Skeleton-free Pose Transfer for Stylized 3D Characters

Zhouyingcheng Liao[1⋆], Jimei Yang[2], Jun Saito[2],
Gerard Pons-Moll[3,4], and Yang Zhou[2]

[1]Saarland University [2]Adobe Research [3]University of Tübingen
[4]Max Planck Institute for Informatics, Saarland Informatics Campus

https://zycliao.github.io/sfpt

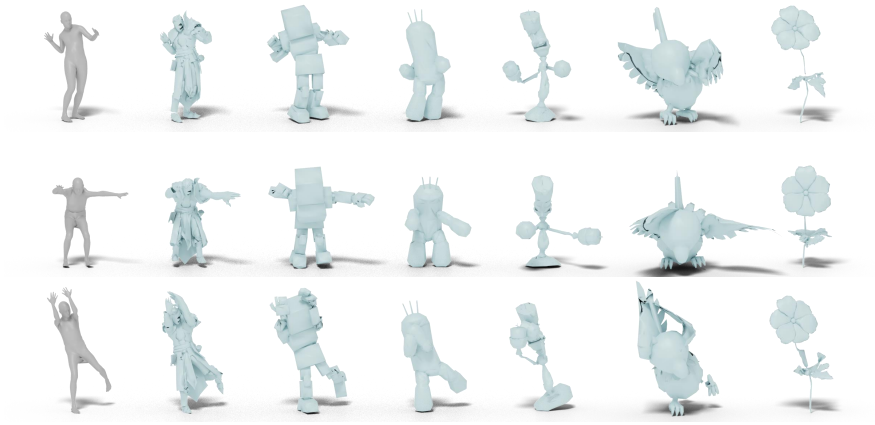

Fig. 1. Stylized 3D characters pose transfer. Given source pose characters as input (left), our model automatically transfers their poses to target subject characters with different body proportions and topologies (right). Our method does not require rigging, skinning, or correspondence labeling for both source and target characters.

Abstract. We present the first method that automatically transfers poses between stylized 3D characters without skeletal rigging. In contrast to previous attempts to learn pose transformations on fixed or topology-equivalent skeleton templates, our method focuses on a novel scenario to handle skeleton-free characters with diverse shapes, topologies, and mesh connectivities. The key idea of our method is to represent the characters in a unified articulation model so that the pose can be transferred through the correspondent parts. To achieve this, we propose a novel pose transfer network that predicts the character skinning weights and deformation transformations jointly to articulate the target character to match the desired pose. Our method is trained in a semi-supervised manner absorbing all existing character data with paired/unpaired poses and stylized shapes. It generalizes well to unseen stylized characters and inanimate objects. We conduct extensive experiments and demonstrate the effectiveness of our method on this novel task.

⋆ Work done during an internship at Adobe.

1 Introduction

Humans and animals evolved naturally with intrinsic articulation structures to facilitate their movements in complex environments. As a result, the articulated poses become an important part of their behaviors and emotions. Based on this observation, 3D artists create highly-stylized characters in movies and games, from human-like, anthropomorphic to even inanimate objects (e.g. Pixar's Luxo Jr.). Posing and animating these characters like humans is key to conveying human-understandable expressions and emotions but actually requires many costly and sophisticated manual processes. Furthermore, once animations are created, artists often want to re-use them in novel characters and scenarios. A large number of existing studies have addressed the problem of automatically transferring poses between human-like characters [47,46,2,24,16], as they often share a similar or same skeletal rig. Very few works were devoted to animating non-human characters from human data [15,7], but they either require correspondence, or need to be trained for every pair of shapes. In this paper, we propose a learning-based method that automatically transfers poses between characters of various proportion and topology without skeletal rigs as a prerequisite (Fig. 1).

Given that most articulated 3D characters have piecewise rigid structures, the animation of characters is usually controlled by a set of sparse deformation primitives (rigs) [20] instead of manipulating the 3D mesh directly. With the deformation primitives, one has to bind them to the character mesh to build the correspondences between the primitives and the mesh vertices. This process is referred to as skinning. The skinning weight is used to describe how each deformation primitive affects the mesh deformation.

For characters deformed by sparse primitives, transferring poses between them faces two challenges. First, rigging and skinning is a laborious manual process, which requires high expertise. Second, rigs must have correspondence to each other so that the primitives are paired to enable pose transfer from source to target characters. Most existing works require the rigs to be exactly the same [47,48,46], e.g., a pre-defined human skeleton template. However, in practice, the rig definition is arbitrary and the rig topology could differ a lot [49]. Thus, most existing pose transfer methods [47,46,2,25] cannot be directly applied to a new character without a rig or with a rig in a different topology. While recent works [49,27] achieve automatic rigging and skinning for characters, their output, i.e., hierarchical skeletons, do not have correspondence across different characters and cannot be directly used for pose transfer.

To address the above issues, we propose a novel pose transfer method to predict correspondence-preserving deformation primitives, skinning weights, and rigid transformations to jointly deform the target character so that its pose is similar to the one of the source character.

Specifically, we define a character articulation model in terms of a set of deformation body parts but without commonly-used hierarchical structures. Each part can be deformed by its own rigid transformation based on skinning weights. Besides, since there is no hierarchical structure to connect the body parts, they can be associated with any part of the character regardless of the shape topol-

ogy. Given such an articulation model, we propose a novel deep learning based method to transfer poses between characters. It first generates skinning weights for both source and target characters, leading to a consistent segmentation of both into a set of corresponding deformation body parts. It then predicts a rigid transformation for each deformation body part. Finally, linear blending skinning (LBS) [23] is applied to deform the target mesh based on predicted rigid transformations. As the deformation body parts and their rigid transformations are automatically generated by the network by analyzing the source and target meshes, they tend to be robust and adaptive to very stylized characters.

