EUROGRAPHICS 2025 / A. Bousseau and A. Dai
(Guest Editors)

COMPUTER GRAPHICS *forum*
Volume 44 (2025), Number 2

ASMR: Adaptive Skeleton-Mesh Rigging and Skinning via 2D Generative Prior

Seokhyeon Hong* Soojin Choi* Chaelin Kim Sihun Cha Junyong Noh

Visual Media Lab, KAIST

Abstract

Despite the growing accessibility of skeletal motion data, integrating it for animating character meshes remains challenging due to diverse configurations of both skeletons and meshes. Specifically, the body scale and bone lengths of the skeleton should be adjusted in accordance with the size and proportions of the mesh, ensuring that all joints are accurately positioned within the character mesh. Furthermore, defining skinning weights is complicated by variations in skeletal configurations, such as the number of joints and their hierarchy, as well as differences in mesh configurations, including their connectivity and shapes. While existing approaches have made efforts to automate this process, they hardly address the variations in both skeletal and mesh configurations. In this paper, we present a novel method for the automatic rigging and skinning of character meshes using skeletal motion data, accommodating arbitrary configurations of both meshes and skeletons. The proposed method predicts the optimal skeleton aligned with the size and proportion of the mesh as well as defines skinning weights for various mesh-skeleton configurations, without requiring explicit supervision tailored to each of them. By incorporating Diffusion 3D Features (Diff3F) as semantic descriptors of character meshes, our method achieves robust generalization across different configurations. To assess the performance of our method in comparison to existing approaches, we conducted comprehensive evaluations encompassing both quantitative and qualitative analyses, specifically examining the predicted skeletons, skinning weights, and deformation quality.

CCS Concepts

• ***Computing methodologies*** → ***Animation; Mesh geometry models;***

1. Introduction

Skeletal data has long been a fundamental representation for character animation. While capturing the skeletal animation was traditionally challenging due to its reliance on costly methods like manual keyframing and motion capture, recent advancements have simplified this process. For example, high-fidelity human motion can now be captured with a minimal setup using video-based 3D pose estimation [PFGA19, ZZM*21, SKHB24, RBH*21] or sensor-based motion capture systems [YZX24, PYA*23, HKA*18]. Additionally, the growing availability of large public motion capture datasets [CMU, KHH*17, SZKS19, HYNP20, MSK22] has made high-quality motion data easily accessible, significantly reducing production costs.

Despite the increased accessibility of skeletal motion data, integrating it to various character meshes remains a significant challenge. To animate a character mesh using skeletal data, the mesh must first undergo a process called rigging, which involves aligning the skeleton with the mesh and defining skinning weights that determine how each vertex of the mesh is influenced by the movement of the underlying joints. Because the motion capture data typically has varying body scales and bone lengths, it is necessary to adjust the skeleton to match the proportions of the mesh, followed by retargeting the original sequence to the adjusted skeleton. Otherwise, unexpected deformation can be made due to joints not being aligned with the mesh, as exemplified by a case when a shoulder joint is located outside the mesh or embedded inside the torso. Furthermore, skeletons and meshes each possesses distinct configurations: skeletons vary in terms of bone length, the number of joints and their hierarchical connectivity, whereas meshes exhibit different vertex connectivity and body volume. These discrepancies and the inherent inhomogeneity between skeleton and mesh data necessitate a careful designation of skinning weights tailored to the specific configurations of both the skeleton and mesh. This complexity underscores the need for an approach that can automatically animate character meshes using existing skeletal data. For clarification, we define **skeleton configuration** as encompassing (i) **shape**, referring to geometric properties such as body scale and bone lengths and (ii) **structure**, which includes the number of joints and their hierarchy. Similarly, **mesh configuration** is defined

* Equal contribution

as (i) **shape**, representing the body proportions and volume and (ii) **structure**, denoting the vertex connectivity.

To address these challenges, auto-rigging methods have been developed to automatically estimate the skeleton and skinning weights that are aligned to a given character mesh. For instance, some approaches predict skinning weights assuming that a skeleton matching the mesh in size and proportions is already provided [MMRH22, WMLL24], while others allow the skeleton to differ in size and proportion from the mesh but limit the number of joints and their connectivity to a pre-defined template [LAH*21]. Yet other approaches derive both the skeleton and skinning weights directly from the mesh, allowing skeletons with varying numbers of joints and connectivity, though with restricted flexibility as they do not allow explicit specification of the exact number of joints or their hierarchical structures [XZK*20]. Pinocchio [BP07] facilitates the use of desired skeletons with varying numbers of joints for animating the character mesh; however, this approach assumes that the input mesh has proportions approximately similar to those of the given skeleton. While the aforementioned approaches have demonstrated the capability of producing promising results, they all fall short in simultaneously addressing variations in both skeletal and mesh configurations, as summarized in Figure 1.

In this work, we propose a novel method for automatic rigging and skinning of character meshes to animate them using skeletal motion data. Our approach is designed to accommodate arbitrary configurations for both the mesh and skeleton, operating through two key modules: **Skeletal Articulation Prediction** and **Skinning Weight Prediction**. The Skeletal Articulation Prediction module first identifies the optimal target skeletal articulation, ensuring that it has the same number of joints and hierarchy as the source skeleton. By employing an attention mechanism to capture the relationships between the input mesh and skeleton, this module is trained to predict a target skeleton that aligns with the mesh while preserving the original joint connectivity. Subsequently, the source skeletal motion is retargeted to the target skeleton, generating the motion that drives the movement of the character mesh. To address the challenge of predicting skinning weights for diverse mesh and skeleton configurations, we introduce a Skinning Weight Prediction module. This module implicitly learns skinning weights from deformed meshes, avoiding the necessity for explicit ground truth skinning weights tailored to each specific mesh and skeleton configuration. Finally, using the predicted skinning weights, the character mesh is animated along with the target skeletal data through Linear Blend Skinning (LBS).

The acquisition of extensive 3D character datasets that adequately encompass the full range of mesh variations presents a significant challenge. Additionally, geometric features of character meshes, such as vertex positions and normals, do not fully capture the semantic information of each vertex, as the same vertex position of different characters could correspond to different body parts depending on their proportions. To address this limitation, we leverage Diffusion 3D Features (Diff3F) [LDP*24], which are semantic descriptors that are derived from foundational vision models pre-trained on large-scale datasets. By incorporating the diffusion-based features that establish consistent semantic correspondences across diverse input meshes, our method effectively associates vertices with skeletal joints, even in unseen mesh configurations. This enables our method to achieve robust generalization across a wide range of character meshes.

	Skeleton		Mesh	
	Shape	Structure	Shape	Structure
Pinocchio	✗	✓	✗	✓
RigNet	N/A	N/A	✓	✓
NBS	✓	✗	✗	✓
Ours	✓	✓	✓	✓

Figure 1: *Comparison of the robustness of different auto-rigging and skinning approaches to variations in input configurations. While each approach has limited robustness or is not applicable in at least one component, the proposed method achieves robustness across all components.*

Our technical contributions can be summarized as follows:

- We propose attention-based Skeletal Articulation Prediction and Skinning Weight Prediction modules that capture the interrelationships between an arbitrary number of skeleton joints and mesh vertices, enabling rigging and skinning character meshes from skeletons having arbitrary configurations.
- We present a self-supervised learning approach that facilitates the implicit learning of skinning weights for diverse mesh-skeleton configurations, eliminating the need for the ground truth skinning weights tailored to each configurations.
- We propose a novel method that leverages an image generative feature as a semantic prior for the task of auto-rigging and skinning character meshes with skeletal data, enhancing the ability to generalize to unseen 3D character meshes.

2. Related Work

2.1. Mesh Deformation with Skeletal Articulations

In computer graphics, deforming a mesh according to a given skeletal animation has long been a significant challenge. Various skinning-based approaches have been developed to tackle this problem. LBS [MLT88] is one of the most widely used techniques due to its simplicity and computational efficiency. Dual Quaternion Skinning (DQS) [Hej04, KCŽO07, LD14] better preserves rotational properties, leading to smoother deformations. Spherical-based skinning [KŽ05] further extends these concepts by leveraging geometric properties to enhance deformation quality. Multi-linear techniques [WP02, MMG06] address non-linear deformations with improved accuracy and effectively reduce artifacts like volume loss. To animate a character mesh using these skinning approaches, the mesh must first be rigged to have an aligned skeleton and corresponding skinning weights for each vertex.

To alleviate the burden involved in rigging and skinning, which often requires expertise, auto-rigging techniques have been proposed. For human-like biped characters, 3D animation applications provide auto-rigging tools that fit a template skeleton to a target character mesh [Aut18, mix]. These tools often require users to

manually specify the positions of joints or key points, which can be cumbersome. Pinocchio [BP07] is the first research to automate the entire rigging and skinning process. It first performs fitting a user-specified skeleton to a character mesh by iteratively contracting the mesh until it converges to a skeleton-like graph, and then calculates skinning weights for each vertex using a heat diffusion model. Subsequent research has expanded to handle character models in a part-wise manner [MAF10, SSW*10] or to accommodate more complex character models with varying proportions or morphologies [BTST12]. FAKIR [FCD20] proposed an iterative algorithm for skeleton registration that reveals the anatomy and pose given raw points, including scanned statues of clothed humans, animals, or hybrid figures. Upon these foundational studies, Neural Blend Shape (NBS) [LAH*21] was introduced as a data-driven deep learning approach. It utilizes a neural network to generate rigged and posed meshes by predicting joint offsets of the template skeleton and pose-dependent blendshapes to enhance the quality of the animated mesh. While NBS offers a flexible and adaptive rigging solution, its effectiveness is constrained by its reliance on a pre-defined skeleton structure and meshes that are closely aligned with the SMPL distribution. Estimating additional bones [MZ23] and separating the skinning and deformation of clothing from the body [WMLL24] improved the mesh deformation quality for highly complex character meshes with apparels and accessories. These methods are designed to work with a fixed template skeleton.

In contrast to approaches that require a specific target template skeleton, some research has focused on estimating the skeleton directly from the geometry data itself. Early methods in this domain often relied on geometric techniques, such as Laplacian contraction methods, to derive a skeleton [ATC*08, CTO*10]. RigMesh [BJD*12] segments the model, generates a skeleton for each segment, and integrates the segmented components into a cohesive structure. More recently, advances in deep learning have enabled predicting skeleton directly from 3D geometry using deep neural networks. RigNet [XZK*20], which employs Graph Neural Network (GNN), provides a robust solution for complex and diverse character models. MoRig [XZYK22] built upon this by incorporating motion features extracted from point cloud animations, enhancing skeleton estimation performance and enabling effective animation retargeting. While these approaches focused on generating random skeletal structures, our method leverages skeleton structures as input, to provide users with control over the generated structure. For this, we utilize attention maps to capture the inter-relationship between vertices and joints, allowing for the flexibility of handling arbitrary numbers of mesh vertices and skeleton joints. Additionally, we leverage generative features obtained from large-scale image models as semantic priors for each vertex, incorporating this semantic information when estimating skinning weights and joint offsets.

2.2. Mesh Deformation Transfer

Another branch of research focuses on animating bipedal character meshes without relying on skeletons, instead directly transferring the deformation from the source to the target mesh by assuming consistent mesh connectivity between the source and target meshes [GFK*18, TGLX18, GYQ*18]. More advanced approaches enable retargeting between character meshes with different connectivities by utilizing local deformations, such as per-triangle Jacobian [AGK*22], implicit skinning weights [LYS*22, WHZ*23], and per-vertex displacement [WLL*23]. While these studies enable the animating and retargeting of unrigged character meshes, the learned latent spaces that represent character poses and deformation are not interpretable, making it difficult to manipulate or adjust the deformed results. To address this, we aim to leverage the intuitive and editable nature of skeleton representations and skinning weights. By estimating the skeleton and skinning weights for the given mesh, we enable both intuitive editing and effective pose transfer.

