Zero-shot Pose Transfer for Unrigged Stylized 3D Characters

Jiashun Wang[1*] Xueting Li[2] Sifei Liu[2] Shalini De Mello[2]
Orazio Gallo[2] Xiaolong Wang[3] Jan Kautz[2]
[1]Carnegie Mellon University [2]NVIDIA [3]UC San Diego

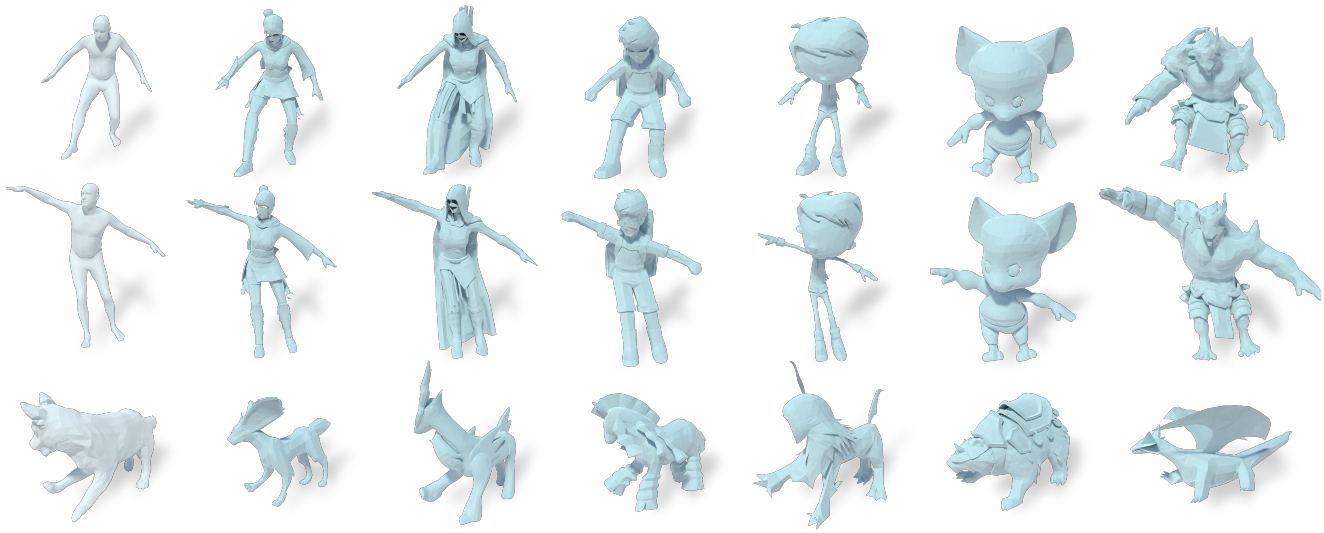

Figure 1. Our algorithm transfers the pose of a reference avatar (source) to stylized characters. Unlike existing methods, at training time our approach needs only the mesh of the source avatar in rest and desired pose, and the mesh of the stylized characater only in rest pose.

Abstract

Transferring the pose of a reference avatar to stylized 3D characters of various shapes is a fundamental task in computer graphics. Existing methods either require the stylized characters to be rigged, or they use the stylized character in the desired pose as ground truth at training. We present a zero-shot approach that requires only the widely available deformed non-stylized avatars in training, and deforms stylized characters of significantly different shapes at inference. Classical methods achieve strong generalization by deforming the mesh at the triangle level, but this requires labelled correspondences. We leverage the power of local deformation, but without requiring explicit correspondence labels. We introduce a semi-supervised shape-understanding module to bypass the need for explicit correspondences at test time, and an implicit pose deformation module that deforms individual surface points to match the target pose. Furthermore, to encourage realistic and accurate deformation of stylized characters, we introduce an efficient volume-based test-time training procedure. Because it does not need rigging, nor the deformed stylized character at training time, our model generalizes to categories with scarce annotation, such as stylized quadrupeds. Extensive experiments demonstrate the effectiveness of the proposed method compared to the state-of-the-art approaches trained with comparable or more supervision. Our project page is available at https://jiashunwang.github.io/ZPT/

1. Introduction

Stylized 3D characters , such as those in Fig. 1, are commonly used in animation, movies, and video games. Deforming these characters to mimic natural human or animal poses has been a long-standing task in computer graphics. Different from the 3D models of natural humans and animals, stylized 3D characters are created by professional artists through imagination and exaggeration. As a result, each stylized character has a distinct skeleton, shape, mesh

[*]Work done during Jiashun Wang's internship at NVIDIA.

topology, and usually include various accessories, such as a cloak or wings (see Fig. 1). These variations hinder the process of matching the pose of a stylized 3D character to that of a reference avatar, generally making manual rigging a requirement. Unfortunately, rigging is a tedious process that requires manual effort to create the skeleton and skinning weights for each character. Even when provided with manually annotated rigs, transferring poses from a source avatar onto stylized characters is not trivial when the source and target skeletons differ. Automating this procedure is still an open research problem and is the focus of many recent works [2, 4, 24, 52]. Meanwhile, non-stylized 3D humans and animals have been well-studied by numerous prior works [35, 40, 54, 62, 68]. A few methods generously provide readily available annotated datasets [11, 12, 41, 68], or carefully designed parametric models [40, 51, 68]. By taking advantage of these datasets [12, 41], several learning-based methods [7, 14, 35, 62, 67] disentangle and transfer poses between human meshes using neural networks. However, these methods (referred to as "part-level" in the following) carry out pose transfer by either globally deforming the whole body mesh [14, 22, 47, 67] or by transforming body parts [35, 48], both of which lead to overfitting on the training human meshes and fail to generalize to stylized characters with significantly different body part shapes. Interestingly, classical mesh deformation methods [55, 56] (referred to as "local" in the following) can transfer poses between a pair of meshes with significant shape differences by computing and transferring per-triangle transformations through correspondence. Though these methods require manual correspondence annotation between the source and target meshes, they provide a key insight that by transforming individual triangles instead of body parts, the mesh deformation methods are more agnostic to a part's shape and can generalize to meshes with different shapes.

We marry the benefits of learning-based methods [7, 14, 35, 62, 67] with the classic local deformation approach [55] and present a model for unrigged, stylized character deformation guided by a non-stylized biped or quadruped avatar. Notably, our model only requires easily accessible posed human or animal meshes for training and can be directly applied to deform 3D stylized characters with a significantly different shape at inference. To this end, we implicitly operationalize the key insight from the local deformation method [55] by modeling the shape and pose of a 3D character with a correspondence-aware shape understanding module and an implicit pose deformation module. The shape understanding module learns to predict the part segmentation label (*i.e.*, the coarse-level correspondence) for each surface point, besides representing the shape of a 3D character as a latent shape code. The pose deformation module is conditioned on the shape code and deforms individual surface point guided by a target pose code sampled from a prior pose latent space [50]. Furthermore, to encourage realistic deformation and generalize to rare poses, we propose a novel volume-based test-time training procedure that can be efficiently applied to unseen stylized characters.

