Neural Human Deformation Transfer

Jean Basset, Adnane Boukhayma, Stefanie Wuhrer, Franck Multon, Edmond Boyer

Neural Human Deformation Transfer

Jean Basset [*] jean.basset@inria.fr
Adnane Boukhayma [†] adnane.boukhayma@inria.fr
Stefanie Wuhrer [*] stefanie.wuhrer@inria.fr
Franck Multon [†] fmulton@irisa.fr
Edmond Boyer [*] edmond.boyer@inria.fr

Abstract

We consider the problem of human deformation transfer, where the goal is to retarget poses between different characters. Traditional methods that tackle this problem assume a human pose model to be available and transfer poses between characters using this model. In this work, we take a different approach and transform the identity of a character into a new identity without modifying the character's pose. This offers the advantage of not having to define equivalences between 3D human poses, which is not straightforward as poses tend to change depending on the identity of the character performing them, and as their meaning is highly contextual. To achieve the deformation transfer, we propose a neural encoder-decoder architecture where only identity information is encoded and where the decoder is conditioned on the pose. We use pose independent representations, such as isometry-invariant shape characteristics, to represent identity features. Our model uses these features to supervise the prediction of offsets from the deformed pose to the result of the transfer. We show experimentally that our method outperforms state-of-the-art methods both quantitatively and qualitatively, and generalises better to poses not seen during training. We also introduce a fine-tuning step that allows to obtain competitive results for extreme identities, and allows to transfer simple clothing.

[*]Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP (Institute of Engineering Univ. Grenoble Alpes), LJK, 38000 Grenoble, France
[†]Univ Rennes, Inria, CNRS IRISA, M2S, France

1 Introduction

Deformation transfer is the process of retargeting poses between characters: Given a source character in a deformed pose, and a target character in a reference pose, the objective is to generate a new shape of the target character in the deformed pose e.g. [30]. It finds interest in digital content creation where it has the potential to drastically reduce animation costs. This is particularly relevant when applied to human shapes, where it can ease animation production from captured motions and enable 3D and 4D data augmentation for data-driven applications.

Deformation transfer between humans requires poses to be defined consistently across different characters. This relies on assumptions on human identities and poses, which are inherently entangled notions. Human poses can be assumed to be identifiable in a coherent way over different characters. In that case, deformation transfer boils down to transferring the given deformed pose to the target character, and the result can thus be seen as the target identity with a new pose. Such a transfer can be achieved through a shared continuous pose parameterisation e.g. [2, 25] or through a discrete correspondence map, as with style transfer e.g. [27]. Another approach is to assume that the human shape can be characterised independently of the pose, i.e. by its identity. In this case, retargeting can be performed by transferring the identity from the target to the source characters, and the result can be seen as the source deformed pose with a new identity.

Both interpretations of the deformation transfer problem are arguably approximations since exact correspondences between character poses are subjective and since human shapes are not fully independent of the pose. They

anyway enable practical solutions as illustrated in the literature. While numerous methods have been proposed to solve the pose transfer problem (e.g. [30, 16, 17, 21, 6]), few works address the alternative solution with identity transfer [7]. However the latter exhibits advantages, in particular the ability to better adapt to any pose by directly considering the correct pose and just modifying identity shape properties.

In this paper we investigate the identity transfer strategy with a data-driven approach. We propose a deep learning architecture that predicts the deformation of the source model so that its identity matches that of the target model. The architecture of the model consists of an encoder that encodes the identity of the target model into a low-dimensional feature vector, and a decoder that consumes the identity feature vector along with the source model and predicts offsets from the source model that transfer the identity. To encode identity information, losses based on classical assumptions of human deformations are used, namely losses based on the hypotheses that two models sharing the same identity should be near-isometric [13] and have body parts that deform near-rigidly between the poses [29]. To structure the latent space, we use a loss that aims to map feature vectors of the same identity to the same location in latent space.

We train our architecture in a weakly supervised way: while we rely on the presence of identity labels for all training data, we only require pose labels for a small subset since 3D models with different characters performing the exact same pose are rare in existing datasets. To have access to high-quality labelled data, we propose an extension of the FAUST dataset [8] that includes additional poses and identities with full label information. We demonstrate experimentally that having access to full label information, and hence a reconstruction loss, on a small proportion of the training data is sufficient to train our architecture.

Inspired by the few-shot learning of generative models literature (e.g. [38, 3, 18]), we observe that fine-tuning our feed forward network at test time improves the results. This is achieved with a few extra training steps on the inputs using a self-supervised loss. In this strategy, the initial network training can be seen as a meta-learning stage, and the fine-tuning can be interpreted as one-shot learning from a single reference pose / target identity pair, which adapts the network weights further to that specific case. To the best of our knowledge, this is the first learning-based deformation transfer work that explores such an idea. Not only does the fine-tuning improve our performance quantitatively, but it also allows us to successfully transfer identity for out of training distribution shapes, such as a shape of a simply clothed person with a hat and a backpack, while the training consists merely of minimally dressed body shapes.

We compare our method to deformation transfer results by the recent deep learning approaches Unsupervised Shape and Pose Disentanglement (USPD) [39] and Neural Pose Transfer (NPT) [35], and show that geometric detail is better preserved with our method when applied to poses not observed during training.

In summary, our contributions are:

- our method better generalises to poses not seen during training than state-of-the-art, achieved by transferring the identity of the target shape to the deformed pose within a deep learning framework.
- our method allows to preserve fine-scale detail linked to the identity of the character and generalizes to characters wearing simple clothing thanks to test time identity transfer refinement with fine-tuning.
- we extend the FAUST dataset [8] to contain more identities and poses with full label information, which can be leveraged for training.

2 Related Work

This section reviews works that solve the deformation transfer problem, generative models for deep learning and recent deep-learning methods for deformation transfer.