2.3. Skeleton-based Motion Retargeting

To reuse existing skeletal motion data for various characters, the motion retargeting technique, which involves adapting the original sequence to fit different skeletons, has been extensively studied. Early work focused on retargeting motions to target skeletons having different bone lengths and proportions through nonlinear optimization with a set of constraints to preserve essential characteristics involved in the source motion [Gle98, CK00, LS99, SLSG01]. With the increased availability of skeletal motion data, deep learning-based methods have addressed this with a data-driven approach, making desired joint transformations to be predicted by neural networks [VYCL18, LCC19, VCH*21, ZWK*23]. Because skeletons can differ not only in body proportions but also in the number of joints and their hierarchy, more advanced neural network architectures have also been proposed to handle a wider range of skeletal configurations. By embedding skeletal motion from different skeletons into a shared latent space, Skeleton-aware networks (SAN) [ALL*20] enabled motion transfer between skeletons that are topologically equivalent but with different numbers of joints. While this work requires a separate network to be trained for each skeletal configuration, SAME [LKP*23] constructed a skeleton-agnostic motion embedding using an autoencoder based on graph convolution networks (GCN), enabling arbitrary skeletons to be processed within a single network. While these studies effectively predict joint transformations for target skeletons to best preserve the source skeletal motion, animating the character meshes with the retargeted results requires aligning the target skeleton to the mesh and predicting corresponding skinning weights. To address this problem, we propose an end-to-end framework that generates proper skeletal articulation and skinning weights of the input mesh simultaneously. Once the articulation is predicted, we transfer the source skeletal motion data to the predicted articulation using a pre-trained SAME [LKP*23]. Subsequently, we apply the retargeted skeletal motion to the character mesh through the predicted skinning weights, allowing the character mesh to be directly animated by the given skeletal motion data.

3. Method

Our goal is to animate a bipedal mesh using available skeletal motion data, with a focus on flexibility and generalization. To achieve this, we accommodate variations in skeletal configurations including differences in body scale, bone lengths, the number of joints,

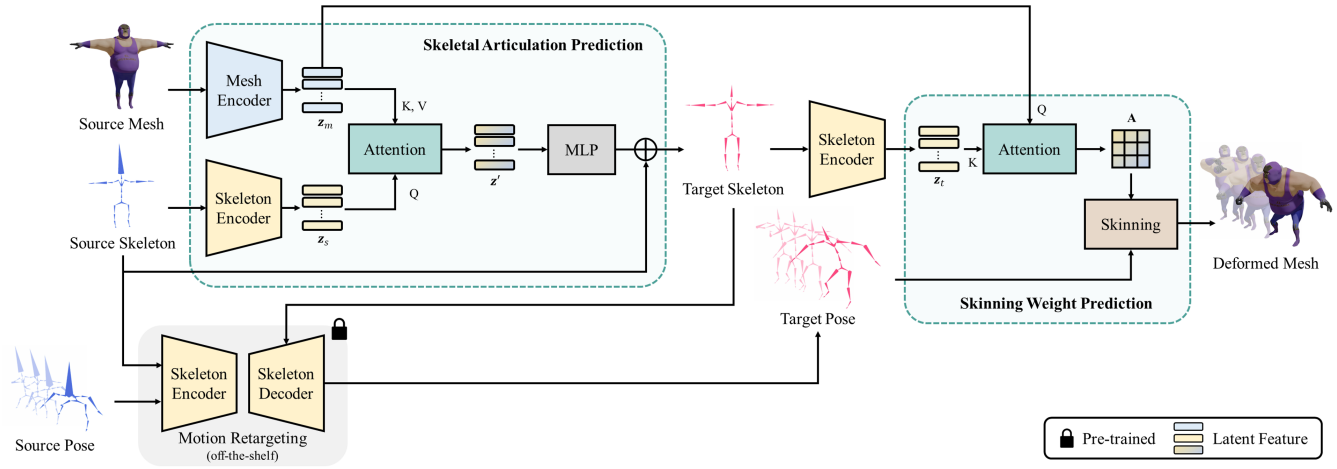

Figure 2: *Overview of our method. Given a source mesh to be animated and a source skeletal motion to derive its movement, our method predicts the target skeleton and the corresponding skinning weights that generate plausible deformation of the character mesh in accordance with the source skeletal movement. To accommodate source skeletons with arbitrary structures, we leverage an off-the-shelf retargeting module that aligns the target skeleton to the source pose, generating the target pose. Finally, the target pose, combined with the predicted skinning weights, is used to deform the character mesh. While our method does not rely on textural information, textures on the character meshes are included only to illustrate different poses.*

and their hierarchies as well as meshes with varying connectivity and shapes. As shown in Figure 2, our approach consists of two main components: **Skeletal Articulation Prediction** and **Skinning Weight Prediction**. The Skeletal Articulation Prediction module predicts a target skeleton that aligns with the input mesh in terms of the sizes and body proportions while preserving the number of joints and their hierarchy of the input skeleton. Note that the source mesh and source skeleton are consistently provided in a T-pose as a reference for calculating the posed state to ensure proper alignment for mesh deformation using LBS across all samples. In this stage, we leverage a 2D generative prior to obtain 3D deep features given the input mesh to improve the understanding of the model in terms of the semantic alignment between mesh and skeleton, resulting in generalizability to unseen shapes of character mesh. Subsequently, the Skinning Weight Prediction module estimates the skinning weights of the input mesh given the target skeleton using an attention mechanism. Finally, we deform the character mesh by applying LBS based on these predicted skinning weights. In the following sections, we outline the data representation used in our approach (Section 3.1), followed by the architecture of Skeletal Articulation Prediction (Section 3.2), and Skinning Weight Prediction (Section 3.3), and finally, the training process with data preparation (Section 3.4).

3.1. Data Representation

This section outlines the input data representation used in our framework, which remains consistent across both the training and inference phases. For each character model, we construct a skeletal motion data $M = (S, D^{1:N_T})$, where S denotes the skeleton, $D^{1:N_T}$ represents the motion data, and N_T is the number of frames. Following Lee et al. [LKP*23], S is represented as follows:

$$S = \{\mathbf{g}_{1:N_J}, \mathbf{o}_{1:N_J}\}, \tag{1}$$

where N_J is the number of joints. Each skeleton consists of $\mathbf{g}_{1:N_J}$ and $\mathbf{o}_{1:N_J}$, where $\mathbf{g}_j \in \mathbb{R}^3$ and $\mathbf{o}_j \in \mathbb{R}^3$ denote the global joint position at the rest pose and local joint position relative to its parent joint coordinate system, respectively. D^t includes skeletal dynamics information at frame t. For more details on its elements and their derivation, please refer to Appendix A.

The character mesh data G is represented as follows:

$$G = \{\mathbf{V}_r, \mathbf{V}_d^{1:N_T}, \mathbf{V}_f\}. \tag{2}$$

Here, $\mathbf{V}_r \in \mathbb{R}^{N_V \times 3}$ represents the vertex positions of the character mesh in the rest pose, where N_V denotes the number of vertices. We then compute the deformed vertex positions $\mathbf{V}_d^{1:N_T}$ over timestep 1 to N_T where $\mathbf{V}_d^t \in \mathbb{R}^{N_V \times 3}$ is obtained by deforming $\mathbf{V}_r$ using D^t through the LBS formulation. Because D^t is expressed with respect to the facing frame of the character, both $\mathbf{V}_r$ and $\mathbf{V}_d^t$ are expressed relative to the facing frame of the character mesh as well.

To obtain 3D diffusion features $\mathbf{V}_f$, we follow the pipeline of Diff3F [DMM24]. Specifically, the character mesh is rendered using multi-view cameras, and we extract the diffusion features [TJW*23] and DINO features [ODM*23]. These features are fused and projected back onto the surface of the character mesh by aggregating the features of multiple views into one for each vertex, resulting in a 3D deep feature representation. By leveraging the large-scale image foundational models, the learned implicit features carry dense and accurate semantic priors [ZHH*24,LDP*24,HSM*24]. These high-level representations at each vertex contribute to significant improvements in both training efficiency and the overall performance of our method.

3.2. Skeletal Articulation Prediction

Mesh Encoder The mesh encoder aims to estimate meaningful features from the input mesh, which effectively captures its correla-

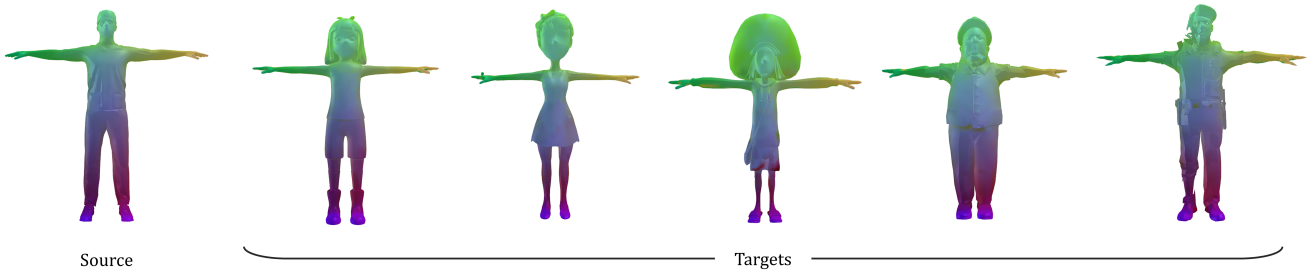

Figure 3: *Visualizations of the vertex correspondences between characters using Diff3F, where corresponding points are similarly colored. The source character is shown on the left, and the target characters are on the right.*

tion with a given skeleton, while being robust to variations in vertex connectivity and shapes. To achieve this, we employ the PointNet architecture [QSMG17] for the mesh encoder, which processes the vertex positions at the rest pose $\mathbf{V}_r$ along with Diff3F $\mathbf{V}_f$, generating mesh latent features $\mathbf{z}_m \in \mathbb{R}^{N_V \times N_D}$, where N_D denotes the dimension of the latent vector. Specifically, PointNet captures both local and global mesh features by combining information from individual vertices with that of the entire mesh, leading to a comprehensive understanding of the input mesh. Additionally, this architecture is flexible and generalizable to various vertex connectivities, even enabling the aggregation of features from unconnected components like the body and eyeballs.

To enhance the capability of the mesh encoder when capturing the semantic information of each vertex, we incorporate Diff3F [DMM24] as an input, which provides accurate correspondences across 3D meshes with arbitrary connectivity by taking advantage of 2D generative prior. While incorporating a large-scale dataset of 3D character meshes could improve the capacity of the mesh encoder, acquiring such datasets is challenging. Instead, we leverage the generative priors of image diffusion models trained on large-scale image datasets, which embed rich semantic information, thereby circumventing the need for preparing such an extensive dataset. As shown in Figure 3, Diff3F effectively captures accurate part-wise correspondences across different characters, even in the presence of highly stylized meshes. By combining both geometric and semantic features, our approach ensures that the encoder can adapt to diverse mesh configurations while accurately reflecting their relationships to the skeleton.

Skeleton Encoder We represent the skeleton as a graph, where the joints correspond to nodes, and the bones, which are connections between them, are represented as edges. In our framework, the body scale, bone length, the number of joints, and their hierarchies can be arbitrary, leading to inhomogeneous graph structures. To handle these variations, we employ GCN that learns via message passing by exchanging and propagating node features through the graph edges. Specifically, we employ graph attention networks [VCC*17] that leverage an attention mechanism to update the node features based on their neighbors. This approach enables us to derive a latent representation of the source skeleton, denoted as $\mathbf{z}_s \in \mathbb{R}^{N_J \times N_D}$, which maintains the original graph structure of the input skeleton while encoding its features and connectivity.

Offset Residual Prediction To ensure the generation of consistent animation when applying skeletal motion data to a character mesh, it is essential to align the size and body proportions of the mesh and those of the skeleton. Mismatches between these two can lead to distortions and artifacts, such as undesired stretching or compression of the animated mesh. This issue is especially critical in our framework because we allow arbitrary skeletons as input, which means that the skeleton and mesh may have different sizes and proportions.

To address this, we predict an offset residual that repositions the joints of the input skeleton by adjusting its local offset while maintaining its hierarchy. This results in a skeletal articulation aligned with the mesh, which we refer to as the target skeleton. Specifically, to capture the relationship between the skeleton and mesh data, we employ a cross-attention mechanism that can model the interactions between different modalities. The cross-attention operation is defined as follows:

$$\mathbf{z}' = \text{softmax}\left(\frac{\mathbf{z}_s \mathbf{z}_m^\top}{\sqrt{N_D}}\right)\mathbf{z}_m, \tag{3}$$

where $\mathbf{z}_s$ and $\mathbf{z}_m$ represent the latent features of the source skeleton and mesh, respectively. The resulting latent feature $\mathbf{z}' \in \mathbb{R}^{N_J \times N_D}$ is then passed through a MLP to predict the residual of the local offset $\Delta\mathbf{o} \in \mathbb{R}^{N_J \times 3}$. To ensure the symmetry of the skeleton with respect to the lateral axis of the character, we update the residual of the local offset as follows:

$$\Delta\mathbf{o}_j = \frac{1}{2}\left(\Delta\mathbf{o}_j + \Delta\mathbf{o}_{\rho(j)} \odot [-1,1,1]\right), \tag{4}$$

where $\rho(j)$ denotes the index of the corresponding symmetrical joint of j-th joint and $\odot$ represents the element-wise multiplication. For example, if j refers to the left arm joint, $\rho(j)$ denotes the index of the corresponding right arm joint, while we define $\rho(j) = j$ for joints that do not have a symmetrical counterpart, such as spine joints. The local offsets of the target skeleton, $\mathbf{o}_{tgt} \in \mathbb{R}^{N_J \times 3}$, are then computed as follows:

$$\mathbf{o}_{tgt} = \mathbf{o}_{src} + \Delta\mathbf{o}, \tag{5}$$

where $\mathbf{o}_{src} \in \mathbb{R}^{N_J \times 3}$ represents the local offsets of the source skeleton. Finally, to ensure that the skeleton is grounded, we translate the target skeleton along the up-axis so that the toe joints rest exactly on the ground.