During inference, by mapping biped or quadruped poses from videos, in addition to meshes to the prior pose latent space using existing works [32, 51, 53], we can transfer poses from different modalities onto unrigged 3D stylized characters. Our main contributions are:

- We propose a solution to a practical and challenging task – learning a model for stylized 3D character deformation with only posed human or animal meshes.
- We develop a correspondence-aware shape understanding module, an implicit pose deformation module, and a volume-based test-time training procedure to generalize the proposed model to unseen stylized characters and arbitrary poses in a zero-shot manner.
- We carry out extensive experiments on both humans and quadrupeds to show that our method produces more visually pleasing and accurate deformations compared to baselines trained with comparable or more supervision.

2. Related Work

Deformation Transfer. Deformation transfer is a long-standing problem in the computer graphics community [3, 6, 8, 9, 55, 65]. Sumner *et al*. [55] apply an affine transformation to each triangle of the mesh to solve an optimization problem that matches the deformation of the source mesh while maintaining the shape of the target mesh. Ben-Chen *et al*. [9] enclose the source and target shapes with two cages and transfer the Jacobians of the source deformation to the target shape. However, these methods need tedious human efforts to annotate the correspondence between the source and target shapes. More recently, several deep learning methods are developed to solve the deformation transfer task. However, they either require manually providing the correspondence [66] or cannot generalize [14, 22, 67] to stylized characters with different shapes. Gao *et al*. [22] propose a VAE-GAN based method to leverage the cycle consistency between the source and target shapes. Nonetheless, it can only work on shapes used in training. Wang *et al*. [62] introduce conditional normalization used in style transfer for 3D deformation transfer. But the method is limited to clothed-humans and cannot handle the large shape variations of stylized characters.

We argue that these learning-based methods cannot generalize to stylized characters because they rely on encoding their global information (*e.g.*, body or parts), which is different from traditional works that focus on local deformation, *e.g.*, the affine transformation applied to each triangle in [55]. Using a neural network to encode the global information easily leads to overfitting. For example, models

trained on human meshes cannot generalize to a stylized humanoid character. At the same time, early works only focus on local information and cannot model global information such as correspondence between the source and target shapes, which is why they all need human effort to annotate the correspondence. Our method tries to learn the correspondence and deform locally at the same time.

Skeleton-based Pose Transfer. Besides mesh deformation transfer, an alternative way to transfer pose is to utilize skeletons. Motion retargeting is also a common name used for transferring poses from one motion sequence to another. Gleicher *et al*. [24] propose a space-time constrained solver aiming to satisfy the kinematics-level constraints and to preserve the characters' original identity. Following works [5, 19, 33] try to solve inverse-kinematics or inverse rate control to achieve pose transfer. There are also dynamics-based methods [4, 59] that consider physics during the retargeting process. Recently, learning-based methods [20, 27, 37, 60, 61] train deep neural networks to predict the transformation of the skeleton. Aberman *et al*. [2] propose a pooling-based method to transfer poses between meshes with different skeletons.

All these works highly rely on the skeleton for pose transfer. Other works try to estimate the rigging of the template shape [7, 39, 52, 63, 64] when a skeleton is not available. But if the prediction of the skinning weights fails, the retargeting fails as well. Liao *et al*. [36] propose a model that learns to predict the skinning weights and pose transfer jointly using ground truth skinning weights and paired motion data as supervision, which limits the generalization of this method to categories where annotations are more scarce compared to humans (*e.g*., quadrupeds). Instead, our method uses posed human or animal meshes for training and deforms stylized characters of different shapes at inference.

Implicit 3D shape representation. Implicit 3D shape representations have shown great success in reconstructing static shapes [13,16,18,21,23,29,42,43,49] and deformable ones [10, 28, 34, 44–48, 58]. DeepSDF [49] proposes to use an MLP to predict the signed distance field (SDF) value of a query point in 3D space, where a shape code is jointly optimized in an auto-decoding manner. Occupancy flow [45] generalizes the Occupancy Networks [42] to learn a temporally and spatially continuous vector field with a NeuralODE [15]. Inspired by parameteric models, NPMs [47] disentangles and represents the shape and pose of dynamic humans by learning an implicit shape and pose function, respectively. Different from these implicit shape representation works that focus on reconstructing static or deformable meshes, we further exploit the inherent continuity and locality of implicit functions to deform stylized characters to match a target pose in a zero-shot manner.

3. Method

We aim to transfer the pose of a biped or quadruped avatar to an unrigged, stylized 3D character. We tackle this problem by modeling the shape and pose of a 3D character using a correspondence-aware shape understanding module and an implicit pose deformation module, inspired by classical mesh deformation methods [55, 56]. The shape understanding module (Sec. 3.1, Fig. 2) predicts a latent shape code and part segmentation label of a 3D character in rest pose, while the pose deformation module (Sec. 3.2, Fig. 3) deforms the character in the rest pose given the predicted shape code and a target pose code. Moreover, to produce natural deformations and generalize to rare poses unseen at training, we introduce an efficient volume-based test-time training procedure (Sec 3.3) for unseen stylized characters. All three modules, trained only with posed, unclothed human meshes, and unrigged, stylized characters in a rest pose, are directly applied to unseen stylized characters at inference. We explain our method for humans, and describe how we extend it to quadrupeds in Sec. 4.6.

3.1. Correspondence-Aware Shape Understanding

Given a 3D character in rest pose, we propose a shape understanding module to represent its shape information as a latent code, and to predict a body part segmentation label for each surface point.

To learn a representative shape code, we employ an implicit auto-decoder [47, 49] that reconstructs the 3D character taking the shape code as input. During training, we jointly optimize the shape code of each training sample and the decoder. Given an unseen character (*i.e*., a stylized 3D character) during inference, we obtain its shape code by freezing the decoder and optimizing the shape code to reconstruct the given character. Specifically, as shown in Fig. 2, given the concatenation of a query point $x \in \mathbb{R}^3$ and the shape code $s \in \mathbb{R}^d$, we first obtain an embedding $e \in \mathbb{R}^d$ via an MLP denoted as $\mathcal{F}$. Conditioned on the embedding e, the occupancy $\hat{o}_x \in \mathbb{R}$ of x is then predicted by another MLP denoted as $\mathcal{O}$. The occupancy indicates if the query point x is inside or outside the body surface and can be supervised by the ground truth occupancy as:

$$\mathcal{L}_{\mathcal{O}} = -\sum_{x}(o_x \cdot log(\hat{o}_x) + (1 - o_x) \cdot log(1 - \hat{o}_x)), \quad (1)$$

where o_x is the ground truth occupancy at point x.