A lack of diverse data makes it hard to train our network. Most public character datasets with ground truth pose transfer data only contain natural human shapes [30,19,54] or characters of limited shape varieties [1]. The datasets with stylized characters [49] contain a single mesh in the rest pose for each character. We propose a semi-supervised training mechanism based on cycle consistency [56] to ease the requirement of ground truth pose transfer data on the stylized characters. This makes it possible to use arbitrary static characters with various shapes and topologies in training to improve the robustness of the method. Overall, our contributions can be summarized as follows:

1. We propose the first automatic method for pose transfer between rig-free 3D characters with diverse shapes, topologies, and mesh connectivities.
2. Our method parameterises the character pose as a set of learned independent body part deformations coherent across characters. We do not require any manual intervention or preprocessing, e.g., rigging, skinning, or correspondence labeling.
3. Our model is trained end-to-end and in a semi-supervised manner. We do not require neither annotations nor mesh correspondences for training and can make use of large amounts of static characters.

2 Related Work

Skeleton-based Pose Transfer. Transferring poses based on skeletal rigs was intensively studied in the past. Gleicher [16] pioneered skeleton-based pose transfer through space-time optimization. Follow-up work [24,44,3,12,4] mostly incorporated various physics and kinematics constraints into this framework. Generalization to arbitrary objects has been proposed by [51,34], but they require example poses from users. Recent deep learning methods [47,46,26] trained neural networks with forward kinematics layers to directly estimate hierarchical transformations of the target skeleton. However, their models require the source and target skeletons to be the same while only allowing for different proportions. [6,25] fit a predefined template skeleton and derive the skinning weights for character pose transfer within the same category, e.g., humanoid, quadruped, etc. [32] relaxed the singular template constraint through a multi-resolution topology graph. [2] proposed skeleton-aware pooling operators which supports skeletal pose transfer with a different number of joints. Yet, these methods still require

the skeletons to be topologically equivalent. However, even these relaxed skeleton constraints cannot be guaranteed through state-of-the-art automated rigging and skinning methods [49,50,27]. Our method does not require skeletal rigging and can transfer poses across characters with different topologies.

Mesh Deformation Transfer. Character pose transfer can also be achieved by mesh deformation transfer without relying on rigs. Traditional methods [43,8,7,5] require accurate correspondences through manual annotation. [52] proposed a keypoint detection method to characterize the deformation, but still required user effort to specify corresponding keypoints from the source to target. Recent deep learning based methods analyzed the mesh deformation primitives and embedded the mesh into latent spaces [45,45,53]. However, their latent spaces are not shared across subjects and thus cannot be used for pose transfer between different characters. [15] trained a GAN-based model with unpaired data in two shape sets to perform pose transfer. But it is limited to shape deformation in the training set and does not generalize to unseen characters. [48,55] disentangled the shape identity and pose information and made it possible to transfer poses to unseen characters. However, they can only handle characters with limited shape varieties, e.g, human bodies with minimal clothing. Our method automatically generates consistent deformation body parts across different character meshes and deforms the target mesh with part-wise rigid transformations in an LBS manner. Hence, no manual correspondence is needed. Once trained, our network can generalize to unseen characters with various shapes and topologies.

Correspondence Learning. Correspondence learning is crucial in pose transfer and motion retargeting tasks [17,2,21,37,38,41,35]. [21,37,38] detected 2D keypoints on images as correspondences for human video reposing. [18,39,40] found corresponding regions and segments for pose transfer. [33] performed analogies on 2D character sprites and transferred animations between them. They worked on image domain by utilizing deep image features to locate corresponding body parts. [28,36,22] proposed unsupervised methods to discover corresponding 3D keypoints as deformation handles. They generate plausible shapes, e.g., chairs, airplanes, via shape deformation but are not suitable for character posture articulation. [41] found per-vertex correspondence between human meshes via correlation matrices. But its generalization is limited to shapes close to training data [10,11]. Our method discovers part-level shape correspondence by learning through the pose transfer task. It does not need correspondence annotation for supervision and can generalize to unseen stylized characters.

3 Method

3.1 Overview

Given a source 3D character mesh $\mathbf{V}^s$ with the desired pose and its mesh $\bar{\mathbf{V}}^s$ in rest pose and a different target 3D character mesh $\bar{\mathbf{V}}^t$ in rest pose, the goal of our

method is to deform the target mesh to a new pose $\hat{\mathbf{V}}^t$ which matches the input source pose $\{\mathbf{V}^s, \bar{\mathbf{V}}^s, \bar{\mathbf{V}}^t\} \mapsto \hat{\mathbf{V}}^t$. Here, we use the bar symbol $\bar{\mathbf{V}}$ to indicate the character in rest pose. To solve this problem, we propose an end-to-end neural network that learns part-level mesh correspondences and transformations between characters to achieve pose transfer. The overview is shown in Fig. 2.

We first define a character articulation model to represent the mesh deformation in terms of a set of deformation parts (Sec. 3.2). Unlike existing methods [2,6,47] requiring skeletal rigging to deform character body parts hierarchically, our model deforms body parts independently without the kinematic chain. Our method parameterises the character pose as a set of learned independent body part deformations coherent across characters, which is the foundation for the following pose transfer network to overcome topology constraints.

We propose a novel skeleton-free pose transfer network to predict the skinning weights and the transformations for each deformation part defined in the above character articulation model so that the target character can be transformed by linear blending skinning (LBS) to match the input pose (Sec. 3.3).

The pose transfer network consists of three modules: *skinning weight predictor*, *mesh encoder*, and *transformation decoder*. The skinning weight predictor estimates per-vertex skinning weights that segment the mesh into K deformation parts (see examples in Fig. 3). The *mesh encoder* encodes the input mesh into a latent feature that embeds both pose and shape information. The *transformation decoder* predicts a set of part-wise rigid transformations, which are further used to articulate the target mesh into the desired pose.