Motion Retargeting Given a source pose, source skeleton, predicted target skeleton, and skinning weights of the source mesh, we aim to deform the source mesh according to the target pose, which follows the source pose with the target skeleton. To obtain the target

pose $\hat{D}^t$, we employ a pre-trained SAME model [LKP*23], which can accommodate heterogeneous input skeletons. While training a separate module for motion retargeting would be possible, it may necessitate complex simultaneous training of motion retargeting, rigging, and skinning. Because our primary focus is specifically on automating the rigging and skinning processes, we rely on an off-the-shelf method for motion retargeting to provide posed skeletons to deform the mesh. During training, we freeze the weights of the SAME model, allowing our network to focus exclusively on optimizing the target skeleton prediction and mesh deformation, while the motion retargeting is handled by the pre-trained capabilities of the SAME model.

3.3. Skinning Weight Prediction

Skinning weights define how the movement of each joint influences the individual vertices of a mesh. Specifically, the deformation of each vertex is computed as a weighted sum of joint transformations relative to the rest pose. To ensure smooth and natural deformations, the skinning weights for each vertex must sum to one, resulting in a convex combination of joint transformations. To model this property within a neural network framework, we leverage the attention mechanism [VSP*17], whose principles align with skinning weights. The attention matrix, computed as the outer product of query and key vectors, encodes the relationships between components, which is analogous to how the skinning weight matrix captures the relationship between vertices and joints. Furthermore, the property that the relationships for each query component sum to one is also similar to the requirement that the skinning weights for each vertex sum to one.

We implement this by trattando the latent feature of the mesh, $\mathbf{z}_m$, as the query, and the latent feature of the target skeleton, $\mathbf{z}_t \in \mathbb{R}^{N_J \times N_D}$, as the key. Note that $\mathbf{z}_m$ is the same latent vector used in Section 3.2. The resulting attention matrix, $\mathbf{A} \in \mathbb{R}^{N_V \times N_J}$, is then computed as follows:

$$\mathbf{A} = \text{softmax}\left(\frac{\mathbf{z}_m \mathbf{z}_t^\top}{\sqrt{N_D}}\right). \tag{6}$$

We interpret the learned attention weights as implicit skinning weights, which we apply within the LBS formulation to deform the character mesh. Specifically, each vertex $\mathbf{V}_{r,i}$ is deformed as follows:

$$\begin{bmatrix} \hat{\mathbf{V}}_{d,i} \\ 1 \end{bmatrix} = \left(\sum_{j=1}^{N_J} \mathbf{A}_{i,j} \mathbf{T}_j\right) \begin{bmatrix} \mathbf{V}_{r,i} \\ 1 \end{bmatrix} \tag{7}$$

where $\mathbf{T} \in \mathbb{R}^{J \times 4 \times 4}$ is the joint transformations derived from $\hat{D}^t$, which are relative to the rest pose. By training the network modules to minimize the differences between the ground truth posed mesh $\mathbf{V}_d$ and the predicted deformed mesh $\hat{\mathbf{V}}_d$, the attention weights are optimized to function effectively as skinning weights.

3.4. Training

3.4.1. Dataset Preparation

To prepare the dataset, we construct a motion database for each character model, following the procedures outlined in Lee et al. [LKP*23]. Specifically, the motion database is represented as follows:

$$\mathcal{M} = \{M_1, M_2, \ldots, M_K\}, \tag{8}$$

where each motion clip $M_k = (S_k, D_k^{1:N_T})$ consists of a skeleton S_k, which has a unique configuration, and corresponding motion data $D_k^{1:N_T}$. Starting with the initial skeletal motion data M_1, represented by skeleton S_1, which matches the size and body proportions of the character mesh, we augment the skeleton by randomly modifying its configuration. This involves adding or removing joints and adjusting bone lengths and root height of S_1. This augmentation process results in a skeleton database:

$$\mathcal{S} = \{S_1, S_2, \ldots, S_K\}. \tag{9}$$

For each skeleton S_k in this database, we retarget the motion data $D_1^{1:T}$ using the off-the-shelf retargeting method provided by MotionBuilder [aut21]. This results in a set of motion clips with diverse skeleton configurations, forming the complete motion database $\mathcal{M}$. Finally, we combine the mesh data and the motion database, yielding the character dataset $C = (G, \mathcal{M})$.

This process is repeated for all character models, producing a collection of character datasets:

$$\mathcal{C} = \{C_1, C_2, ..., C_{N_C}\}, \tag{10}$$

where N_C is the number of character models. Because each character model has a distinct initial skeleton and mesh configurations, the number of vertices N_V and the number of joints N_J may differ between characters. However, N_T remains consistent across the entire dataset because we use the same motion clips for all characters.

3.4.2. Training Procedures

During training, we randomly sample a pose at frame t along with the corresponding geometry features, and we omit t for brevity in this section. For the motion and mesh data, we randomly sample the input source pose $M_{src} = (S_{src}, D_{src})$ and the source mesh $G_{src} = \{\mathbf{V}_r, \mathbf{V}_d, \mathbf{V}_f\}$. Notably, M_{src} and G_{src} do not need to originate from the same character model:

$$M_{src} \in C_i, \quad G_{src} \in C_j, \quad i, j \in \{1, 2, \ldots, N_C\}, \tag{11}$$

where M_{src} is randomly sampled from the motion database $\mathcal{M}$, which is part of the character dataset C_i. Given the sampled motion and mesh, our model predicts the deformed mesh vertices $\hat{\mathbf{V}}_d$ using Equation 7.

The objective terms to train the model are as follows:

$$\mathcal{L} = \mathcal{L}_{vtx} + \mathcal{L}_{edge} + \mathcal{L}_{skel} + \mathcal{L}_{sdf}. \tag{12}$$

la perdita di ricostruzione del vertice perdita di bordo
The vertex reconstruction loss $\mathcal{L}_{vtx}$ and edge loss $\mathcal{L}_{edge}$ measure the differences between the ground truth and predicted mesh in terms of deformed vertex positions and edges, respectively:

$$\mathcal{L}_{vtx} = \frac{1}{N_V} \sum_{i=1}^{N_V} \|\mathbf{V}_{d,i} - \hat{\mathbf{V}}_{d,i}\|^2, \tag{13}$$

$$\mathcal{L}_{edge} = \frac{1}{|\mathcal{E}|} \sum_{i,j \in \mathcal{E}} \|(\mathbf{V}_{d,i} - \mathbf{V}_{d,j}) - (\hat{\mathbf{V}}_{d,i} - \hat{\mathbf{V}}_{d,j})\|^2, \tag{14}$$

where $\mathcal{E}$ represents the set of edges connecting adjacent vertices

and $\hat{\mathbf{V}}_d$ represents the deformed vertices predicted by our model. The edge loss ensures that the predicted mesh preserves the local rigidity of the ground truth mesh by accurately reconstructing local deformations along the edges. The skeleton loss $\mathcal{L}_{skel}$ and signed distance function (SDF) loss $\mathcal{L}_{sdf}$ guide the learning of a target skeleton that aligns with the mesh, which are defined as follows:

$$\mathcal{L}_{skel} = \mathrm{CD}(\mathbf{g}_{gt}, \mathbf{g}_{tgt}), \tag{15}$$

$$\mathcal{L}_{sdf} = \mathrm{SDF}(\mathbf{V}_r, \mathbf{g}_{tgt}), \tag{16}$$

where $\mathbf{g}_{gt}$ and $\mathbf{g}_{tgt}$ represent the global offsets of the ground truth and predicted target skeleton, respectively. $\mathrm{CD}(\mathbf{g}_{gt}, \mathbf{g}_{tgt})$ computes the Chamfer distance between these two, which is defined as follows:

$$\mathrm{CD}(\mathbf{g}_{gt}, \mathbf{g}_{tgt}) = \frac{1}{|\mathbf{g}_{gt}|} \sum_{x \in \mathbf{g}_{gt}} \min_{y \in \mathbf{g}_{tgt}} \|x-y\|_2^2 + \frac{1}{|\mathbf{g}_{tgt}|} \sum_{y \in \mathbf{g}_{tgt}} \min_{x \in \mathbf{g}_{gt}} \|x-y\|_2^2. \tag{17}$$

The intuition behind using the Chamfer distance for the skeleton loss lies in its ability to compare the shapes of two different skeletons with varying numbers of joints by treating each joint as a point. Because the ground truth skeleton that fits the source mesh, which corresponds to S_1 of C_j, can possess a different number of joints from the target skeleton, we use the Chamfer distance to ensure that the predicted target skeleton has similar shape to the ground truth even when the number of joints and joint hierarchy are different. $\mathrm{SDF}(\mathbf{V}_r, \mathbf{g}_{tgt})$ computes the signed distance for each component of $\mathbf{g}_{tgt}$ with respect to the mesh at the rest pose $\mathbf{V}_r$. The SDF loss assigns negative values for joints inside the mesh and positive values for joints outside, encouraging the joints to be properly embedded within the character mesh.

One notable aspect of our approach is that we do not explicitly supervise the estimation of deformation parameters, such as skinning weights and bone offset residual factors. The reason for this is that the source skeletal structure S_{src} can be arbitrary, with varying numbers of joints and hierarchical connections between them. As a result, it is infeasible to prepare a set of skinning weights that generalize across arbitrary skeletal structures. Instead, our method employs a self-supervised learning strategy through the vertex reconstruction loss. This encourages the model to implicitly learn the correct deformation parameters because the reconstruction loss can only be minimized when these deformation parameters are accurately estimated.

4. Experiments

4.1. Implementation Details

To obtain the character mesh data G, we randomly selected 17 characters for training and 9 characters for testing from Mixamo [mix]. For the motion data M, we used 14 motion sequences from the LaFAN1 dataset [HYNP20], comprising a total of 75,351 frames sampled at 30fps. Using G and M, we generated the deformed mesh vertices $\mathbf{V}_d$ via LBS, utilizing the skinning weights embedded in each Mixamo character. Additionally, to enhance training efficiency, we decimated the mesh vertices to fewer than 5,000 for characters with a higher vertex count.

Our method was implemented in PyTorch and executed on a single NVIDIA RTX A5000 GPU with 24GB of VRAM. We used the Adam optimizer [Kin14], with a fixed learning rate of 10^{-4} and a weight decay of 10^{-6}. Each network module is followed by a batch normalization layer [Iof15] and the ReLU activation. The model was trained for 50 epochs with a batch size of 64 frames, which required approximately 30 minutes of computation per epoch.

4.2. Baselines for Comparison

We compared our method with other auto-rigging approaches, including Pinocchio [BP07], RigNet [XZK*20], and NBS [LAH*21]. We evaluated the performance of each method both qualitatively and quantitatively in terms of rigging, skinning, and mesh deformation. Pinocchio is the most relevant to our work because it generates plausible skeletal articulations and skinning weights based on the input skeleton and mesh. NBS also addresses a similar problem by generating skinning weights and mesh deformations given a mesh input, but it is limited by the fixed skeleton template. Consequently, we evaluated NBS only for skeletons that included all the joints corresponding to the predefined skeleton template. RigNet, a representative method for learning-based rig estimation, generates target skeletons for the given mesh without requiring skeleton inputs.

For evaluation, we utilized the augmented skeleton database of each character, which corresponds to $\mathcal{S}$ described in Equation 9, to validate the ability of our method to handle arbitrary skeletons and meshes. Each character was paired with its original skeleton S_1 and five additional augmented skeletons, denoted as $S_{2:6}$, which vary in number of joints and offset scales. To ensure a fair comparison across different methods, each of which has specific requirements for skeleton and mesh input, we took several steps to fully leverage this inhomogeneous skeleton dataset.

