MAPConNet: Self-supervised 3D Pose Transfer with Mesh and Point Contrastive Learning

Jiaze Sun[1]* Zhixiang Chen[2] Tae-Kyun Kim[1,3]
[1]Imperial College London [2]University of Sheffield [3]Korea Advanced Institute of Science and Technology

Abstract

3D pose transfer is a challenging generation task that aims to transfer the pose of a source geometry onto a target geometry with the target identity preserved. Many prior methods require keypoint annotations to find correspondence between the source and target. Current pose transfer methods allow end-to-end correspondence learning but require the desired final output as ground truth for supervision. Unsupervised methods have been proposed for graph convolutional models but they require ground truth correspondence between the source and target inputs. We present a novel self-supervised framework for 3D pose transfer which can be trained in unsupervised, semi-supervised, or fully supervised settings without any correspondence labels. We introduce two contrastive learning constraints in the latent space: a mesh-level loss for disentangling global patterns including pose and identity, and a point-level loss for discriminating local semantics. We demonstrate quantitatively and qualitatively that our method achieves state-of-the-art results in supervised 3D pose transfer, with comparable results in unsupervised and semi-supervised settings. Our method is also generalisable to unseen human and animal data with complex topologies[†].

1. Introduction

3D pose transfer [35, 40, 47, 33] is a challenging generation task in which the pose of a source geometry is transferred to a target geometry whilst preserving the identity of the target geometry (see top half of Figure 1). This has many potential applications in areas including animation, human modelling, virtual reality, and more. It also provides a more affordable way of generating synthetic 3D data which can be expensive to produce in the real world.

One of the main challenges of 3D pose transfer is that current methods still put certain requirements on their training data making it difficult to collect and expensive to annotate. One requirement is having correspondence labels, which are pairs of vertices that correspond to each other semantically between two point clouds or meshes. Many prior pose transfer methods either require ground truth correspondence [35, 4, 2, 43, 47] or simply neglect the issue [40]. Requiring ground truth correspondence is costly, and neglecting correspondence would adversely impact model performance. [33] proposed learning a correspondence module based on optimal transport in an end-to-end fashion through the pose transfer task. However, their method is supervised and requires the mesh with the desired pose and identity as ground truth. This puts another requirement on the dataset: having multiple subjects performing exactly the same set of poses, which is unfeasible. [47] proposed an unsupervised approach for registered meshes which only requires each subject to perform multiple poses and they do not have to align exactly across subjects – a more practical requirement for real datasets [6, 22]. However, their approach is based on graph convolutional networks (GCNs) and ARAP deformation which require ground truth correspondence.

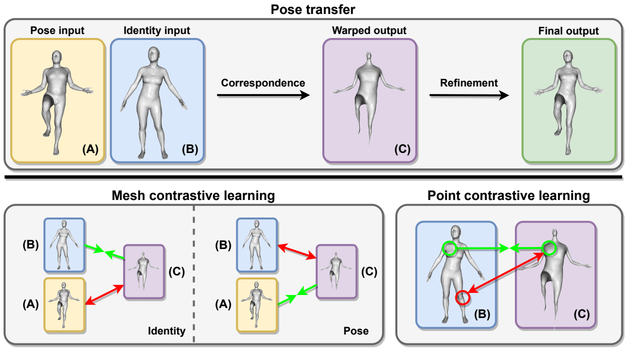

Figure 1. **Overview of our framework.** Top: our pose transfer pipeline. Bottom: our contrastive learning scheme, with a mesh-level loss for disentangling global pose and identity and a point-level loss for discriminating local semantics.

We propose a self-supervised framework (Figure 1) for 3D pose transfer with *Mesh And Point Contrastive learning*, *MAPConNet*, requiring no correspondence labels or

*Corresponding author email: `j.sun19@imperial.ac.uk`.
[†]Code: `https://github.com/justin941208/MAPConNet`.

target outputs with the desired pose and identity as ground truth for supervision. It can be applied in supervised, unsupervised, and semi-supervised settings, and does not need the pose and identity inputs to have the same ordering or number of points. We propose to adapt the unsupervised approach by [47] for pose transfer on unaligned meshes, using [33] as our baseline. To prevent the network from exploiting shortcuts in unsupervised learning that circumvent the desired objective, we introduce disentangled latent pose and identity representations. To strengthen the disentanglement and guide the learning process more effectively, we propose *mesh-level contrastive learning* to force the model's intermediate output to have matching latent identity and pose representations with the respective inputs. To further improve the quality of the model's intermediate output as well as the correspondence module, we also propose *point-level contrastive learning*, which enforces similarity between representations of corresponding points and dissimilarity between non-corresponding ones. In summary:

- We present *MAPConNet*, a novel self-supervised network for 3D pose transfer with *Mesh And Point Contrastive* learning. Our model requires no ground truth correspondence or target outputs for supervision.

- We introduce two levels of contrastive learning constraints, with a mesh-level loss for global disentanglement of pose and identity and a point-level loss for discrimination between local semantics.

- We achieve state-of-the-art results through extensive experiments on human and animal data, and demonstrate competitive and generalisable results in supervised, unsupervised, and semi-supervised settings.

2. Related work

Deep learning on 3D data. Recent years have seen a surge in deep learning methods on 3D data such as point clouds, meshes and voxels. [41] and [23] operate on voxels using 3D convolutions which would be computationally expensive for high-dimensional data such as ours. [12] and [38] are designed for meshes but include fully-connected layers which are memory-intensive. Some GCN models [29, 21, 47] include down- and up-sampling layers and others [19, 15] incorporate novel operations, both of which improve efficiency. However, all their architectures rely on a template structure to implement and is not suitable in our scenario. As for point clouds, [27] and [28] use shared weights across points and adopt aggregation strategies to enforce order-invariance. We use shared weights without aggregation to preserve detailed identity information.

3D pose/deformation transfer. Pose transfer aims to transfer the pose of a source geometry to a target without changing the target's identity. Deformation transfer methods [35, 4, 2, 43, 44, 18] require a template pose across different identities and sometimes also correspondence labels, which are unavailable in our and most realistic settings. Image-to-image translation methods [48, 8, 16, 24, 37, 36] have also been repurposed for pose transfer due to their relevance. [13] used CycleGAN [48] for pose transfer but requires retraining for each new pair of identities. [3] reformulated the problem as "identity transfer" but required vertex correspondence. [40] used SPADE normalisation [24] to inject target identity into the source. [33] added correspondence learning to [40] and improved the normalisation method. However, both require ground truth outputs. [47] proposed an unsupervised framework but requires correspondence labels. We do not require ground truth outputs or correspondence labels. [34] proposed a dual reconstruction objective in a similar spirit to [47] to enable unsupervised learning in [33]. In contrast, our approach is to force the model to learn disentangled latent pose and identity codes and impose mesh- and point-level contrastive losses on them, which improves performance in both supervised and unsupervised settings.

Self-supervised learning on 3D data. Self-supervised learning is the paradigm of automatically generating supervisory signals in training and has been successful in 2D [32, 9, 25, 46, 14, 7, 37]. Some 2D approaches have also been adopted for 3D data, such as rotation [26] and completion [39], but we choose contrastive learning as it can be easily adapted to suit our needs. Most existing contrastive learning approaches for 3D data [45, 42, 31, 10, 1] focus on learning invariance across views or rigid transformations for scenes or simple objects, whereas we address more fine-grained patterns such as identity, pose, and correspondence for complex shapes including humans and animals.