Deformation transfer. Deformation transfer aims to deform a character to make it mimic the pose of a source character. This is classically done either by adapting the skeleton movement to a new skeleton [16, 22, 21, 17], or by directly deforming the surface of the characters [30, 40, 6, 24]. Recently, Basset *et al.* [7] showed that transferring the shape, while preserving the pose, requires simpler deformations than deforming the pose of the target character. We follow a similar direction by predicting the shape deformation from the source to the result, instead of predicting coordinates from scratch. These methods can

give good results, but often come at an important computational cost. Instead of using an optimisation based technique, we thus chose to leverage recent advances in deep learning.

Generative deep learning models. Deep learning methods have demonstrated a strong generative capability. Recent works have focused on learning to generate 3D data, in particular with autoencoders. Jiang *et al.* [19] represent 3D data by encoding the vertices of their meshes in a lower dimensional feature, based on an anatomical hierarchical segmentation, and train their model on those features. We make use of similar information by segmenting the meshes into body parts. Similarly, Tan *et al.* [31] represent meshes with rotation invariant features and train a variational encoder with fully connected layers on these features. Ranjan *et al.* [28] define a spectral convolutional layer to generate 3D human faces. They also introduce down and up sampling layers adapted to 3D mesh data based on quadratic edge collapse. Bouritsas *et al.* [10] use a spiral ordering of the neighbourhood of each vertex to apply convolutions to 3D meshes. These methods give satisfying results for generating 3D data, and our architecture is based on the building blocks they provide.

Tretschk *et al.* [32] predict the rigid deformation from a template mesh to their output, which results in higher quality reconstructions of poses. We use a similar idea but take it a step further; instead of predicting the deformation from a common template, we predict the deformation directly from the pose we want to preserve.

Deep learning for deformation transfer. Recently, deep learning methods have been applied to the deformation transfer problem, on various categories of inputs. Numerous works explored transferring 2D video between animated characters e.g. [11, 34] or videos of real people e.g. [23, 14, 20]. Some works explored transferring the style between rigid 3D objects. For instance, Wang *et al.* [36] propose to transfer the style of 3D objects by conditioning their decoder on the vertex coordinates of the object to be deformed, and predicting offsets from this source to the result. This is close to the shape transfer idea by Basset *et al.* [7], and we use a similar architecture.

Gao *et al.* [15] trained two autoencoders for the source and the target shapes, and used a GAN to map the latent code of a deformed source to the latent code of the deformed target. This leads to satisfying deformation transfer results, but the model needs to be retrained for each new shape pair.

Closer to our method, some works [27, 13] describe the instrinsic shape, or style, of a 3D model using the spectrum of its Laplace Beltrami operator (LBO). This is made possible by the observation that the LBO spectrum is invariant to isometric deformation. Style transfer is performed by aligning the spectrum of the source pose to the one of the style target. However the LBO spectrum is known to be sensitive to noise, and struggles to encode fine details of the shape, which limits the transfer results to simple shape classes or to only smoothed versions of the target.

Recently a lot of interest has been given to autoencoders to disentangle shape and pose parameters [12, 39, 19, 5]. These methods naturally allow deformation transfer by reconstructing a character from the shape and pose parameters of two different characters. A common problem these methods encounter is the lack of large real world datasets with pose labels. To remedy this, Cosmo *et al.* [12] present LIMP, a supervised model that allows to train from small-scale datasets. LIMP is built on the hypothesis that human pose deformations are near-isometric, and preserve geodesic distances. The computational cost of the metrics makes LIMP unscalable to large datasets. Another supervised deformation transfer method uses spatially adaptive instance normalisation to perform pose transfer between human characters [35]. While this method, called Neural Pose Transfer (NPT), has access to identity and pose labels for training, the training 3D models do not need to be in correspondence, unlike other methods. Another approach to address the data problem is to train in an unsupervised manner. Aumentado-Armstrong *et al.* [5, 4] use the LBO spectrum to define intrinsic shapes in a way invariant to isometric pose deformation. This allows for unsupervised decoupling of shapes and poses, but the method is sensible to the aforementioned limitations of the LBO spectrum. Zhou *et al.* [39] create pseudo ground truth for the pose transfer between two characters, by applying on the fly As Rigid As Possible deformations [29] during training. This method, Unsupervised Shape and Pose Disentanglement (USPD), constitutes the state-of-the-art for unsupervised methods.

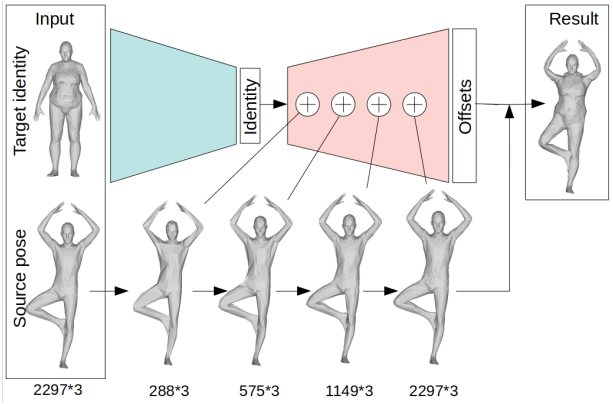


Figure 1: Overview of the proposed approach. The encoder (green) generates an identity code for the target. We feed this code to the decoder (red) along with the source, which is concatenated with the decoder features at all resolution stages. The decoder finally outputs per vertex offsets from the input source towards the identity transfer result.

3 Neural Identity Transfer

This section describes our neural identity transfer method. We address the problem of deformation transfer between 3D shapes described by triangle meshes with the same topology, i.e. all meshes have the same connectivity and vertex to vertex correspondence. We assume a dataset of such shapes i.e. meshes $\{\mathcal{M}\}$ where some have ground-truth identity and/or pose labels, and denote them by $\mathcal{M}_p^{id}$, id being the identity label and p the pose label.

