Computer Vision and Image Understanding

journal homepage: www.elsevier.com

MoMa: Skinned Motion Retargeting Using Masked Pose Modeling

Giulia Martinelli[a,b], Nicola Garau[a,b], Niccoló Bisagno[a,b], Nicola Conci[a,b,**]

[a] University of Trento,Via Sommarive 14, Trento, 38123, Italy
[b] CNIT - Consorzio Nazionale Interuniversitario per le Telecomunicazioni, Via Sommarive 14, Trento, 38123, Italy

ABSTRACT

Motion retargeting requires to carefully analyze the differences in both skeletal structure and body shape between source and target characters. Existing skeleton-aware and shape-aware approaches can deal with such differences, but they struggle when the source and target characters exhibit significant dissimilarities in both skeleton (like joint count and bone length) and shape (like geometry and mesh properties). In this work we introduce MoMa, a novel approach for skinned motion retargeting which is both skeleton and shape-aware. Our skeleton-aware module learns to retarget animations by recovering the differences between source and target using a custom transformer-based auto-encoder coupled with a spatio-temporal masking strategy. The auto-encoder can transfer the motion between input and target skeletons by reconstructing the masked skeletal differences using shared joints as a reference point. Surpassing the limitations of previous approaches, we can also perform retargeting between skeletons with a varying number of leaf joints. Our shape-aware module incorporates a novel face-based optimiser that adapts skeleton positions to limit collisions between body parts. In contrast to conventional vertex-based methods, our face-based optimizer excels in resolving surface collisions within a body shape, resulting in more accurate retargeted motions. The proposed architecture outperforms the state-of-the-art results on the Mixamo dataset, both quantitatively and qualitatively. Our code is available at: [Github link upon acceptance, see supplementary materials].

1. Introduction

Motion retargeting between two characters has recently gained popularity in computer vision and graphics (Chan et al. (2019); Yang et al. (2020)), with applications ranging from animation (Tak and Ko (2005)) to human-computer interaction (Hecker et al. (2008)).

A character is commonly defined by a skeleton and a mesh, modeling the animation rigging parameters and the body shape, respectively. A skeleton consists of joints and bones, with its outermost parts often referred to as end-effectors (or leaf joints); a mesh consists of vertices, edges and faces. To animate a character, the mesh surface is attached to an animated skeleton via a process called skinning (Kavan (2014)).

In the domain of motion retargeting with *unpaired motion* (Aberman et al. (2020)), the goal is to transfer movements from one character to another while preserving the dynamics of the motion without having neither the explicit mapping between skeletons nor paired motion of the source and the target characters. This retargeting procedure becomes more complex as variations in skeletal structure and body shape between input and target occur.

The common steps of motion retargeting consist of (I) trans-

[**] Corresponding author.
e-mail: giulia.martinelli-2@unitn.it (Giulia Martinelli), nicola.garau@unitn.it (Nicola Garau), niccolo.bisagno@unitn.it (Niccoló Bisagno), nicola.conci@unitn.it (Nicola Conci)

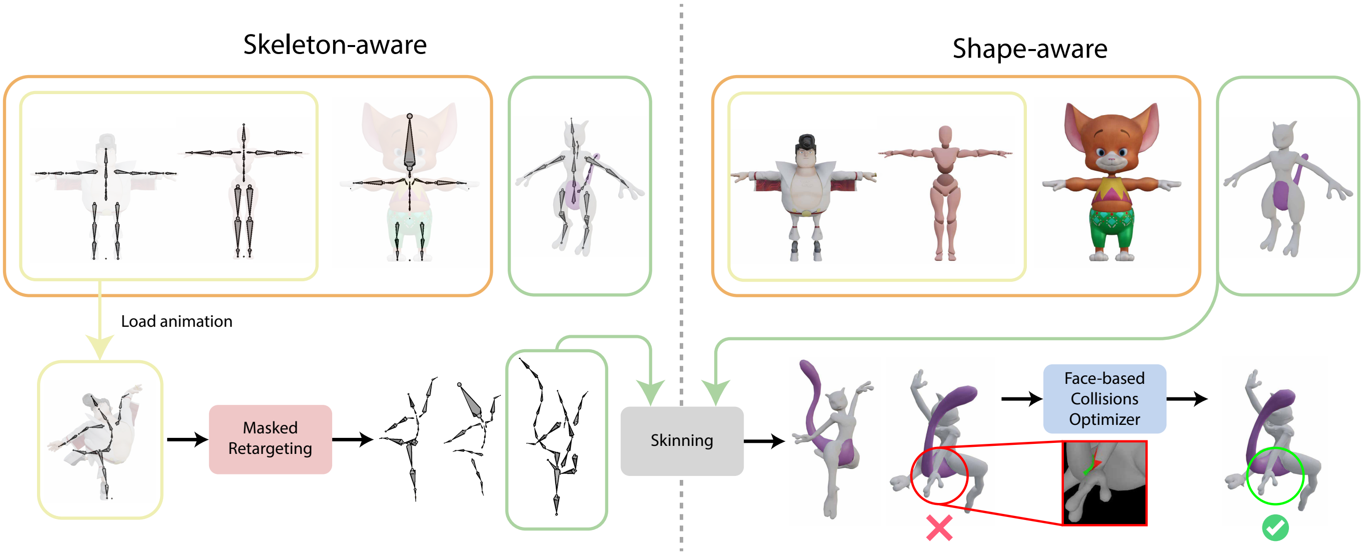

Fig. 1: **Overview of the MoMa architecture.** On the left: the skeleton-aware module can retarget motion between 'isomorphic' (yellow), 'homeomorphic' (orange) as well as 'non-homeomorphic' skeletons (green). On the right: after the skinning, the shape-aware module can adapt the skeleton position to avoid mesh collisions, yet preserving the dynamics of the motion.

ferring the animation from the source to the target skeleton, (II) bounding the mesh to the skeleton via skinning, (III) adjusting the skeleton position to avoid interpenetration between parts of the mesh, and (IV) optionally applying an interpolation filter between frames to obtain a smooth motion. The retargeting process is complex and time consuming, traditionally carried out manually by 3D artists. Thus, researchers have started investigating the possibility to automate the motion retargeting through neural approaches. Existing methods can be generally grouped into two categories: *skeleton-aware* methods focusing solely on the skeleton (Aberman et al. (2020); Lim et al. (2019); Villegas et al. (2018)), and *shape-aware* methods, that also consider the body shape (Zhang et al. (2023); Villegas et al. (2021)).

Skeleton-aware methods model skeletons as graphs with diverse topologies. An input and a target skeleton are *homeomorphic* when they share the same number of end-effectors and have different topology. They are considered *non-homeomorphic* when they don't. *Isomorphic* skeletons are homeomorphic skeletons that, additionally, also share the same number of joints. In contrast to existing methodologies designed to handle only isomorphic skeletons (Villegas et al. (2018, 2021); Zhang et al. (2023); Lim et al. (2019)) or homeomorphic ones (Aberman et al. (2020)), our approach can also deal with non-homeomorphic skeletons.

Shape-aware methods optimize the motion considering body shapes as meshes with faces and vertices. The mesh skinning for each frame of an animation might result in collisions between body parts, which need to be solved to obtain a good retargeted motion. While available methods (Zhang et al. (2023); Villegas et al. (2021)) use vertices to solve collisions, we use faces instead. Resolving collisions at face level minimizes unwanted mesh surface deformation: in fact, vertices might respond individually to collision detection, leading to erratic movements or distortions in the mesh.

With **MoMa**, we propose a novel approach that handles disparities in both skeleton and shape domains for motion retargeting, as shown in Fig. 1.