For Pinocchio, we pre-processed the character meshes to meet the requirement of being a single connected component, ensuring all vertices were linked within one graph. However, this process alters the number of vertices in the mesh, potentially affecting a fair comparison. To address this, we mapped the skinning weights predicted by Pinocchio back to the original mesh by finding the closest corresponding vertices. To deform the mesh with the predicted skeletal articulations and skinning weights, we utilized the SAME model to retarget the source motion to the predicted skeleton. RigNet does not allow for explicitly specifying the desired number of joints, but it can generate multiple skeletons from a single input mesh. Therefore, we randomly generated five different skeletons using RigNet for evaluation, addressing variability in shape and structure of the generated skeletons. To ensure a fair comparison, we evaluated NBS only on S_1, which is the skeleton configuration that meets the requirements of NBS across all characters, while we evaluated Pinocchio and our method on all skeleton variations, $S_{1:6}$. Throughout the evaluations of RigNet and NBS, we used the pre-trained model provided by the authors.

4.3. Evaluation

Rigging To evaluate rigging quality, we compared the similarity between the predicted and ground truth skeletons, which corresponds to S_1 for each character. The ground truth skeletons were derived from the character models obtained from Mixamo [mix].

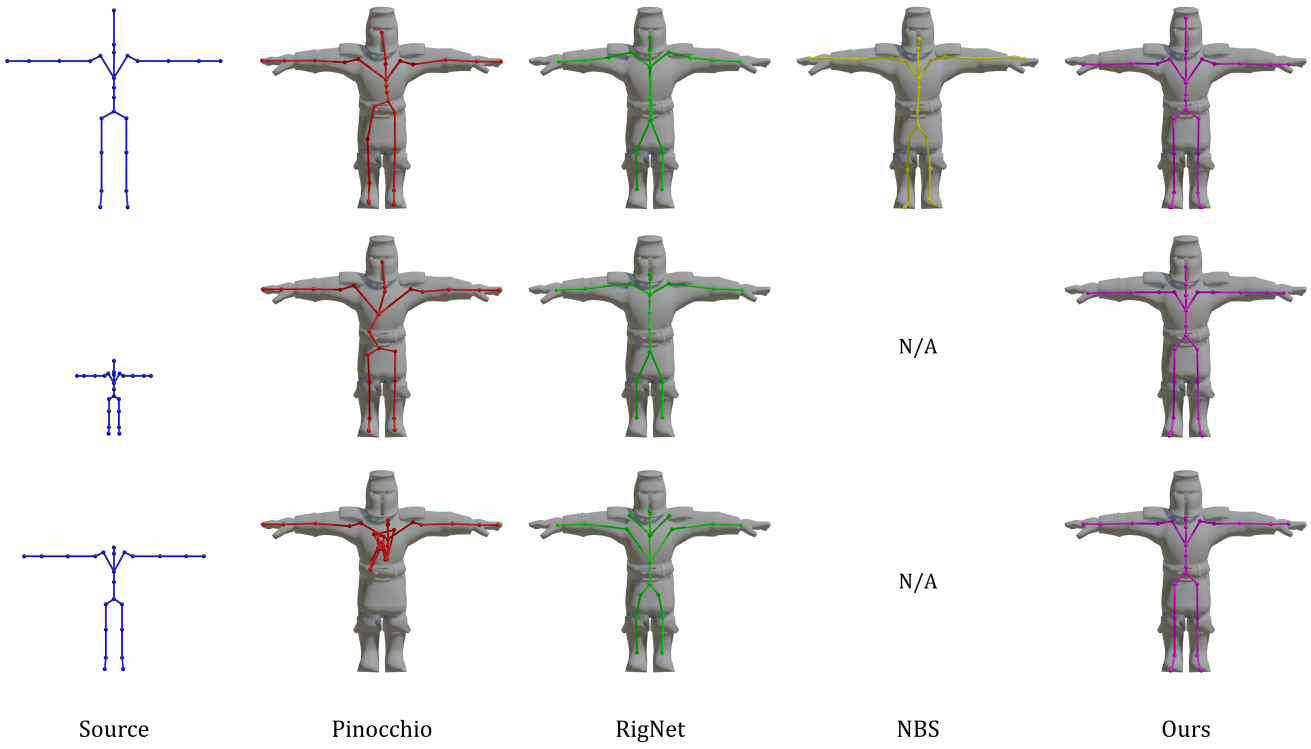

Figure 4: *Comparison to baselines on skeleton prediction results given the same mesh with different source skeletons. Each skeleton has distinct body scales and bone lengths, with varying numbers of joints: from the top 25, 24, and 23 joints.*

For evaluation, we employed three Chamfer distance-based metrics, making them suitable for comparing skeletons with different numbers of joints: CD-J2J (joint-to-joint), CD-J2B (joint-to-bone), and CD-B2B (bone-to-bone) [XZK*20]. These metrics quantify the spatial discrepancies between the predicted and ground truth skeletons, where lower values indicate a closer alignment between the predicted and ground truth skeletal shapes. For more details, please refer to RigNet [XZK*20].

Table 1 presents that our method yielded better results than the baselines across most metrics. While RigNet achieved slightly better results in CD-B2B than ours, the difference was marginal compared to the differences observed in other metrics. Moreover, considering that our approach enables greater flexibility in handling diverse skeleton and mesh configurations while maintaining the integrity of the given skeletal structure, these superior results are highly significant.

Figure 4 shows that our method outperformed the baselines with significantly higher flexibility in the number of joints and bone lengths. Pinocchio struggled to preserve structural integrity given discrepancies in size and proportion between the skeleton and mesh, leading to asymmetric joint predictions along the up-axis. RigNet faced difficulties in accurately identifying the pelvis joint, and introduced unnecessary joints in bulky regions, such as the shoulders, causing the skeleton to exhibit significant deviations from the source skeletal structure. While NBS successfully generated skeletons that fit the size and proportions of the mesh, it altered the source skeleton shape into predefined skeletal offsets, which resulted in deviations from the source. Furthermore, NBS was not operable when the source skeleton did not contain the joints defined in the predefined template, as indicated by the N/A in the last two rows of Figure 4. In contrast, our method successfully predicted the target skeleton, showing consistent rig predictions even in cases with substantial variations in size and proportion between the mesh and skeleton.

Table 1: *Quantitative results on rigging prediction. The best result for each column is in bold.*

	CD-J2J↓	CD-J2B↓	CD-B2B↓
Pinocchio	34.03	26.81	25.10
RigNet	18.88	13.01	**8.64**
NBS	20.15	15.70	11.92
Ours	**15.25**	**10.87**	9.07

Skinning and Deformation To evaluate the accuracy of skinning weight prediction, we measured the difference between predicted and ground truth skinning weights by computing the average L1 distance (Skinning L1) between them. Because the ground truth skinning weight is only available for S_1, we measured the L1 distance exclusively for S_1, while other metrics were measured across all skeleton variations $S_{1:6}$. Additionally, to evaluate the deformation quality driven by the predicted skinning weights, we measured the Chamfer distance (CD), average distance error (ADE), and max distance error (MDE) between the predicted and ground truth deformed meshes. These metrics were measured between vertices de-

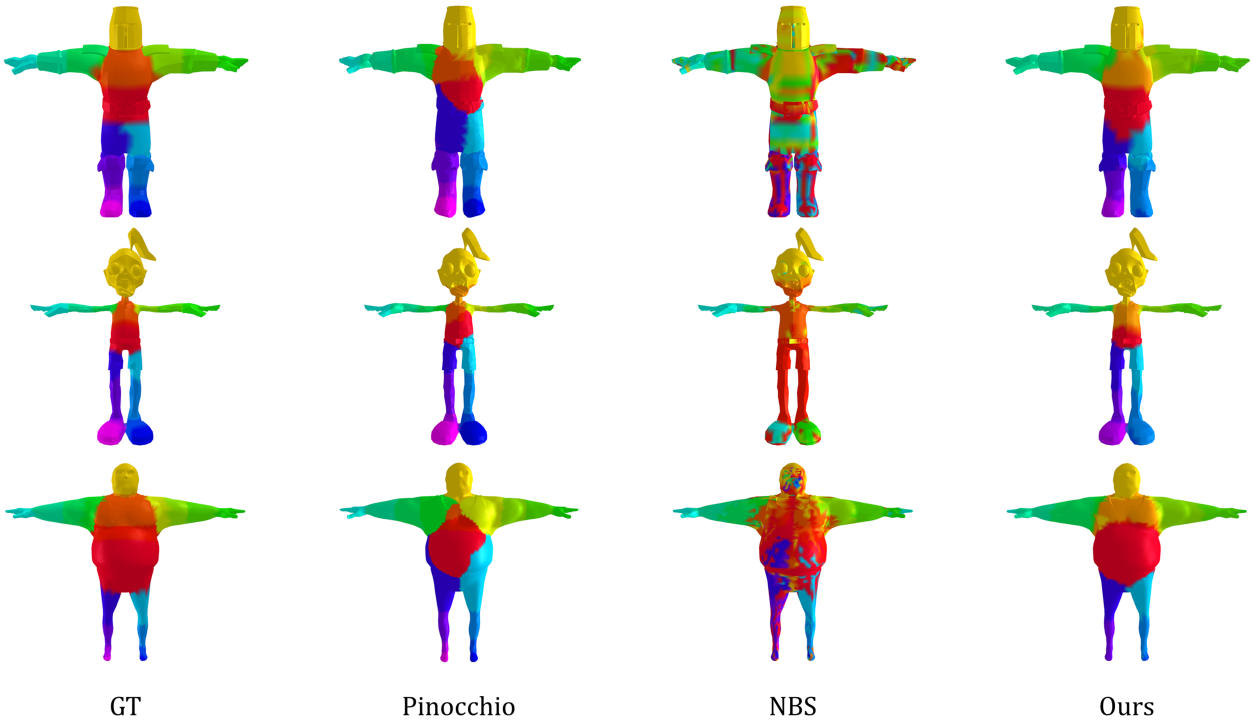

Figure 5: *Skinning weight results predicted from a source skeleton with a fixed number of joints.*

Table 2: *Quantitative results on skinning and deformation. The best result for each column is in bold.*

	Skinning L1↓	CD↓	ADE↓	MDE↓	ELS↑
Pinocchio	**0.0236**	15.14	16.78	52.68	**0.94**
NBS	0.0599	18.11	31.60	154.40	-5.65
Ours	0.0449	**7.15**	**8.09**	**26.18**	0.89

formed using the predicted skinning weights and those deformed with the ground truth weights. To further evaluate the smoothness of the deformed meshes, we measured Edge Length Score (ELS) [WLL*23]. This score compares the length of each edge in the predicted mesh to that of the corresponding edge in the ground truth mesh, which is calculated as follows:

$$\mathrm{ELS}(\mathbf{V}_d, \hat{\mathbf{V}}_d) = \frac{1}{|\mathcal{E}|} \sum_{\{i,j\}\sim\mathcal{E}} 1 - \left| \frac{\|\hat{\mathbf{V}}_{d,i} - \hat{\mathbf{V}}_{d,j}\|_2}{\|\mathbf{V}_{d,i} - \mathbf{V}_{d,j}\|_2} - 1 \right|, \quad (18)$$

where $\mathcal{E}$ denotes the entire edges of the mesh, $\hat{\mathbf{V}}_{d,i}$ and $\hat{\mathbf{V}}_{d,j}$ are the predicted vertices in the deformed mesh, and $\mathbf{V}_{d,i}$ and $\mathbf{V}_{d,j}$ are the vertices in the ground truth mesh.

