retargeting with residual perception of motion semantics & geometry. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13864–13872 (2023) 4, 8, 9, 10
29. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: Proceedings of the IEEE international conference on computer vision. pp. 2223–2232 (2017) 4