We train our framework end-to-end in a semi-supervised manner (Sec. 3.4). For characters with pairwise animation sequences [30,1], i.e. different subjects with the same animation poses, we train our network directly with cross-subject reconstruction. There also exist datasets with stylized characters of diverse shapes, topologies, and mesh connectivities. However, such data usually contains only a static rest pose and thus cannot be directly used in training. We propose a cycle-consistency loss to train on such data unsupervised, which turns out to improve our model robustness significantly.

3.2 Character Articulation Model

We propose to represent the mesh deformation in a unified way so that the pose can be easily transferred between characters with various shapes and topologies. We define K deformation parts for a mesh $\bar{\mathbf{V}}$ with N vertices. Each part can be deformed based on the skinning weight $\mathbf{W} \in \mathbb{R}^{N \times K}$ associated with it. The K deformation parts are not character-specific but consistent across characters (Fig. 3). $\mathbf{W}$ satisfies the partition of unity condition where $0 \leq w_{i,k} \leq 1$ and $\sum_{k=1}^{K} w_{i,k} = 1$. Here i is the vertex index and k is the deformation part index. Different characters may have different shapes and topologies, so ideally the number of deformation parts should vary. We define $K = 40$ as the maximum number of parts for all the characters in our experiment. Depending on the shape of the character, we allow some parts to be degenerate, i.e. having zero coverage:

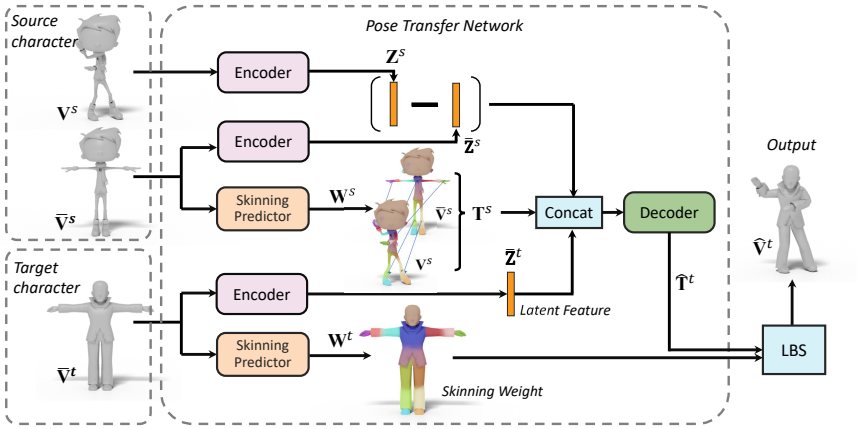

Fig. 2. Overview. Given a posed source character and a target character as input, the pose transfer network estimates character skinning weights and part-wise transformations which articulate the target character through LBS to match the pose of the source.

$w_{i,k} = 0, \forall i$. Meanwhile, the number of vertices N is not fixed and can vary from character to character during either training or testing phases.

Given rigid transformations for K body parts $\mathbf{T} = \{\mathbf{T}_1, ..., \mathbf{T}_K\}$, we use LBS [23] to deform the character mesh,

$$\mathbf{V}_i = \sum_{k=1}^{K} w_{i,k} \mathbf{T}_k (\bar{\mathbf{V}}_i - \mathbf{C}_k), \quad \forall \bar{\mathbf{V}}_i \in \bar{\mathbf{V}} \tag{1}$$

where $\mathbf{C}_k$ is the center of deformation part k in terms of the average position of vertices weighted by the skinning weight,

$$\mathbf{C}_k = \frac{\sum_{i=1}^{N} w_{i,k} \bar{\mathbf{V}}_i}{\sum_{i=1}^{N} w_{i,k}} \tag{2}$$

We do not connect the center of deformations parts $\mathbf{C}_k$ to form a skeleton since the skeleton connectivity varies in characters with different topology [50]. Our transformation $\mathbf{T}_k$ is applied independently on each deformation part without the kinematic chain. A consistent deformation part segmentation together with part-wise transformations forms a more general way of articulation than the commonly-used skeleton-based methods [2], which is crucial for achieving the skeleton-free pose transfer.

3.3 Skeleton-Free Pose Transfer Network

We propose a skeleton-free pose transfer network to transfer the pose from a source character to a different target character (see Fig. 2). It predicts skinning

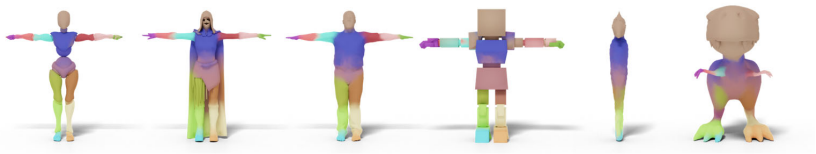

Fig. 3. Visualization of deformation parts based on predicted skinning weights. Each color represents a deformation part. The deformation part is semantically consistent across characters with various shapes and topologies.

weights of both source and target characters through the skinning weight predictor and estimates the corresponding transformation matrices jointly from the mesh encoder and transformation decoder network.

Skinning Weight Predictor Given a character mesh $\bar{\mathbf{V}}$, we design a graph convolution network g_s to predict its skinning weight $\mathbf{W} \in \mathbb{R}^{N \times K}$,

$$\mathbf{W} = g_s(f(\bar{\mathbf{V}}); \phi_s) \tag{3}$$

where $f(\bar{\mathbf{V}}) \in \mathbb{R}^{N \times 6}$ is the vertex feature vector consisting of position and normal. ϕ_s are learnable parameters. Each row of $\mathbf{W}$ indicates each vertex skinning weight association to K deformation parts. The network architecture follows [49] and can process meshes with arbitrary number of vertices and connectivities. We modify the last layer as a softmax layer to satisfy the skinning weight convex condition. The detailed structure can be found in the supplementary.