Table 2 shows the quantitative results on skinning and deformation. In Skinning L1, Pinocchio outperformed ours. The suboptimal performance of ours can be attributed to the challenge of predicting accurate skinning weights for joints with minimal movements, such as the chest, especially without ground truth supervision during training. Specifically, these skinning weights are indirectly optimized through self-supervised learning, which can be less accurate

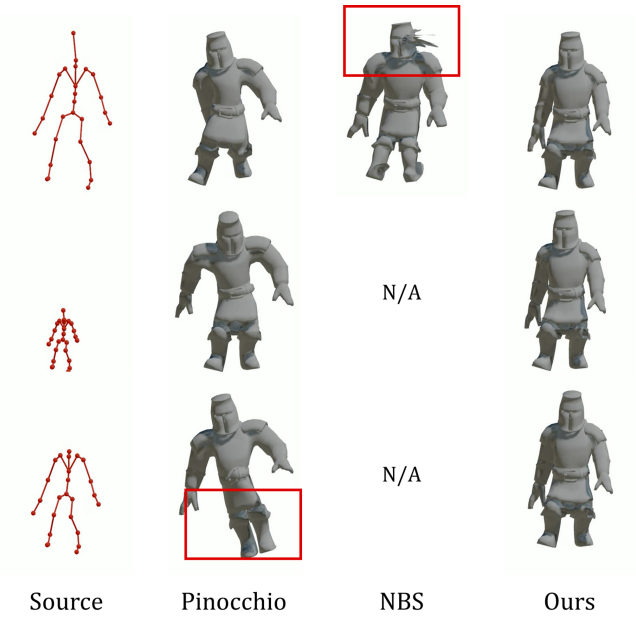

Figure 6: *Qualitative comparison with baselines on mesh deformation.*

when there are no significant vertex displacements associated with those joints. Nonetheless, our method consistently produced visually plausible skinning weights that aligned well with the relevant body parts, as shown in Figure 5. While Pinocchio produced rea-

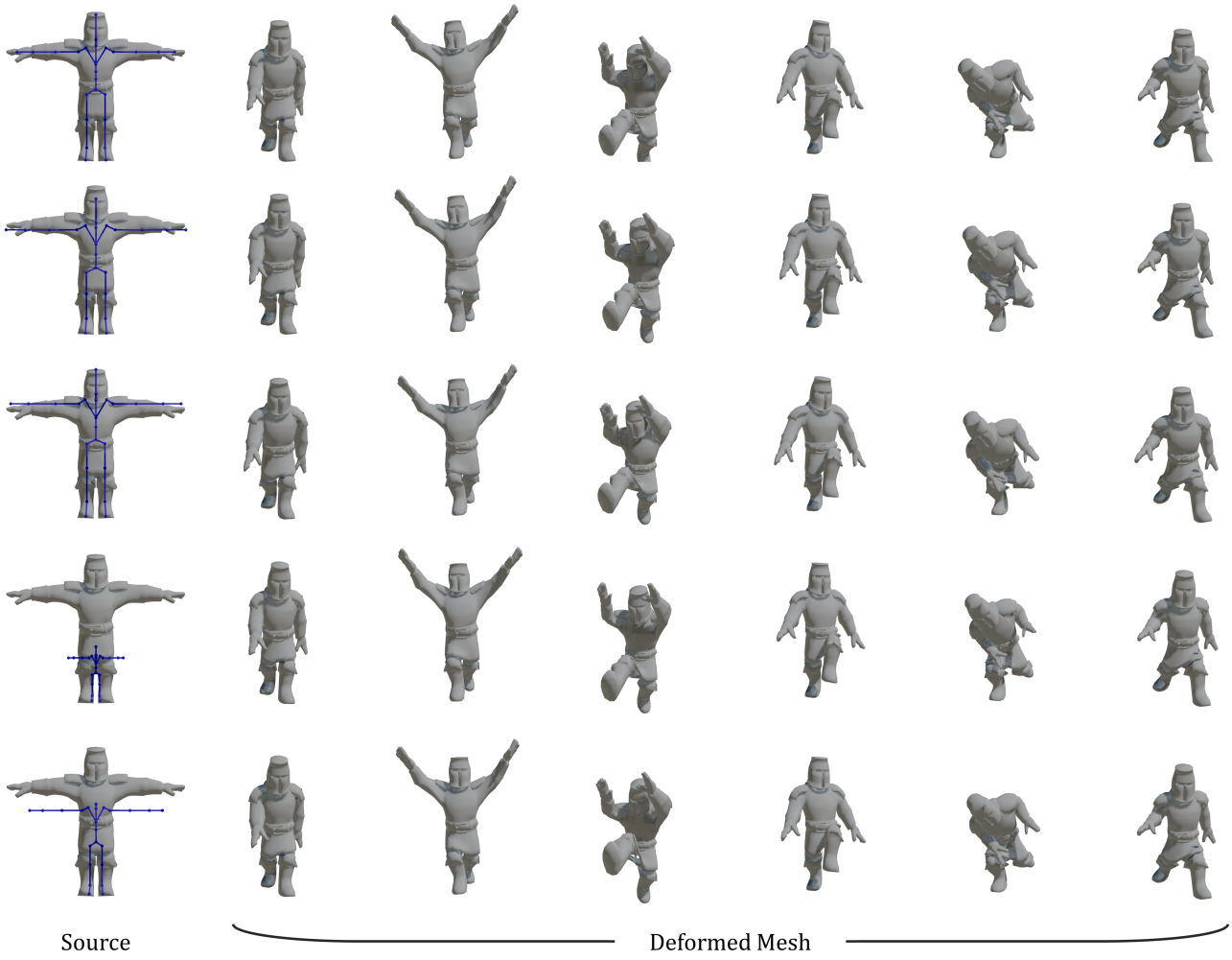

Figure 7: *Given the same source mesh with different source skeletons, our method robustly generates consistently deformed meshes in accordance with skeletal movements, even when the source skeletons have diverse configurations with varying number of joints: from the top 25, 22, 20, 24, and 23 joints are respectively used. The target skeletons derived from the source are presented in the last column of Figure 4.*

Table 3: *Deformation metrics with different skeletal configurations. S_1 represents the case in which the input skeleton is exactly aligned with the mesh, while $S_{2:6}$ represents cases with augmented skeletons. Despite the misalignment, $S_{2:6}$ still produces competitive results, demonstrating the robustness of our method.*

	CD↓	ADE↓	MDE↓	ELS↑
S_1	7.15	8.09	26.18	0.89
$S_{2:6}$	7.54	8.63	28.38	0.89

sonable results for the arms and head, it incorrectly assigned irrelevant vertices from the lower body and pelvis joint to other joints. This is because Pinocchio relies on geometric processing for skeleton estimation, which does not account for the high-level semantic relationships that define which vertices should be bound to specific joints. On the other hand, NBS produced unstable skinning predictions with inconsistent results across body parts. This is because NBS requires the source mesh to be strictly aligned with the SMPL distribution due to its dependency on the training dataset, limiting its performance on stylized, non-human characters. In contrast, ours demonstrated consistent and accurate skinning weight predictions that closely resemble the ground truth, indicating its ability to effectively capture the relationships between mesh vertices and skeleton joints.

When evaluating deformation metrics across all skeleton variations $S_{1:6}$, our method outperformed the baselines in CD, ADE, and MDE, as shown in Table 2, although the ELS results were slightly lower than those of Pinocchio. Additionally, Figure 6 presents the visual results of deformed meshes based on the predicted skeletons shown in Figure 4. While Pinocchio produced plausible results in general, it failed to retarget the given pose when the predicted skeleton significantly deviated from the shape of the source skeleton. Similarly, although NBS predicted a plausible target skeleton, unstable skinning weights employed by NBS affected the quality of

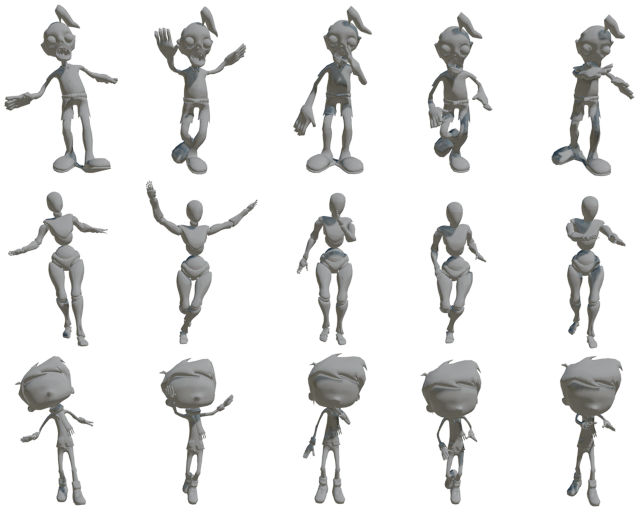

Figure 8: *Deformation results for multiple characters in the same pose. Each row shows different characters, while each column represents the same input pose applied to all characters.*

the mesh deformation. In contrast, ours robustly predicted the target skeleton given variations of source skeletons and generated consistently deformed results across varying skeletal configurations. For animation results, please see the supplementary video. To evaluate the robustness of our method in handling diverse skeletal configurations for a given input mesh, we measured the deformation metrics in two scenarios: (i) when the input skeleton is perfectly aligned with the mesh, which corresponds to S_1, and (ii) when the input skeleton differs from the mesh, which corresponds to $S_{2:6}$. As presented in Table 3, S_1 consistently produced better deformation compared to $S_{2:6}$, indicating that a more accurate initial alignment of the skeleton yields improved results. However, the performance gap between S_1 and $S_{2:6}$ was not substantial. Furthermore, even the mismatched skeletons $S_{2:6}$ produced outperforming or comparable results to baselines whose results are reported in Table 2. This result highlights the robustness of our method on adaptive rigging and skinning even when the input skeleton does not perfectly align with the mesh. The visual results are presented in Figure 7, where the same source mesh, paired with different skeleton configurations, produced consistently deformed results. In addition, Figure 8 demonstrates that our method reliably produces consistent poses across different character meshes. For animation results, please see the supplementary video.

4.4. Ablation Studies

To evaluate the contribution of each loss term, we conducted an ablation study by progressively removing individual loss terms, and the results are shown in Table 4 and Figure 9. We observed that the edge loss $\mathcal{L}_{edge}$ consistently improved the evaluation results of all the metrics compared to the case of training solely with $\mathcal{L}_{vtx}$, showing its contribution to enhanced overall performance. Notably, the models trained with $\mathcal{L}_{vtx}$ and $\mathcal{L}_{vtx} + \mathcal{L}_{edge}$ yielded slightly better quantitative results in terms of deformation, measured by CD, ADE, and MDE, compared to the full model. However, their quantitative results on rigging reflected in CD-J2J, CD-J2B, and CD-B2B, were significantly worse. This suggests that optimizing solely on deformation-related objectives does not adequately account for skeletal articulations, leading to distortions like candy-wrapper artifacts around the arm joints, as shown in Figure 9. Therefore, we emphasize the significance of rigging-related loss terms, which are $\mathcal{L}_{skel}$ and $\mathcal{L}_{sdf}$, for improving the overall deformation quality.

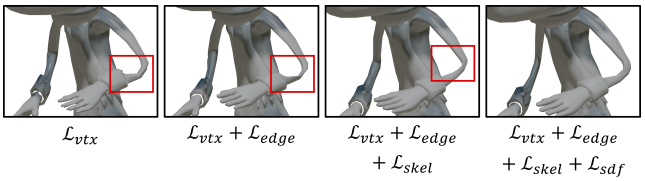

Figure 9: *Qualitative results of the ablation study for each loss term.*

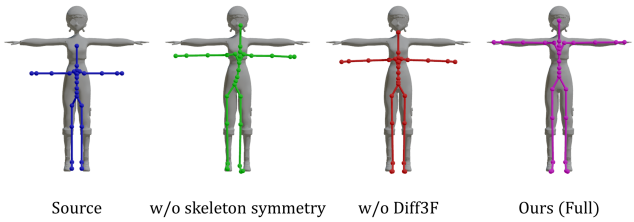

Figure 10: *Ablation on target skeleton prediction.*

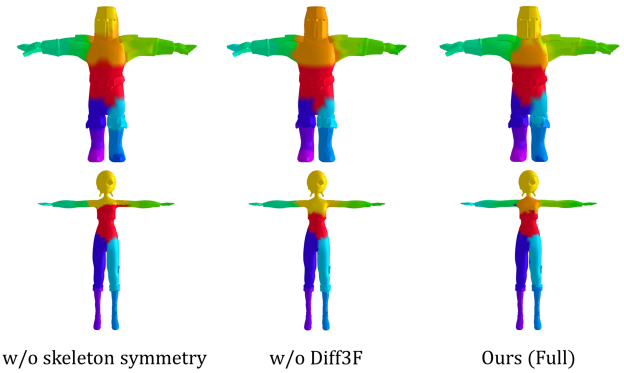

Figure 11: *Ablation on skinning prediction.*

The skeleton loss $\mathcal{L}_{skel}$ significantly enhanced the rigging quality, as indicated by all values of CD-J2J, CD-J2B, and CD-B2B, demonstrating its effectiveness in aligning character rigs with the mesh proportions. Additionally, it mitigated deformation artifacts, leading to smoother mesh shapes compared to the models trained with deformation objectives alone. The SDF loss $\mathcal{L}_{sdf}$ enhanced both rigging accuracy and deformation quality, with the full loss combination yielding the best results. These findings indicate the importance of using all the loss terms to achieve the best rigging results without compromising the deformation quality.