3.1 Overview

Fig. 1 provides a visual overview of our approach. Given two meshes, $\mathcal{M}_p$ with an input source pose and $\mathcal{M}^{id}$ with an input target identity, our goal is to generate a third mesh $\tilde{\mathcal{M}}_p^{id}$ representing the shape of the target identity id in the source pose p. We formulate this problem using a deep learning framework and an encoder-decoder architecture.

Our neural architecture implements the identity transfer by predicting the deformation of the source model $\mathcal{M}_p$ so that its identity matches that of the target model $\mathcal{M}^{id}$. This does not require to encode explicitly pose information as the pose is naturally preserved by predicting identity deformations only.

The encoder Enc takes $\mathcal{M}^{id}$ as input and encodes its identity information into a low-dimensional feature vector $\boldsymbol{z}^{id}$ as

$$\boldsymbol{z}^{id} = Enc(\mathcal{M}^{id}). \tag{1}$$

The decoder Dec takes as input a latent code $\boldsymbol{z}^{id}$ along with $\mathcal{M}_p$ and outputs offsets from $\mathcal{M}_p$ to $\tilde{\mathcal{M}}_p^{id}$ as

$$\tilde{\mathcal{M}}_p^{id} = Dec(\boldsymbol{z}^{id}, \mathcal{M}_p) + \mathcal{M}_p. \tag{2}$$

The architectures of both Enc and Dec are based on spiral convolutions at gradually decreasing/increasing mesh resolutions through pooling/unpooling layers [10]. Note that if the pose information is not explicitly encoded, Dec is anyway conditioned on the pose $\mathcal{M}_p$. This is achieved in practice by concatenating channel-wise at every convolution and unpooling layer of Dec the current vertex features and the 3D coordinates of $\mathcal{M}_p$ at the corresponding mesh resolution.

The two main differences in terms of architecture compared to state-of-the-art deformation transfer methods, such as [12, 39], result from the identity transfer strategy. First, rather than encoding pose information, our decoder is conditioned on the input model $\mathcal{M}_p$ that has the desired pose. Second, rather than predicting 3D vertex information directly, our decoder predicts offsets from $\mathcal{M}_p$.

At inference time, we fine-tune our feed-forward network to improve the results. This strategy, where network training can be seen as a meta-learning stage and fine-tuning as one-shot learning from a single $\mathcal{M}_p, \mathcal{M}^{id}$ pair, is inspired by the few-shots learning of generative models literature (e.g. [38, 3, 18]).

3.2 Training

The model is trained in a weakly supervised way because labeled 3D models of different characters performing the exact same poses are rare in existing datasets. In particular, while each model is equipped with an identity label, only a small subset of all models is equipped with a pose label. For training, we sample triplets of distinct meshes of the form $(\mathcal{M}_{p_1}^{id_1}, \mathcal{M}_{p_2}^{id_2}, \mathcal{M}_{p_1}^{id_2})$ for fully labeled data, and of the form $(\mathcal{M}_{p_1}^{id_1}, \mathcal{M}_{p_2}^{id_2}, \mathcal{M}_{p_3}^{id_2})$ for data with only identity labels (with unknown pose labels p_1, p_2, p_3). Note that while fully labeled data contains the ground

truth of the deformation transfer result $\mathcal{M}_{p_1}^{id_2}$, this information is not available for data with identity labels only.

These triplets are used to train the network based on the following losses

$$\begin{aligned} l_{sup} &= \alpha_{lat} l_{lat} + \alpha_{rec} l_{rec}, \\ l_{weaksup} &= \alpha_{lat} l_{lat} + \alpha_{lap} l_{lap} + \alpha_{rig} l_{rig}, \end{aligned} \quad (3)$$

where l_{sup} is the supervised loss used when full label information is available and $l_{weaksup}$ is the weakly supervised loss used when merely identity labels are known. Let $\tilde{\mathcal{M}}_{p_1}^{id_2}$ denote the transfer result predicted by our method for inputs $\mathcal{M}_{p_1}^{id_1}$ as source pose and $\mathcal{M}_{p_2}^{id_2}$ as target identity.

We use three types of losses to train the network. First, a latent loss l_{lat}, which helps structuring the latent space, is used during both full and weak supervision. Second, in case of full supervision, a standard L_2 penalty reconstruction loss l_{rec} is employed. Finally, in case of weak supervision, two self supervised identity losses l_{lap} and l_{rig} are used that measure identity distances based on predefined deformation models. These losses are weighted using the weights α_{lat}, α_{rec}, α_{lap}, and α_{rig}. Details on these losses follow.

Latent loss This loss uses of the identity label of our data, and constrains the identity latent space by encouraging pairs of shapes that share the same identity to have similar latent representations as

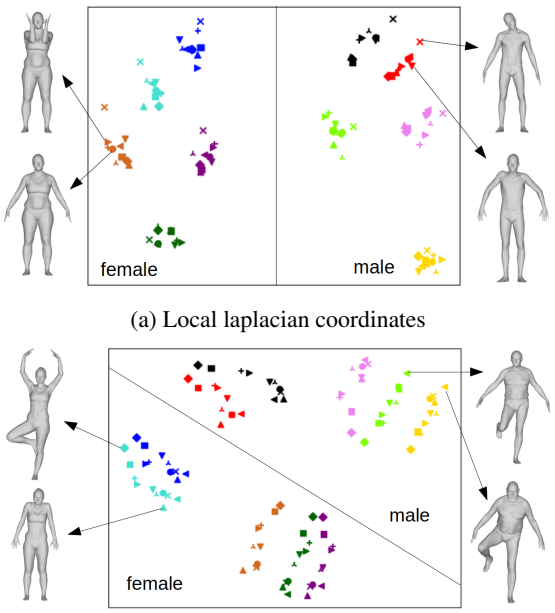
$$l_{lat}(\mathcal{M}_{p_1}^{id_1}, \mathcal{M}_{p_2}^{id_1}) = \|Enc(\mathcal{M}_{p_1}^{id_1}) - Enc(\mathcal{M}_{p_2}^{id_1})\|_2^2. \quad (4)$$

This loss is evaluated for the two input meshes of the triplet that share the same identity code, namely $\mathcal{M}_{p_2}^{id_2}, \mathcal{M}_{p_1}^{id_2}$ for fully labeled data and $\mathcal{M}_{p_2}^{id_2}, \mathcal{M}_{p_3}^{id_2}$ for data with identity labels only.