Since our shape code eventually serves as a condition for the pose deformation module, we argue that it should also capture the part correspondence knowledge across different instances, in addition to the shape information (*e.g*., height, weight, and shape of each body part). This insight has been utilized by early local mesh deformation method [55], which explicitly utilizes correspondence to transfer local transformations between the source and target meshes. Our

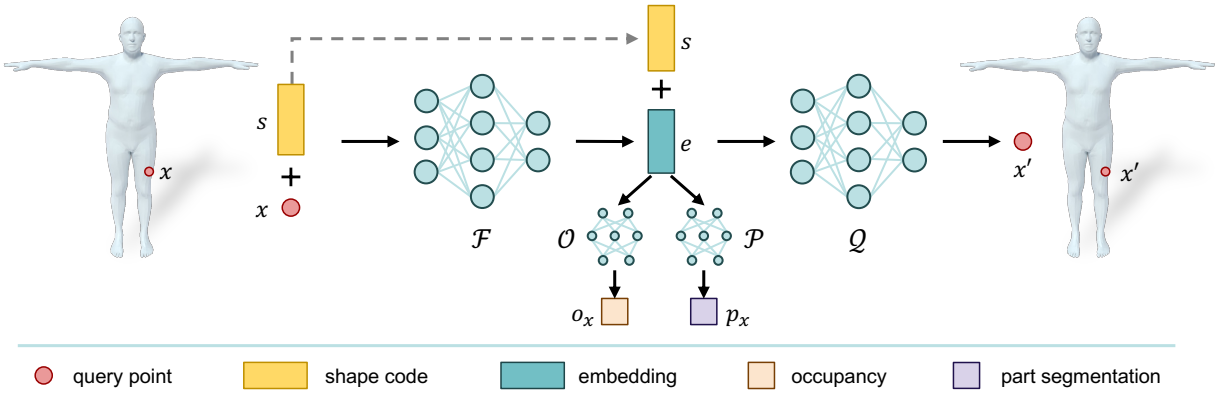

Figure 2. **The shape understanding module (Sec. 3.1).** Given a query point and a learnable shape code, we take MLPs to predict the occupancy, part segmentation label and further use an inverse MLP to regress the query point.

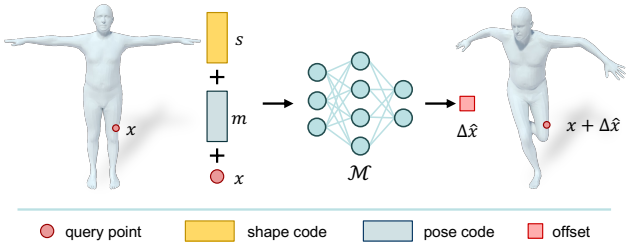

Figure 3. **The pose deformation module (Sec. 3.2).** Given a query point on the surface, the learned shape code and a target pose code, we use an MLP to predict the offset of the query point.

pose deformation process could also benefit from learning part correspondence. Take the various headgear, hats, and horns on the stylized characters's heads in Fig. 1 as an example. If these components can be "understood" as extensions of the character's heads by their shape codes, they will move smoothly with the character's heads during pose deformation. Thus, besides mesh reconstruction, we effectively task our shape understanding module with an additional objective: predicting part-level correspondence instantiated as the part segmentation label. Specifically, we propose to utilize an MLP $\mathcal{P}$ to additionally predict a part label $p_x = (p_x^1, ..., p_x^K)^T \in \mathbb{R}^K$ for each surface point x. Thanks to the densely annotated human mesh dataset, we can also supervise part segmentation learning with ground truth labels via:

$$\mathcal{L}_{\mathcal{P}} = \sum_x (-\sum_{k=1}^{K} \mathbb{1}_x^k log(p_x^k)), \tag{2}$$

where K is the total number of body parts, and $\mathbb{1}_x^k = 1$ if x belongs to the k^{th} part and $\mathbb{1}_x^k = 0$ otherwise.

To prepare the shape understanding module for stylized characters during inference, besides unclothed human meshes, we also include *unrigged* 3D stylized characters in rest pose during training. These characters in rest pose are easily accessible and do not require any annotation. For shape reconstruction, Eq. 1 can be similarly applied to the stylized characters. However, as there is no part segmentation annotation for stylized characters, we propose a self-supervised inverse constraint inspired by correspondence learning methods [17,38] to facilitate part segmentation prediction on these characters. Specifically, we reconstruct the query point's coordinates from the concatenation of the shape code s and the embedding e through an MLP $\mathcal{Q}$ and add an auxiliary objective as:

$$\mathcal{L}_{\mathcal{Q}} = ||\mathcal{Q}(s, e) - x||^2. \tag{3}$$

Intuitively, for stylized characters without part annotation, the model learned without this objective may converge to a trivial solution where similar embeddings are predicted for points with the same occupancy value, even when they are far away from each other, and belong to different body parts. Tab. 4 further quantitatively verifies the effectiveness of this constraint. Beyond facilitating shape understanding, the predicted part segmentation label is further utilized in the volume-based test-time training module which will be introduced in Sec. 3.3.

3.2. Implicit Pose Deformation Module

Given the learned shape code and a target pose, the pose deformation module deforms each surface point of the character to match the target pose. In the following, we first describe how we represent a human pose and then introduce the implicit function used for pose deformation.

Instead of learning a latent pose space from scratch as in [36, 47], we propose to represent a human pose by the corresponding pose code in the latent space of VPoser [51]. Our intuition is that VPoser is trained with an abundance of posed humans from the large-scale AMASS dataset [41]. This facilitates faster training and provides robustness to overfitting. Furthermore, human poses can be successfully estimated from different modalities (*e.g.*, videos or meshes), and mapped to the latent space of VPoser by existing methods [32, 51, 53]. By taking advantage of these works, our

model can be applied to transfer poses from various modalities to an unrigged stylized character without any additional effort. A few examples can be found in the supplementary.