We also analyzed the impact of Diff3F and skeleton symmetry by excluding each component. As shown in Figure 10, while our full model produced accurate rigging results, with joints precisely embedded within the character mesh, the model trained without Diff3F

Table 4: *Ablation study results with a source skeleton having **varying** number of joints and **differing** size and proportion from the source mesh.*

Metric	CD-J2J↓	CD-J2B↓	CD-B2B↓	Skinning L1↓	CD↓	ADE↓	MDE↓	ELS↑
$\mathcal{L}_{vtx}$	18.82	14.26	11.59	0.0461	7.54	8.63	27.98	0.87
$\mathcal{L}_{vtx} + \mathcal{L}_{edge}$	17.99	13.52	10.70	0.0449	**7.39**	**8.41**	**27.63**	0.88
$\mathcal{L}_{vtx} + \mathcal{L}_{edge} + \mathcal{L}_{skel}$	16.54	12.52	10.37	**0.0418**	7.80	9.04	29.86	0.88
$\mathcal{L}_{vtx} + \mathcal{L}_{edge} + \mathcal{L}_{skel} + \mathcal{L}_{sdf}$	**16.28**	**11.85**	**9.90**	0.0449	7.48	8.54	28.01	0.89
w/o Diff3F	16.43	12.16	10.23	0.0503	7.56	8.69	29.60	**0.91**
w/o skeleton symmetry	17.90	13.70	11.41	0.0505	8.14	9.52	31.86	0.90

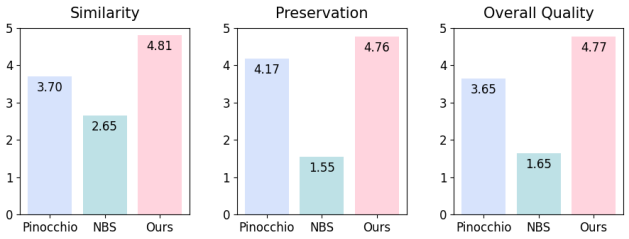

Figure 12: *Results of user study in terms of Similarity, Preservation, and Overall Quality.*

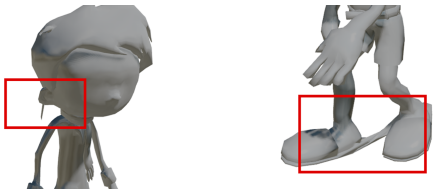

Figure 13: *When the skinning weights are inaccurately predicted, artifacts may appear during the deformation process.*

produced joints that deviate from the character mesh, emphasizing its role in maintaining joint-vertex correspondences. Moreover, the model trained without skeletal symmetry processing struggled to position the joints correctly in relation to the mesh. We also present the importance of both components for accurate character skinning in Figure 11. Excluding Diff3F led to the leakage of skinning weights to unintended body parts, demonstrating the importance of Diff3F in capturing accurate joint-vertex correspondences. Additionally, removing skeletal symmetry produced asymmetric skinning weights, which could cause inaccurate deformations when the character poses are applied. These results are also numerically evident in Table 4, with the worst scores for the Skinning L1. Overall, these results validate the effectiveness of incorporating both Diff3F and skeletal symmetry to improve the quality of rigging and skinning, which is crucial for high-quality mesh deformations. For animation results, please see the supplementary video.

4.5. User Study

We conducted a user study to evaluate the deformation quality resulted from the predicted skeleton and skinning weights of our method. We compared our approach with other skeleton-based auto-rigging methods: Pinocchio and NBS. The study was designed to compare the quality of the deformed mesh animations generated by transferring a given source skeletal animation to a source mesh in the rest pose. We sampled 11 source skeleton-mesh pairs from our test dataset, including augmented skeletons. As a result, we generated a total of 33 tasks, each comprising 4-5 seconds of animations created by each method. For each resulting mesh animation that is presented in a random order, participants were asked to evaluate the following three criteria: *Similarity* with the source skeletal animation, *Preservation* of the original mesh shape compared to its rest pose state, and *Overall Quality* of the animation. All questions were rated on a 5-point Likert scale, ranging from 1 (Strongly disagree) to 5 (Strongly agree). We recruited 16 participants (9 males and 7 females; ages: 24 to 33).

The results of the study are shown in Figure 12. Our method achieved the highest scores for all three criteria, with 4.81 in *Similarity*, 4.76 in *Preservation*, and 4.77 in *Overall Quality*, demonstrating its superior performance in both rigging and skinning quality. Pinocchio scored high in *Preservation* but showed relatively lower performance in the other two criteria. This low performance is due to the suboptimal rigging quality created for the augmented skeletons, as shown in Figure 4. Specifically, the predicted skeleton frequently collapsed into a localized region of the mesh, resulting in limited deformation across the remaining body parts. NBS produced even lower scores, particularly in *Preservation* and *Overall Quality*, due to the presence of noticeable artifacts such as abnormal stretching, as shown in Figure 6. This along with the inaccurate prediction of skinning weights shown in Figure 5 also negatively impacted the *Similarity* score, making it relatively lower compared to other methods. In contrast, the robust performance of our method in predicting both skeleton and skinning weights, even with variations in skeleton configurations, resulted in high-quality pose-conditioned mesh deformations that reflect the naturalness on human perception.

5. Discussion and Conclusion

Despite the flexibility and high-quality results, we observed some limitations of our method. While our method consistently produced plausible results across diverse mesh and skeleton variations, there was occasional leakage of skinning weights, due to the bounding of vertices to unintended joints. Furthermore, the predicted skinning weights were sometimes distributed evenly across multiple joints, instead of being concentrated on a few specific joints, which can cause indistinct deformations that fail to follow the given pose adequately. This can result in artifacts in the deformed meshes, as shown in Figure 13, significantly reducing the overall deformation quality. Additionally, our method struggles to handle auxiliary

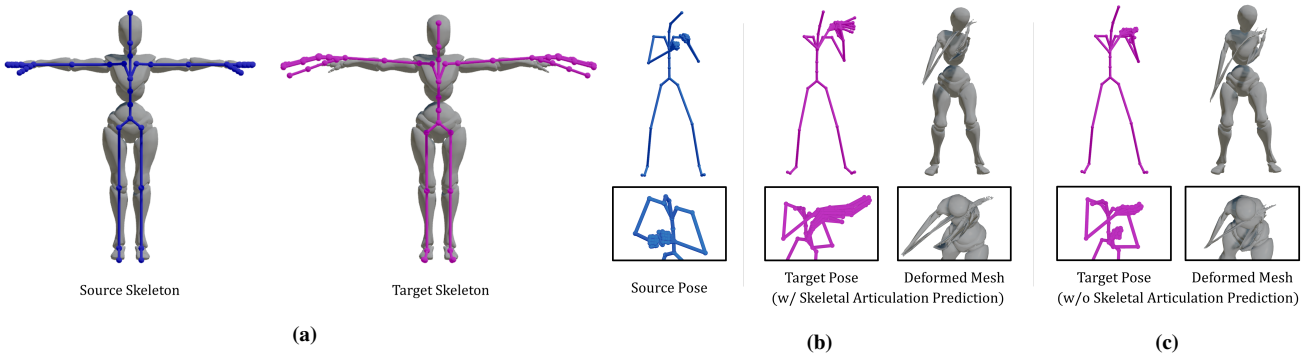

Figure 14: *(a) Source skeleton with finger joints (left) and the target skeleton predicted by the Skeletal Articulation Prediction module (right). Skeletons are overlapped on source meshes to demonstrate alignments between them. (b) Target pose and deformed mesh obtained by retargeting the source pose to the predicted target skeleton. (c) Target pose and deformed mesh obtained by directly using the source skeleton as the target skeleton without Skeletal Articulation Prediction.*

joints, such as fingers. As shown in Figure 14a, the predicted target skeleton deviates significantly from the source mesh, resulting in undesirable mesh deformation caused by inaccurate motion retargeting and skinning prediction, as illustrated in Figure 14b. Even when the source skeleton is directly given to the Skinning Weight Prediction module, the deformation still yields unnatural results, as shown in Figure 14c. This issue stems partly from the limitations of SAME in accurately retargeting motions for auxiliary joints, and partly from the difficulty in predicting appropriate skinning weights for finer skeletal joints like fingers. Finally, to compute the SDF loss, we first obtain unsigned distances from the joints to the nearest surface, and then assign signs based on inside and outside classification, enabling SDF computation for non-watertight meshes. However, this simplified approach lacks robustness, particularly for complex geometries.

For future work, we aim to improve rigging quality by ensuring that each joint is exactly embedded within the character mesh. Enhancing the accuracy of motion retargeting by incorporating more advanced techniques and utilizing more robust prior features will also be an interesting direction to improve performance. Additionally, incorporating additional regularization to encourage sharp changes in skinning weights between different joints may further enhance deformation quality.

In this paper, we introduced a novel end-to-end framework for adaptive rigging and skinning of stylized character meshes using skeletal motion data. Our approach consists of two key stages: Skeletal Articulation Prediction, which adjusts the given skeleton to align with the source mesh, and Skinning Weight Prediction, which generates plausible skinning weights for the predicted target skeleton and given source mesh. One of the key strengths of our method is its flexibility to handle a wide variety of input formats, including variations in both skeletal and mesh configurations. By incorporating Diff3F as a semantic prior, our method can effectively model the correspondences between mesh vertices and skeleton joints, improving its generalizability to unseen characters. We demonstrated that our method outperforms previous rigging and skinning methods, along with the robustness of our method on rigging, skinning, and deformation given varying skeleton and mesh input.

Acknowledgements

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (RS-2024-00333478).

References

[AGK*22] AIGERMAN N., GUPTA K., KIM V. G., CHAUDHURI S., SAITO J., GROUEIX T.: Neural jacobian fields: Learning intrinsic mappings of arbitrary meshes. *arXiv preprint arXiv:2205.02904* (2022). 3

[ALL*20] 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 (2020), 62–1. 3

[ATC*08] AU O. K.-C., TAI C.-L., CHU H.-K., COHEN-OR D., LEE T.-Y.: Skeleton extraction by mesh contraction. *ACM transactions on graphics (TOG) 27*, 3 (2008), 1–10. 3

[Aut18] AUTODESK L.: Quick rig tool. https://knowledge.autodesk.com/support/maya/learn-explore/caas/CloudHelp/cloudhelp/2022/ENU/Maya-CharacterAnimation/files/GUID-DC29C982-D04F-4C20-9DBA-4BBB33E027EF-htm.html/, 2018. 2

[aut21] Motion builder - a 3d character animation software. https://www.autodesk.com, 2021. 6

[BJD*12] BOROSÁN P., JIN M., DECARLO D., GINGOLD Y., NEALEN A.: Rigmesh: automatic rigging for part-based shape modeling and deformation. *ACM Transactions on Graphics (TOG) 31*, 6 (2012), 1–9. 3

[BP07] BARAN I., POPOVIĆ J.: Automatic rigging and animation of 3d characters. *ACM Transactions on graphics (TOG) 26*, 3 (2007), 72–es. 2, 3, 7, 16

[BTST12] BHARAJ G., THORMÄHLEN T., SEIDEL H.-P., THEOBALT C.: Automatically rigging multi-component characters. In *Computer Graphics Forum* (2012), vol. 31, Wiley Online Library, pp. 755–764. 3

[CK00] CHOI K.-J., KO H.-S.: Online motion retargetting. *The Journal of Visualization and Computer Animation 11*, 5 (2000), 223–235. 3

[CMU] Cmu graphics lab motion capture database. http://mocap.cs.cmu.edu/. Accessed: 2024-08-21. 1

[CTO*10] CAO J., TAGLIASACCHI A., OLSON M., ZHANG H., SU Z.: Point cloud skeletons via laplacian based contraction. In *2010 Shape Modeling International Conference* (2010), IEEE, pp. 187–197. 3

[DMM24] DUTT N. S., MURALIKRISHNAN S., MITRA N. J.: Diffusion 3d features (diff3f): Decorating untextured shapes with distilled semantic features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2024), pp. 4494–4504. 4, 5

[FCD20] FU T., CHAINE R., DIGNE J.: Fakir: An algorithm for revealing the anatomy and pose of statues from raw point sets. In *Computer Graphics Forum* (2020), vol. 39, Wiley Online Library, pp. 375–385. 3

[GFK*18] GROUEIX T., FISHER M., KIM V. G., RUSSELL B. C., AUBRY M.: 3d-coded: 3d correspondences by deep deformation. In *Proceedings of the european conference on computer vision (ECCV)* (2018), pp. 230–246. 3

[Gle98] GLEICHER M.: Retargetting motion to new characters. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 1998), Association for Computing Machinery, p. 33–42. URL: https://doi.org/10.1145/280814.280820, doi:10.1145/280814.280820. 3

[GYQ*18] GAO L., YANG J., QIAO Y.-L., LAI Y., ROSIN P., XU W., XIA S.: Automatic unpaired shape deformation transfer. *ACM Transactions on Graphics 37*, 6 (2018), 1–15. 3