Reconstruction Loss When pose labels are available, we use a standard reconstruction loss that measures the vertex-to-vertex L_2 distance between the ground truth $\mathcal{M}_{p_1}^{id_2}$ and the predicted result $\tilde{\mathcal{M}}_{p_1}^{id_2}$ as

$$l_{rec}(\tilde{\mathcal{M}}_{p_1}^{id_2}, \mathcal{M}_{p_1}^{id_2}) = \|\tilde{\mathcal{M}}_{p_1}^{id_2} - \mathcal{M}_{p_1}^{id_2}\|_2^2. \quad (5)$$

This strong constraint, that is effective for training, is only required for a small subset of our training data.



(a) Local laplacian coordinates

(b) Distances between vertices belonging to the same body part

Figure 2: T-SNE dimensionality reduction applied to local Laplacian (2a) and intra body part distances (2b). All parameters are computed on the FAUST dataset, containing 10 identities performing 10 poses. In each figure, marker colors indicate identities, and marker shapes indicate poses.

Identity Losses When only identity labels are available, we evaluate differences in identity using two common hypotheses on human deformations.

The first hypothesis is that two characters with the same identity are near-isometric [13]. We use in particular an unsupervised loss that measures differences of the local Laplacian coordinates [37] between our prediction and the target identity. We validate the suitability of this loss using the T-SNE dimensionality reduction method [33] on the FAUST dataset [8] that contains 10 identities in 10 different poses each. Fig. 2a shows the result, which clearly groups models with the same identity. This validates our hypothesis that near-isometric meshes, approximated with local Laplacian coordinates, are likely to represent the same identity. To formally define our loss, let

L be the Laplacian matrix with uniform weights associated with our common mesh connectivity. We compute the Laplacian coordinates of a mesh as $\Delta = L\mathcal{M}$. We then make these coordinates pose invariant by expressing them in per-vertex local coordinate frames, and denote the result by Δ_{loc}. Meshes that preserve the resulting local coordinates are near-isometric. Our loss between two meshes $\mathcal{M}^{id}_{p_1}$ and $\mathcal{M}^{id}_{p_2}$ of the same identity is

$$l_{lap}(\mathcal{M}^{id}_{p_1}, \mathcal{M}^{id}_{p_2}) = \|\Delta_{loc}{}^{id}_{p_1} - \Delta_{loc}{}^{id}_{p_2}\|_2^2. \quad (6)$$

We use this loss between our prediction $\tilde{\mathcal{M}}^{id_2}_{p_1}$ and the given model $\mathcal{M}^{id_2}_{p_2}$ during training.

The second hypothesis is that body parts of a same identity deform near-rigidly between different poses. To validate this hypothesis, Fig. 2b shows a T-SNE plot for FAUST data between all Euclidean intra body part distances. Note that identities are well clustered, which validates our hypothesis. Our loss penalizes distances between vertices belonging to the same body part being inconsistent between our prediction and the target identity. The unsupervised rigidity loss between two models $\mathcal{M}^{id}_{p_1}$ and $\mathcal{M}^{id}_{p_2}$ of the same identity is then

$$l_{rig}(\mathcal{M}^{id}_{p_1}, \mathcal{M}^{id}_{p_2}) = \sum_{P\in\mathcal{P}} \sum_{i,j\in P} \|d(\boldsymbol{v}_{i,1}, \boldsymbol{v}_{j,1}) - d(\boldsymbol{v}_{i,2}, \boldsymbol{v}_{j,2})\|^2, \quad (7)$$

where $\mathcal{P}$ is the set of mesh body parts, $\{\boldsymbol{v}_{i,k}\}$ are the vertices of $\mathcal{M}^{id}_{p_k}$ and $d(.,.)$ is the Euclidean distance. We use this loss between our prediction $\tilde{\mathcal{M}}^{id_2}_{p_1}$ and the given model $\mathcal{M}^{id_2}_{p_2}$ during training.

Note that the two identity losses are evaluated between our prediction and the target identity used for the prediction, and are thus fully unsupervised.

3.3 Fine-Tuning

We introduce a fine-tuning step that is performed systematically at test time. At a small additional computational cost, this step allows to improve results, and enables identity transfer to new shapes considerably different from those seen during training, as demonstrated experimentally.

This step acts as an additional adaptation of the weights of our pre-trained network to a specific input. Given a target identity $\mathcal{M}^{id}$ and a source pose $\mathcal{M}_p$, we first generate our result $\tilde{\mathcal{M}}^{id}_p$ using the trained model as described in Eq. 2. This result is used as initialisation for further optimisation. We fine-tune our model for a few more iterations, using as input identity the target identity $\mathcal{M}^{id}$, and as input pose the initial inference result $\tilde{\mathcal{M}}^{id}_p$. For these extra training steps, we use a self-supervised loss, combining the Laplacian and the rigidity losses to maintain the target identity, in addition to a regularization loss l_{reg} in the form of a L_2 penalty between the vertices of the initial result $\tilde{\mathcal{M}}^{id}_p$ and those of the final fine-tuned mesh

$$l_{ft} = \alpha_{lap} l_{lap} + \alpha_{rig} l_{rig} + \alpha_{reg} l_{reg}. \quad (8)$$

3.4 Implementation Details

For the main model training, we generate training triplets as follows: a triplet $(\mathcal{M}_1, \mathcal{M}_2, \mathcal{M}_3)$ is created for each mesh sample $\mathcal{M}_1$ in the training data. The third and second meshes of the triplet $\mathcal{M}_3$ and $\mathcal{M}_2$ are then randomly sampled from the same identity as the first mesh, and other identities, respectively. If $\mathcal{M}_1$ comes from the portion of our data with pose labels, we restrict the choice of $\mathcal{M}_2$ to meshes with pose labels too, in order to be able to select $\mathcal{M}_3$ with the identity label of $\mathcal{M}_1$ and the pose label of $\mathcal{M}_2$. Every 5 epochs, we re-sample a tenth of our training triplets chosen at random. This way, while each triplet is likely to be seen multiple times by the model, which helps lowering the loss values for these specific triplets, the re-sampling allows the model to see new triplets, which helps to better capture the variety of the training set.