In the *skeleton-aware* module, our new **pose masking auto-encoder** can retarget motion between isomorphic, homemorphic and non-homeomorphic skeletons. To do so, we draw inspiration from masked image modeling (He et al. (2022); Xie et al. (2022); Tong et al. (2022); Feichtenhofer et al. (2022)) used in transformer auto-encoders (Dosovitskiy et al. (2020); Vaswani et al. (2017)) to reconstruct masked portions of the input data, extending the same principle to skeleton motion modeling. In our approach, we predict the missing joints in the target skeleton similarly to how masked patches are reconstructed starting from visible ones. Our approach aims at providing a simple baseline for motion retargeting that alleviates all the complexities of previous state-of-the-art methods, like the need for sophisticated cycle consistency or adversarial loss (Zhang et al. (2023); Villegas et al. (2021)), while outperforming them. We believe that masked transformers can be more suitable for the task compared to the previous proposed architectures, unlocking new possibilities, such as motion retargeting between non-homeomorphic characters.

In the *shape-aware* module, our method solves collisions between body parts through a *quasi-Newton* **face-based optimizer**, outperforming existing vertex-based methodologies. Inspired by SMPL approaches (Pavlakos et al. (2019); Tzionas et al. (2016)), we develop an optimizer that can selectively solve collisions for a variety of different complex meshes.

The novelties of our work can be summarised as follows:

- we propose a novel pipeline for skinned motion retargeting, which is both *skeleton-aware* and *shape-aware*;
- we propose the first motion retargeting approach that deals with non homeomorphic skeletons without paired motion;
- we introduce a novel **pose masking auto-encoder** to reconstruct absent joints in the target;
- we implement a novel a **face-based optimizer** to solve collisions in the body mesh, which is a more consistent and precise approach compared to vertices-based solutions;
- we obtain state-of-the-art results on motion retargeting on the Mixamo dataset (Adobe (2020)) and provide a framework for transferring the motion from real videos to synthetic characters.

2. Related work

Masked modeling for representation learning. Masked language modeling (Devlin et al. (2018); Liu et al. (2019)) and masked image modeling (Xie et al. (2022); He et al. (2022); Bao et al. (2021); Dosovitskiy et al. (2020); Chen et al. (2020); Xie et al. (2022)) have recently demonstrated how self-supervised approaches can achieve better performances compared to fully supervised ones, making them scalable representation learners for multiple tasks. Among image-based approaches, SimMIM (Xie et al. (2022)) has proven to be a simple yet effective strategy, masking image patches by replacing them with random token vectors of the same dimension. In the video domain, where data is not only spatially but also temporally correlated (Tong et al. (2022); Feichtenhofer et al. (2022)), the relationship between tokens is strong enough to allow the network to reconstruct the input with up to 90% of masked frames.

In our solution, we apply a similar strategy by randomly masking a subset of skeleton joints both in space and time, demonstrating the effectiveness of the learned skeleton representation on the motion retargeting task. In the pose domain, it is crucial to accurately model the relationships between joints, a task that holds greater significance than modeling dependencies between patches in the image domain. This distinction arises because an image patch encapsulates considerable information from multiple pixels, whereas each joint is characterized by a limited set of numerical values representing positions or rotations.

Skeleton-aware motion retargeting. Skeleton-based motion retargeting aims at transferring the motion from an input skeleton to a target one (Zhang et al. (2023); Villegas et al. (2021); Lim et al. (2019); Villegas et al. (2018); Lee et al. (2023)). Input and target skeletons can be isomorphic, homeomorphic or non-homeomorphic. A basic approach to 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 taking care of changes in scale or bone length, as described by Aberman et al. (2020). Most approaches (Lim et al. (2019); Villegas et al. (2018); Zhang et al. (2023); Villegas et al. (2021); Lee et al. (2023)) can retarget unpaired motion between isomorphic skeletons, but cannot generalize to homeomorphic ones. The work by Aberman et al. (2020) introduces a Skeleton-Aware Network (SAN) for motion retargeting between homeomorphic skeletons. In Lee et al. (2023), they solve various animation tasks in a skeleton-agnostic manner, also tackling the retargeting between isomorphic and homeomorphic skeletons. However, both methods cannot generalize to *non-homeomorphic* skeletons with different topologies where the number of leaf joints changes. Motion retargeting with non-homeomorphic skeletons has been investigated in the past with non-neural approaches (Yamane et al. (2010); Seol et al. (2013)), relying on paired motions or explicit skeleton mappings.

To the best of our knowledge, our proposed solution is the first one that can deal with skeleton-based motion retargeting between non-homeomorphic skeletons without paired motion or ad-hoc mapping.

Shape-aware motion retargeting. Meshes are fundamental when evaluating the animation retargeting performances between two characters. The first methods dealing with skinned characters (Villegas et al. (2018); Lim et al. (2019)) cannot be considered shape-aware because they do not consider the character's mesh to optimize the motion retargeting, resulting in unrealistic movements of skinned characters. Neural Kinematic Networks (NKN) by Villegas et al. (2018) and PMNet by Lim et al. (2019) use simple Linear Blend Skinning (LBS) (Kavan (2014)) before evaluating their performances on the retargeted body shape. More recent methods (Zhang et al. (2023); Villegas et al. (2021)) can be considered *shape-aware* because they explicitly include the mesh in the retargeting process. Other methods tackle either the motion generation task starting from the mesh generation Yang et al. (2024) or the human motion transfer between meshes Regateiro and Boyer (2022); Chen et al. (2021). Shape-aware methods aim to avoid mesh interpenetration by detecting contacts between different parts of the mesh using vertices (Villegas et al. (2021); Zhang et al. (2023)). The work proposed by Villegas et al. (2021) focuses on detecting contacts between different parts of the mesh, as well as foot-ground contact using an encoder-decoder network optimisation. Zhang et al. (2023) define a skeleton-aware and a shape-aware module, the latter being trained employing an attractive/repulsive field mechanism to solve collisions, while adhering to the target motion. Despite the ability to generalise to diverse body shapes, this solution only solves body-limbs collisions but not limbs-limbs nor body-body ones.

In a similar fashion to (Pavlakos et al. (2019); Tzionas et al. (2016)), we introduce a generalized face-based optimizer to solve all possible collisions.

3. Method

We present MoMa, a novel skeleton-aware pose masking auto-encoder combined with a shape-aware face-based optimizer to tackle the skinned motion retargeting task.

Given a set of possible characters $\mathbf{C} = \{C_1, \ldots, C_k, \ldots, C_K\}$, our goal is to retarget the motion A from an input character C_k to a different one C_t.

The motion retargeting from C_k to C_t follows these steps:

1. *Skeleton motion retargeting.* We perform the retargeting of the chosen motion from the input skeleton of C_k to the target skeleton of C_t using our **skeleton-aware pose masking auto-encoder**.
2. *Skinning and collisions detection.* For each frame of the retargeted animation A, we apply Linear Blend Skinning (LBS) to attach the target mesh of C_t to the retargeted skeleton. Then, we detect collisions between mesh parts, which correspond to possible interpenetration and artifacts.
3. *Mesh optimization to solve collisions.* We adapt the joints positions of the skeleton to minimise the collisions of the mesh of C_t using our **shape-aware face-based optimizer**.