[Hej04] HEJL J.: Hardware skinning with quaternions. *Game Programming Gems 4*, 487-495 (2004), 3. 2

[HKA*18] HUANG Y., KAUFMANN M., AKSAN E., BLACK M. J., HILLIGES O., PONS-MOLL G.: Deep inertial poser: Learning to reconstruct human pose from sparse inertial measurements in real time. *ACM Transactions on Graphics (TOG) 37*, 6 (2018), 1–15. 1

[HKS17] HOLDEN D., KOMURA T., SAITO J.: Phase-functioned neural networks for character control. *ACM Transactions on Graphics (TOG) 36*, 4 (2017), 1–13. 15

[HSM*24] HEDLIN E., SHARMA G., MAHAJAN S., ISACK H., KAR A., TAGLIASACCHI A., YI K. M.: Unsupervised semantic correspondence using stable diffusion. *Advances in Neural Information Processing Systems 36* (2024). 4

[HYNP20] HARVEY F. G., YURICK M., NOWROUZEZAHRAI D., PAL C.: Robust motion in-betweening. 1, 7

[Iof15] IOFFE S.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167* (2015). 7

[KCŽO07] KAVAN L., COLLINS S., ŽÁRA J., O'SULLIVAN C.: Skinning with dual quaternions. In *Proceedings of the 2007 symposium on Interactive 3D graphics and games* (2007), pp. 39–46. 2

[KHH*17] KOMURA T., HABIBIE I., HOLDEN D., SCHWARZ J., YEARSLEY J.: A recurrent variational autoencoder for human motion synthesis. In *The 28th British Machine Vision Conference* (2017). 1

[Kin14] KINGMA D. P.: Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014). 7

[KŽ05] KAVAN L., ŽÁRA J.: Spherical blend skinning: a real-time deformation of articulated models. In *Proceedings of the 2005 symposium on Interactive 3D graphics and games* (2005), pp. 9–16. 2

[LAH*21] LI P., ABERMAN K., HANOCKA R., LIU L., SORKINE-HORNUNG O., CHEN B.: Learning skeletal articulations with neural blend shapes. *ACM Transactions on Graphics (TOG) 40*, 4 (2021), 1–15. 2, 3, 7, 16

[LCC19] LIM J., CHANG H. J., CHOI J. Y.: Pmnet: Learning of disentangled pose and movement for unsupervised motion retargeting. In *BMVC* (2019), vol. 2, p. 7. 3

[LD14] LE B. H., DENG Z.: Robust and accurate skeletal rigging from mesh sequences. *ACM Transactions on Graphics (TOG) 33*, 4 (2014), 1–10. 2

[LDP*24] LUO G., DUNLAP L., PARK D. H., HOLYNSKI A., DARRELL T.: Diffusion hyperfeatures: Searching through time and space for semantic correspondence. *Advances in Neural Information Processing Systems 36* (2024). 2, 4

[LKP*23] LEE S., KANG T., PARK J., LEE J., WON J.: Same: Skeleton-agnostic motion embedding for character animation. In *SIGGRAPH Asia 2023 Conference Papers* (2023), pp. 1–11. 3, 4, 6, 15, 16

[LS99] LEE J., SHIN S. Y.: A hierarchical approach to interactive motion editing for human-like figures. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques* (1999), pp. 39–48. 3

[LYS*22] LIAO Z., YANG J., SAITO J., PONS-MOLL G., ZHOU Y.: Skeleton-free pose transfer for stylized 3d characters. In *European Conference on Computer Vision* (2022), Springer, pp. 640–656. 3

[MAF10] MILLER C., ARIKAN O., FUSSELL D.: Frankenrigs: building character rigs from multiple sources. In *Proceedings of the 2010 ACM SIGGRAPH symposium on Interactive 3D Graphics and Games* (2010), pp. 31–38. 3

[mix] Mixamo. https://www.mixamo.com/. Accessed: 2024-08-21. 2, 7, 16

[MLT88] MAGNENAT T., LAPERRIÈRE R., THALMANN D.: Joint-dependent local deformations for hand animation and object grasping. In *Proceedings of Graphics Interface'88* (1988), Canadian Inf. Process. Soc, pp. 26–33. 2

[MMG06] MERRY B., MARAIS P., GAIN J.: Animation space: A truly linear framework for character animation. *ACM Transactions on Graphics (TOG) 25*, 4 (2006), 1400–1423. 2

[MMRH22] MOSELLA-MONTORO A., RUIZ-HIDALGO J.: Skinningnet: Two-stream graph convolutional neural network for skinning prediction of synthetic characters. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), pp. 18593–18602. 2

[MSK22] MASON I., STARKE S., KOMURA T.: Real-time style modelling of human locomotion via feature-wise transformations and local motion phases. *Proceedings of the ACM on Computer Graphics and Interactive Techniques 5*, 1 (may 2022). doi:10.1145/3522618. 1

[MZ23] MA J., ZHANG D.: Tarig: Adaptive template-aware neural rigging for humanoid characters. *Computers & Graphics 114* (2023), 158–167. 3

[ODM*23] OQUAB M., DARCET T., MOUTAKANNI T., VO H., SZAFRANIEC M., KHALIDOV V., FERNANDEZ P., HAZIZA D., MASSA F., EL-NOUBY A., ET AL.: Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193* (2023). 4

[PFGA19] PAVLLO D., FEICHTENHOFER C., GRANGIER D., AULI M.: 3d human pose estimation in video with temporal convolutions and semi-supervised training. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2019), pp. 7753–7762. 1

[PYA*23] PONTON J. L., YUN H., ARISTIDOU A., ANDUJAR C., PELECHANO N.: Sparseposer: Real-time full-body motion reconstruction from sparse data. *ACM Transactions on Graphics 43*, 1 (2023), 1–14. 1

[QSMG17] QI C. R., SU H., MO K., GUIBAS L. J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2017), pp. 652–660. 5

[RBH*21] REMPE D., BIRDAL T., HERTZMANN A., YANG J., SRIDHAR S., GUIBAS L. J.: Humor: 3d human motion model for robust pose estimation. In *Proceedings of the IEEE/CVF international conference on computer vision* (2021), pp. 11488–11499. 1

[SKHB24] SHIN S., KIM J., HALILAJ E., BLACK M. J.: Wham: Reconstructing world-grounded humans with accurate 3d motion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2024), pp. 2070–2080. 1

[SLSG01] SHIN H. J., LEE J., SHIN S. Y., GLEICHER M.: Computer puppetry: An importance-based approach. *ACM Transactions on Graphics (TOG) 20*, 2 (2001), 67–94. 3

[SSW*10] SEO J., SEOL Y., WI D., KIM Y., NOH J.: Rigging transfer. *Computer Animation and Virtual Worlds 21*, 3-4 (2010), 375–386. 3

[SZKS19] STARKE S., ZHANG H., KOMURA T., SAITO J.: Neural state machine for character-scene interactions. *ACM Transactions on Graphics 38*, 6 (2019), 178. 1

[TGLX18] TAN Q., GAO L., LAI Y.-K., XIA S.: Variational autoencoders for deforming 3d mesh models. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2018), pp. 5841–5850. 3

[TJW*23] TANG L., JIA M., WANG Q., PHOO C. P., HARIHARAN B.: Emergent correspondence from image diffusion. *Advances in Neural Information Processing Systems 36* (2023), 1363–1389. 4

[VCC*17] VELIČKOVIĆ P., CUCURULL G., CASANOVA A., ROMERO A., LIO P., BENGIO Y.: Graph attention networks. *arXiv preprint arXiv:1710.10903* (2017). 5

[VCH*21] 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* (2021), pp. 9720–9729. 3

[VSP*17] VASWANI A., SHAZEER N., PARMAR N., USZKOREIT J., JONES L., GOMEZ A. N., KAISER L. U., POLOSUKHIN I.: Attention is all you need. In *Advances in Neural Information Processing Systems* (2017), Guyon I., Luxburg U. V., Bengio S., Wallach H., Fergus R., Vishwanathan S., Garnett R., (Eds.), vol. 30, Curran Associates, Inc. URL: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf. 6

[VYCL18] 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* (2018), pp. 8639–8648. 3

[WHZ*23] WANG H., HUANG S., ZHAO F., YUAN C., SHAN Y.: Hmc: Hierarchical mesh coarsening for skeleton-free motion retargeting. *arXiv preprint arXiv:2303.10941* (2023). 3

[WLL*23] WANG J., LI X., LIU S., DE MELLO S., GALLO O., WANG X., KAUTZ J.: Zero-shot pose transfer for unrigged stylized 3d characters. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2023), pp. 8704–8714. 3, 9

[WMLL24] WANG R., MAO W., LU C., LI H.: Towards high-quality 3d motion transfer with realistic apparel animation. *arXiv preprint arXiv:2407.11266* (2024). 2, 3

[WP02] WANG X. C., PHILLIPS C.: Multi-weight enveloping: least-squares approximation techniques for skin animation. In *Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation* (2002), pp. 129–138. 2

[XZK*20] XU Z., ZHOU Y., KALOGERAKIS E., LANDRETH C., SINGH K.: Rignet: Neural rigging for articulated characters. *arXiv preprint arXiv:2005.00559* (2020). 2, 3, 7, 8, 16

[XZYK22] XU Z., ZHOU Y., YI L., KALOGERAKIS E.: Morig: Motion-aware rigging of character meshes from point clouds. In *SIGGRAPH Asia 2022 conference papers* (2022), pp. 1–9. 3

[YZX24] YI X., ZHOU Y., XU F.: Physical non-inertial poser (pnp): Modeling non-inertial effects in sparse-inertial human motion capture. In *ACM SIGGRAPH 2024 Conference Papers* (2024), pp. 1–11. 1

[ZBL*19] ZHOU Y., BARNES C., LU J., YANG J., LI H.: On the continuity of rotation representations in neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2019), pp. 5745–5753. 15

[ZHH*24] ZHANG J., HERRMANN C., HUR J., POLANIA CABRERA L., JAMPANI V., SUN D., YANG M.-H.: A tale of two features: Stable diffusion complements dino for zero-shot semantic correspondence. *Advances in Neural Information Processing Systems 36* (2024). 4

[ZWK*23] 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* (2023), pp. 13864–13872. 3

[ZZM*21] ZHENG C., ZHU S., MENDIETA M., YANG T., CHEN C., DING Z.: 3d human pose estimation with spatial and temporal transformers. In *Proceedings of the IEEE/CVF international conference on computer vision* (2021), pp. 11656–11665. 1

Appendix A: Detailed explanation of D^T

A skeletal motion data $M = (S, D^{1:N_T})$, includes S denoting the skeleton and $D^{1:N_T}$ representing the motion data with N_T frames. Following Lee et al. [LKP*23], D^t is represented as follows:

$$D^t = \{\mathbf{q}^t_{1:N_J}, \mathbf{p}^t_{1:N_J}, \mathbf{p}^{t-1}_{1:N_J}, \mathbf{v}^t_{1:N_J}, \mathbf{r}^t, \mathbf{c}^t_{1:N_J}\}, \tag{19}$$

where $\mathbf{q}_j \in \mathbb{R}^6$ represents the local joint rotation with respect to the parent joint in the 6D representation [ZBL*19]. This representation is derived by taking the first two columns of a 3×3 rotation matrix, which ensures continuity and resolves ambiguities, stabilizing neural networks training. $\mathbf{p}_j \in \mathbb{R}^3$ represents joint positions in the character's facing frames [HKS17]. Specifically, a lateral vector is computed by averaging the vector between the hip joints and the vector between the shoulder joints, and then the facing direction is calculated as the cross product of the lateral and upward directions. Subsequently, the lateral vector is orthogonalized via the Gram-Schmidt process using the facing and upward directions, establishing a consistent 3D orthonormal basis. The origin of this frame is computed by projecting the root joint position onto the horizontal plane. The linear velocity $\mathbf{v}^t_j$ is computed by $\mathbf{v}_j = (\mathbf{p}^t_j - \mathbf{p}^{t-1}_j)/\Delta t$ as a translational difference between two consecutive frames. $\mathbf{r}^t = (\Delta x, \Delta z, \Delta\theta, h)$ denotes the root movement, where $(\Delta x, \Delta z)$ and $\Delta\theta$ are translational velocities on the horizontal plane and rotational velocity around the up-axis with respect to the facing frame at the previous frame, while h is the height of the root joint from the ground. Finally, $\mathbf{c}_j$ represents a contact label, indicating whether the j-th joint is in contact with the ground or not. The generated target pose at frame t is defined as follows:

$$\hat{D}^t = \{\hat{\mathbf{q}}^t_{1:N_J}, \hat{\mathbf{r}}^t, \hat{\mathbf{c}}^t_{1:N_J}\}, \tag{20}$$

by omitting redundant elements such as joint positions, which can be derived by solving forward kinematics using $\hat{\mathbf{q}}$ and $\hat{\mathbf{r}}$.