Our network takes as input a list of 3D points that correspond to the vertices of the input meshes. We preprocess all meshes by aligning them rigidly and down-sampling them to 2297 vertices using a quadratic error criterion following [28]. This down-sampling balances the computing cost of our losses, while keeping a reasonable level of precision. We propose a simple two step up-sampling for better qualitative visualization. First, we up-sample the meshes to 6890 vertices, by placing the new vertices at the centroid of their neighbours [28]. Then, we move the new vertices to the local Laplacian coordinates (see Section 3.2) computed on the unprocessed target identity,

while preserving the coordinates of the 2297 vertices predicted by our model.

We use the ADAM optimiser. For the main training, we use a learning rate of 0.001 and a learning rate decay of 0.99 per epoch, and train for 500 epochs. We use batches of size 32. We set the loss weights in Eq. 3 as $\alpha_{rec} = 10$, $\alpha_{lap} = 1000$, $\alpha_{rig} = 1$ and $\alpha_{lat} = 1000$. For the fine-tuning, we use a learning rate of 0.0001, and fine-tune for 50 iterations. We set the loss weights in Eq. 8 as $\alpha_{lap} = 10$, $\alpha_{rig} = 1$ and $\alpha_{reg} = 0.1$.

4 Evaluation

In this section we evaluate our model's ability to achieve deformation transfer both quantitatively and qualitatively. We perform an ablation study to evaluate the effects of supervising and fine-tuning our model, and compare our method quantitatively to state-of-the-art deformation transfer methods. In particular, we choose to compare to the supervised NPT [35] and the unsupervised USPD [39] as they achieve the best results in the literature. Finally, we present qualitative results of deformation transfer using our method to extreme identities and characters with simple clothing. We apply our method to animations on a frame-by-frame basis, and to an identity morphing scenario, where the pose stays constant while the identity changes. For additional visualizations, please refer to the supplementary material.

To evaluate the results numerically, we use input pairs of shapes $\mathcal{M}_p$ and $\mathcal{M}^{id}$ for which the ground truth transfer $\mathcal{M}_p^{id}$ is known. As our meshes are in point-to-point correspondences, the error is measured using the mean of the L_2 distances between corresponding vertices of the ground truth $\mathcal{M}_p^{id}$ and the result $\tilde{\mathcal{M}}_p^{id}$ after Procrustes alignment.

4.1 Data

ExtFAUST To obtain labelled data for supervision, we create a new dataset with full identity and pose labels by augmenting the FAUST dataset [8] with additional pseudo-ground-truth. FAUST contains 10 identities performing the same 10 poses each, providing us with 100 meshes with full identity and pose labels. We extend this data by adding meshes with new poses and identities from other datasets, and then applying an optimization based deformation transfer [7] to transfer every new identity and pose to all pre-existing poses and identities in FAUST. For the new poses and identities added to FAUST, we choose meshes from Dynamic FAUST (DFAUST) [9], SMPL [25], and Mixamo [1]. We add 11 identities and 17 poses to the original FAUST data, yielding 540 meshes with pose and identity labels after manually removing a few outliers. We refer to the resulting dataset as Extended FAUST (ExtFAUST) in the remainder of this paper. We created a test split by removing all occurrences of 4 poses and 4 identities from this dataset. This leaves 369 shapes for training and 171 for testing.

DFAUST We also use the Dynamic FAUST dataset (DFAUST) [9] for training. This dataset contains 10 identities performing between 11 and 14 motions, each of them containing a few hundred frames. It comes with identity labels. However, even if the motions are semantically equivalent across subjects, the poses differ in timing and style, and no pose labels are available. The dataset contains 41220 shapes, which are used for training.

Test sets We use three different test sets in our evaluations. The *ExtFAUST pose test set* consists of 4 identities in 4 poses, all of which were unseen during training. This allows to evaluate the method's ability to generalize to both new identities and poses. When combining all possible triplets of target identity, source pose, and transfer ground truth, 240 triplets are available for testing. The *ExtFAUST id test set* consists of 4 identities unseen during training in 4 poses that were seen during training. This allows to evaluate the method's ability to generalize to identities on poses that it has been trained on. A total of 240 ground truth triplets are available for testing. The *AMASS test set* is used for evaluation w.r.t. the state-of-the-art. It contains 100 triplets generated from motion capture data used in AMASS [26] combined with random SMPL [25] shape parameters.

4.2 Ablation study

To evaluate the necessity and effectiveness of our supervision scheme (Eq. 3), we train our model without any full supervision, with only the FAUST data as full supervision (approximately 0.2% of the training data is labeled), and with all the ExtFAUST data as full supervision (approxi-

Supervision	None	FAUST	ExtFAUST
Mean error (mm)	29.19	24.83	20.19

Table 1: Ablation study on supervision.

mately 1% of the training data is labeled). Tab. 1 reports the errors in mm on the ExtFAUST pose test set. Note that a small percentage of labeled training data allows to improve our results by almost $1cm$.

To evaluate the influence of the fine-tuning at inference time, we run our method with and without fine-tuning on the ExtFAUST test set. While the error without fine-tuning is $31.51mm$, it decreases significantly to $20.19mm$ when fine-tuning is used.