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

Character representation. Each character is represented by a skeleton (S_k, Q_k) and a mesh M_k, as described in Fig. 2. The skeleton is divided into a static representation S_k and a motion representation $A(Q_k)$. The static representation S_k contains the

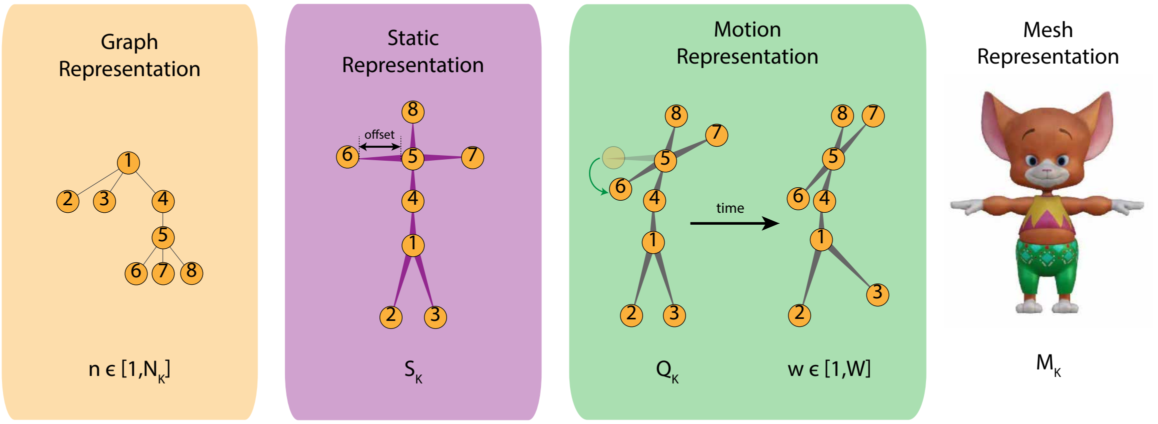

Fig. 2: **Animation representation.** Each skeleton can be represented as a graph with $n \in [1, N_k]$ nodes (joints) where the parents-child relationship is defined by the kinematic chain. Moreover, each skeleton is described by a static representation S_k containing the offsets (bone lengths), and a motion representation $A(Q_k)$.

offsets between joints. Instead, Q_k represents the motion representation at a specific frame of the animation. It consists of N_k 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. The mesh in T-pose to be skinned to the skeleton is defined as M_k. While the static representation S_k does not vary for a character during an animation, the motion representation $A(Q_k)$ and the mesh M_k are subject to changes to obtain the animation.

Animation representation. An animation is composed by three main parts:

- S_k **(Static Representation)**. This component contains information about the offsets and inherently includes the skeleton's topology information, such as the edges. The static representation remains constant and serves as the structural framework for the skeleton. S_k is a $N \times d$ vector, where N is the number of joints, and $d = 3$ represents the xyz coordinates.

- $A(Q_k)$ **(Motion Representation)**. This component involves the rotations expressed using quaternions. These rotations change over time, creating the animation by driving the movements of the skeleton. The motion representation is dynamic, as it varies with each frame of the animation sequence. $A(Q_k)$ has dimensions $N \times w \times d$, where N is the number of joints, w is the window length and $d = 4$, since the motion is expressed in quaternions format.

- M_k **(Mesh Representation)**. This component represents the mesh of the character. Through the skinning process, the animation of the skeleton is bound to the mesh. As the skeleton moves, the mesh vertices change position, making the mesh dynamic and altering its shape over the course of the animation. M_k is expressed as a set of vertices, with each vertex having its own xyz coordinates, whose dimension is $N_v \times d$, where N_v is the number of vertices and $d = 3$.

In summary, the static representation (S_k) remains constant, providing the structural framework of the skeleton. The motion representation captures the dynamic rotations that drive the animation, while the mesh representation (M_k) changes as the skeleton moves, reflecting the animated character's shape.

We define $A((S_k, Q_k), M_k)$ as the motion (animation) of the input character C_k to be retargeted. Our goal is to obtain $A((S_t, Q_t), M_t)$, which corresponds to the same motion applied to the target character C_t. The length of the animation is a window of size $w \in [1, W]$ frames. We define the skeletal-only motion as $A(S_k, Q_k)$, namely the same motion without the mesh component. We can further divide the skeletal motion between the skeletal motion representation $A(Q_k)$, that consists of a varying Q_k for each of the W frames, and the static representation S_k.

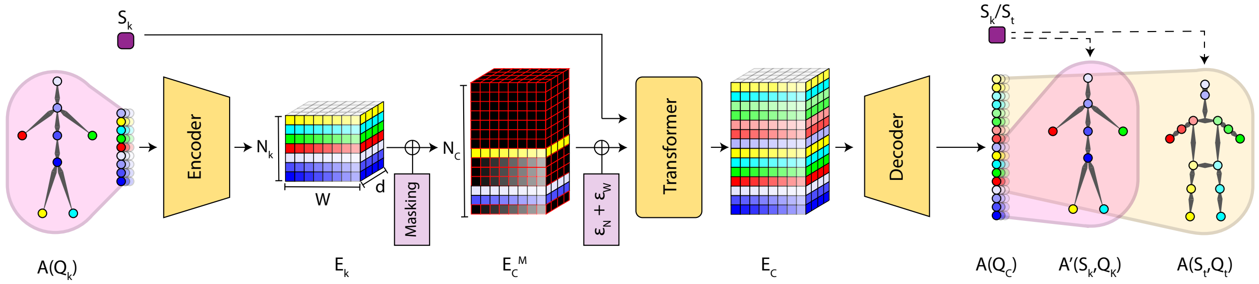

Fig. 3: **Skeleton-aware pose masking auto-encoder.** Starting from an input animation $A(Q_k)$, an **encoder** embeds each input joint into a set of tokens E_k. Next, we randomly **mask** (black squares) a subset of E_k and concatenate the remaining missing joints to include all the possible topologies, resulting in E_C^M. To model the relationships between the embedded joints in E_C^M, we add a **spatio-temporal positional embedding** $\varepsilon_N + \varepsilon_W$. To this representation, we concatenate the learnable token S_k representing the static information to form the input of the **encoding transformer**. The latter learns to map the masked input to a latent feature representation containing the embedded motion E_C, where all the masked joints have been predicted. Finally, the **decoder** extracts the super-skeleton motion $A(Q_C)$ from the latent space using the learnt token (S_k at training time and S_t at test time), from which we can derive the reconstructed input motion $A'(S_k, Q_k)$ and the retargeted motion $A(S_t, Q_t)$. We train our auto-encoder to predict the masked joints by enforcing a MSE loss between the input $A(S_k, Q_k)$ and the reconstructed $A'(S_k, Q_k)$.

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.

Objective. Given a skeletal motion $A(S_k, Q_k)$ on an input character C_k, our pose masking auto-encoder performs the retargeting to obtain the same motion $A(S_t, Q_t)$ in a target skeleton format of the character C_t.

Encoder. Since transformers process chunks of data (tokens) such as words in NLP (Vaswani et al. (2017)), or image patches in computer vision (Dosovitskiy et al. (2020)) within their multi-headed architecture, we apply a similar strategy by tokenizing the dynamic part $A(Q_k)$ of the input animation. Thus, for every frame w of $A(Q_k)$, we embed each joint j_n of Q_k as a token. The embedding is applied to each joint using a single linear layer followed by a Leaky ReLU to obtain a vector of size d for each joint. After the tokenization, each embedded animation E_k sequence has dimensions $W \times N_k \times d$.

Pose masking strategy. In analogy with image-based masking approaches (Tong et al. (2022); Feichtenhofer et al. (2022)), also in our case the goal is to train an auto-encoder by taking as input masked data and reconstruct the missing parts of it. Given an embedded animation E_k of size $W \times N_k \times d$, we mask a subset of joints M to obtain the masked embedded animation. As introduced by Xie et al. (2022), we replace each masked token with a vector of the same size d, which can contain different values depending on the chosen masking strategy.

Next, we concatenate a set of $N_C - N_k$ empty tokens to the masked embedded animation joints to obtain E_C^M, where N_C represents the set of all possible joints in the skeletons of characters in $\mathbf{C}$. This operation ensures that every possible input animation can be mapped into a latent representation big enough to contain all the possible skeleton topologies in $\mathbf{C}$. The latent representation E_C^M has size $W \times N_C \times d$ and each token is initialised as a masked vector of size d.