Mesh Encoder. We use another graph convolution network g_e to encode the input mesh $\mathbf{V}$ into a latent feature $\mathbf{Y} \in \mathbb{R}^{N \times C}$,

$$\mathbf{Y} = g_e(f(\mathbf{V}); \phi_e) \tag{4}$$

where C is the dimension of the latent space and ϕ_e are learnable parameters.

Instead of pooling $\mathbf{Y}$ into a global latent feature, we multiply it with the predicted skinning weight as an attention map to convert the feature dimension $\mathbb{R}^{N \times C} \mapsto \mathbb{R}^{K \times C}$. This conversion can be interpreted as an aggregation function to gather deformation part features from relevant vertices. After that, a 1D convolution layer is further applied to transform the feature to be the attended latent feature $\mathbf{Z} \in \mathbb{R}^{K \times C}$,

$$\mathbf{Z} = \text{Conv1d}(\mathbf{W}^\intercal \cdot \mathbf{Y}, \phi_c) \tag{5}$$

where ϕ_c are learnable parameters and $C = 128$ in our experiment. Note that the mesh encoder is applied to all three input meshes $\mathbf{V}^s, \bar{\mathbf{V}}^s, \bar{\mathbf{V}}^t$ to obtain their attended latent features $\mathbf{Z}^s, \bar{\mathbf{Z}}^s, \bar{\mathbf{Z}}^t$ with corresponding skinning weights.

Transformation Decoder. The goal of the decoder is to predict transformations $\hat{\mathbf{T}}^t = \{\hat{\mathbf{T}}_1^t, ..., \hat{\mathbf{T}}_K^t\}$ on each deformation part of the target mesh $\bar{\mathbf{V}}^t$. Hence, the target mesh $\bar{\mathbf{V}}^t$ can be reposed to $\hat{\mathbf{V}}^t$ which matches the desired pose mesh $\mathbf{V}^s$. The decoder takes as input three component:

- the latent feature of the target mesh $\bar{\mathbf{Z}}^t$. It is derived from the target mesh $\bar{\mathbf{V}}^t$ with the mesh encoder. It encodes the target mesh shape information.
- the difference between the latent features of the posed source mesh and itself in rest pose $\mathbf{Z}^s - \bar{\mathbf{Z}}^s$.
- the transformation of each deformation part $\mathbf{T}^s = \{\mathbf{T}_1^s, ..., \mathbf{T}_K^s\}$ between the pair of source meshes. This explicit transformation serves as an initial guess and helps the network focus on estimating residuals. It is analytically calculated by [9].

To summarize, the decoder takes the concatenation of $\mathbf{Z}^t, \mathbf{Z}^s - \bar{\mathbf{Z}}^s, \mathbf{T}^s$ as input and predicts the transformation $\hat{\mathbf{T}}^t$:

$$\hat{\mathbf{T}}^t = g_d(\bar{\mathbf{Z}}^t, \mathbf{Z}^s - \bar{\mathbf{Z}}^s, \mathbf{T}^s; \phi_d) \tag{6}$$

where ϕ_d are learnable parameters for the decoder. With the predicted skinning weights $\mathbf{W}^t$ and transformation $\hat{\mathbf{T}}^t$, we can use the proposed articulation model to deform the target mesh $\bar{\mathbf{V}}^t$ to the new pose $\hat{\mathbf{V}}^t$.

3.4 Training and Losses

We propose the following losses to train our network in a semi-supervised manner to make the best use of all possible data.

Mesh reconstruction loss. For characters with paired pose data in [30,1], we use a reconstruction loss as direct supervision. We apply a per-vertex L1 loss between the predicted mesh $\hat{\mathbf{V}}^t$ and the ground truth mesh $\mathbf{V}^t$,

$$L_{rec} = ||\hat{\mathbf{V}}^t - \mathbf{V}^t||_1 \tag{7}$$

Transformation loss. With the predicted skinning weight $\mathbf{W}^t$ of the target mesh, we group the vertices into K deformation parts by performing $\text{argmax}_k\, w_{i,k}$. Then we calculate the ground truth transformation $\mathbf{T}^t$ on the approximated parts between the input rest pose mesh $\bar{\mathbf{V}}^t$ and the ground truth mesh $\mathbf{V}^t$. We apply an L1 loss between the ground truth and predicted transformation $\hat{\mathbf{T}}^t$,

$$L_{trans} = ||\hat{\mathbf{T}}^t - \mathbf{T}^t||_1 \tag{8}$$

Cycle loss. When paired data are not available, or just a single rest pose mesh is provided, e.g., in [49], we use the cycle consistency loss for training [56]. Given a pair of source meshes $\mathbf{V}^s, \bar{\mathbf{V}}^s$ and a target mesh $\bar{\mathbf{V}}^t$ in rest pose, we first transfer the pose from source to target: $\{\mathbf{V}^s, \bar{\mathbf{V}}^s, \bar{\mathbf{V}}^t\} \mapsto \hat{\mathbf{V}}^t$, and then transfer the pose from the predicted target back to the source mesh: $\{\hat{\mathbf{V}}^t, \bar{\mathbf{V}}^t, \bar{\mathbf{V}}^s\} \mapsto \hat{\mathbf{V}}^s$. The predicted source mesh $\hat{\mathbf{V}}^s$ should be the same as $\mathbf{V}^s$. We apply L1 loss between them for the cycle reconstruction.