To deform a character to match the given pose, we learn a neural implicit function $\mathcal{M}$ that takes the sampled pose code $m \in \mathbb{R}^{32}$, the learned shape code, and a query point x around the character's surface as inputs and outputs the offset (denoted as $\Delta\hat{x} \in \mathbb{R}^3$) of x in 3D space. Given the densely annotated human mesh dataset, we directly use the ground truth offset Δx as supervision. The training objective for our pose deformation module is defined as:

$$\mathcal{L}_{\mathcal{D}} = \sum_{x} ||\Delta\hat{x} - \Delta x||^2. \tag{4}$$

Essentially, our implicit pose deformation module is similar in spirit to early local mesh deformation methods [55] and has two key advantages compared to the part-level pose transfer methods [22, 36, 62]. First, our implicit pose deformation network is agnostic to mesh topology and resolution. Thus our model can be directly applied to unseen 3D stylized characters with significantly different resolutions and mesh topology compared to the training human meshes during inference. Second, stylized characters often include distinct body part shapes compared to humans. For example, the characters shown in Fig. 1 include big heads or various accessories. Previous part-level methods [36] that learn to predict a bone transformation and skinning weight for each body part usually fail on these unique body parts, since they are different from the corresponding human body parts used for training. In contrast, by learning to deform individual surface point, implicit functions are more agnostic to the overall shape of a body part and thus can generalize better to stylized characters with significantly different body part shapes. Fig. 4 and Fig. 6 show these advantages.

3.3. Volume-based Test-time Training

The shape understanding and pose deformation modules discussed above are trained with only posed human meshes and unrigged 3D stylized characters in rest pose. When applied to unseen characters with significantly different shapes, we observe surface distortion introduced by the pose deformation module. Moreover, it is challenging for the module to fully capture the long tail of the pose distribution. To resolve these issues, we propose to apply test-time training [57] and fine-tune the pose deformation module on unseen stylized characters.

To encourage natural pose deformation, we further propose a volume-preserving constraint during test-time training. Our key insight is that preserving the volume of each part in the rest pose mesh during pose deformation results in less distortion [35, 62]. However, it is non-trivial to compute the precise volume of each body part, which can have complex geometry. Instead, we propose to preserve the Euclidean distance between pairs of vertices sampled from the surface of the mesh, as a proxy for constraining the volume. Specifically, given a mesh in rest pose, we randomly sample two points x_i^c and x_j^c on the surface within the same part c using the part segmentation prediction from the shape understanding module. We calculate the offset of these two points $\Delta\hat{x}_i^c$ and $\Delta\hat{x}_j^c$ using our pose deformation module and minimize the change in the distance between them by:

$$\mathcal{L}_v = \sum_c \sum_i \sum_j (||x_i^c - x_j^c|| - ||(x_i^c + \Delta\hat{x}_i^c) - (x_j^c + \Delta\hat{x}_j^c)||)^2. \tag{5}$$

By sampling a large number of point pairs within a part and minimizing Eq. 5, we can approximately maintain the volume of each body part during pose deformation.

Furthermore, in order to generalize the pose deformation module to long-tail poses that are rarely seen during training, we propose to utilize the source character in rest pose and its deformed shape as paired training data during test-time training. Specifically, we take the source character in rest pose, its target pose code, and its optimized shape code as inputs and we output the movement $\Delta\hat{x}^{dr}$, where x^{dr} is a query point from the source character. We minimize the L2 distance between the predicted movement $\Delta\hat{x}^{dr}$ and the ground truth movement Δx^{dr},

$$\mathcal{L}_{dr} = \sum_{x^{dr}} ||\Delta\hat{x}^{dr} - \Delta x^{dr}||^2. \tag{6}$$

Besides the volume-preserving constraint and the reconstruction of the source character, we also employ the edge loss $\mathcal{L}_e$ used in [25, 36, 62]. Overall, the objectives for the test-time training procedure are $\mathcal{L}_{\mathcal{T}} = \lambda_v\mathcal{L}_v + \lambda_e\mathcal{L}_e + \lambda_{dr}\mathcal{L}_{dr}$, where λ_v, λ_e, and λ_{dr} are hyper-parameters balancing the loss weights.

4. Experiments

4.1. Datasets

To train the shape understanding module, we use 40 human meshes sampled from the SMPL [40] parametric model. We use both the occupancy and part segmentation label of these meshes as supervision (see Sec. 3.1). To generalize the shape understanding module to stylized characters, we further include 600 stylized characters from RigNet [63]. Note that we *only* use the rest pose mesh (*i.e.*, occupancy label) of the characters in [63] for training. To train our pose deformation module, we construct paired training data by deforming each of the 40 SMPL characters discussed above with 5000 pose codes sampled from the VPoser's [50] latent space. In total, we collect 200,000 training pairs, with each pair including an unclothed human mesh in rest pose and the same human mesh in target pose.

After training the shape understanding and pose deformation modules, we test them on the Mixamo [1] dataset,

which includes challenging stylized characters, and the MGN [11] dataset, which includes clothed humans. The characters in both datasets have different shapes compared to the unclothed SMPL meshes we used for training, demonstrating the generalization ability of the proposed method. Following [36], we test on 19 stylized characters, with each deformed by 28 motion sequences from the Mixamo dataset. For the MGN dataset, we test on 16 clothed characters, with each deformed by 200 target poses. Both the testing characters and poses are unseen during training.

For quadrupeds, since there is no dataset including large-scale paired stylized quadrupeds for quantitative evaluation, we split all characters from the SMAL [68] dataset and use the first 34 shapes (*i.e.*, cats, dogs, and horses) for training. We further collect 81 stylized quadrupeds in rest pose from the RigNet [63] to improve generalization of the shape understanding module. Similarly to the human category, we use occupancy and part segmentation supervision for the SMAL shapes and only the occupancy supervision for RigNet meshes. To train the pose deformation module, we deform each of the 34 characters in SMAL by 2000 poses sampled from the latent space of BARC [54], a 3D reconstruction model trained for the dog category. We quantitatively evaluate our model on the hippo meshes from the SMAL dataset, which have larger shape variance compared to the cats, dogs, and horses used for training. We produce the testing data by deforming each hippo mesh with 500 unseen target poses from SMAL [68]. We show qualitative pose transfer on stylized quadrupeds in Fig. 1.

4.2. Implementation Details

We use the ADAM [30] optimizer to train both the shape understanding and pose deformation modules. For the shape understanding module, we use a learning rate of $1e-4$ for both the decoder and shape code optimization, with a batch size of 64. Given a new character at inference time, we fix the decoder and only optimize the shape code for the new character with the same optimizer and learning rate. For the pose deformation module, we use a learning rate of $3e-4$ with a batch size of 128. For test-time training, we use a batch size of 1 and a learning rate of $5e-3$ with the ADAM optimizer. We set λ_v, λ_e, and λ_{dr} (See Sec. 3.3) as 0.05, 0.01, and 1 respectively.