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 Feichtenhofer et al. (2022), we adopt two separable positional embeddings, in the spatial and in the temporal domains, respectively. The goal of the spatial embedding ε_N is to learn the hierarchical graph representation shown in Fig. 2 of the N_C joints for all the possible characters $\mathbf{C}$. The goal of the temporal embedding ε_W is to learn the spatial relation between the dynamic part of the skeleton Q_k over the $w \in [1, W]$ time frames.

The spatial embedding ε_N with dimensions $N_C \times d$ models the spatial relations between joints, and it is repeated for each of the W frames. The temporal embedding ε_W with dimensions $W \times d$ models the time relationship between frames, and it is repeated for each of the N_C joints. The total spatial-temporal embedding is given by $\varepsilon_N + \varepsilon_W$, with dimensions $W \times N_C \times d$. The spatial-temporal embedding is added to the embedded animation to retain positional information. This separable implementation prevents the size of positional embeddings growing too large in 3D. Similarly to Dosovitskiy et al. (2020), the values for both ε_N and ε_W are learnable.

Encoding transformer. The input of the encoding transformer is the motion sequence plus the spatial-temporal encoder expressed as $E^M_C + \varepsilon_N + \varepsilon_W$. To this input, we concatenate a learnable token vector S_k of size d that models the static part of the skeleton for each possible character in **C**. The transformer is a custom ViT architecture (Dosovitskiy et al. (2020)), with different activation functions and number of attention heads that process spatio-temporal joints information. At the output, the latent space contains the embedded reconstructed motion E_C.

Decoder. Similarly to Xie et al. (2022), our prediction head (decoder) is replaced by a simple linear layer. The output of the linear layer is an animation $A(Q_{\mathbf{C}})$ which can be seen as the animation of a super-skeleton $Q_{\mathbf{C}}$ containing all the possible topologies in **C**.

At training time, from $A(Q_{\mathbf{C}})$ we need to extract the motion $A'(Q_k)$, which corresponds to the input motion $A(Q_k)$ as reconstructed by the auto-encoder. This is possible thanks to the learned static token S_k, which acts as a selector for the joints corresponding to a given character. The network is trained to reconstruct all the input joints (both masked and unmasked) using the Mean Square Error (MSE) loss on each single frame of the animation. The training loss is defined as:

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

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 (Aberman et al. (2020)) that allows to express the joints as 3D spatial positions computed from the quaternions. Similarly to Aberman et al. (2020), the loss is not enforced directly on the quaternions in order to avoid accumulation of error along the kinematic chain. This is possible as the loss is enforced over the joint position.

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 $A(S_t, Q_t) = FK(S_t, j'_n)$.

3.2. Shape-aware face-based optimizer

For our shape-aware module, we design a face-based optimizer that applies an iterative process to obtain a refined animation as shown in Fig. 4. Inspired by Pavlakos et al. (2019), we design a quasi-Newton optimizer that applies to any mesh with triangular faces. Differently from neural vertex-based approaches (Zhang et al. (2023); Villegas et al. (2021)), our optimizer starts from triangular faces of the mesh to build 3D volumes, as shown in Fig. 5.

In the following paragraphs we describe the details of each module.

Skinning. Starting from the retargeted skeletal motion $A(S_t, Q_t)$, we apply the skinning process to the target mesh M_t and skeleton (S_t, Q_t) for each frame to obtain $A((S_t, Q_t), M_t)$. A target mesh in T-pose is defined by its vertices v. The target mesh M_t in a different pose (S_t, Q_t) is defined by its vertices v', which positions can be computed by a linear transformation of the vertices v, called Linear Blend Skinning (LBS) (See Supplementary Materials).

By adjusting the positions of the joints j_n of the motion representation Q_t, the optimizer can adapt the positions of the vertices v' of the mesh M_t to avoid collisions and obtain the resulting $A((S_t, Q_t), M_t)$.

Collision penalizer. Given $A((S_t, Q_t), M_t)$, we detect all the colliding triangles $\Delta((S_t, Q_t))$ of vertices using Bounding Volume Hierarchy (Karras (2012); Pavlakos et al. (2019)). Differently from vertex-based approaches, which directly operate on the colliding vertices (Zhang et al. (2023); Villegas et al. (2021)), we build a conic 3D volumetric distance field Ψ on the external face of each triangular vertex face Δ_Ψ and its normal vector $\hat{v}$. Given two colliding triangles, Δ_i and Δ_r the interpenetration is defined as bi-directional, where the vertices v_i of Δ_i

Isomorphic				
	Methods	**MSE**	**FIE%↓**	**SCE↓**
	GT (Mixamo)	-	4.10	-
	Copy	0.045	3.23	0.145
Skeleton-aware	NKN*	0.575	-	-
	PMnet*	0.281	-	-
	SAN	0.141	*1.53*	0.216
	SAME	0.176	1.21	0.213
	Ours (no opt)	*0.043*	3.13	*0.134*
Shape- aware	R^{2}ET	**0.042**	3.96	0.166
	SAN opt	0.163	1.02	0.166
	Ours	0.049	**1.01**	**0.049**

Table 1: Results for the retargeting between isomorphic skeletons.

Homeomorphic			
Methods	**MSE**	**FIE%↓**	**SCE↓**
SAN	0.108	1.25	0.135
SAME	0.122	1.12	0.137
Ours (no opt)	**0.025**	3.6	0.090
SAN opt	0.117	0.91	0.106
Ours	0.031	**0.9**	**0.028**

Table 2: Results for the retargeting between homeomorphic skeletons.

are the intruders of the distance field Ψ_{Δ_r} of the receiver triangle Δ_r, and vice-versa. The collision term ξ to be minimised is defined as:

$$\xi(S_t, Q_t) = \sum_{(\Delta_i,\Delta_r)} \Big\{ \sum_{v_i \in \Delta_i} \| -\Psi_{\Delta_r}(v_i)\hat{v}_i\|^2 + \sum_{v_r \in \Delta_r} \| -\Psi_{\Delta_i}(v_r)\hat{v}_r\|^2 \Big\} \tag{2}$$

The collision term $\xi(S_t, Q_t)$ indicates how much two volumetric distance fields built on faces are colliding. Being grouped in triangular faces, a vertex cannot respond individually to a collision, leading to an improved mesh consistency. For further mathematical details please refer to Pavlakos et al. (2019); Tzionas et al. (2016).

To help the optimization process we need to detect and exclude self-contacts (e.g. eyes colliding with the skull) from the collision term. Similarly to Villegas et al. (2021), and differently from the SMPL approaches where the self-contact points are consistent, we do not have any prior of the body shape because the characters in the available datasets have a much wider spectrum of possible meshes with highly varying features (e.g. hair, clothes, accessories). Thus, we need to detect and filter out all the self-contacts for each mesh M_k. To do so, we set all the input meshes in T-pose and we exclude the collisions between all the body parts with self-contacts from the collision term. Moreover, we also filter out collisions between meshes of neighbor body parts in the kinematic chain (e.g. neck and torso) and self-collisions between parts of the same limb (e.g. the upper arm and the lower arm).

Optimization. We choose a Limited-memory BFGS optimizer (L-BFGS) (Nocedal and Wright (1999)) because it is a quasi-Newton method that approximates the BFGS algorithm, a highly efficient tool for optimizing smooth, convex functions. It is used to optimize the following loss for each frame of the animation:

$$\mathcal{L}_O(S_t, Q_t) = \lambda\xi(S_t, Q_t) + (1-\lambda)\mathbf{L}_{MSE}((S_t, Q_t), (S_t, Q_{t-1})) \tag{3}$$

where λ is the balancing weight of the loss and (S_t, Q_{t-1}) represents the same skeleton at the previous time step. The term $\xi(S_t, Q_t)$ aims at solving the collisions, while the second term $\mathcal{L}_{MSE}$ aims at providing consistency between frames and enforcing the motion to be as close as possible to the input.