Through experiments, we found that training only with this loss leads to mode collapse. In existing datasets, characters with multiple poses are usually human characters, while most stylized characters are only in rest pose. These stylized characters can only be used as the target mesh $\bar{\mathbf{V}}^t$ in the cycle loss instead of interchangeably as $\mathbf{V}^s$. Thus the network tends to collapse and results in $\hat{\mathbf{V}}^t$ with limited pose variance. To solve this problem, we apply the transformations calculated from the source meshes $\mathbf{T}^s$ to the target mesh $\bar{\mathbf{V}}^t$ to obtain a pseudo-ground truth $\tilde{\mathbf{V}}^t$ for $\hat{\mathbf{V}}^t$. The complete cycle loss is

$$L_{cyc} = ||\hat{\mathbf{V}}^s - \mathbf{V}^s||_1 + w_{\text{pseudo}}||\hat{\mathbf{V}}^t - \tilde{\mathbf{V}}^t||_1 \tag{9}$$

where $w_{pseudo} = 0.3$ is used in our experiment. To note that $\hat{\mathbf{V}}^t$ is an approximated pseudo-ground truth mesh and sometimes may not be well-deformed if the transformation $\mathbf{T}^s$ is large. We introduce this term as a regularization which helps prevent the model from collapsing.

Skinning weight loss. In existing rigged character datasets [30,1,49], the skeletons and skinning weights are defined independently for each character. Therefore, we cannot use them directly as ground truth to supervise the training of our skinning weight predictor because of their lack of consistency. We thus propose a contrastive learning method to make use of such diverse skinning data. Our assumption is if two vertices belong to the same body part based on the ground truth skinning weight, they should also belong to the same deformation part in the predicted skinning. We select vertices with $w_{i,k} > 0.9,\ \exists k$ and use the KL divergence to enforce similarity between skinning weights of two vertices,

$$L_{skin} = \gamma_{i,j} \sum_{k=1}^{K} (w_{i,k} log(w_{i,k}) - w_{i,k} log(w_{j,k})) \tag{10}$$

where i, j indicate two randomly sampled vertices. γ is an indicator function: $\gamma_{i,j} = 1$ if vertices i and j belong to the same part in the ground truth skinning weight and $\gamma_{i,j} = -1$ if not. This loss holds only when the ground truth skinning is available.

Edge length loss. The desired deformation should be locally rigid and preserve the character shape, e.g., limb lengths and other surface features. Thus, we apply an edge length loss between the predicted mesh and input target mesh to prevent

undesired non-rigid deformations,

$$L_{edge} = \sum_{\{i,j\}\in\mathcal{E}} \mid ||\hat{\mathbf{V}}_i^t - \hat{\mathbf{V}}_j^t||_2 - ||\mathbf{V}_i^t - \mathbf{V}_j^t||_2 \mid \tag{11}$$

where $\mathcal{E}$ denotes the set of edges on the mesh.

4 Experiments

4.1 Datasets

We train our model on three datasets: AMASS [30], Mixamo [1] and RigNet [49]. We additionally use MGN [10] for evaluation.

AMASS [30] is a large human motion dataset that fits SMPL [29] to real-world human motion capture data. SMPL disentangles and parameterizes the human pose and shape. Therefore, we can obtain paired pose data by switching different shape parameters while keeping the pose parameter the same. We follow the train-test split protocol in [55].

Mixamo [1] contains over a hundred humanoid characters and over two thousand corresponding motion sequences. Since the motion sequences are shared across all characters, the paired pose data is also available. We use 98 characters for training and 20 characters for testing. The detailed split can be found in the supplementary.

RigNet [49] contains 2703 rigged characters with a large shape variety, including humanoids, quadrupeds, birds, fish, etc. All characters have their skeletal rigging. Each character only has one mesh in the rest pose. We remove character categories that can not be animated by the human pose, e.g., fish. We follow the train-test split protocol in [49] on the remaining 1268 characters.

MGN [10] contains 96 scanned clothed human registered to SMPL topology. We evaluation on this dataset to further demonstrate the robustness on human model with larger shape variety.

4.2 Comparison Methods

Pinocchio [6] is a pioneering work in automatic 3D character rigging and skinning. It fits a predefined skeleton to the target mesh and calculates skinning weights. As the skeleton is fixed, it achieves pose transfer by retargeting the joint rotations estimated by [9]. Pinocchio has a strict requirement on the mesh connectivity: non-watertight, multi-component, or non-manifold meshes are not

allowed. We manually preprocessed meshes to match its requirement in our evaluation. **Skeleton-aware Network (SAN)** [2] transfers the pose between two humanoid characters with the same skeleton topology. However, they require the motion statistics for both source and target characters to remap the predicted motion, e.g., an animation sample for the test subject mesh. This is not available in our task where only one instance of the subject character is provided. To make a fair comparison, we used the average statistics from the training set for test meshes. The ground truth skeleton and skinning is also assumed given for this method. **Neural Blend Shape (NBS)** [25] is the state-of-the-art method that achieves pose transfer between skeleton-free human shapes. They adopt SMPL skeleton template and can only work on human characters. **Shape Pose Disentanglement (SPD)** [55] can disentangle pose and shape information for 3D meshes from the same category. The pose transfer can be achieved by applying the pose information from one mesh to the other while keeping the shape information. Their model can only work on meshes with the same connectivity and thus we evaluate it only on SMPL-based dataset. **Ours (AMASS)** represents the reposing results from our proposed framework. It is trained only with AMASS data with limited character shapes. It is used for evaluating the generalization of the proposed network architecture. **Ours (full)** is our full result trained on all three datasets mentioned above.

4.3 Pose Transfer Evaluation

We evaluate our skeleton-free pose transfer results on different stylized characters both qualitatively and quantitatively.

Fig. 4 shows comparison results of the reposed characters from each of the comparison methods. SAN [2] fails on our task where only a single test mesh is given. It relies a lot on the motion statistics for test characters. Pinocchio [6] does not preserve the character shape well, e.g., the limbs have undesired non-rigid deformations. NBS [25] results in collapsed shapes and cannot generalize well to stylized characters. Our results match the source pose the best and work well on various character shapes. More visual comparisons and animation videos can be found in the supplementary.