3. Methodology

We now present our problem setting and proposed MAPConNet in detail. Given a pose (i.e. source) mesh $\mathbf{x}^{A1} \in \mathbb{R}^{N_{pose} \times 3}$ and identity (i.e. target) mesh $\mathbf{x}^{B2} \in \mathbb{R}^{N_{id} \times 3}$, where the letters $A, B, \ldots$ and numbers $1, 2, \ldots$ denote the identities and poses of the meshes respectively, our goal is to train a network G to produce a new mesh $\hat{\mathbf{x}}^{B1} = G(\mathbf{x}^{A1}, \mathbf{x}^{B2}) \in \mathbb{R}^{N_{id} \times 3}$ which inherits pose 1 and identity B. The integers N_{pose} and N_{id} are the numbers of vertices in the pose and identity meshes, respectively. The meshes are treated as point clouds by our model, but connectivity is required for training. In addition, we do not require the vertices of both inputs to share the same order, but the output mesh $\hat{\mathbf{x}}^{B1}$ would follow the same order as $\mathbf{x}^{B2}$.

3.1. Preliminaries

For our baseline, we choose 3D-CoreNet [33] – a prior state-of-the-art 3D pose transfer model that requires no

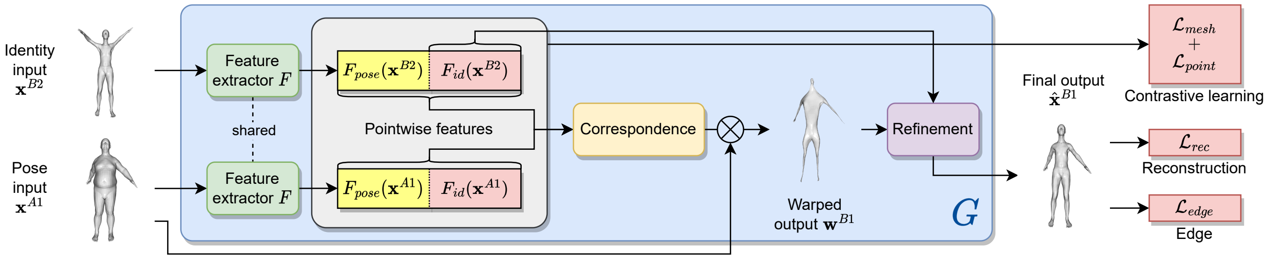

Figure 2. **Our supervised learning pipeline**. The feature extractor F embeds the inputs $\mathbf{x}^{B2}$ and $\mathbf{x}^{A1}$ into a shared latent space. The correspondence module warps $\mathbf{x}^{A1}$ into $\mathbf{w}^{B1}$ which should have the same vertex order as $\mathbf{x}^{B2}$. In addition, we separate the latent codes into identity and pose, where only the identity channels are used as style conditioning to refine $\mathbf{w}^{B1}$ into the final output $\hat{\mathbf{x}}^{B1}$. We propose mesh and point contrastive learning on top of the existing reconstruction and edge losses. Our model is based on 3D-CoreNet [33].

ground truth correspondence. It has two modules: correspondence and refinement. The correspondence module produces an intermediate "warped" output $\mathbf{w}^{B1} \in \mathbb{R}^{N_{id} \times 3}$ inheriting the pose from $\mathbf{x}^{A1}$ but the vertex order of $\mathbf{x}^{B2}$. Specifically, the warped output is obtained by $\mathbf{w}^{B1} = \mathbf{T}\mathbf{x}^{A1}$, where $\mathbf{T} \in \mathbb{R}_{+}^{N_{id} \times N_{pose}}$ is an optimal transport (OT) matrix learned based on the latent features of both inputs. The refinement module then uses the features of the identity input $\mathbf{x}^{B2}$ as style condition for the warped output $\mathbf{w}^{B1}$, refining it through elastic instance normalisation and producing the final output $\hat{\mathbf{x}}^{B1}$. The training of 3D-CoreNet is supervised: given the model output $\hat{\mathbf{x}}^{B1}$ and the ground truth output $\mathbf{x}^{B1}$, it minimises the reconstruction loss

$$\mathcal{L}_{rec}(\hat{\mathbf{x}}^{B1}; \mathbf{x}^{B1}) = \frac{1}{3N_{id}} \|\hat{\mathbf{x}}^{B1} - \mathbf{x}^{B1}\|_F^2, \tag{1}$$

where $\|\cdot\|_F$ is the Frobenius norm. For this loss to work properly, the ground truth $\mathbf{x}^{B1}$ and identity input $\mathbf{x}^{B2}$ must have the same dimensions and vertex order. In addition to $\mathcal{L}_{rec}$, an edge loss is used to help generate smoother surfaces and prevent flying vertices [40, 33]. Given the model output $\hat{\mathbf{x}}^{B1}$ and identity input $\mathbf{x}^{B2}$, which should have matching vertex orders, the edge loss is given by

$$\mathcal{L}_{edge}(\hat{\mathbf{x}}^{B1}; \mathbf{x}^{B2}) = \frac{1}{|\mathcal{E}|} \sum_{(j,k)\in\mathcal{E}} \left| \frac{\|\hat{\mathbf{x}}_j^{B1} - \hat{\mathbf{x}}_k^{B1}\|_2}{\|\mathbf{x}_j^{B2} - \mathbf{x}_k^{B2}\|_2} - 1 \right|, \tag{2}$$

where $\mathcal{E}$ is the set of all index pairs representing vertices that are connected by an edge, and $\hat{\mathbf{x}}_j^{B1}, \mathbf{x}_j^{B2} \in \mathbb{R}^3$ are the coordinates of the j-th (similarly, k-th) vertices of $\hat{\mathbf{x}}^{B1}$ and $\mathbf{x}^{B2}$ respectively. Finally, the overall supervised loss is given by

$$\mathcal{L}_s = \lambda_{rec}\mathcal{L}_{rec}(\hat{\mathbf{x}}^{B1}; \mathbf{x}^{B1}) + \lambda_{edge}\mathcal{L}_{edge}(\hat{\mathbf{x}}^{B1}; \mathbf{x}^{B2}), \tag{3}$$

where λ_{rec} and λ_{edge} are the weights for the two losses.

3.2. Latent disentanglement of pose and identity

Our supervised pipeline is shown in Figure 2 with proposed loss terms $\mathcal{L}_{mesh}$ and $\mathcal{L}_{point}$ which are discussed in detail in Section 3.4 and 3.5. In Section 3.3, we present our unsupervised pipeline with the self- and cross-consistency losses by [47]. However, directly using these losses leads to suboptimal results due to 3D-CoreNet not having a disentangled latent space. Hence, given an input mesh $\mathbf{x}$, we further separate its latent representation $F(\mathbf{x}) \in \mathbb{R}^{N \times D}$ into identity $F_{id}(\mathbf{x}) \in \mathbb{R}^{N \times D_{id}}$ and pose $F_{pose}(\mathbf{x}) \in \mathbb{R}^{N \times D_{pose}}$ channels. Here, $F(\cdot)$ is the feature extractor, and F_{id} and F_{pose} are the components of F corresponding to the pose and identity channels, and $D_{id} + D_{pose} = D$. Furthermore, we feed *only* the identity channels $F_{id}(\mathbf{x}^{B2})$ to the refinement module as input, but *both* identity and pose channels to the correspondence module as input.

3.3. Unsupervised pose transfer

It is clear that training 3D-CoreNet requires the ground truth output with the desired pose and identity. This requires training samples with: (i) the same identity in different poses, *and* (ii) different identities in the same pose. Whilst (i) is easily satisfied, (ii) is more difficult in practice. For instance, if the two inputs come from separate datasets with different sets of identities and poses, there would be no ground truth for the reconstruction loss (Equation 1).