Thus, the optimizer learns to adapt the skeleton $Q(S_t, Q_t)$ to minimise both terms, to obtain $A((S_t, Q_t), M_t)$ and preserve the dynamics of the input motion without collisions.

4. Experiments

To evaluate MoMa, we perform the retargeting between isomorphic, homeomorphic and non-homeomorphic characters with varying body shapes. Throughout the result section we use *Ours (no opt)*, when referring to MoMa with only the skeleton-aware module, while *Ours* corresponds to the full approach, which also includes the shape-aware optimizer.

Datasets. We evaluate MoMa on the Mixamo dataset (Adobe (2020)). For the retargeting between isomorphic skeletons, we

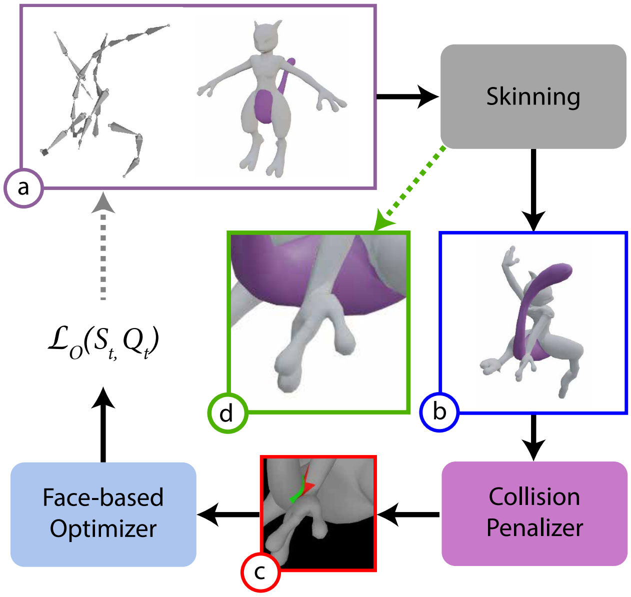

Fig. 4: **Shape-aware face-based optimizer.** (a) Given the retargeted skeletal motion $A(S_t, Q_t)$ and the corresponding mesh in T-pose, we apply the **skinning** to obtain the full animation $A((S_t, Q_t), M_k)$. (b) During each iteration the mesh can display collisions, which (c) are detected and weighted by the **collision penalizer**. The **face-based optimizer** minimizes the loss $L_o(S_t, Q_t)$ by adapting the skeleton position (S_t, Q_t) until obtaining (d) a retargeted motion without collisions.

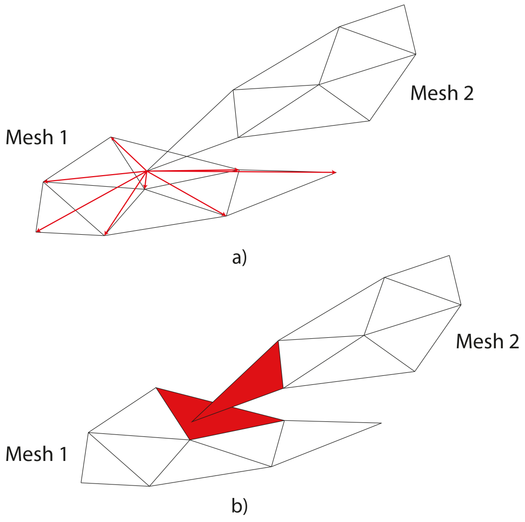

Fig. 5: (a) Vertex-based methods use a global distance field between each vertex and the others, thus providing a computationally expensive one-to-many metric. (b) Face-based methods identify collisions with colliding triangle surfaces, thus a more efficient one-to-one metric.

follow the same setup as Zhang et al. (2023) and Villegas et al. (2018), collecting 7 characters for training and 11 for testing. For homeomorphic skeletons, we follow the same protocol as in Aberman et al. (2020), but for 3 characters (Liam, Pearl and Jasper), which are no more available on Mixamo. For non-homeomorphic skeletons, we selected 2 CC-Licensed characters available on Sketchfab, animated using Mixamo, and we combined them with the characters used for the isomorphic experiments. 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 contains seen and unseen characters with only unseen motions. Given the encountered inconsistencies of the train/test splittings of the Mixamo dataset for different methods (Zhang et al. (2023); Lim et al. (2019); Villegas et al. (2018); Aberman et al. (2020)), we provide the full list of characters and animations used in training and testing in the Supplementary Materials, hoping to establish a relevant baseline for future research.

Furthermore for non-homeomorphic character, we utilize two datasets: the Ubisoft La Forge Animation Dataset (LAFAN1) from Harvey et al. (2020), and the quadruped dataset presented in Zhang et al. (2018). The LAFAN1 dataset comprises human motion data, featuring 5 subjects and 77 different motion sequences. In contrast, the quadruped dataset includes 52 unique sequences of dog motions, encompassing various activities such as idle, walk, run, sit, stand, and several jumps.

We test our method also on the Carnegie Mellon University (CMU) Motion Capture Database CMU. This dataset captures real human motion with 144 actors performing a diverse array of movements. Unlike the Mixamo dataset, where retargeting is performed by copying rotations, the CMU dataset is collected using an Optical Motion Capture System. This introduces a more complex scenario due to the presence of real actors, each exhibiting unique movement patterns and behaviors, aspects that the Mixamo dataset does not account for due to its reliance on copy rotation.

Quantitative evaluation of retargeting between the CMU and

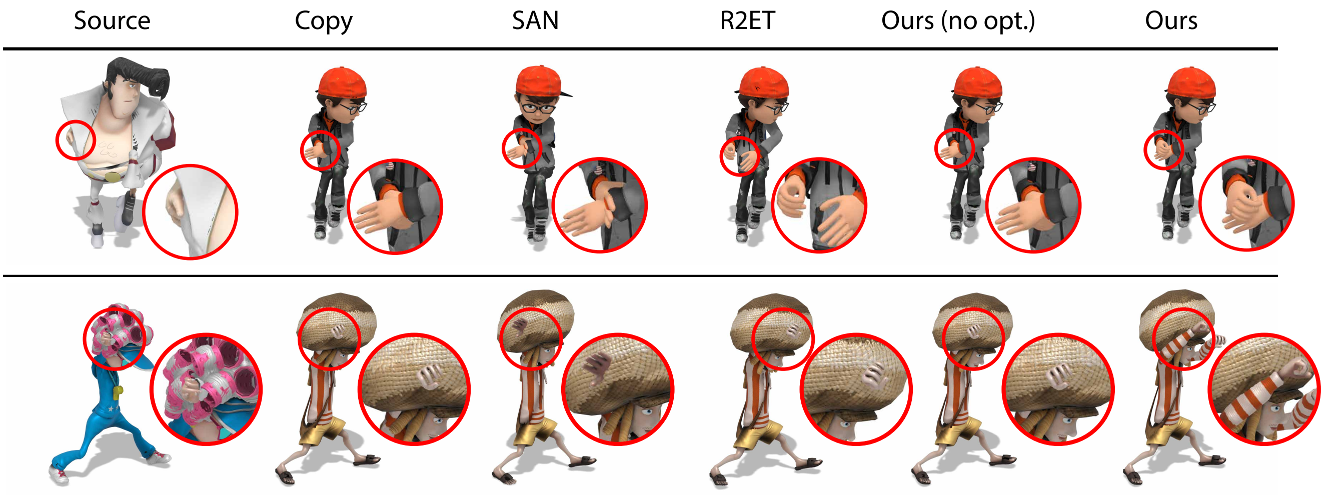

Fig. 6: Qualitative results for the retargeting between **isomorphic skeletons**. Our method (last column) successfully minimizes the Skeleton Collision Error (SCE) compared to state-of-the-art results.