Quantitatively, we use the Point-wise Mesh Euclidean Distance (PMD) [55,48] as the evaluation metric. We first evaluate the results on MGN dataset [10] with all competing methods (the first row of Table. 1). SAN [2] is not compared because it is trained on Mixamo [1] and cannot generalize to the unseen skeleton. SPD [55] trained only on naked human data fails to generalize to clothed human. NBS [25] directly trained on MGC and thus achieves relatively good result. Our full model achieves the best result by using all possible stylized characters.

In addition, we evaluate our method and competing methods on a more challenging dataset Mixamo [1], with more stylized test characters. The results are reported in the second row of Table. 1. SPD [55] is not compared since it can only handle meshes with the same mesh connectivity. All competing methods cannot generalize well to stylized characters in Mixamo and fail significantly in terms of PMD. Our full model results in the lowest PMD which demonstrates

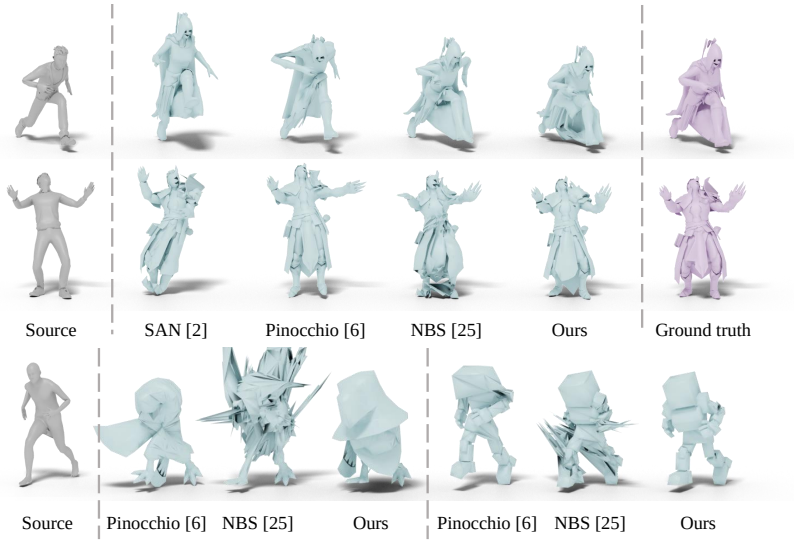

Fig. 4. Pose transfer results for human (top) and stylized characters (bottom).

	Pinocchio [6]	SAN [2]	NBS [25]	SPD [55]	Ours (AMASS)	Ours (full)
PMD ↓ on MGN [10]	3.145	-	1.498	5.649	2.878	**1.197**
PMD ↓ on Mixamo [1]	6.139	5.581	3.875	-	3.412	**2.393**

Table 1. Quantitative comparison of pose transfer results on MGN and Mixamo.

the performance of our model on more stylized characters. Our ablation model trained only on AMASS data also scores better than other methods. It shows the generalization of our method when being only trained on limited data.

4.4 Deformation Part Semantic Consistency

Our predicted deformation parts denote the same body region across different characters. Although this can be demonstrated by our pose transfer results, we further validate it by conducting an explicit semantic consistency evaluation.

In Sec. 3.4, we define the vertex belongings to each deformation part by selecting the vertex maximum skinning weight. Therefore, each deformation part can be defined as a mesh semantic part with a group of vertices belonging to it. Then our goal is to evaluate such semantic part consistency across subjects. Because existing ground truth annotation, i.e., traditional skeletal skinning weights, cannot be directly used for evaluation, we design an evaluation protocol similar to [13,22] for semantic part consistency. More specifically, first, we get our and ground truth mesh semantic parts based on the predicted and ground truth

Fig. 5. Visualization of deformation parts based on predicted skinning weights from each method (in row). Each part is denoted by a unique color.

	Pinocchio [6]	NBS [25]	Ours (AMASS)	Ours (full)
Pred → GT ↑ on Mixamo [1]	0.833	0.870	0.886	**0.993**
GT → Pred ↑ on Mixamo [1]	0.886	0.808	0.827	**0.947**

Table 2. Semantic consistency scores for deformation part prediction. We compare with Pinocchio [6] and NBS [25] on Mixamo [1] in both correlation directions.

skinning weights respectively. Second, we calculate the *correlation* from the semantic parts of prediction to the ones of ground truth: a predicted semantic part is *correlated* with the ground truth part with the most number of overlapped vertices. Then we can derive the *"consistency score"* of each predicted part as the maximum percentage of the correlated ground truth parts with the same semantics from all characters in the dataset. The final metric is the average of the consistency score over all predicted parts. We denote the above evaluation metric as Pred → GT since we find the correlation from predictions to ground truth. GT → Pred can be calculated in the opposite direction, i.e., for each ground truth part, we find the most correlated predicted part.

We compare our results with Pinocchio and NBS which rely on a fixed skeleton template and thus can predict skinning weights with the same semantic parts for different characters. The comparison result on Mixamo characters is shown in Table. 2. Our full model achieves the best and close to 1 correlation accuracy compared to others. Ours trained only on AMASS achieves a similar average performance to comparison methods. We note that NBS used predefined skeleton [29] for training, while ours is not supervised by any body part labels.

We also visualize the skinning weights predicted from ours and comparison methods in Fig. 5. For each method, we used consistent color for deformation parts to reflect the same semantic. Our skinning paints characters consistently on semantic regions while the comparison methods fail on some body parts.

4.5 Ablation Study

We conduct ablation studies on Mixamo dataset to investigate the effectiveness of each component. **w/o edge loss** is trained without the edge length constraint. **w/o pseudo** is trained without cycle loss from the pseudo-ground truth. **w/o skinning** is trained with out skinning weight loss. Table. 3 shows the evaluation results on Mixamo data. Our full model achieves the best performance.

	w/o edge loss	w/o pseudo	w/o skinning loss	Ours (full)
PMD ↓ on Mixamo [1]	2.450	2.601	2.978	**2.393**

Table 3. Quantitative evaluation results on ablation methods.