Unsupervised pose transfer with only condition (i) was shown to be possible on registered meshes by [47], whose main idea is that the network should arrive at the same output in two sub-tasks: (a) when both inputs share a common identity, and (b) when the pose input in (a) is replaced by a different identity with the same pose. Task (a) is called "cross-consistency" and is readily available from the dataset. The pose input in task (b) is unavailable but can be generated by the network itself, i.e. "self-consistency". However, their GCN model and ARAP deformation require the vertices of both input meshes to be pre-aligned.

Despite these differences, we propose incorporating the cross- and self- consistency losses from [47] into the task of pose transfer on unaligned meshes to enable unsupervised training (see Figure 3). Following [47], meshes $\mathbf{x}^{A1}, \mathbf{x}^{A2} \in$

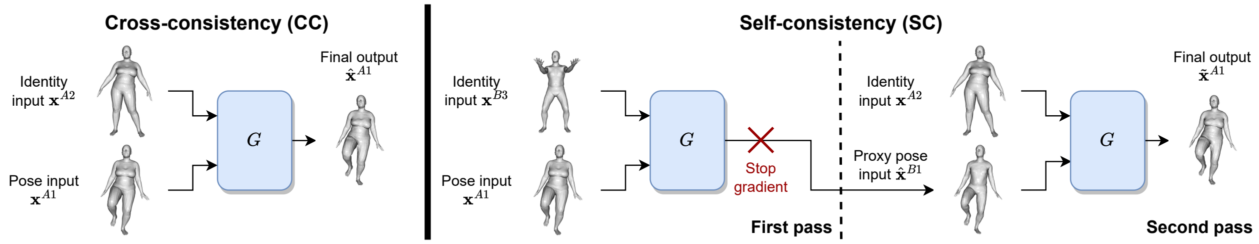
Cross-consistency (CC)
Identity input $\mathbf{x}^{A2}$
Pose input $\mathbf{x}^{A1}$
G
Final output $\hat{\mathbf{x}}^{A1}$
Self-consistency (SC)
Identity input $\mathbf{x}^{B3}$
Pose input $\mathbf{x}^{A1}$
G
Stop gradient
First pass
Identity input $\mathbf{x}^{A2}$
Proxy pose input $\hat{\mathbf{x}}^{B1}$
G
Final output $\tilde{\mathbf{x}}^{A1}$
Second pass

Figure 3. **Our unsupervised learning pipeline.** During CC, the model receives two inputs with the same identity. During SC, the pose input in the second pass is of a different identity but in the same pose as that in CC, generated by the model itself in the first pass. The model should learn to produce the same output in both stages. This framework is based on [47].

$\mathbb{R}^{N_{pose}\times 3}$ and $\mathbf{x}^{B3} \in \mathbb{R}^{N_{id}\times 3}$ are required as inputs during training. In addition, whilst vertex order can vary across identities, it must be the same between $\mathbf{x}^{A1}$ and $\mathbf{x}^{A2}$.

Cross-consistency (CC). Given pose input $\mathbf{x}^{A1}$ and identity input $\mathbf{x}^{A2}$, which have the same identity but different poses, the network should reconstruct $\mathbf{x}^{A1}$ through $\hat{\mathbf{x}}^{A1} = G(\mathbf{x}^{A1}, \mathbf{x}^{A2})$ (Figure 3, left). This is enforced via

$$\mathcal{L}_{cc} = \lambda_{rec}\mathcal{L}_{rec}(\hat{\mathbf{x}}^{A1}; \mathbf{x}^{A1}) + \lambda_{edge}\mathcal{L}_{edge}(\hat{\mathbf{x}}^{A1}; \mathbf{x}^{A2}). \quad (4)$$

Self-consistency (SC). CC alone is insufficient as it does not train the network to perform transfers between meshes with different identities. As mentioned previously, the model should reconstruct the same output as that in CC when its pose input is replaced by a different identity with the same pose – which is usually not available from the training data. [47] proposed to let the network itself generate such examples as proxy inputs for training in a two-pass manner (Figure 3, centre and right). In the first pass, given pose input $\mathbf{x}^{A1}$ and identity input $\mathbf{x}^{B3}$, the network generates a proxy $\hat{\mathbf{x}}^{B1} = G(\mathbf{x}^{A1}, \mathbf{x}^{B3})$. This is then used as the pose input in the second pass to reconstruct the initial pose input $\tilde{\mathbf{x}}^{A1} = G(SG(\hat{\mathbf{x}}^{B1}), \mathbf{x}^{A2})$, where SG stops the gradient from passing through. The purpose of SG is to prevent the model from exploiting shortcuts such as using the input $\mathbf{x}^{A1}$ from the first pass to directly reconstruct the output in the second pass. Incidentally, this also avoids extra computational overhead. The SC loss is given by

$$\mathcal{L}_{sc} = \lambda_{rec}\mathcal{L}_{rec}(\tilde{\mathbf{x}}^{A1}; \mathbf{x}^{A1}) + \lambda_{edge}\mathcal{L}_{edge}(\tilde{\mathbf{x}}^{A1}; \mathbf{x}^{A2}). \quad (5)$$

Finally, the overall unsupervised loss is given by

$$\mathcal{L}_{us} = \mathcal{L}_{cc} + \mathcal{L}_{sc}. \quad (6)$$

3.4. Mesh contrastive learning

As mentioned in 3.2, we disentangle the latents into pose and identity channels. This allows us to impose direct constraints on the meaning of these channels to improve the accuracy of the model output. For instance, we can compare the output against the inputs in terms of pose and identity and impose losses to enforce consistency. In addition, during *unsupervised* learning, the network may also exploit potential shortcuts such as taking both pose and identity information from one input only and ignoring the other input. Disentangling the latent space and imposing additional constraints makes these shortcuts more difficult to exploit.

For these purposes, we propose mesh-level contrastive learning losses for pose and identity. As we cannot compare pose and identity directly in the mesh space, we take a self-supervised approach by feeding meshes through the feature extractor F and imposing the triplet loss [32] on the latent representations (see Figure 4). Specifically, given an anchor latent $\mathbf{a}$, a positive latent $\mathbf{p}$, and a negative latent $\mathbf{n}$ which are all in $\mathbb{R}^{N\times D}$, our mesh triplet loss is given by

$$l(\mathbf{a}, \mathbf{p}, \mathbf{n}) = \left(m + \frac{1}{N}\sum_{j=1}^{N} d(\mathbf{a}_j, \mathbf{p}_j, \mathbf{n}_j) \right)^{+}, \quad (7)$$

where $(\cdot)^{+} = \max(0, \cdot)$, m is the margin, and

$$d(\mathbf{a}_j, \mathbf{p}_j, \mathbf{n}_j) = \|\mathbf{a}_j - \mathbf{p}_j\|_2 - \|\mathbf{a}_j - \mathbf{n}_j\|_2, \quad (8)$$

where $\mathbf{a}_j, \mathbf{p}_j, \mathbf{n}_j \in \mathbb{R}^{D}$ are the latents of the j-th vertex from $\mathbf{a}, \mathbf{p}, \mathbf{n}$ respectively. In equation 7, we enforce the margin m on the whole mesh rather than on individual points since pose and identity are global patterns. We will now discuss how equation 7 is incorporated into our unsupervised and supervised pipelines.