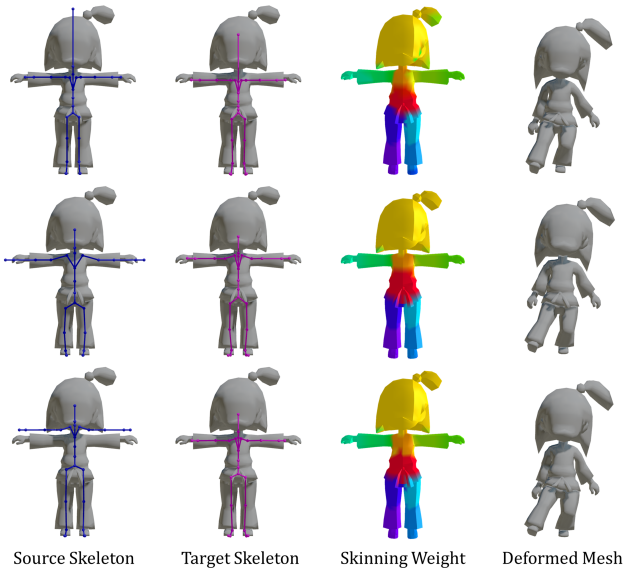

Figure 15: *Qualitative results on out-of-domain characters from the RigNet-v1 dataset. Skeletons are overlaid on source meshes to demonstrate alignments between them.*

Appendix B: Architecture Details

Tables 5 and 6 show the detailed network architectures of the Skeletal Articulation Prediction module and Skinning Weight Prediction module, respectively. The names correspond to those in Figure 2 of the main paper, except the Skeleton Decoder of Skeletal Articulation Prediction module in Table 5 refers to the combination of the Attention and MLP layers.

Appendix C: Additional Experiments

Evaluations on Out-of-Domain Characters

While our test dataset includes a wide range of stylized characters from Mixamo [mix], we conducted additional experiments on out-of-domain characters to further evaluate the generalization capabilities of our method. Specifically, we employed characters in the T-pose from the RigNet-v1 dataset [XZK*20], which are unseen during training. These characters were used as source meshes and deformed using source skeletons and motions from our database.

As shown in Figure 15, our method showed robust performance in generating target skeletons that align with the character mesh despite variations in the structures and shapes of the input skeletons. Furthermore, our method produced plausible skinning weights aligned seamlessly with both the source mesh and the predicted target skeletons, deriving plausible deformations for out-of-domain characters, which have shapes and body ratios distinct from Mixamo characters. These results highlight the robustness and generalizability of our approach on unseen characters. For animation results with additional characters, please see the supplementary video.

Evaluations on Individual Impact

To rigorously evaluate the impact of individual changes in each independent variable, we conducted two additional experiments: (i) comparison of the performance of the skinning weight prediction module across different baselines, and (ii) analysis on the impacts of individual changes in each element of the skeletal configuration, including body scale, bone lengths, and the number of joints.

Performance of Skinning Weight Prediction To solely compare the performance of the skinning weight prediction components across different baselines, we measured the skinning and deformation metrics using identical skeletons for all methods. Specifically, we used the source skeleton precisely aligned with the source mesh as input to the skinning weight prediction modules of each baseline to generate skinning weights, while bypassing their skeleton prediction modules. For deformation metrics, we used the source poses directly, instead of retargeting poses using SAME [LKP*23]. NBS [LAH*21] was excluded from this experiment because its skinning weight prediction relies solely on the mesh and does not utilize skeleton inputs. To obtain the results of Pinocchio [BP07], we followed the experimental setup of NBS [LAH*21] that uses the auto-skinning tool provided by Blender, which is implemented based on the algorithm of Pinocchio.

As shown in Table 7, Pinocchio [BP07] achieved the best performance across all metrics, with our method producing comparable results. While ours did not achieve the best quantitative scores, the discrepancies were minor with visually imperceptible variations as shown in Figure 16. Furthermore, our method still has an advantage in generalizability of rigging and skinning for various skeletal configurations, in that our method produced results consistent to Table 2 of the main paper, whereas Pinocchio produced results with significant deviation. RigNet [XZK*20] achieved comparable results in CD and ADE metrics to other methods, but its results exhibited noticeable artifacts, such as stretched vertices, which were reflected in significantly higher MDE and lower ELS values than those from other methods.

Changes of Individual Skeletal Configuration Starting with an initial source skeleton containing 25 joints, precisely aligned in size and proportion with the source mesh, we modified three key elements of the skeletal configuration to generate new source skeletons, as follows:

- Body scale: A uniform scaling factor of 0.5 was applied to all bones to adjust the overall body size.
- Bone length: A non-uniform scaling was applied using scaling factors of 0.8 and 1.2 along the vertical and lateral axes of each joint, respectively.
- Number of joints: The number of joints was randomly adjusted, resulting in two additional joints to the initial skeleton.

Figure 17 shows the generated source skeletons with their corresponding poses, along with the deformed meshes driven by each method based on the source mesh. Our approach consistently produced plausible deformations that follow the source poses, regardless of changes in the skeletal configuration. In contrast, Pinocchio [BP07] failed to preserve the volume of the source mesh, resulting in excessive expansion around shoulders and contraction around the spine. NBS [LAH*21] resulted in distorted meshes due

Table 5: *Network architectures of the Skeletal Articulation Prediction module.*

Name	Layers	Channels	Attention Heads
Mesh Encoder	Linear - BatchNorm - ReLU - Dropout	$35 \rightarrow 256$	-
	Linear - BatchNorm - ReLU - Dropout	$256 \rightarrow 256$	-
	Linear - BatchNorm - ReLU - Dropout	$256 \rightarrow 32$	-
	Pooling & Concatenation	$32 \rightarrow 64$	-
	Linear	$64 \rightarrow 32$	-
Skeleton Encoder	GAT - BatchNorm - ReLU - Dropout	$6 \rightarrow 16$	16
	GAT - BatchNorm - ReLU - Dropout	$256 \rightarrow 16$	16
	GAT - BatchNorm - ReLU - Dropout	$256 \rightarrow 16$	16
	GAT - BatchNorm - Dropout	$256 \rightarrow 32$	1
Skeleton Decoder	CrossAttention (QKV) - Dropout - Residual Connection	$32 \rightarrow 2$	16
	Linear - ReLU	$32 \rightarrow 32$	-
	Linear - ReLU	$32 \rightarrow 32$	-
	Linear - BatchNorm	$32 \rightarrow 3$	-

Table 6: *Network architectures of the Skinning Weight Prediction module.*

Name	Layers	Channels	Attention Heads
Skeleton Encoder	GAT - BatchNorm - ReLU - Dropout	$6 \rightarrow 16$	16
	GAT - BatchNorm - ReLU - Dropout	$256 \rightarrow 16$	16
	GAT - BatchNorm - ReLU - Dropout	$256 \rightarrow 16$	16
	GAT - BatchNorm - Dropout	$256 \rightarrow 32$	1
Skinnning Weight Predictor	CrossAttention (QK)	$N_V \times 32$ and $N_J \times 32 \rightarrow N_V \times N_J$	1

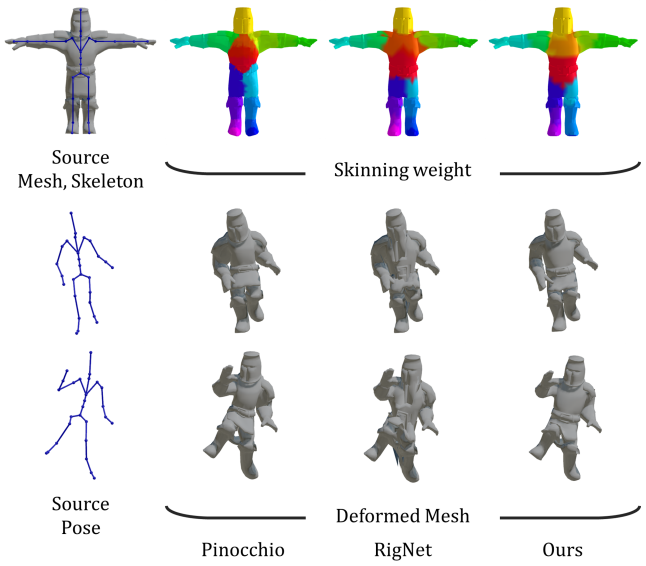

Figure 16: *Qualitative comparison with baselines on predicted skinning weights and mesh deformation. The first column shows the source mesh and skeleton given to the skinning weight prediction module of each baseline, with source poses to deform the mesh. In the second to fourth columns, the first row demonstrates the skinning weights predicted by each baseline, while the second and third rows show the resulting deformed meshes.*

Table 7: *Quantitative results on skinning and deformation using identical source skeletons that precisely align with the source meshes. The best result for each column is in bold.*

	Skinning L1↓	CD↓	ADE↓	MDE↓	ELS↑
Pinocchio	**0.0188**	**2.83**	**2.56**	**13.74**	**0.96**
RigNet	0.0178	2.95	4.71	95.60	0.10
Ours	0.0469	4.72	4.72	21.52	0.89

to improper skinning weights applied to certain body parts. Because NBS relies on a pre-defined set of joints, the results for the last source skeleton, which contains additional joints, were excluded.

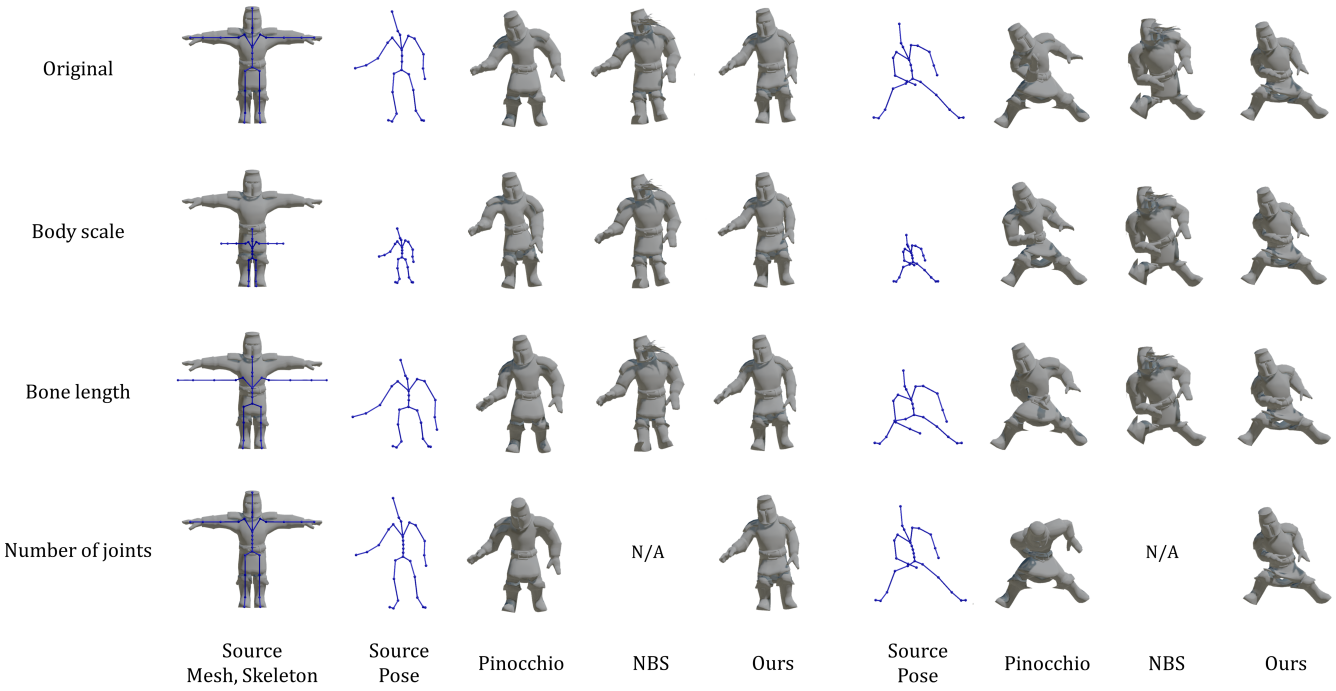

Figure 17: *Qualitative comparison of deformation results under varying skeletal configurations. Beginning with a source skeleton that precisely aligns with the source mesh, each subsequent row includes the source skeleton generated by modifying one factor: body scale, bone length, or the number of joints, respectively, while maintaining the other elements fixed.*