

Unsupervised Dense Deformation Embedding Network for Template-Free Shape Correspondence

Ronghan Chen^{1,2,3}Yang Cong^{1,2*}Jiahua Dong^{1,2,3}¹State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences[†]²Institutes for Robotics and Intelligent Manufacturing, Chinese Academy of Sciences³University of Chinese Academy of Sciences

chenronghan@sia.cn, congyang81@gmail.com, dongjiahua1995@gmail.com

Abstract

Shape correspondence from 3D deformation learning has attracted appealing academy interests recently. Nevertheless, current deep learning based methods require the supervision of dense annotations to learn per-point translations, which severely over-parameterize the deformation process. Moreover, they fail to capture local geometric details of original shape via global feature embedding. To address these challenges, we develop a new Unsupervised Dense Deformation Embedding Network (i.e., UD^2E -Net), which learns to predict deformations between non-rigid shapes from dense local features. Since it is non-trivial to match deformation-variant local features for deformation prediction, we develop an Extrinsic-Intrinsic Autoencoder to first encode extrinsic geometric features from source into intrinsic coordinates in a shared canonical shape, with which the decoder then synthesizes corresponding target features. Moreover, a bounded maximum mean discrepancy loss is developed to mitigate the distribution divergence between the synthesized and original features. To learn natural deformation without dense supervision, we introduce a coarse parameterized deformation graph, for which a novel trace and propagation algorithm is proposed to improve both the quality and efficiency of the deformation. Our UD^2E -Net outperforms state-of-the-art unsupervised methods by 24% on *Faust Inter* challenge and even supervised methods by 13% on *Faust Intra* challenge.

1. Introduction

Alignment of deformable 3D shapes is a ubiquitous challenge in the field of computer vision and graphics with

*The corresponding author is Prof. Yang Cong.

[†]This work is supported in part by the National Key Research and Development Program of China under Grant 2019YFB1310300 and the National Nature Science Foundation of China under Grant 61821005.

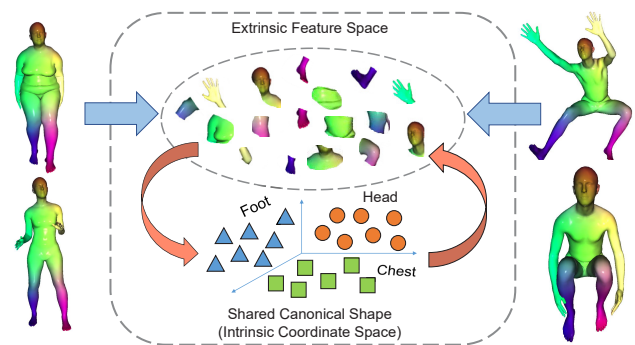


Figure 1. Illustration of our UD^2E -Net model, which achieves the bidirectional mapping between extrinsic feature space and intrinsic coordinate space for dense deformation embedding.

many applications, including non-rigid reconstruction, deformation transfer and texture mapping. Traditional methods [33, 23] infer parametric deformation by optimizing correspondences construction and objective minimization iteratively. However, the optimization heavily depends on initialization and is prone to being stuck in local minima, especially for large articulated deformations. With the advent of 3D deep learning techniques [28, 29], deep deformation learning methods [13, 14, 39, 43, 21] leveraging a large amount of data have been applied for large deformation prediction. Generally, these techniques directly regress dense translations or positions for all input 3D points, which neglects the implicit deformation disciplines and severely over-parameterizes the deformation leading to high-frequency artifacts. Unfortunately, these approaches require strong supervision of dense correspondences (e.g., usually more than 5000 per human body shape), which consumes huge labor-intensive efforts.

Furthermore, most existing state-of-the-art methods [21, 13, 14, 5, 39, 44, 15] encode the target shape into a global feature for deformation learning, which neglects the low-level geometric details and fails to infer fine-grained defor-

mation. To leverage local features for deformation prediction, connections between both sides should be established to enable feature communication, which is a challenging correspondence problem for deformable shapes.