5 Conclusion and Future Work

We present a novel learning-based framework to automatically transfer poses between stylized 3D characters without skeletal rigging. Our model can handle characters with diverse shapes, topologies, and mesh connectivities. We achieve this by representing the characters in a unified articulation model and predicting the deformation skinning and transformations when given the desired pose. Our model can utilize all types of existing character data, e.g., with motion or static, and thus can have great generalization to various unseen stylized characters.

Limitations and Future Work. Our model focuses on pose transfer and is not optimized for motion transfer in the temporal domain. Jittering and penetration problems [46] may occur when using our proposed method for animation. We apply the edge length constraint to prevent the mesh from breaking but no other explicit controls are involved. Data-driven deformation constraints [14] and better geometric regularizations [42] could prevent the implausible deformations further. Our current framework requires the rest pose mesh as an additional input. Canonicalization methods [31] might be helpful to ease this requirement. We are looking for automating the process of the character pose transfer, yet in real content authoring scenarios, user input is still desired to increase tool usability and content diversity. We look forward to future endeavors on expressive pose transfer animations with intuitive user controls.

Acknowledgement. This work is funded by a gift from Adobe Research and the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) - 409792180 (Emmy Noether Programme, project: Real Virtual Humans). Gerard Pons-Moll is a member of the Machine Learning Cluster of Excellence, EXC number 2064/1 – Project number 390727645.

References

1. Mixamo, https://www.mixamo.com (Feb 2022), https://www.mixamo.com
2. Aberman, K., Li, P., Lischinski, D., Sorkine-Hornung, O., Cohen-Or, D., Chen, B.: Skeleton-aware networks for deep motion retargeting. ACM Transactions on Graphics (TOG) **39**(4), 62–1 (2020)
3. Al Borno, M., Righetti, L., Black, M.J., Delp, S.L., Fiume, E., Romero, J.: Robust physics-based motion retargeting with realistic body shapes. In: Computer Graphics Forum. vol. 37, pp. 81–92. Wiley Online Library (2018)
4. Aristidou, A., Lasenby, J.: Fabrik: A fast, iterative solver for the inverse kinematics problem. Graphical Models **73**(5), 243–260 (2011)
5. Avril, Q., Ghafourzadeh, D., Ramachandran, S., Fallahdoust, S., Ribet, S., Dionne, O., de Lasa, M., Paquette, E.: Animation setup transfer for 3d characters. In: Computer Graphics Forum. vol. 35, pp. 115–126. Wiley Online Library (2016)
6. Baran, I., Popović, J.: Automatic rigging and animation of 3d characters. ACM Transactions on graphics (TOG) **26**(3), 72–es (2007)
7. Baran, I., Vlasic, D., Grinspun, E., Popović, J.: Semantic deformation transfer. In: ACM SIGGRAPH 2009 papers, pp. 1–6 (2009)
8. Ben-Chen, M., Weber, O., Gotsman, C.: Spatial deformation transfer. In: Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation. pp. 67–74 (2009)
9. Besl, P.J., McKay, N.D.: Method for registration of 3-d shapes. In: Sensor fusion IV: control paradigms and data structures (1992)
10. Bhatnagar, B.L., Tiwari, G., Theobalt, C., Pons-Moll, G.: Multi-garment net: Learning to dress 3d people from images. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 5420–5430 (2019)
11. Bogo, F., Romero, J., Loper, M., Black, M.J.: Faust: Dataset and evaluation for 3d mesh registration. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3794–3801 (2014)
12. Choi, K.J., Ko, H.S.: Online motion retargetting. The Journal of Visualization and Computer Animation **11**(5), 223–235 (2000)
13. Fernandez-Labrador, C., Chhatkuli, A., Paudel, D.P., Guerrero, J.J., Demonceaux, C., Gool, L.V.: Unsupervised learning of category-specific symmetric 3d keypoints from point sets. In: European Conference on Computer Vision. pp. 546–563. Springer (2020)
14. Gao, L., Lai, Y.K., Yang, J., Zhang, L.X., Xia, S., Kobbelt, L.: Sparse data driven mesh deformation. IEEE TVCG (2019)
15. Gao, L., Yang, J., Qiao, Y.L., Lai, Y.K., Rosin, P.L., Xu, W., Xia, S.: Automatic unpaired shape deformation transfer. ACM Transactions on Graphics (TOG) **37**(6), 1–15 (2018)
16. Gleicher, M.: Retargetting motion to new characters. In: Proceedings of the 25th annual conference on Computer graphics and interactive techniques. pp. 33–42 (1998)
17. 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). pp. 230–246 (2018)
18. Hung, W.C., Jampani, V., Liu, S., Molchanov, P., Yang, M.H., Kautz, J.: Scops: Self-supervised co-part segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 869–878 (2019)