Contrastive learning in CC. Recall that in CC, the network tries to predict the output from two inputs with the same identity. Given identity input $\mathbf{x}^{A2}$ and pose input $\mathbf{x}^{A1}$, let $\mathbf{w}^{A1}$ be the resulting warped output. By intuition, the pose representation of $\mathbf{x}^{A1}$ should be closer to that of $\mathbf{w}^{A1}$ than $\mathbf{x}^{A2}$, as the pose of $\mathbf{x}^{A1}$ should be inherited by $\mathbf{w}^{A1}$. Whilst $\mathbf{w}^{A1}$, $\mathbf{x}^{A2}$, and $\mathbf{x}^{A1}$ should all have the same identity, the identity representation of $\mathbf{x}^{A1}$ should be closer to that of $\mathbf{x}^{A2}$ than $\mathbf{w}^{A1}$ as $\mathbf{w}^{A1}$ should generally avoid inheriting the identity representation from the pose input. Hence, the triplet loss formulation for CC is

$$\begin{aligned}\mathcal{L}^{cc}_{mesh} = & l\left(F_{pose}(\mathbf{x}^{A1}), F_{pose}(\mathbf{w}^{A1}), F_{pose}(\mathbf{x}^{A2})\right) \\ & + l\left(F_{id}(\mathbf{x}^{A1}), F_{id}(\mathbf{x}^{A2}), F_{id}(\mathbf{w}^{A1})\right)\end{aligned} \quad (9)$$

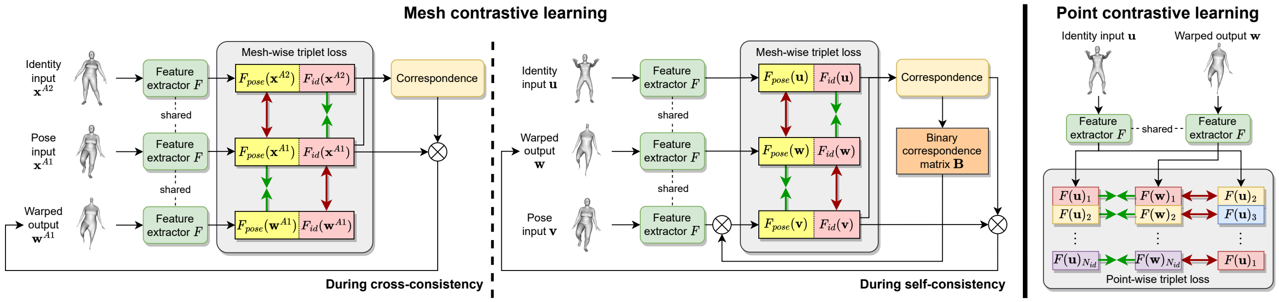
Mesh contrastive learning
Identity input $\mathbf{x}^{A2}$
Pose input $\mathbf{x}^{A1}$
Warped output $\mathbf{w}^{A1}$
Feature extractor F
shared
Mesh-wise triplet loss
$F_{pose}(\mathbf{x}^{A2})$
$F_{id}(\mathbf{x}^{A2})$
$F_{pose}(\mathbf{x}^{A1})$
$F_{id}(\mathbf{x}^{A1})$
$F_{pose}(\mathbf{w}^{A1})$
$F_{id}(\mathbf{w}^{A1})$
Correspondence
During cross-consistency
Identity input $\mathbf{u}$
Warped output $\mathbf{w}$
Pose input $\mathbf{v}$
$F_{pose}(\mathbf{u})$
$F_{id}(\mathbf{u})$
$F_{pose}(\mathbf{w})$
$F_{id}(\mathbf{w})$
$F_{pose}(\mathbf{v})$
$F_{id}(\mathbf{v})$
Binary correspondence matrix $\mathbf{B}$
During self-consistency
Point contrastive learning
Identity input $\mathbf{u}$
Warped output $\mathbf{w}$
$F(\mathbf{u})_1$
$F(\mathbf{w})_1$
$F(\mathbf{u})_2$
$F(\mathbf{w})_2$
$F(\mathbf{u})_3$
$F(\mathbf{u})_{N_{id}}$
$F(\mathbf{w})_{N_{id}}$
Point-wise triplet loss

Figure 4. **Mesh and point contrastive learning.** Left: the triplet formulation for CC, where $\mathbf{x}^{A1}$ and $\mathbf{x}^{A2}$ have the same identity and vertex order but different poses. Centre: the triplet formulation for SC (and supervised pipeline), where $\mathbf{u}$ and $\mathbf{v}$ do not have the same identity, poses, or vertex order. Right: the triplet formulation for point contrastive learning, where $\mathbf{u}$ and $\mathbf{w}$ share the same vertex order.

We use the warped output here (and later in SC) instead of the final output for two reasons. First, this forces the warped output to be closer to the desired final output, making it easier to be refined. Second, this has a lower computational cost as it does not involve the refinement module.

Contrastive learning in SC. Recall that in SC, the model should produce the same output as that in CC through two consecutive passes, and the two inputs for each pass have different identities and poses. Given identity input $\mathbf{u} \in \mathbb{R}^{N_{id} \times 3}$ and pose input $\mathbf{v} \in \mathbb{R}^{N_{pose} \times 3}$, let $\mathbf{w} \in \mathbb{R}^{N_{id} \times 3}$ be the resulting warped output. Naturally, the pose representation of $\mathbf{w}$ should be closer to that of $\mathbf{v}$ than $\mathbf{u}$, and the identity representation of $\mathbf{w}$ should be closer to that of $\mathbf{u}$ than $\mathbf{v}$. This logic applies to *both* passes within SC.

However, unlike CC, the vertex orders of $\mathbf{u}$ and $\mathbf{v}$ in SC are not aligned. As a result, equations 7 and 8 cannot be applied directly as they only compare distances between *aligned* vertices. We again take a self-supervised approach to "reorder" the feature of $\mathbf{v}$ by utilising the OT matrix $\mathbf{T}$. As the cost matrix for OT is based on the pairwise similarities between point features of $\mathbf{u}$ and $\mathbf{v}$, most entries in $\mathbf{T}$ are made close to zero except for a small portion which are more likely to be corresponding points. Therefore, we use the following binary version of $\mathbf{T}$ to select vertices from $\mathbf{v}$

$$\mathbf{B}_{jk} = I\{\mathbf{T}_{jk} = \max_{l} \mathbf{T}_{jl}\}, \tag{10}$$

where I is the indicator function. In other words, each row of $\mathbf{B} \in \{0,1\}^{N_{id} \times N_{pose}}$ is a binary vector marking the location of the maximum value in the corresponding row of $\mathbf{T}$. We further constrain the rows of $\mathbf{B}$ to be one-hot in case of multiple maximum entries. Now, the triplet loss for SC with the "reordered" pose feature is given by

$$\begin{aligned} \mathcal{L}^{ss}_{mesh} =& l\left(F_{pose}(\mathbf{w}), \mathbf{B}F_{pose}(\mathbf{v}), F_{pose}(\mathbf{u})\right) \\ &+ l\left(F_{id}(\mathbf{w}), F_{id}(\mathbf{u}), \mathbf{B}F_{id}(\mathbf{v})\right). \end{aligned} \tag{11}$$

Contrastive learning in supervised pipeline. Equation 11 can also be applied to our supervised learning pipeline by replacing $\mathbf{u}, \mathbf{v}, \mathbf{w}$ with $\mathbf{x}^{B2}, \mathbf{x}^{A1}, \mathbf{w}^{B1}$, respectively.