4.3. Metrics and Baselines for Comparison

Metrics. We use Point-wise Mesh Euclidean Distance (PMD) [36, 62] to evaluate pose transfer error. The PMD metric reveals pose similarity of the predicted deformation compared to its ground truth. However, as shown in Fig. 4, PMD can not fully show the smoothness and realism of the generated results. Thus, we adopt an edge length score (ELS) metric to evaluate the character's smoothness after the deformation. Specifically, we compare each edge's length in the deformed mesh with the corresponding edge's length in the ground truth mesh. We define the score as

$$\frac{1}{|\mathcal{E}|}\sum_{\{i,j\}\sim\mathcal{E}} 1-\left|\frac{||\hat{V}_i-\hat{V}_j||_2}{||V_i-V_j||_2}-1\right|, \tag{7}$$

where $\mathcal{E}$ indicates all edges of the mesh, $|\mathcal{E}|$ is the number of the edges in the mesh. $\hat{V}_i$ and $\hat{V}_j$ are the vertices in the deformed mesh. V_i and V_j are the vertices in the ground truth mesh. For all the evaluation metrics, we scale the template character to be 1 meter tall, following [36].

Dataset	Metric	SPT*(full) [36]	NBS [35]	SPT [36]	Ours
MGN [11]	PMD ↓	1.62	1.33	1.82	0.99
	ELS ↑	0.86	0.70	0.85	0.89
Mixamo [1]	PMD ↓	3.05	7.04	5.29	5.06
	ELS ↑	0.61	0.66	0.59	0.88

Table 1. **Quantitative comparison on MGN and Mixamo.** Our method achieves the lowest PMD with the highest ELS. We provide the performance of the SPT*(full) method, which uses more supervision than the other methods as a reference. Our method is even better or comparable to it.

Baselines. We compare our method with Neural Blend Shapes (NBS) [35] and Skeleton-free Pose Transfer (SPT) [36]. NBS is a rigging prediction method trained on the SMPL and MGN datasets, which include naked and clothed human meshes with ground truth rigging information. For SPT, we show the results of two versions, one is trained only on the AMASS dataset, named SPT, which has a comparable level of supervision to our method. We also test the SPT*(full) version, which is trained on the AMASS, RigNet and Mixamo datasets, using both stylized characters' skinning weights as supervision and paired stylized characters in rest pose and target pose.

4.4. Human-like Character Pose Transfer

We report the PMD metric on the MGN and Mixamo datasets in Tab. 1. We also include the performance of SPT*(full) for reference. On the MGN dataset which includes clothed humans, our method which is trained with only unclothed humans achieve the best PMD score than all baseline methods, including baselines trained with more supervision (*i.e.*, the NBS [35] learned with clothed humans and the SPT*(full) [36] learned with skinning weight and paired motion data). For the stylized characters, our method outperforms the SPT baseline learned with a comparable amount of supervision and gets competitive results with the NBS [35] and SPT*(full) baseline trained with more supervision. Furthermore, when testing on the more challenging, less human-like characters (*e.g.*, a mouse with a big head in Fig. 1), the baselines produce noticeable artifacts and rough surfaces, which can be observed in the qualitative comparisons in Fig. 4. We provide the PMD value for each character in the supplementary.

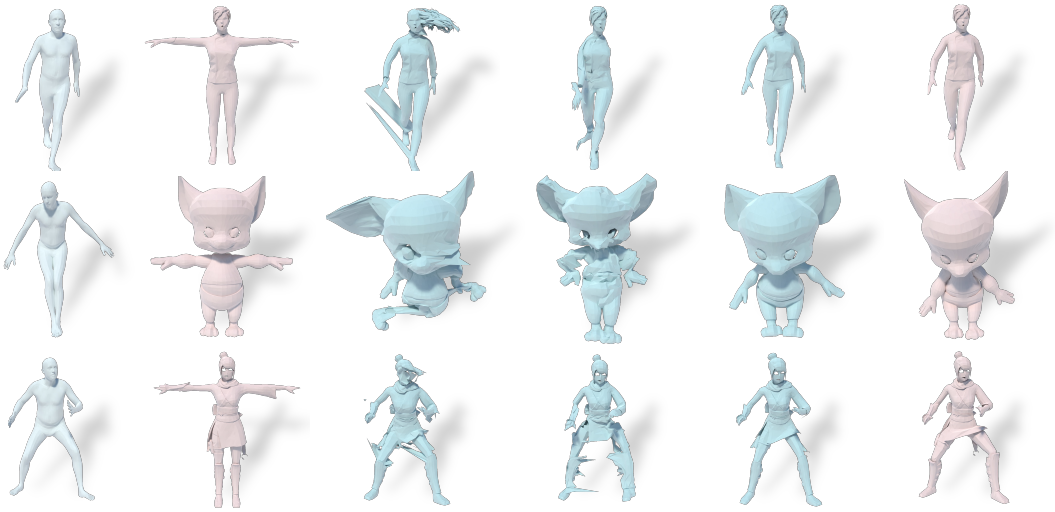

Figure 4. **Qualitative comparison on Mixamo.** The average PMD of these three results for NBS, SPT, and Ours are 8.16, 6.13, and 5.16 respectively and the average ELS for NBS, SPT, and Ours are 0.65, 0.78, and 0.93 respectively. Our method can successfully transfer the pose to challenging stylized characters (e.g., the mouse with a big head in the second row).

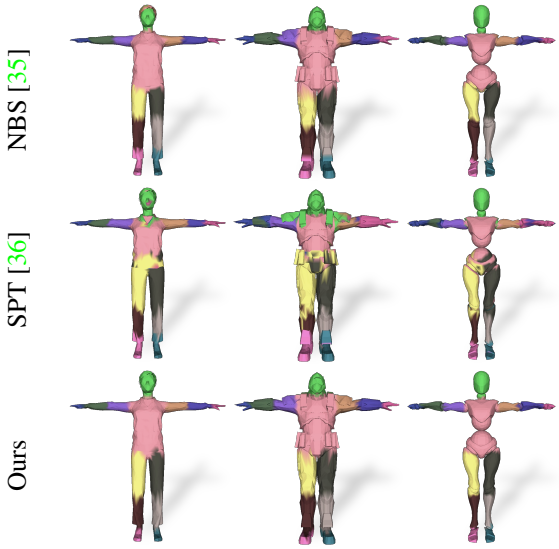

Figure 5. **Part segmentation visualization.** NBS makes wrong predictions for hair while SPT may mix the upper legs.