Mixamo datasets is not feasible since the motions in these datasets are not paired. Instead, we focus on the network's ability to reconstruct segments of the input real motion. This approach simulates optical capture sessions where occlusions may lead to undetected markers, resulting in non-smooth animations. By showcasing our network's capacity to reconstruct motion segments from noisy observations, we emphasize its potential to enhance the post-processing of motion capture data.

Implementation details. Our auto-encoder is trained using PyTorch Lightning on a single NVIDIA 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 encoder and the transformer. 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 Pavlakos et al. (2019) and $\lambda = 0.7$. As Aberman et al. (2020), we set the window length to $W = 64$.

Evaluation metrics. To evaluate the performances of our approach we use the Mean Square Error (MSE) between joints for each frame of an animated skeleton. Similarly to Aberman et al. (2020); Zhang et al. (2023), the MSE is calculated aligning the root of the retargeted motion to the ground truth (GT), normalizing by each character height. The Mixamo dataset does not always provide clean GT; in fact, in a few cases, we noticed that the motions display interpenetration or contact-missing issues. Thus, a low MSE does not necessarily guarantee an accurate retargeting without collisions (Zhang et al. (2023)).

Therefore, to provide a more comprehensive evaluation, we introduce the Face Interpenetration Error (FIE) indicating the percentage of faces colliding in a given animation as:

$$FIE = \frac{1}{W}\sum_{w=1}^{W}\frac{\#\Delta(S_t, Q_t)}{\#F(M_t)}\% \tag{4}$$

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.

In an ideal scenario, retargeting would return a score equal to 0 for both MSE and FIE. However, in practice, achieving this is not possible, as reducing MSE often leads to an increase in FIE and vice versa. Thus, combining both the skeleton-based metric MSE and the shape-based FIE, we propose a comprehensive metric, which takes into account both collisions and joints errors, called Skeleton and Collisions Error (SCE) computed as $SCE = MSE \times FIE$. In designing the SCE, we aimed to capture the dependency between the skeletal and shape aspects of motion retargeting. Choosing a weighted sum, multiplication, or another relation for the SCE is a matter of preference and does not alter its fundamental significance.

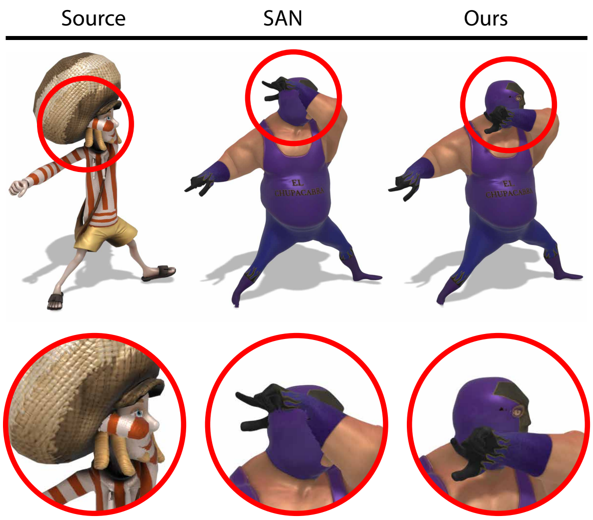

Fig. 7: Qualitative results for the retargeting between **homeomorphic skeletons**.

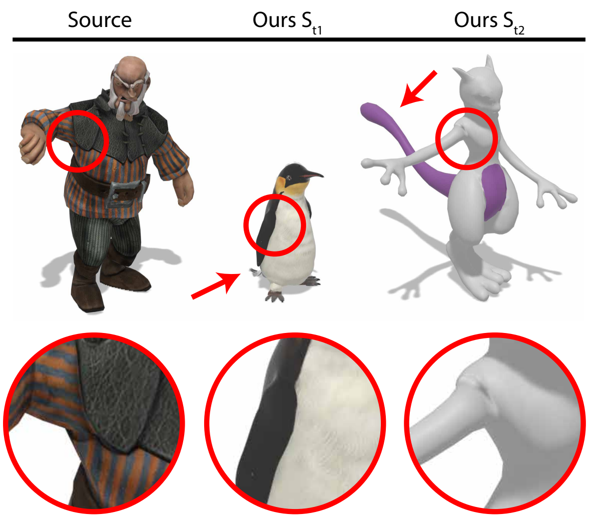

Fig. 8: Qualitative results for the retargeting between **non-homeomorphic skeletons**.

4.1. Quantitative results

Isomorphic skeletons. We report the quantitative results for the retargeting between isomorphic skeletons in Tab. 1 with $N_k = 22$, $N_t = 22$ and $N_C = 25$. We compare against both shape-aware and skeleton-aware methods. We run copy-rotation, SAN (Aberman et al. (2020)), SAME (Lee et al. (2023)) and R^{2}ET (Zhang et al. (2023)) on our data for fairness. For the GT data, we report the FIE, indicating the number of collisions happening, further demonstrating the limits of the Mixamo dataset. Looking at the MSE values, both our method and R^{2}ET outperform the baseline copy-rotations, meaning that they can better retarget the skeletal animation between input and target character. However, the MSE does not provide a comprehensive evaluation metric, since low values can correspond to many collisions in the mesh. Looking at the FIE, our non-optimised skeleton-aware module already achieves state-of-the-art performances, with a similar MSE and a lower number of collisions compared to all other approaches, meaning that the low MSE corresponds to a good retargeted motion with fewer collisions than other approaches. For fairness, we apply our shape-aware module on top of SAN method (denoted as SAN opt), the number of collisions decreases. This proves that our skeleton-aware and shape-aware modules are effective independently and can be utilized separately. Looking at the combined metric SCE, our optimized approach with both the skeleton-aware and shape-aware method obtains the lowest number of collisions, while maintaining a pose closer to the original, indicating the best retargeted motion.

Homeomorphic skeletons. We report the quantitative results for the retargeting between isomorphic skeletons in Tab. 1 with $N_k = 22$ and $N_t = 25$ or vice-versa $N_k = 25$ and $N_t = 22$, and $N_C = 25$. We compare with SAN, the only other method able to deal with homeomorphic skeletons. Our approach outperforms SAN across all the metrics, thus achieving motions with less collisions and better preserving the semantics of the original ones, using either our skeleton-aware module or combined with the shape-aware one.

Motion Reconstruction. We evaluated our approach using the CMU dataset, which comprises real motion capture data from a diverse group of actors performing a wide range of motions. This evaluation on a realistic and varied dataset allows us to validate our method comprehensively, in contrast to the Mixamo dataset, which involves motion generated through copy rotation between characters.

Due to the absence of paired motions between the CMU and Mixamo datasets, we assessed our network's ability to reconstruct the original data by masking various joints. This process simulates real capturing sessions, where researchers may face challenges such as occlusions or prediction errors. As shown in Table 4, our network effectively reconstructs real motion cap-

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 **skeleton-aware module**. 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.

ture data, yielding results consistent with those obtained from the Mixamo dataset. We observed similar performances on the LAFAN1 dataset, further corroborating the robustness and generalization abilities of our approach.

Dataset	Masked Joints	MSE
LAFAN1	5	0.042
	8	0.032
	10	0.050
	15	0.070
CMU	5	0.047
	8	0.035
	10	0.081
	15	0.15

Table 4: Results on LAFAN1 and CMU dataset with different number of masked joints.

4.2. *Qualitative results*

We provide a demo video, showcasing the results of the motion sequence retargeting for all the possible scenarios, with many different characters at the following link (`https://youtu.be/gWQJ1ltQGnQ`).

Isomorphic skeletons. In Fig. 6, we report the a subset of the qualitative results for our approach, compared to copy-rotations, SAN, and R^{2}ET.