3.5. Point contrastive learning

Intuitively, the final output would be more accurate if the warped output more closely resembles the ground truth. However, as later experiments will demonstrate, both 3D-CoreNet and $\mathcal{L}_{mesh}$ have a shrinking effect on the warped output, particularly on the head and lower limbs. We propose to address this problem by enforcing similarity between corresponding points and dissimilarity between non-corresponding points across different meshes. Specifically, given identity input $\mathbf{u}$ and warped output $\mathbf{w}$, we propose the following triplet loss for point-level contrastive learning

$$\mathcal{L}_{point} = \frac{1}{N_{id}} \sum_{j=1}^{N_{id}} \left(m + d(F(\mathbf{w})_j, F(\mathbf{u})_j, F(\mathbf{u})_k)\right)^{+}, \tag{12}$$

where d is from equation 8 and $k \in \{1, \dots, N_{id}\} \setminus \{j\}$. Unlike equation 7, the margin m here is enforced on individual points instead of the whole mesh. In our implementation, we set $k = j + 1$, and consequently set $F(\mathbf{u})_{N_{id}+1} := F(\mathbf{u})_1$. In other words, the negative points are simply $\mathbf{u}$ with its first point moved to the end and all others shifted down in index by 1. As all input points of a mesh are randomly re-ordered during pre-processing (see Section 4), the negative point can come from any region of the feature $F(\mathbf{u})$ throughout training. In addition, this loss can be applied in all cases including supervised learning and both CC and SC in unsupervised learning.

3.6. Overall training losses

The overall objective of our framework when ground truth is available is given by the labelled loss:

$$\mathcal{L}_L = \mathcal{L}_s + \lambda_{m,s}\mathcal{L}^{ss}_{mesh} + \lambda_p \mathcal{L}_{point}. \tag{13}$$

When ground truth is unavailable, we instead minimise:

$$\mathcal{L}_U = \mathcal{L}_{us} + \lambda_{m,c}\mathcal{L}^{cc}_{mesh} + \lambda_{m,s}\mathcal{L}^{ss}_{mesh} + \lambda_p \mathcal{L}_{point}. \tag{14}$$

In our full models, we set $\lambda_{m,s} = \lambda_{m,c} = \lambda_p = 1$.

Test set	*Mode*	*Method*	*PMD* ↓	*CD* ↓	*EMD* ↓
SMPL	S	(*A*) DT [35]	1.50	3.50	22.10
		(*B*) NPT [40]	6.60	14.20	42.20
		(*C*) 3D-CoreNet (Baseline) [33]	0.36	1.18	1.35
		(*D*) MAPConNet (Ours)	**0.30**	**1.04**	**1.15**
		(*E*) 3D-CoreNet (Baseline) [33], 50% labelled	4.99	10.34	10.13
		(*F*) MAPConNet (Ours), 50% labelled	**3.69**	**6.67**	**9.25**
	SS	(*G*) MAPConNet (Ours), 50% labelled + 50% unlabelled	**0.40**	**1.32**	**1.45**
	US	(*H*) 3D-CoreNet (Baseline) [33], without LD	14.09	32.34	18.85
		(*I*) 3D-CoreNet (Baseline) [33], with LD	0.76	2.20	2.58
		(*J*) MAPConNet (Ours)	**0.56**	**1.71**	**1.83**
SMAL	S	(*K*) DT [35]	133.70	357.70	159.00
		(*L*) NPT [40]	67.50	145.20	116.50
		(*M*) 3D-CoreNet (Baseline) [33]	27.04	51.55	29.31
		(*N*) MAPConNet (Ours)	**25.34**	**47.81**	**26.22**
		(*O*) 3D-CoreNet (Baseline) [33], 50% labelled	59.66	130.99	49.26
		(*P*) MAPConNet (Ours), 50% labelled	**51.36**	**112.25**	**43.80**
	SS	(*Q*) MAPConNet (Ours), 50% labelled + 50% unlabelled	**29.80**	**55.16**	**30.85**
	US	(*R*) MAPConNet (Ours)	**29.29**	**54.64**	**30.00**
DFAUST	US	(*S*) 3D-CoreNet (Baseline) [33], without LD	468.10	441.74	128.08
		(*T*) 3D-CoreNet (Baseline) [33], with LD	92.45	176.21	55.71
		(*U*) MAPConNet (Ours)	**61.22**	**60.97**	**32.07**
	S	(*C*) 3D-CoreNet (Baseline) [33], trained on SMPL only	138.56	159.11	58.95
		(*D*) MAPConNet (Ours), trained on SMPL only	**120.10**	**133.09**	**49.75**
	SS	(*V*) MAPConNet (Ours), SMPL labelled + DFAUST unlabelled	**36.23**	**35.69**	**23.30**

Table 1. **Quantitative results.** PMD and CD are in units of 10^{-4}, and EMD is in units of 10^{-3}. Lower values are better. The modes “S”, “SS”, and “US” are short for “supervised”, “semi-supervised”, and “unsupervised”, respectively. “LD” is short for “latent disentanglement” which is described in Section 3.2. *Note: Training and test sets are from the same dataset unless otherwise specified.*

4. Experiments

4.1. Data

SMPL. This is a synthetic dataset [40] of 24,000 human meshes generated by SMPL [20] from 30 identities and 800 poses. Each mesh has 6,890 vertices. Ground truths are available for evaluation and supervised learning as all identities share a common set of poses. Following [33], we randomly sample a training set of 4,000 meshes from a randomly sampled list of 16 identities and 400 poses. For evaluation, we use the same fixed set of 400 mesh pairs as [33] which are randomly sampled from the remaining 14 identities and 200 poses that are unseen during training.

SMAL. This is a synthetic dataset of 24,600 animal meshes generated by SMAL [49] from 41 identities and 600 poses. Each mesh has 3,889 vertices. All identities also share a common set of poses. Following [33], we randomly select a training pool of 11,600 meshes from a random list of 29 identities and 400 poses. For evaluation, we again use the same fixed set of 400 mesh pairs as [33] from the unseen 12 identities and 200 poses. Compared to SMPL, this is a more challenging dataset as it consists of a wider variety of shapes and sizes of animals.

DFAUST. Unlike SMPL and SMAL, which are synthetic datasets, DFAUST [6] is a more challenging collection of registered human meshes obtained from real 3D scans which allows us to validate our model in realistic unsupervised settings. The dataset consists of 10 subjects, 5 males and 5 females, each performing multiple motion sequences. There are no direct ground truths available since different subjects are not in precisely the same pose, and as a result we use the SMPL+H model [30] to generate pseudo ground truths for evaluation. We randomly select 4,000 meshes from 3 males and 3 females for training and 399 meshes from the unseen subjects for evaluation.

MG. The Multi-Garment (MG) dataset [5] includes registered meshes of humans in clothing from real 3D scans. Each mesh has 27,554 vertices – significantly more than SMPL and DFAUST. We use MG to qualitatively validate our method’s ability to handle complex unseen topologies.

4.2. Implementation details

Pre-processing. We pre-process all inputs in two steps. First, the vertices of each pair of pose and identity inputs are *randomly and independently re-ordered* to remove the correspondence between them. Second, they are zero-centred

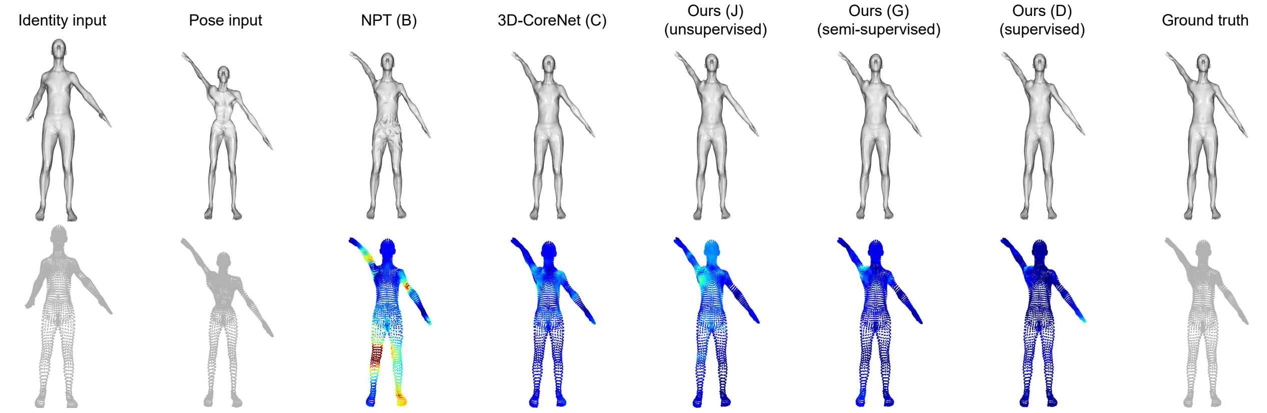

Figure 5. **Qualitative comparison on unseen SMPL inputs.** This shows pose transfer results using prior and our methods (labels defined in Table 1) trained on SMPL. The first and second rows show the rendered surfaces and point clouds respectively. The colours of the model output point clouds represent PMDs against the ground truth, with dark red and dark blue indicating high and low PMDs respectively.