4.3 Comparison to state of the art

To the best of our knowledge none of the existing deformation transfer methods operate in a weakly supervised way and we compare therefore our method to the state-of-the-art supervised method NPT [35] and unsupervised method USPD [39]. This results in three methods that make different assumptions on their training supervision. Moreover, while our method and USPD assume full correspondence of the 3D input models, NPT is more general and can handle 3D models without correspondence or fixed topology. These differences make a completely fair comparison difficult. To make the comparison as fair as possible, we train each method in its optimal supervision setting, with the training data presented in the original papers.

We evaluate the errors on our three test sets: one that requires pose generalization from all methods, and two that require pose generalization from some of the methods. For the ExtFAUST pose test set, none of the methods have seen during training any of the poses or identities presented at test time. This test set therefore evaluates all method's abilities to generalize to new poses and new identities, and can be considered the hardest test set for all methods. For the ExtFAUST identity test set, none of the methods have seen any of the idenitites presented at test time. However, our method has seen the poses, coupled with other identities, during training. This test set therefore requires NPT and USPD to generalize to new poses, while this is not the case for our method. For the AMASS test set, none of the methods have seen any of the idenitites presented at test time. However, NPT and USPD have seen the poses, coupled with other identities, during training. This test set therefore requires our method to generalize to new poses, while this is not the case for NPT and USPD.

Fig. 3 shows cumulative error plots for each method on each validation set. Note that our method and USPD obtain significantly better results for the first two validation sets, that require a generalization ability to new poses from NPT. This is because NPT does not use correspondence information, and treats points on the 3D human model that are close-by as neighbors and aims to deform them using similar deformations. In cases where different body parts are close-by or in contact in one input pose but not the other, this creates stretching artifacts that explain the high errors. For the AMASS test set, where NPT does not need to generalize to new poses and the results provide a meaningful measure for NPT's performance, their result is better, but our method still outperforms NPT on average and in the fine details.

Our method and USPD perform respectively better than the other when one method has seen the poses during training and not the other. For the ExtFAUST pose test set with poses unseen by all methods, both methods have similar performances, but our method gives slightly better results for the low error range, showing that details are better preserved. This can be observed in Fig. 4, where USPD's result has the correct overall body shape, but the details of the identity are overly smooth, whereas our method better transfers the fine-scale geometric details of the identity. It is also noteworthy to mention that USPD uses approximately 3 times more training data than we do.

4.4 Qualitative evaluation

Figures 5, 6 and 7 present results upsampled with the method described in Section 3.4, for visualization purposes.

Fig. 5 shows results of transferring an unrealistic identity, unseen at training, to new poses. This figure demonstrates that our method is able to transfer extreme identities, while our method only saw realistic identities during training. Fig. 6 shows results of transferring a character with simple clothing and accessories to new poses. Note that during training, no clothed characters or accessories

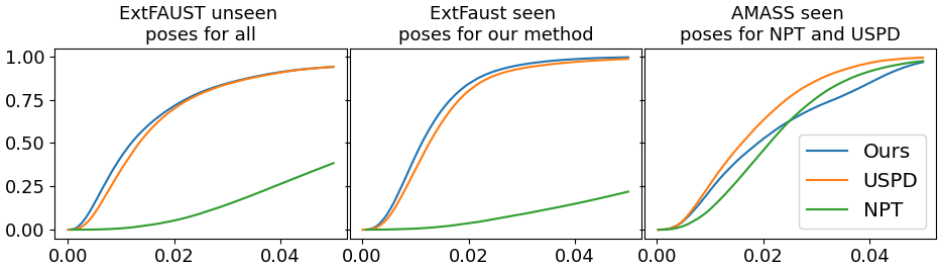


Figure 3: Cumulative errors for our method, USPD and NPT on 3 validation sets. The x-axis shows per-vertex errors (mm). The y-axis is the proportion of all error values below the corresponding error value.

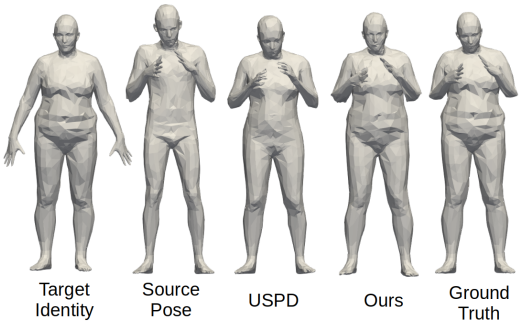


Figure 4: Qualitative comparison to USPD.

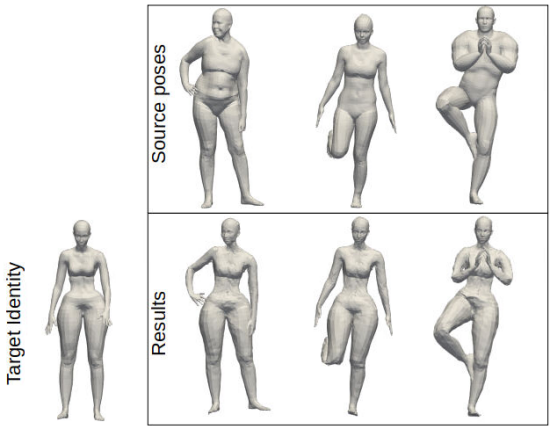


Figure 5: Transferring the identity of an unrealistic character to new poses.

are seen. These results show our method's ability to generalize to data that is far from the distribution of the training data while preserving geometric detail. This property is achieved in large part by the fine-tuning at inference.

To demonstrate the potential of our method, we apply it to two problems arising in automatic content creation. First, Fig. 7 shows our method applied to solve the motion retargeting problem. Given an input animation and a new identity, we apply our method to the animation on a frame-by-frame basis. Note that although no temporal information is used by our method, the resulting animation is consistent and does not suffer from a significant jitter (best observed in supplementary video).

Second, Fig. 8 shows our method applied to solve the morphing problem. For this result, the identity code of two input characters is linearly interpolated in the latent space before being passed to the decoder. Our method is able to interpolate smoothly between identity codes while keeping the pose consistent. In addition to being an interesting application, this result shows that the latent space learned by our method is well structured.