To address these challenges, a new Unsupervised Dense Deformation Embedding Network (*i.e.*, UD²E-Net) endowed with the traditional *Embedded Deformation* (ED) technique [33] is designed to predict deformation between arbitrary source and target shape pairs. With the local rigidity regularization provided by ED, UD²E-Net can learn a more natural deformation space, which we expect to mitigate the strong reliance of deep learning models on abundant annotated data. Moreover, our network employs dense local feature embedding and fusion to reason about deformation parameters for each node within the deformation graph constructed via ED. Specifically, with the fine-grained geometric features extracted via the Siamese mesh encoder, an *Extrinsic-Intrinsic Autoencoder* (EI-AE) is developed to first encode source features into intrinsic coordinates of a shared canonical shape. With the coordinates, corresponding target features are synthesized via the decoder. To minimize the distribution gap between the synthesized and original target features, we design a Bounded Maximum Mean Discrepancy loss, which further provides the self-supervised guidance for canonical shape construction. To eliminate artifacts caused by wrong adjacent relationships between the input shape and its deformation graph via Euclidean measure, a trace and propagation algorithm is developed to improve both efficiency and accuracy by leveraging a pre-constructed mesh hierarchy. Extensive experiments demonstrate that UD²E-Net shows stable performance under data volume reduction. On Faust benchmark [1], the proposed UD²E-Net outperforms state-of-the-art unsupervised methods by 24%~37% on Inter challenge, and shows 13% improvement on Intra challenge over supervised methods. Furthermore, experiments also demonstrate the potential of our UD²E-Net in several challenging applications, *e.g.*, shape retrieval and human pose transfer.

In conclusion, the main contributions of our work are:

- A new Unsupervised Dense Deformation Embedding Network (*i.e.*, UD²E-Net) is developed to learn large articulated deformations between arbitrary shape pairs without supervision of ground-truth correspondences.
- We design an Extrinsic-Intrinsic Autoencoder to encode extrinsic geometric features from source shape into a shared canonical shape, which is utilized to decode corresponding synthesized target features. Meanwhile, a bounded maximum mean discrepancy loss is introduced to mitigate the distribution divergence between the synthesized and original target features.
- A trace and propagation algorithm is developed to avoid artifacts brought by incorrect node-to-vertex as-

ignment in Embedded Deformation, which improves both quality and efficiency of the deformation process.

2. Related Work

Non-rigid Shape Matching. Generally, methods for shape correspondence can be divided into intrinsic methods and extrinsic methods. Intrinsic methods rely on intrinsic property of shapes, such as spectral [34, 4] or learning based [37] descriptors or preservation of geodesic distances [24] to directly obtain correspondences. However, most intrinsic methods are not robust enough under perturbation, such as topology changes and incompleteness. Other extrinsic methods achieve correspondences by explicitly deforming one shape to align the other [22, 23], where hand-crafted deformation models [33, 32] are applied for regularization. However, they are prone to being stuck in local minima.

Deformation Representation for Learning. A direct and most commonly applied way to deform a 3D model is to assign each point a vector, representing translation or position [14, 13, 39, 5]. Such deformation is endowed with high degree of freedom, which binds the number of predicted parameters to resolution of the input model. For low-dimensional deformation representations, free-form deformation [31] is utilized in [18] to deform meshes by manipulating several predefined vertices of structured grids. LBS autoencoder [21] uses predefined LBS model to deform human body and hands, which cannot adapt to general objects. [44] utilizes the cage-based deformation technique [19] for detail-preserving deformation and cannot achieve perfect alignment. DEMEA [35] proposes an embedded deformation layer based on ED [33] to learn mesh representations. Obviously, ED is vastly limited by fully connected layers, which are known to be permutation variant and cannot deal with unordered point clouds and meshes.

Deep Deformation Embedding. Existing methods typically embed the target shape into a global feature, and then rely on a vector it concatenates with for location cues to express different deformation information [14, 13, 5, 44, 39, 16]. However, they cannot provide fine geometric details. Other methods infer deformation from local features. Pixel2Mesh [38, 42] assigns local features of 2D image to vertices of an ellipsoid template by 3D-2D projection. Flownet3d [25] mixes features from two point clouds based on Euclidean neighbours, which are not reliable under large articulated deformations. Unlike them, our model can learn stable feature pairs via the canonical shape built by EI-AE.