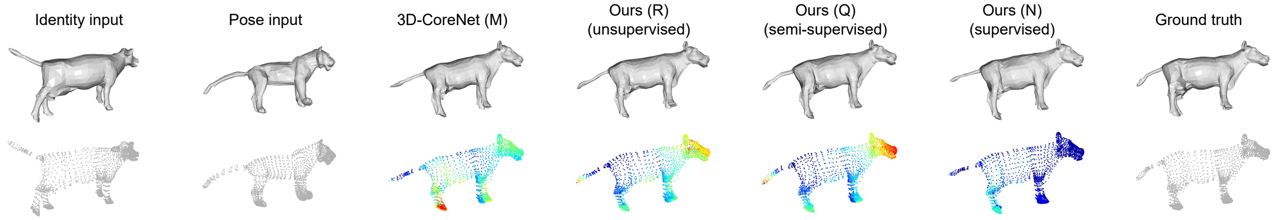

Figure 6. **Qualitative comparison on unseen SMAL inputs.** Similar to Figure 5 (with method labels defined in Table 1), the first and second rows are the rendered surfaces and point clouds respectively, with PMD heatmaps visualised on model output point clouds.

based on their bounding boxes. We perform the same procedure for *both* training and evaluation.

Parameter settings. Our experiments are based on the official implementation of 3D-CoreNet [33]. We set a batch size of 2 for all experiments. Following [33], all hyperparameters are kept at their default values including $\lambda_{rec} = 1000$ and $\lambda_{edge} = 0.5$. As for $\mathcal{L}_{mesh}$ and $\mathcal{L}_{point}$, we set the margin at $m = 1$. The network is trained for 200 epochs using the Adam optimiser [17] with an initial learning rate of 1×10^{-4} which is kept constant for the first 100 epochs and then linearly decayed to 0 in the last 100 epochs.

Training. The detailed training procedures for supervised, unsupervised, and semi-supervised settings are in Algorithms 1, 2, and 3 in the Appendix, respectively. For semi-supervised learning in SMPL or SMAL (Table 1 methods (G) and (Q)), we alternate iterations between optimising $\mathcal{L}_L$ and $\mathcal{L}_U$. On the other hand, for semi-supervised learning on SMPL and DFAUST (Table 1 method (V)), as we would like to run inference on DFAUST with the final model, we train the first 100 epochs on SMPL in a supervised manner and the remaining epochs on DFAUST in an unsupervised manner. During an unlabelled iteration, we also allow either pose or identity input (not both) to come from the labelled dataset (see Algorithm 3 in the Appendix).

4.3. Evaluation

We evaluate model performance at epoch 200 by comparing its outputs against ground truths using three metrics.

Pointwise Mesh Distance (PMD). The PMD [40, 33] is in the same form as Equation 1 and takes into account both the positions and ordering of the vertices measured.

Chamfer Distance (CD). The CD [11] measures the discrepancy between two point clouds without taking into account the ordering of their points. Given point clouds $P, Q \subset \mathbb{R}^3$, the CD between P and D is given by

$$\frac{1}{|P|}\sum_{\mathbf{p}\in P}\min_{\mathbf{q}\in Q}\|\mathbf{p}-\mathbf{q}\|_2^2 + \frac{1}{|Q|}\sum_{\mathbf{q}\in Q}\min_{\mathbf{p}\in P}\|\mathbf{p}-\mathbf{q}\|_2^2. \quad (15)$$

Earth Mover's Distance (EMD). The EMD [11] first solves the assignment problem between two point clouds before taking the pointwise distance by aligning the two point clouds using the resulting assignment function. Specifically, given point clouds $P, Q \subset \mathbb{R}^3$ such that $|P| = |Q|$, the EMD between P and Q is given by

$$\min_{\phi:P\to Q}\frac{1}{|P|}\sum_{\mathbf{p}\in P}\|\mathbf{p}-\phi(\mathbf{p})\|_2, \quad (16)$$

where ϕ is a bijective mapping from P to Q.

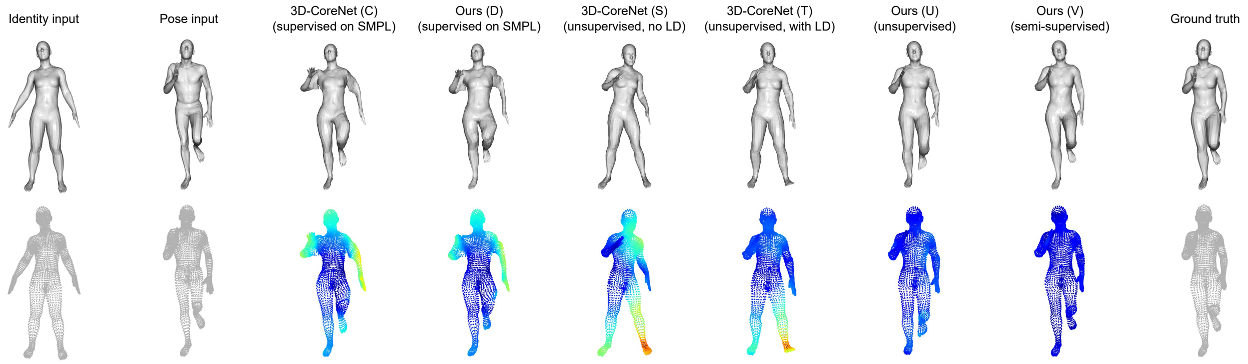
Identity input
Pose input
3D-CoreNet (C) (supervised on SMPL)
Ours (D) (supervised on SMPL)
3D-CoreNet (S) (unsupervised, no LD)
3D-CoreNet (T) (unsupervised, with LD)
Ours (U) (unsupervised)
Ours (V) (semi-supervised)
Ground truth

Figure 7. **Qualitative comparison on unseen DFAUST inputs.** Similar to Figure 5 (with method labels defined in Table 1), the first and second rows are the rendered surfaces and point clouds respectively, with PMD heatmaps visualised on model output point clouds.

For all metrics, lower values are better. As mentioned in section 4.2, we randomly and independently re-order the input vertices in evaluations (similar to training) but with a *fixed* seed for all experiments to ensure fairness.

4.4. Quantitative results

We compare our method with previous state-of-the-art pose transfer models using the aforementioned metrics and evaluation procedure (see Table 1). For DT [35] and NPT [40], we quote the results obtained by [33]. For 3D-CoreNet [33], we retrain their model from scratch using their official implementation. As we use a smaller batch size during training than that used in the original 3D-CoreNet work [33] but keep the total number of epochs unchanged, there are more gradient descent steps during our training which causes the baseline numbers we obtained to be different from the original numbers reported by [33]. However, our replicated results are close to the original ones or even better. We also emphasise that we keep all data pre-processing, training, and evaluation procedures identical across experiments to ensure any comparisons between the baseline and our method are fair.