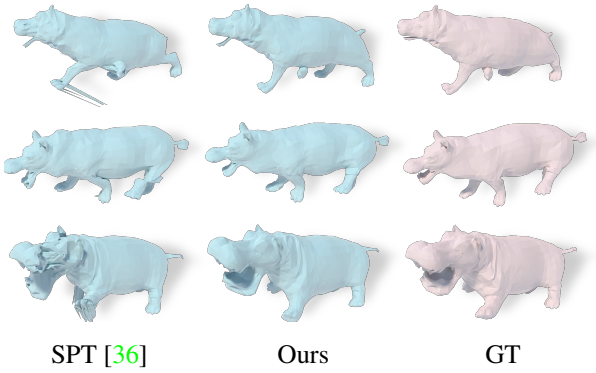

Figure 6. **Quadrupedal pose transfer visualization.** Our method can achieve smooth and accurate pose transfer while SPT fails on the mouth and leg regions.

Metric	NBS [35]	SPT [36]	Ours
Accuracy ↑	67.8%	75.6%	86.9%

Table 2. **Part prediction accuracy on Mixamo [1].** Our method achieves the best part segmentation accuracy.

We show the ELS score comparison of different methods on the MGN and Mixamo datasets in Tab. 1. For both clothed humans and stylized characters, our method can generate more realistic results which are consistent with the target mesh and achieves the best ELS score.

We visually compare our method and the baseline methods in Fig. 4 on the Mixamo dataset. Although NBS is trained with a clothed-human dataset, when testing on the human-like characters, it still fails on parts that are separate from the body such as the hair and the pants. When using only naked human meshes as supervision, SPT cannot generalize to challenging human-like characters, producing rough mesh surface with spikes.

4.5. Part Understanding Comparison

As discussed in Sec. 3.1, part segmentation plays an important role in both shape understanding and pose deformation. Though NBS [35] and SPT [36] do not explicitly predict part segmentation label, they are both skinning weight-based methods and we can derive the part segmentation label from the predicted skinning weights. Specifically, by selecting the maximum weight of each vertex, we can convert the skinning weight prediction to part segmentation labels for the vertices. We compare our part prediction results with those derived from SPT and NBS. We report the part segmentation accuracy on the Mixamo datasets in Tab. 2

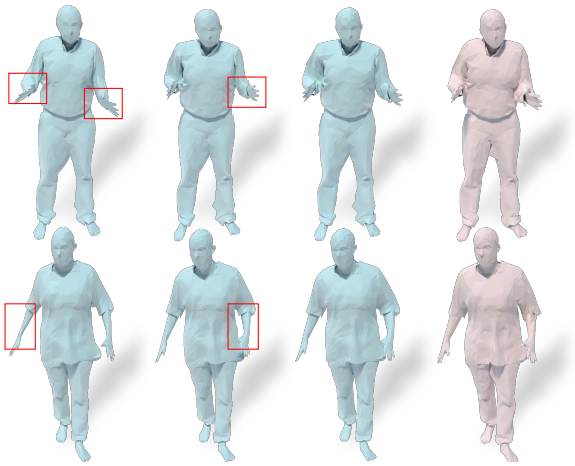

Figure 7. **Qualitative comparison for ablation study.** Removing the constraint (eq. 1) in shape understanding leads to wrong pose deformation results. The volume preserving loss (eq. 5) helps to maintain the identity, *e.g.*, the thickness of the arms in first row.

Metric	SPT [36]	Ours	Metric	SPT [36]	Ours
PMD ↓	10.28	8.28	ELS ↑	0.28	0.86

Table 3. **Comparison on Hippos from SMAL [68].** Our method achieves better pose transfer accuracy with more smooth results.

and visualize the part segmentation results in Fig. 5. Even trained with only part segmentation supervision of human meshes, our method can successfully segment each part for the stylized characters. On the contrary, SPT uses graph convolution network [31] to predict the skinning weights. When training only with human meshes, it often fails to distinguish different parts. As shown in Fig. 5, it mixes up the right and left upper legs, and incorrectly classifies the shoulder as the head. Though NBS is trained with clothed humans, it always classifies human hair as the human body for characters from Mixamo. This is because that NBS uses the MeshCNN [26] as the shape encoder. As a result, it is sensitive to mesh topology and cannot generalize to meshes with disconnected parts (*e.g.*, disconnected hair and head). Tab. 2 further quantitatively demonstrates that our method achieves the best part segmentation accuracy, demonstrating its ability to correctly interpret the shape and part information in stylized characters.

4.6. Quadrupedal Pose Transfer Comparison

To further show the generalization ability of our method, we conduct experiments on quadrupeds. We report the PMD and ELS score of our method and the SPT [36] in Tab. 3. When testing on hippos with large shape gap from the training meshes, SPT has a hard time generalizing both in terms of pose transfer accuracy and natural deformation. While our method achieves both better qualitative and quantitative results. We visualize the qualitative comparisons in Fig. 6. SPT produces obvious artifacts on the hippo's mouth and legs, while our method achieves accurate pose transfer and maintains the shape characteristics of the original character at the same time. We provide more results in the supplementary. We also show the part segmentation results on stylized characters by our method in Fig. 8. Even for unique parts such as the hats and antlers, our method correctly assigns them to the head part.

Figure 8. **Part prediction on stylized quadrupeds.** Our method successfully predicts the parts of unseen stylized quadrupeds.

Metric	Ours w/o inv	Ours w/o volume	Ours
PMD ↓	1.26	1.02	0.99
ELS ↑	0.88	0.88	0.89

Table 4. **Ablation study on inverse MLP and volume preserving loss.** The inverse MLP and volume preserving loss helps to improve pose transfer accuracy and produce smooth deformation.

4.7. Ablation Study

To evaluate the key components of our method, we conduct ablation studies on the MGN dataset by removing the inverse constraint (Eq. 3) in the shape understanding module and the volume-preserving loss (Eq. 5) used during the test-time training produce, we name them as "ours w/o inv" and "ours w/o v" respectively. We report the PMD and ELS metrics in Tab. 4. The model learned without the inverse constraint or volume-preserving loss has worse PMD and ELS score than our full model, indicating the contribution of these two objectives. We also provide qualitative results in Fig. 7. We use red boxes to point out the artifacts. As shown in Fig. 7, our model trained without the inverse constraint produces less accurate pose transfer results. Moreover, adding the volume-preserving loss helps to maintain the character's local details such as the thickness of the arms.