4.5 Limitations

While our method gives state-of-the-art results for the deformation transfer problem, limitations remain. First, we require meshes in vertex-to-vertex correspondence. This limits our method to meshes with a given template, and requires solving a correspondence problem for other inputs. Another limitation of our method is the need for supervision, as training our network without supervision results in deformed poses due to the near-isometry hypothesis. An interesting direction for future work is to extend our method to be fully unsupervised by exploring other iden-

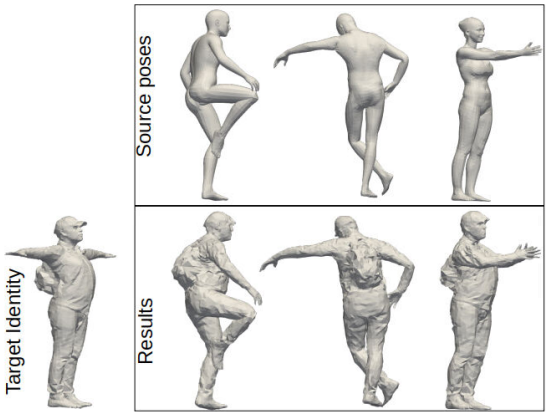


Figure 6: Transferring the identity of a clothed character to new poses.

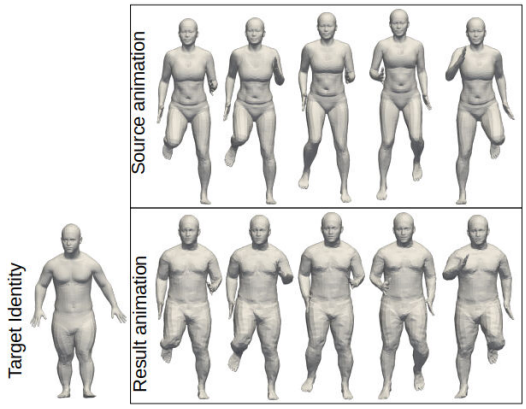


Figure 7: Transferring a new identity to an animation.

tity losses.

5 Conclusion

In this paper, we introduced a neural deformation transfer method that predicts the identity deformation from a source character to a character with the same pose and a new identity. We used geometric properties of meshes to describe identity in a pose invariant way. We introduced a large dataset of human models with full identity and pose labels, which we use in addition to a larger unlabeled dataset to supervise our training. Experiments

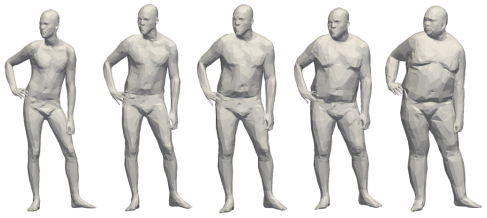


Figure 8: Interpolating the identity latent code between the leftmost and rightmost models.

demonstrate our model's ability to generalize to unseen poses when using around 1% of supervision at training time. A fine tuning step, inspired by the few-shot learning methods, is shown to allow for the transfer of fine-scale geometric details of the identity. The method generalizes well to new identities, and even allows to transfer simple clothing and accessories.

Acknowledgments

This project was funded by the Inria IPL AVATAR project, and by the EU's Horizon 2020 research and innovation program under grant agreement No 952147. We also want to thank João Regateiro for the helpful discussions.

References

[1] Adobe's mixamo. https://www.mixamo.com/. Accessed: 2020-01-02. 7

[2] Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis. Scape: shape completion and animation of people. In *ACM SIGGRAPH 2005 Papers*, pages 408–416. 2005. 1

[3] Sercan O Arik, Jitong Chen, Kainan Peng, Wei Ping, and Yanqi Zhou. Neural voice cloning with a few samples. In *Conference on Neural Information Processing Systems*, 2018. 2, 4

[4] Tristan Aumentado-Armstrong, Stavros Tsogkas, Sven Dickinson, and Allan Jepson. Disentangling geometric deformation spaces in generative latent shape models. *arXiv preprint arXiv:2103.00142*, 2021. 3

[5] Tristan Aumentado-Armstrong, Stavros Tsogkas, Allan Jepson, and Sven Dickinson. Geometric disentanglement for generative latent shape models. In *Proceedings of the*

IEEE/CVF International Conference on Computer Vision, pages 8181–8190, 2019. 3

[6] Ilya Baran, Daniel Vlasic, Eitan Grinspun, and Jovan Popović. Semantic deformation transfer. In *ACM SIGGRAPH 2009 papers*, pages 1–6. 2009. 2

[7] Jean Basset, Stefanie Wuhrer, Edmond Boyer, and Franck Multon. Contact preserving shape transfer: Retargeting motion from one shape to another. *Computers & Graphics*, 89:11–23, 2020. 2, 3, 7

[8] Federica Bogo, Javier Romero, Matthew Loper, and Michael J Black. Faust: Dataset and evaluation for 3d mesh registration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3794–3801, 2014. 2, 5, 7

[9] Federica Bogo, Javier Romero, Gerard Pons-Moll, and Michael J Black. Dynamic faust: Registering human bodies in motion. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6233–6242, 2017. 7

[10] Giorgos Bouritsas, Sergiy Bokhnyak, Stylianos Ploumpis, Michael Bronstein, and Stefanos Zafeiriou. Neural 3d morphable models: Spiral convolutional networks for 3d shape representation learning and generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7213–7222, 2019. 3, 4

[11] Caroline Chan, Shiry Ginosar, Tinghui Zhou, and Alexei A Efros. Everybody dance now. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5933–5942, 2019. 3

[12] Luca Cosmo, Antonio Norelli, Oshri Halimi, Ron Kimmel, and Emanuele Rodola. Limp: Learning latent shape representations with metric preservation priors. *arXiv preprint arXiv:2003.12283*, 2, 2020. 3, 4