We outperform prior state-of-the-art methods in the supervised setting on SMPL and SMAL as shown by method (D) and (N) in Table 1. Our method also enables unsupervised learning in (J) and (R) where we achieve results comparable to the supervised ones while still outperforming earlier supervised methods DT and NPT. Training 3D-CoreNet [33] directly using the CC and SC losses for unsupervised learning [47] does not yield ideal results as shown by (H), but applying our latent space disentanglement leads to a substantial improvement as shown by (I) and (J). Similar improvement can also be observed on the more challenging DFAUST dataset as shown by (S), (T), and (U). This demonstrates that it can be difficult for the model to disentangle pose and identity effectively in an unsupervised setting without proper architectural constraints.

Table 1 also demonstrates that our method is more effective in settings with limited labelled data. In methods (E), (F), (O), and (P), we randomly remove 50% identities and 50% poses from the training set, which means the model only sees 25% of all available meshes during training. As expected, this leads to a dramatic increase in PMD and CD which indicates reduced model generalisability. However, our model is still able to outperform 3D-CoreNet under this limitation. Since our method also supports unsupervised learning, we can introduce meshes with the remaining 50% identities and 50% poses to our model as unlabelled samples in addition to the labelled ones. Note that the model still only sees 50% of all available training meshes since meshes whose identity is in the labelled set and whose pose is in the unlabelled set (and vice versa) are not included. Under this semi-supervised setting, our model achieves substantial improvement in accuracy as shown by (G) and (Q), approaching the fully supervised results. In addition, by using all SMPL training data as the labelled set and all DFAUST training data as the unlabelled set, we achieve further improvement when testing on DFAUST as shown by (V).

4.5. Qualitative results

We visually compare the outputs of various methods on SMPL and SMAL in Figures 5 and 6, respectively. We can observe that the output surfaces of our method are smoother compared to NPT which produces numerous concavities. We visualise the PMDs between the outputs and ground truths through heatmaps which show that our method generates more accurate outputs indicated by the darker blue. We also visualise various models on DFAUST in Figure 7, which clearly shows that our unsupervised model (U) is more accurate than (S) and (T) which do not employ our contrastive losses, and our semi-supervised model (V) trained on both SMPL and DFAUST outperforms the supervised models (C) and (D) trained only on SMPL. Finally, Figure 8 demonstrates that our model can handle complex

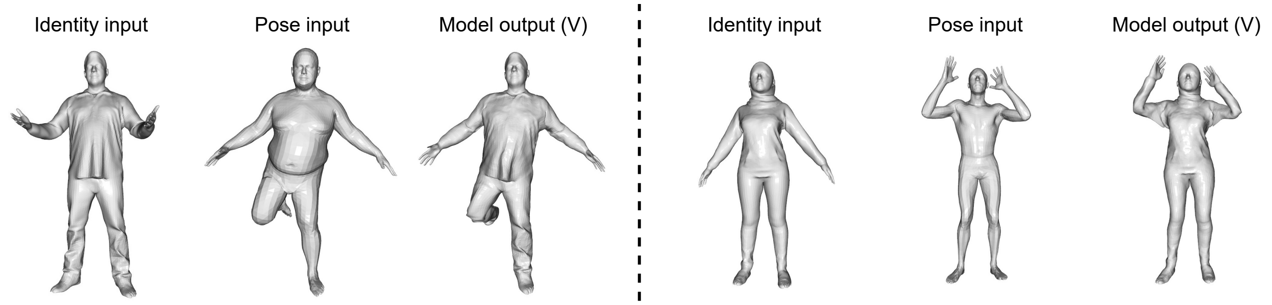

Figure 8. **Qualitative results on unseen MG and DFAUST inputs.** The above shows two instances of pose transfer using our semi-supervised model (V) defined in Table 1, with MG meshes as identity inputs and DFAUST meshes as pose inputs.

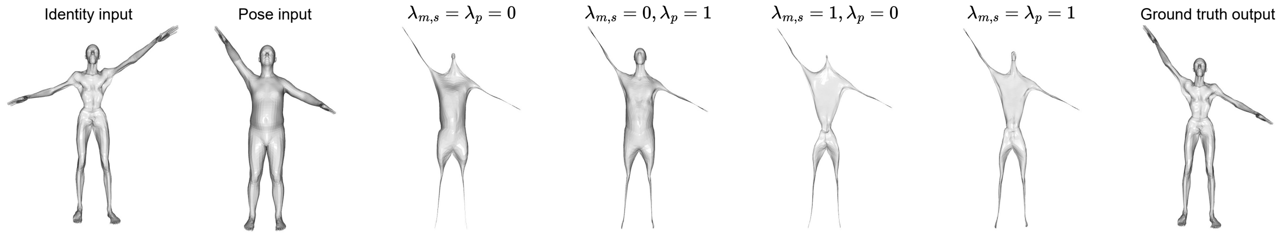
Identity input
Pose input
$\lambda_{m,s} = \lambda_p = 0$
$\lambda_{m,s} = 0, \lambda_p = 1$
$\lambda_{m,s} = 1, \lambda_p = 0$
$\lambda_{m,s} = \lambda_p = 1$
Ground truth output

Figure 9. **Warped outputs in the ablation study (supervised).** This is a visual comparison of the warped outputs from the supervised models in the ablation study (Table 2), with the input and ground truth output meshes as references.

Mode	$\lambda_{m,c}$	$\lambda_{m,s}$	λ_p	*PMD* ↓	*CD* ↓	*EMD* ↓
S	N/A	0	0	0.36	1.18	1.35
	N/A	0	1	0.35	1.16	1.26
	N/A	1	0	0.31	1.08	1.17
	N/A	1	1	**0.30**	**1.04**	**1.15**
US	0	0	0	0.76	2.20	2.58
	0	0	1	0.59	1.81	2.00
	1	0	0	0.60	1.81	1.98
	1	1	0	0.59	1.79	1.96
	1	1	1	**0.56**	**1.71**	**1.83**

Table 2. **Ablation study.** PMD and CD are in units of 10^{-4}, and EMD is in units of 10^{-3}. The modes "S" and "US" are short for "supervised" and "unsupervised" respectively.

unseen input meshes with different topologies and vertex numbers. Additional results can be found in the Appendix.

4.6. Ablation study

In Table 2, we study the individual effectiveness of our proposed losses $\mathcal{L}_{mesh}$ and $\mathcal{L}_{point}$ on SMPL by changing their associated weights $\lambda_{m,c}$, $\lambda_{m,s}$, and λ_p. It can be seen that setting any of them to 1 can improve the result, and setting all to 1 yields the best result. In the unsupervised setting, we also observe that setting $\lambda_{m,c} = \lambda_{m,s} = 1$ is more effective than setting $\lambda_{m,c} = 1$ only. The warped outputs of the supervised experiments are shown in Figure 9. Setting $\lambda_p = 1$ leads to a visible improvement in the quality of the warped output particularly in the head and lower limbs, mitigating the shrinking effect as mentioned in Section 3.5. On the other hand, setting $\lambda_{m,s} = 1$ allows the warped output to have a better resemblance with the ground truth. This is shown by the corresponding warped outputs in Figure 9 which are thinner compared to the ones with $\lambda_{m,s} = 0$. However, this causes a side effect where the head and limbs are drastically shrunk. Combining both losses leads to an improved resemblance between the warped output and ground truth whilst reducing excessive shrinking.