5. Conclusion

In this paper, we present a model that deforms unrigged, stylized characters guided by a biped or quadruped avatar. Our model is trained with only easily accessible posed human or animal meshes, yet can be applied to unseen stylized characters in a zero-shot manner during inference. To this end, we draw key insights from classic mesh deformation method and develop a correspondence-aware shape understanding module, an implicit pose deformation module and a volume-based test-time training procedure. We carry out extensive experiments on both the biped and quadruped category and show that our method produces more realistic and accurate deformation compared to baselines learned with comparable or more supervision.

References

[1] Mixamo. http://www.mixamo.com/. Accessed on November 09th, 2022. 5, 6, 7

[2] Kfir Aberman, Peizhuo Li, Dani Lischinski, Olga Sorkine-Hornung, Daniel Cohen-Or, and Baoquan Chen. Skeleton-aware networks for deep motion retargeting. In *ACM Transactions on Graphics (SIGGRAPH)*, 2020. 2, 3

[3] Noam Aigerman, Kunal Gupta, Vladimir G Kim, Siddhartha Chaudhuri, Jun Saito, and Thibault Groueix. Neural jacobian fields: Learning intrinsic mappings of arbitrary meshes. *arXiv preprint arXiv:2205.02904*, 2022. 2

[4] Mazen Al Borno, Ludovic Righetti, Michael J Black, Scott L Delp, Eugene Fiume, and Javier Romero. Robust physics-based motion retargeting with realistic body shapes. In *Computer Graphics Forum*. Wiley Online Library, 2018. 2, 3

[5] Andreas Aristidou and Joan Lasenby. FABRIK: A fast, iterative solver for the inverse kinematics problem. *Graphical Models*, 2011. 3

[6] Quentin Avril, Donya Ghafourzadeh, Srinivasan Ramachandran, Sahel Fallahdoust, Sarah Ribet, Olivier Dionne, Martin de Lasa, and Eric Paquette. Animation setup transfer for 3D characters. In *Computer Graphics Forum*, 2016. 2

[7] Ilya Baran and Jovan Popović. Automatic rigging and animation of 3D characters. In *ACM Transactions on Graphics (SIGGRAPH)*, 2007. 2, 3

[8] Ilya Baran, Daniel Vlasic, Eitan Grinspun, and Jovan Popović. Semantic deformation transfer. In *ACM Transactions on Graphics (ToG)*. 2009. 2

[9] Mirela Ben-Chen, Ofir Weber, and Craig Gotsman. Spatial deformation transfer. In *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2009. 2

[10] Bharat Lal Bhatnagar, Cristian Sminchisescu, Christian Theobalt, and Gerard Pons-Moll. Combining implicit function learning and parametric models for 3D human reconstruction. In *European Conference on Computer Vision (ECCV)*, 2020. 3

[11] Bharat Lal Bhatnagar, Garvita Tiwari, Christian Theobalt, and Gerard Pons-Moll. Multi-garment net: Learning to dress 3D people from images. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2, 6

[12] Federica Bogo, Javier Romero, Matthew Loper, and Michael J Black. FAUST: Dataset and evaluation for 3D mesh registration. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 2

[13] Rohan Chabra, Jan E Lenssen, Eddy Ilg, Tanner Schmidt, Julian Straub, Steven Lovegrove, and Richard Newcombe. Deep local shapes: Learning local sdf priors for detailed 3D reconstruction. In *European Conference on Computer Vision (ECCV)*, 2020. 3

[14] Haoyu Chen, Hao Tang, Henglin Shi, Wei Peng, Nicu Sebe, and Guoying Zhao. Intrinsic-extrinsic preserved gans for unsupervised 3D pose transfer. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2

[15] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in Neural Information Processing Systems (NeurIPS)*, 2018. 3

[16] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 3

[17] An-Chieh Cheng, Xueting Li, Min Sun, Ming-Hsuan Yang, and Sifei Liu. Learning 3D dense correspondence via canonical point autoencoder. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 4

[18] Julian Chibane, Thiemo Alldieck, and Gerard Pons-Moll. Implicit functions in feature space for 3D shape reconstruction and completion. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 3

[19] Kwang-Jin Choi and Hyeong-Seok Ko. Online motion retargetting. *Comput. Animat. Virtual Worlds*, 2000. 3

[20] Brian Delhaisse, Domingo Esteban, Leonel Rozo, and Darwin Caldwell. Transfer learning of shared latent spaces between robots with similar kinematic structure. In *International Joint Conference on Neural Networks (IJCNN)*, 2017. 3

[21] Philipp Erler, Paul Guerrero, Stefan Ohrhallinger, Niloy J Mitra, and Michael Wimmer. Points2surf learning implicit surfaces from point clouds. In *European Conference on Computer Vision (ECCV)*, 2020. 3

[22] Lin Gao, Jie Yang, Yi-Ling Qiao, Yu-Kun Lai, Paul L Rosin, Weiwei Xu, and Shihong Xia. Automatic unpaired shape deformation transfer. *ACM Transactions on Graphics (ToG)*, 2018. 2, 5

[23] Kyle Genova, Forrester Cole, Daniel Vlasic, Aaron Sarna, William T Freeman, and Thomas Funkhouser. Learning shape templates with structured implicit functions. In *IEEE International Conference on Computer Vision (ICCV)*, 2019. 3

[24] Michael Gleicher. Retargetting motion to new characters. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, 1998. 2, 3

[25] Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. 3D-CODED: 3D correspondences by deep deformation. In *European Conference on Computer Vision (ECCV)*, 2018. 5

[26] Rana Hanocka, Amir Hertz, Noa Fish, Raja Giryes, Shachar Fleishman, and Daniel Cohen-Or. MeshCNN: a network with an edge. *ACM Transactions on Graphics (ToG)*, 2019. 8

[27] Hanyoung Jang, Byungjun Kwon, Moonwon Yu, Seong Uk Kim, and Jongmin Kim. A variational U-Net for motion retargeting. In *Comput. Animat. Virtual Worlds*, 2020. 3

[28] Boyan Jiang, Yinda Zhang, Xingkui Wei, Xiangyang Xue, and Yanwei Fu. Learning compositional representation for 4D captures with neural ode. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 3

[29] Chiyu Jiang, Avneesh Sud, Ameesh Makadia, Jingwei Huang, Matthias Nießner, Thomas Funkhouser, et al. Local implicit grid representations for 3D scenes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 3

[30] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations (ICLR)*, 2015. 6

[31] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*, 2017. 8

[32] Muhammed Kocabas, Nikos Athanasiou, and Michael J Black. Vibe: Video inference for human body pose and shape estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2, 4

[33] Jehee Lee and Sung Yong Shin. 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. 3

[34] Jiahui Lei and Kostas Daniilidis. CaDeX: Learning canonical deformation coordinate space for dynamic surface representation via neural homeomorphism. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 3