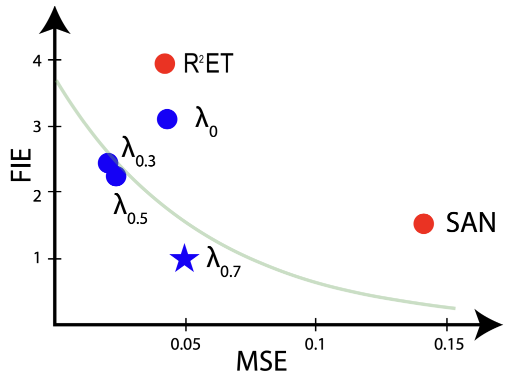

Fig. 9: Ablation studies for the λ value in the shape-aware module. The best results are indicated by points closer to the origin; we can see that our MoMa can outperform both shape-aware (R2ET) and skeleton-aware (SAN) methods depending on the value of λ. We obtain our best result with $\lambda = 0.7$.

In the first row, we show how other methods such as copy-rotations and SAN cannot fully solve the collisions between the hands. R^{2}ET solves the collisions, but spreads the hands far apart from each other, losing the original dynamics of the motion. Our shape-aware module is able to refine the retargeting of the skeleton-aware module and avoid interpenetrations between limbs, resulting in a collision-free motion, while preserving the original poses.

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. 7, we show how, being shape-aware, 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 position of the arm, also visible in the source taken from Mixamo.

Non-homeomorphic skeletons. In Fig. 8, we show how we can retarget from a source skeleton with $N_k = 22$ to skeletons Q_{t1} with $N_{t1} = 26$ and, Q_{t2} with $N_{t2} = 25$. Both Q_{t1} and Q_{t2} present extra end-effectors in the tail, which are animated by the retargeting. Also, having different ”arms” configurations, it

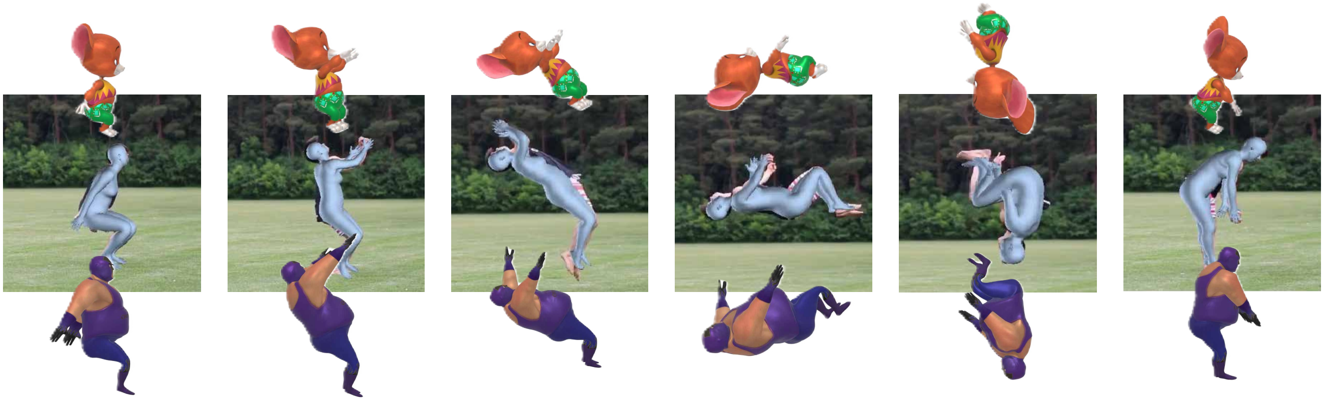

Fig. 10: Our MoMa approach is able to retarget from real world character, which motion can be modeled by a common SMPL mesh, even with challenging and unseen motion such as backflip.

is interesting to notice how our approach can adapt to different ranges of motion during the retargeting process.

Retargeting from real-world character. To demonstrate the robustness and the generalization ability of MoMa, we show some results from real-world complex motions taken from the Skills from Video (SFV) data (Peng et al. (2018)) in Fig. 10. This unlock the possibility of transferring to simulated characters virtually any human motion video processed by the common SMPL mesh (Kocabas et al. (2020)). Moreover, this demonstrates the ability of MoMa to generalise at test time to unseen skeletons that can be closely represented by one of the skeletons in the training set (e.g., similar body proportions and number of joints) or by a combination of multiple skeletons from the training set (e.g., maintaining the same number of joints and matching body proportions as a combination of multiple skeletons). Therefore, if the training set is sufficiently diverse, we can generalize to nearly all isomorphic and homeomorphic skeletons, but not to all possible non-homeomorphic ones.

4.3. Ablation studies

We conduct the ablation studies using the isomorphic skeletons setup.

Skeleton-aware module. In Tab. 3, we show the ablation studies for our skeleton aware module. 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 of 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 is able to model the relationships across both space and time.

Shape-aware module. In Fig. 9, we report how our method performs by varying the λ parameter in the optimizer. Since in the ideal case, one would like to achieve 0 in both MSE and FIE, the closest a method is to the origin, the better. The parameter λ allows to choose the preferred configuration, balancing between collision avoidance and preserving the original skeletal motion. Regardless of the value of λ, our method always performs better than the state-of-the-art solutions.

Computational cost of the optimization step. In the Table

Time of execution (seconds)		Avg. number of collisions per frame	Length of the animation (frames)
SGD	**L-BFGS**		
144.2	73.1	522	56
345.3	180.39	450	144

Table 5: Computational cost of the optimization step.

5, we present the performance comparison of different optimizers, namely SGD and L-BFG,S for two Big Vegas animations of different lengths. The mesh's joint and face numbers have minimal impact on performance, as computation load is primarily determined by the collisions to be resolved per frame.

5. Conclusions

In this work we presented MoMa, a novel approach for skeleton-aware and shape-aware skinned motion retargeting. To our knowledge, MoMa 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.

References

, . Cmu graphics lab motion capture database. http://mocap.cs.cmu.edu/ .

Aberman, K., Li, P., Lischinski, D., Sorkine-Hornung, O., Cohen-Or, D., Chen, B., 2020. Skeleton-aware networks for deep motion retargeting. ACM Transactions on Graphics (TOG) 39, 62–1.

Adobe, 2020. Mixamo .

Bao, H., Dong, L., Wei, F., 2021. Beit: Bert pre-training of image transformers. arXiv preprint arXiv:2106.08254 .

Chan, C., Ginosar, S., Zhou, T., Efros, A.A., 2019. Everybody dance now, in: Proceedings of the IEEE/CVF international conference on computer vision, pp. 5933–5942.

Chen, H., Tang, H., Shi, H., Peng, W., Sebe, N., Zhao, G., 2021. Intrinsic-extrinsic preserved gans for unsupervised 3d pose transfer, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 8630–8639.

Chen, M., Radford, A., Child, R., Wu, J., Jun, H., Luan, D., Sutskever, I., 2020. Generative pretraining from pixels, in: International conference on machine learning, PMLR. pp. 1691–1703.

Devlin, J., Chang, M.W., Lee, K., Toutanova, K., 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 .

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

Feichtenhofer, C., Fan, H., Li, Y., He, K., 2022. Masked autoencoders as spatiotemporal learners. arXiv preprint arXiv:2205.09113 .

Harvey, F.G., Yurick, M., Nowrouzezahrai, D., Pal, C., 2020. Robust motion in-betweening. ACM Transactions on Graphics (TOG) 39, 60–1.

He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R., 2022. Masked autoencoders are scalable vision learners, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 16000–16009.

Hecker, C., Raabe, B., Enslow, R.W., DeWeese, J., Maynard, J., van Prooijen, K., 2008. Real-time motion retargeting to highly varied user-created morphologies. ACM Transactions on Graphics (TOG) 27, 1–11.