19. Ionescu, C., Papava, D., Olaru, V., Sminchisescu, C.: Human3. 6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. IEEE transactions on pattern analysis and machine intelligence **36**(7), 1325–1339 (2013)
20. Jacobson, A., Deng, Z., Kavan, L., Lewis, J.P.: Skinning: Real-time shape deformation (full text not available). In: ACM SIGGRAPH 2014 Courses, pp. 1–1 (2014)
21. Jakab, T., Gupta, A., Bilen, H., Vedaldi, A.: Unsupervised learning of object landmarks through conditional image generation. Advances in neural information processing systems **31** (2018)
22. Jakab, T., Tucker, R., Makadia, A., Wu, J., Snavely, N., Kanazawa, A.: Keypointdeformer: Unsupervised 3d keypoint discovery for shape control. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12783–12792 (2021)
23. Kavan, L.: Direct skinning methods and deformation primitives. In: ACM SIGGRAPH Courses (2014)
24. 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. pp. 39–48 (1999)
25. 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), 1–15 (2021)
26. Lim, J., Chang, H.J., Choi, J.Y.: Pmnet: Learning of disentangled pose and movement for unsupervised motion retargeting. In: BMVC. vol. 2, p. 7 (2019)
27. Liu, L., Zheng, Y., Tang, D., Yuan, Y., Fan, C., Zhou, K.: Neuroskinning: Automatic skin binding for production characters with deep graph networks. ACM TOG (2019)
28. Liu, M., Sung, M., Mech, R., Su, H.: Deepmetahandles: Learning deformation meta-handles of 3d meshes with biharmonic coordinates. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12–21 (2021)
29. Loper, M., Mahmood, N., Romero, J., Pons-Moll, G., Black, M.J.: Smpl: A skinned multi-person linear model. ACM transactions on graphics (TOG) **34**(6), 1–16 (2015)
30. Mahmood, N., Ghorbani, N., Troje, N.F., Pons-Moll, G., Black, M.J.: AMASS: Archive of motion capture as surface shapes. In: International Conference on Computer Vision. pp. 5442–5451 (Oct 2019)
31. Musoni, P., Marin, R., Melzi, S., Castellani, U.: Reposing and retargeting unrigged characters with intrinsic-extrinsic transfer. Smart Tools and Applications in Graphics (2021)
32. Poirier, M., Paquette, E.: Rig retargeting for 3d animation. In: Graphics interface. pp. 103–110 (2009)
33. Reed, S.E., Zhang, Y., Zhang, Y., Lee, H.: Deep visual analogy-making. Proc. NeurIPS (2015)
34. Rhodin, H., Tompkin, J., Kim, K.I., de Aguiar, E., Pfister, H., Seidel, H.P., Theobalt, C.: Generalizing wave gestures from sparse examples for real-time character control. ACM Trans. Graph. **34**(6), 1–12 (Oct 2015)
35. Saito, S., Yang, J., Ma, Q., Black, M.J.: Scanimate: Weakly supervised learning of skinned clothed avatar networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2886–2897 (2021)

36. Shi, R., Xue, Z., You, Y., Lu, C.: Skeleton merger: an unsupervised aligned keypoint detector. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 43–52 (2021)
37. Siarohin, A., Lathuilière, S., Tulyakov, S., Ricci, E., Sebe, N.: Animating arbitrary objects via deep motion transfer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2377–2386 (2019)
38. Siarohin, A., Lathuilière, S., Tulyakov, S., Ricci, E., Sebe, N.: First order motion model for image animation. Advances in Neural Information Processing Systems **32** (2019)
39. Siarohin, A., Roy, S., Lathuilière, S., Tulyakov, S., Ricci, E., Sebe, N.: Motion-supervised co-part segmentation. arXiv preprint arXiv:2004.03234 (2020)
40. Siarohin, A., Woodford, O.J., Ren, J., Chai, M., Tulyakov, S.: Motion representations for articulated animation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13653–13662 (2021)
41. Song, C., Wei, J., Li, R., Liu, F., Lin, G.: 3d pose transfer with correspondence learning and mesh refinement. Advances in Neural Information Processing Systems **34** (2021)
42. Sorkine, O., Alexa, M.: As-rigid-as-possible surface modeling. In: Symposium on Geometry Processing (2007)
43. Sumner, R.W., Popović, J.: Deformation transfer for triangle meshes. ACM Transactions on graphics (TOG) **23**(3), 399–405 (2004)
44. Tak, S., Ko, H.S.: A physically-based motion retargeting filter. ACM Transactions on Graphics (TOG) **24**(1), 98–117 (2005)
45. 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. pp. 5841–5850 (2018)
46. Villegas, R., Ceylan, D., Hertzmann, A., Yang, J., Saito, J.: Contact-aware retargeting of skinned motion. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9720–9729 (2021)
47. Villegas, R., Yang, J., Ceylan, D., Lee, H.: Neural kinematic networks for unsupervised motion retargetting. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 8639–8648 (2018)
48. Wang, J., Wen, C., Fu, Y., Lin, H., Zou, T., Xue, X., Zhang, Y.: Neural pose transfer by spatially adaptive instance normalization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5831–5839 (2020)
49. Xu, Z., Zhou, Y., Kalogerakis, E., Landreth, C., Singh, K.: Rignet: Neural rigging for articulated characters. arXiv preprint arXiv:2005.00559 (2020)
50. Xu, Z., Zhou, Y., Kalogerakis, E., Singh, K.: Predicting animation skeletons for 3d articulated models via volumetric nets. In: 3DV (2019)
51. Yamane, K., Ariki, Y., Hodgins, J.: Animating non-humanoid characters with human motion data. In: Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation. pp. 169–178 (2010)
52. Yang, J., Gao, L., Lai, Y.K., Rosin, P.L., Xia, S.: Biharmonic deformation transfer with automatic key point selection. Graphical Models **98**, 1–13 (2018)
53. Yang, J., Gao, L., Tan, Q., Huang, Y., Xia, S., Lai, Y.K.: Multiscale mesh deformation component analysis with attention-based autoencoders. arXiv preprint arXiv:2012.02459 (2020)
54. Zhang, X., Bhatnagar, B.L., Starke, S., Guzov, V., Pons-Moll, G.: Couch: Towards controllable human-chair interactions. In: European Conference on Computer Vision (ECCV). Springer (October 2022)

55. Zhou, K., Bhatnagar, B.L., Pons-Moll, G.: Unsupervised shape and pose disentanglement for 3d meshes. In: European Conference on Computer Vision. pp. 341–357. Springer (2020)
56. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: Proceedings of the IEEE international conference on computer vision. pp. 2223–2232 (2017)