[13] Luca Cosmo, Mikhail Panine, Arianna Rampini, Maks Ovsjanikov, Michael M Bronstein, and Emanuele Rodola. Isospectralization, or how to hear shape, style, and correspondence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7529–7538, 2019. 2, 3, 5

[14] Patrick Esser, Ekaterina Sutter, and Björn Ommer. A variational u-net for conditional appearance and shape generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8857–8866, 2018. 3

[15] Lin Gao, Jie Yang, Yi-Ling Qiao, Yu-Kun Lai, Paul L Rosin, Weiwei Xu, and Shihong Xia. Automatic unpaired shape deformation transfer. In *SIGGRAPH Asia 2018*, page 237. ACM, 2018. 3

[16] Michael Gleicher. Retargetting motion to new characters. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 33–42, 1998. 2

[17] Edmond SL Ho, Taku Komura, and Chiew-Lan Tai. Spatial relationship preserving character motion adaptation. In *ACM SIGGRAPH 2010 papers*, pages 1–8. 2010. 2

[18] Ye Jia, Yu Zhang, Ron J Weiss, Quan Wang, Jonathan Shen, Fei Ren, Zhifeng Chen, Patrick Nguyen, Ruoming Pang, Ignacio Lopez Moreno, et al. Transfer learning from speaker verification to multispeaker text-to-speech synthesis. In *Conference on Neural Information Processing Systems*, 2018. 2, 4

[19] Boyi Jiang, Juyong Zhang, Jianfei Cai, and Jianmin Zheng. Disentangled human body embedding based on deep hierarchical neural network. *IEEE transactions on visualization and computer graphics*, 26(8):2560–2575, 2020. 3

[20] Moritz Kappel, Vladislav Golyanik, Mohamed Elgharib, Jann-Ole Henningson, Hans-Peter Seidel, Susana Castillo, Christian Theobalt, and Marcus Magnor. High-fidelity neural human motion transfer from monocular video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1541–1550, 2021. 3

[21] Richard Kulpa, Franck Multon, and Bruno Arnaldi. Morphology-independent representation of motions for interactive human-like animation. In *Eurographics*, 2005. 2

[22] 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*, pages 39–48, 1999. 2

[23] Wen Liu, Zhixin Piao, Jie Min, Wenhan Luo, Lin Ma, and Shenghua Gao. Liquid warping gan: A unified framework for human motion imitation, appearance transfer and novel view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5904–5913, 2019. 3

[24] Zhiguang Liu, Antonio Mucherino, Ludovic Hoyet, and Franck Multon. Surface based motion retargeting by preserving spatial relationship. In *Proceedings of the 11th Annual International Conference on Motion, Interaction, and Games*, pages 1–11, 2018. 2

[25] 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)*, 34(6):1–16, 2015. 1, 7

[26] Naureen Mahmood, Nima Ghorbani, Nikolaus F. Troje, Gerard Pons-Moll, and Michael J. Black. AMASS: Archive of motion capture as surface shapes. In *International Conference on Computer Vision*, pages 5442–5451, 2019. 7

[27] Riccardo Marin, Arianna Rampini, Umberto Castellani, Emanuele Rodola, Maks Ovsjanikov, and Simone Melzi. Instant recovery of shape from spectrum via latent space connections. In *2020 International Conference on 3D Vision (3DV)*, pages 120–129. IEEE, 2020. 1, 3

[28] Anurag Ranjan, Timo Bolkart, Soubhik Sanyal, and Michael J Black. Generating 3d faces using convolutional mesh autoencoders. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 704–720, 2018. 3, 6

[29] Olga Sorkine and Marc Alexa. As-rigid-as-possible surface modeling. In *Symposium on Geometry processing*, volume 4, pages 109–116, 2007. 2, 3

[30] Robert W Sumner and Jovan Popović. Deformation transfer for triangle meshes. *ACM Transactions on graphics (TOG)*, 23(3):399–405, 2004. 1, 2

[31] Qingyang Tan, Lin Gao, Yu-Kun Lai, and Shihong Xia. Variational autoencoders for deforming 3d mesh models. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5841–5850, 2018. 3

[32] Edgar Tretschk, Ayush Tewari, Michael Zollhöfer, Vladislav Golyanik, and Christian Theobalt. Demea: Deep mesh autoencoders for non-rigidly deforming objects. In *European Conference on Computer Vision*, pages 601–617. Springer, 2020. 3

[33] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008. 5

[34] Ruben Villegas, Jimei Yang, Duygu Ceylan, and Honglak Lee. Neural kinematic networks for unsupervised motion retargetting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8639–8648, 2018. 3

[35] Jiashun Wang, Chao Wen, Yanwei Fu, Haitao Lin, Tianyun Zou, Xiangyang Xue, and Yinda Zhang. Neural pose transfer by spatially adaptive instance normalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5831–5839, 2020. 2, 3, 7, 8

[36] Weiyue Wang, Duygu Ceylan, Radomir Mech, and Ulrich Neumann. 3dn: 3d deformation network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1038–1046, 2019. 3

[37] Stefanie Wuhrer, Chang Shu, and Pengcheng Xi. Posture-invariant statistical shape analysis using laplace operator. *Computers & Graphics*, 36(5):410–416, 2012. 5

[38] Egor Zakharov, Aliaksandra Shysheya, Egor Burkov, and Victor Lempitsky. Few-shot adversarial learning of realistic neural talking head models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9459–9468, 2019. 2, 4

[39] Keyang Zhou, Bharat Lal Bhatnagar, and Gerard Pons-Moll. Unsupervised shape and pose disentanglement for 3d meshes. In *European Conference on Computer Vision*, pages 341–357. Springer, 2020. 2, 3, 4, 7, 8

[40] Kun Zhou, Weiwei Xu, Yiying Tong, and Mathieu Desbrun. Deformation transfer to multi-component objects. In *Computer Graphics Forum*, volume 29, pages 319–325. Wiley Online Library, 2010. 2