[35] Peizhuo Li, Kfir Aberman, Rana Hanocka, Libin Liu, Olga Sorkine-Hornung, and Baoquan Chen. Learning skeletal articulations with neural blend shapes. In *ACM Transactions on Graphics (SIGGRAPH)*, 2021. 2, 5, 6, 7

[36] Zhouyingcheng Liao, Jimei Yang, Jun Saito, Gerard Pons-Moll, and Yang Zhou. Skeleton-free pose transfer for stylized 3D characters. In *European Conference on Computer Vision (ECCV)*, 2022. 3, 4, 5, 6, 7, 8

[37] Jongin Lim, Hyung Jin Chang, and Jin Young Choi. PMnet: Learning of disentangled pose and movement for unsupervised motion retargeting. In *British Machine Vision Conference (BMVC)*, 2019. 3

[38] Feng Liu and Xiaoming Liu. Learning implicit functions for topology-varying dense 3D shape correspondence. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 4

[39] Lijuan Liu, Youyi Zheng, Di Tang, Yi Yuan, Changjie Fan, and Kun Zhou. Neuroskinning: Automatic skin binding for production characters with deep graph networks. *ACM Transactions on Graphics (ToG)*, 2019. 3

[40] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. SMPL: A skinned multi-person linear model. *ACM Transactions on Graphics (ToG)*, 2015. 2, 5

[41] Naureen Mahmood, Nima Ghorbani, Nikolaus F. Troje, Gerard Pons-Moll, and Michael J. Black. AMASS: Archive of motion capture as surface shapes. In *IEEE International Conference on Computer Vision (ICCV)*, 2019. 2, 4

[42] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3D reconstruction in function space. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 3

[43] Mateusz Michalkiewicz, Jhony K Pontes, Dominic Jack, Mahsa Baktashmotlagh, and Anders Eriksson. Deep level sets: Implicit surface representations for 3D shape inference. *arXiv preprint arXiv:1901.06802*, 2019. 3

[44] Marko Mihajlovic, Yan Zhang, Michael J Black, and Siyu Tang. LEAP: Learning articulated occupancy of people. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 3

[45] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Occupancy flow: 4D reconstruction by learning particle dynamics. In *IEEE International Conference on Computer Vision (ICCV)*, 2019. 3

[46] Atsuhiro Noguchi, Umar Iqbal, Jonathan Tremblay, Tatsuya Harada, and Orazio Gallo. Watch it move: Unsupervised discovery of 3D joints for re-posing of articulated objects. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 3

[47] Pablo Palafox, Aljaž Božič, Justus Thies, Matthias Nießner, and Angela Dai. NPMs: Neural parametric models for 3D deformable shapes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2, 3, 4

[48] Pablo Palafox, Nikolaos Sarafianos, Tony Tung, and Angela Dai. SPAMs: Structured implicit parametric models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2, 3

[49] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. DeepSDF: Learning continuous signed distance functions for shape representation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 3

[50] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed AA Osman, Dimitrios Tzionas, and Michael J Black. Expressive body capture: 3D hands, face, and body from a single image. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2, 5

[51] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed A. A. Osman, Dimitrios Tzionas, and Michael J. Black. Expressive body capture: 3D hands, face, and body from a single image. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2, 4

[52] Martin Poirier and Eric Paquette. Rig retargeting for 3d animation. In *Proceedings of the Graphics Interface 2009 Conference*, 2009. 2, 3

[53] Davis Rempe, Tolga Birdal, Aaron Hertzmann, Jimei Yang, Srinath Sridhar, and Leonidas J Guibas. Humor: 3D human motion model for robust pose estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2, 4

[54] Nadine Rüegg, Silvia Zuffi, Konrad Schindler, and Michael J Black. BARC: Learning to regress 3D dog shape from images by exploiting breed information. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2, 6

[55] Robert W Sumner and Jovan Popović. Deformation transfer for triangle meshes. *ACM Transactions on Graphics (ToG)*, 2004. 2, 3, 5

[56] Robert W Sumner, Johannes Schmid, and Mark Pauly. Embedded deformation for shape manipulation. In *ACM Transactions on Graphics (SIGGRAPH)*. 2007. 2, 3

[57] Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *International Conference on Machine Learning (ICML)*, 2020. 5

[58] Ramana Sundararaman, Gautam Pai, and Maks Ovsjanikov. Implicit field supervision for robust non-rigid shape matching. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part III*, pages 344–362. Springer, 2022. 3

[59] Seyoon Tak and Hyeong-Seok Ko. A physically-based motion retargeting filter. In *ACM Transactions on Graphics (ToG)*, 2005. 3

[60] Ruben Villegas, Duygu Ceylan, Aaron Hertzmann, Jimei Yang, and Jun Saito. Contact-aware retargeting of skinned motion. In *IEEE International Conference on Computer Vision (ICCV)*, 2021. 3

[61] Ruben Villegas, Jimei Yang, Duygu Ceylan, and Honglak Lee. Neural kinematic networks for unsupervised motion retargetting. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 3

[62] Jiashun Wang, Chao Wen, Yanwei Fu, Haitao Lin, Tianyun Zou, Xiangyang Xue, and Yinda Zhang. Neural pose transfer by spatially adaptive instance normalization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2, 5, 6

[63] Zhan Xu, Yang Zhou, Evangelos Kalogerakis, Chris Landreth, and Karan Singh. RigNet: Neural rigging for articulated characters. In *ACM Transactions on Graphics (SIGGRAPH)*, 2020. 3, 5, 6

[64] Zhan Xu, Yang Zhou, Evangelos Kalogerakis, and Karan Singh. Predicting animation skeletons for 3D articulated models via volumetric nets. In *International Conference on 3D Vision*, 2019. 3

[65] Jie Yang, Lin Gao, Yu-Kun Lai, Paul L Rosin, and Shihong Xia. Biharmonic deformation transfer with automatic key point selection. *Graphical Models*, 2018. 2

[66] Wang Yifan, Noam Aigerman, Vladimir G Kim, Siddhartha Chaudhuri, and Olga Sorkine-Hornung. Neural cages for detail-preserving 3D deformations. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[67] Keyang Zhou, Bharat Lal Bhatnagar, and Gerard Pons-Moll. Unsupervised shape and pose disentanglement for 3D meshes. In *European Conference on Computer Vision (ECCV)*, 2020. 2

[68] Silvia Zuffi, Angjoo Kanazawa, David W Jacobs, and Michael J Black. 3D Menagerie: Modeling the 3D shape and pose of animals. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2, 6, 8