5. Conclusion

We proposed MAPConNet, a self-supervised 3D pose transfer framework without requiring correspondence labels or ground truth outputs for supervision. We introduce disentangled latent spaces for pose and identity to improve unsupervised learning, and mesh and point level contrastive learning to improve the model's intermediate output and in turn, final output. We achieve state-of-the-art results in supervised learning, and competitive results in unsupervised and semi-supervised settings that are generalisable to unseen human and animal data with complex topologies.

6. Acknowledgements

This work was supported by the Croucher Foundation, Huawei Consumer Business Group, NST grant (CRC 21011, MSIT), and KOCCA grant (R2022020028, MCST).

References

[1] Mohamed Afham, Isuru Dissanayake, Dinithi Dissanayake, Amaya Dharmasiri, Kanchana Thilakarathna, and Ranga Rodrigo. CrossPoint: Self-supervised cross-modal contrastive learning for 3d point cloud understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9902–9912, June 2022.

[2] Ilya Baran, Daniel Vlasic, Eitan Grinspun, and Jovan Popović. Semantic deformation transfer. *ACM Trans. Graph.*, 28(3), July 2009.

[3] J. Basset, A. Boukhayma, S. Wuhrer, F. Multon, and E. Boyer. Neural human deformation transfer. In *2021 International Conference on 3D Vision (3DV)*, pages 545–554, Los Alamitos, CA, USA, dec 2021. IEEE Computer Society.

[4] Mirela Ben-Chen, Ofir Weber, and Craig Gotsman. Spatial deformation transfer. In *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '09, page 67–74, New York, NY, USA, 2009. Association for Computing Machinery.

[5] Bharat Lal Bhatnagar, Garvita Tiwari, Christian Theobalt, and Gerard Pons-Moll. Multi-Garment Net: Learning to dress 3D people from images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[6] Federica Bogo, Javier Romero, Gerard Pons-Moll, and Michael J. Black. Dynamic FAUST: Registering human bodies in motion. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5573–5582, 2017.

[7] Ting Chen, Xiaohua Zhai, Marvin Ritter, Mario Lucic, and Neil Houlsby. Self-supervised GANs via auxiliary rotation loss. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[8] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. StarGAN: Unified generative adversarial networks for multi-domain image-to-image translation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[9] Carl Doersch, Abhinav Gupta, and Alexei A. Efros. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[10] Bi'an Du, Xiang Gao, Wei Hu, and Xin Li. Self-contrastive learning with hard negative sampling for self-supervised point cloud learning. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 3133–3142, 2021.

[11] Haoqiang Fan, Hao Su, and Leonidas J. Guibas. A point set generation network for 3D object reconstruction from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[12] Yutong Feng, Yifan Feng, Haoxuan You, Xibin Zhao, and Yue Gao. MeshNet: Mesh neural network for 3D shape representation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8279–8286, 2019.

[13] 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*, 37(6), dec 2018.

[14] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.

[15] Shunwang Gong, Lei Chen, Michael Bronstein, and Stefanos Zafeiriou. SpiralNet++: A fast and highly efficient mesh convolution operator. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 0–0, 2019.

[16] Z. He, W. Zuo, M. Kan, S. Shan, and X. Chen. AttGAN: Facial attribute editing by only changing what you want. *IEEE Transactions on Image Processing*, 28(11):5464–5478, Nov 2019.

[17] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[18] 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)*. Springer, October 2022.

[19] Isaak Lim, Alexander Dielen, Marcel Campen, and Leif Kobbelt. A simple approach to intrinsic correspondence learning on unstructured 3D meshes. In *Computer Vision – ECCV 2018 Workshops: Munich, Germany, September 8-14, 2018, Proceedings, Part III*, page 349–362, Berlin, Heidelberg, 2019. Springer-Verlag.

[20] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *ACM Transaction on Graphics*, 34(6):248:1–248:16, Oct. 2015.

[21] Qianli Ma, Jinlong Yang, Anurag Ranjan, Sergi Pujades, Gerard Pons-Moll, Siyu Tang, and Michael J. Black. Learning to dress 3D people in generative clothing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

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

[23] Daniel Maturana and Sebastian Scherer. VoxNet: A 3D convolutional neural network for real-time object recognition. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 922–928, 2015.

[24] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[25] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A. Efros. Context encoders: Feature

learning by inpainting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[26] O. Poursaeed, T. Jiang, H. Qiao, N. Xu, and V. G. Kim. Self-supervised learning of point clouds via orientation estimation. In *2020 International Conference on 3D Vision (3DV)*, pages 1018–1028, Los Alamitos, CA, USA, nov 2020. IEEE Computer Society.

[27] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. PointNet: Deep learning on point sets for 3D classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[28] Charles R. Qi, Li Yi, Hao Su, and Leonidas J. Guibas. PointNet++: Deep hierarchical feature learning on point sets in a metric space. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 5105–5114, Red Hook, NY, USA, 2017. Curran Associates Inc.

[29] 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)*, September 2018.

[30] Javier Romero, Dimitrios Tzionas, and Michael J. Black. Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 36(6), Nov. 2017.

[31] Aditya Sanghi. Info3D: Representation learning on 3D objects using mutual information maximization and contrastive learning. In *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIX*, page 626–642, Berlin, Heidelberg, 2020. Springer-Verlag.

[32] Florian Schroff, Dmitry Kalenichenko, and James Philbin. FaceNet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[33] Chaoyue Song, Jiacheng Wei, Ruibo Li, Fayao Liu, and Guosheng Lin. 3D pose transfer with correspondence learning and mesh refinement. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021.

[34] Chaoyue Song, Jiacheng Wei, Ruibo Li, Fayao Liu, and Guosheng Lin. Unsupervised 3D pose transfer with cross consistency and dual reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–13, 2023.

[35] Robert W. Sumner and Jovan Popović. Deformation transfer for triangle meshes. *ACM Transactions on Graphics*, 23(3):399–405, aug 2004.

[36] Jiaze Sun, Binod Bhattarai, Zhixiang Chen, and Tae-Kyun Kim. SeCGAN: Parallel conditional generative adversarial networks for face editing via semantic consistency. In *AI for Content Creation Workshop at CVPR*, 2022.

[37] Jiaze Sun, Binod Bhattarai, and Tae-Kyun Kim. MatchGAN: A self-supervised semi-supervised conditional generative adversarial network. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*, November 2020.

[38] 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 (CVPR)*, June 2018.

[39] Hanchen Wang, Qi Liu, Xiangyu Yue, Joan Lasenby, and Matt J. Kusner. Unsupervised point cloud pre-training via occlusion completion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9782–9792, October 2021.

[40] 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 (CVPR)*, June 2020.

[41] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3D ShapeNets: A deep representation for volumetric shapes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[42] Saining Xie, Jiatao Gu, Demi Guo, Charles R. Qi, Leonidas Guibas, and Or Litany. PointContrast: Unsupervised pre-training for 3D point cloud understanding. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 574–591, Cham, 2020. Springer International Publishing.

[43] Jie Yang, Lin Gao, Yu-Kun Lai, Paul L. Rosin, and Shihong Xia. Biharmonic deformation transfer with automatic key point selection. *Graphical Models*, 98:1–13, 2018.

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

[45] Ling Zhang and Zhigang Zhu. Unsupervised feature learning for point cloud understanding by contrasting and clustering using graph convolutional neural networks. In *2019 International Conference on 3D Vision (3DV)*, pages 395–404, 2019.

[46] Richard Zhang, Phillip Isola, and Alexei A. Efros. Colorful image colorization. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, pages 649–666, Cham, 2016. Springer International Publishing.

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

[48] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017.

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