Karras, T., 2012. Maximizing parallelism in the construction of bvhs, octrees, and k-d trees, in: Proceedings of the Fourth ACM SIGGRAPH/Eurographics conference on High-Performance Graphics, pp. 33–37.

Kavan, L., 2014. Part i: direct skinning methods and deformation primitives, in: ACM SIGGRAPH, pp. 1–11.

Kocabas, M., Athanasiou, N., Black, M.J., 2020. 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.

Lee, S., Kang, T., Park, J., Lee, J., Won, J., 2023. Same: Skeleton-agnostic motion embedding for character animation, in: SIGGRAPH Asia 2023 Conference Papers, pp. 1–11.

Lim, J., Chang, H.J., Choi, J.Y., 2019. Pmnet: Learning of disentangled pose and movement for unsupervised motion retargeting., in: BMVC, p. 7.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V., 2019. Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692 .

Nocedal, J., Wright, S.J., 1999. Numerical optimization. Springer.

Pavlakos, G., Choutas, V., Ghorbani, N., Bolkart, T., Osman, A.A.A., Tzionas, D., Black, M.J., 2019. 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.

Peng, X.B., Kanazawa, A., Malik, J., Abbeel, P., Levine, S., 2018. Sfv: Reinforcement learning of physical skills from videos. ACM Transactions On Graphics (TOG) 37, 1–14.

Regateiro, J., Boyer, E., 2022. Temporal shape transfer network for 3d human motion, in: 2022 International Conference on 3D Vision (3DV), IEEE. pp. 424–432.

Seol, Y., O'Sullivan, C., Lee, J., 2013. Creature features: online motion puppetry for non-human characters, in: Proceedings of the 12th ACM SIGGRAPH Symposium on Computer Animation, pp. 213–221.

Tak, S., Ko, H.S., 2005. A physically-based motion retargeting filter. ACM

Transactions on Graphics (TOG) 24, 98–117.

Tong, Z., Song, Y., Wang, J., Wang, L., 2022. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. arXiv preprint arXiv:2203.12602 .

Tzionas, D., Ballan, L., Srikantha, A., Aponte, P., Pollefeys, M., Gall, J., 2016. Capturing hands in action using discriminative salient points and physics simulation. International Journal of Computer Vision 118, 172–193.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I., 2017. Attention is all you need. Advances in neural information processing systems 30.

Villegas, R., Ceylan, D., Hertzmann, A., Yang, J., Saito, J., 2021. Contact-aware retargeting of skinned motion, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 9720–9729.

Villegas, R., Yang, J., Ceylan, D., Lee, H., 2018. Neural kinematic networks for unsupervised motion retargetting, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 8639–8648.

Xie, Z., Zhang, Z., Cao, Y., Lin, Y., Bao, J., Yao, Z., Dai, Q., Hu, H., 2022. Simmim: A simple framework for masked image modeling, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9653–9663.

Yamane, K., Ariki, Y., Hodgins, J., 2010. Animating non-humanoid characters with human motion data, in: Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pp. 169–178.

Yang, Z., Zhou, M., Shan, M., Wen, B., Xuan, Z., Hill, M., Bai, J., Qi, G.J., Wang, Y., 2024. Omnimotiongpt: Animal motion generation with limited data, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1249–1259.

Yang, Z., Zhu, W., Wu, W., Qian, C., Zhou, Q., Zhou, B., Loy, C.C., 2020. Transmomo: Invariance-driven unsupervised video motion retargeting, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5306–5315.

Zhang, H., Starke, S., Komura, T., Saito, J., 2018. Mode-adaptive neural networks for quadruped motion control. ACM Transactions on Graphics (TOG) 37, 1–11.

Zhang, J., Weng, J., Kang, D., Zhao, F., Huang, S., Zhe, X., Bao, L., Shan, Y., Wang, J., Tu, Z., 2023. 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.

Supplementary Material

6. Datasets splits used for Animations and Characters

In this section, we provide the full list of characters and animations used in training and testing for the Mixamo dataset. The list of animations is presented in a text file, with details on the training and testing characters provided below.

6.1. Training

The characters used for training are the following:

- **Isomorphic Characters**: Aj, BigVegas, Goblin, Kaya, Mousey, Warrok, PeasantMan.
- **Homemorphic Characters**: Aj, Malcolm_m, BigVegas, Kaya, SportyGranny, Remy_m, Maria_m, Knight_m, Liam_m, Parasite, Michelle_m, LolaB_m, Pumpkinhulk_m, Ortiz_m, Paladin_m, James_m, Joe_m, Olivia_m, Yaku_m, Timmy_m, Racer_m, Abe_m.
- **Extramorphic Characters**: Mewtwo, Penguin, Aj.

With *_m* we indicate the characters with 25 joints.

6.2. Testing

The characters used for testing are the following:

- **Isomorphic Characters**: Ortiz, SportyGranny, Aj, BigVegas, Goblin, Kaya, Mousey, Warrok, PeasantMan, XBot, Man, CastleGuard.
- **Homemorphic Characters**:Aj, BigVegas, Goblin_m, Kaya, Mousey_m, Mremireh_m, SportyGranny, Vampire_m, Mutant.
- **Extramorphic Characters**: Mewtwo, Penguin, Xbot.

With *_m* we indicate the characters with 25 joints.

7. Qualitative Ablation of lambda values

Our shape optimizer efficiently resolves collisions while preserving the character's original pose semantics. Figure 11 illustrates the impact of varying the lambda parameter during

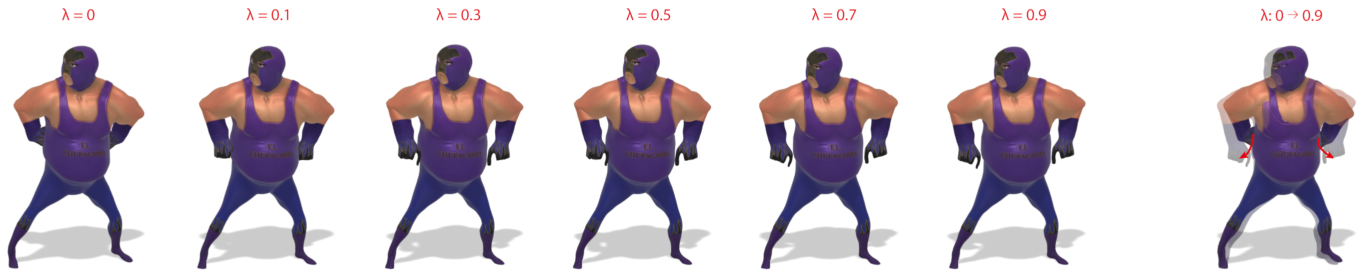

Fig. 11: Qualitative Ablation of λ values. Higher λ values correspond to a lower number of collisions but higher deviation from the original pose.

the optimization results. Notably, higher lambda values yield reduced collisions at the expense of increased deviation from the original pose. Our experiments show that setting $\lambda = 0.7$ leads to the best trade-off between collision resolution and pose fidelity in terms of both FIE and SCE. However, since the retargeting style largely depends on the animators' artistic preferences, our method allows the user to tailor the λ parameter based on the specific characteristics of the motion sequence and the character being retargeted. This additional degree of flexibility aligns with the creative nuances inherent in the artistic dimension of motion retargeting.

8. Linear Blend Skinning (LBS)

The Linear Blend Skinning (LBS) (Kavan (2014)) is defined as:

$$v' = \sum_{n=1}^{N_t} \omega_n T(j_n) v \tag{5}$$

where ω_n are the weights that define the influence of joint j_n on vertex v, and $T(j_n)$ are a set of matrices that define the spatial transformations to align the T-pose of joint j_n with its current (animated) pose. Please refer to Kavan (2014) for further mathematical details.