3. Our Proposed Model

3.1. Preliminary

Embedded Deformation. We build upon the *Embedded Deformation* (ED) algorithm [33] to model non-rigid deformation. To deform a mesh $M = (V, E)$ with N vertices

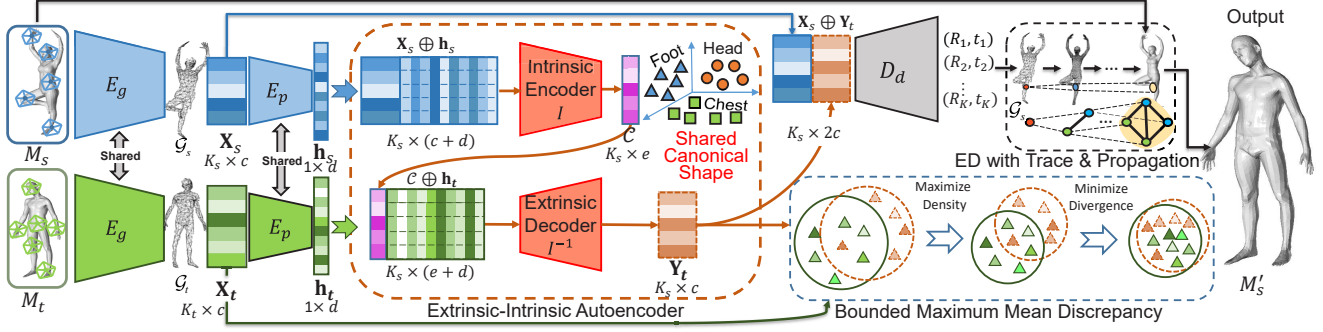


Figure 2. Overview architecture of UD²E-Net. After extracting local features $\mathbf{X}_s, \mathbf{X}_t$ via the Siamese mesh encoder E_g and global features $\mathbf{h}_s, \mathbf{h}_t$ via a PointNet-like network E_p , the intrinsic encoder \mathcal{I} in EI-AE encodes \mathbf{X}_s into the coordinates in the shared canonical shape \mathcal{C} , which is concatenated with \mathbf{h}_t and further decoded by the extrinsic decoder \mathcal{I}^{-1} to obtain corresponding local features \mathbf{Y}_t . The bounded maximum mean discrepancy is applied to eliminate the domain gap between \mathbf{X}_t and \mathbf{Y}_t . Then the deformation decoder D_d decodes the concatenation of \mathbf{Y}_t and \mathbf{X}_s into deformation parameters in \mathcal{G}_s to output the deformation prediction of M_s .

$V = \{\mathbf{v}_i\}_{i=1}^N$ and edges E , ED constructs a coarse embedded deformation graph \mathcal{G} to manipulate M . $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ is a low-resolution M and is composed of a node set $\mathcal{N} = \{\mathbf{g}_i\}_{i=1}^K$ and an edge set \mathcal{E} . Each node $\mathbf{g}_i \in \mathbb{R}^3$ is parameterized with transformation $\mathbf{T}_i = (\mathbf{R}_i, \mathbf{t}_i)$ to control the local deformation, where \mathbf{R}_i is a rotation matrix and $\mathbf{t}_i \in \mathbb{R}^3$ is a translation vector. With local deformations $T = \{(\mathbf{R}_i, \mathbf{t}_i)\}_{i=1}^K$, M can be deformed by:

$$\mathbf{v}'_j = \sum_{\mathbf{g}_i \in \mathcal{A}(\mathbf{v}_j)} w_{i,j} (\mathbf{R}_i(\mathbf{v}_j - \mathbf{g}_i) + \mathbf{g}_i + \mathbf{t}_i), \quad (1)$$

where \mathbf{v}_j is the position of a vertex on M and \mathbf{v}'_j is the deformed \mathbf{v}_j . $\mathcal{A}(\mathbf{v}_j)$ is the set of nodes controlling the deformation of vertex \mathbf{v}_j , and $w_{i,j}$ is the node-to-vertex weight determined by distance between \mathbf{v}_j and \mathbf{g}_i . Finally, an as-rigid-as-possible loss is utilized to further balance the deformation and smoothness:

$$\mathcal{L}_{\text{ARAP}} = \sum_i \sum_{j:(i,j) \in \mathcal{E}} \|(\mathbf{R}_i(\mathbf{g}_j - \mathbf{g}_i) + \mathbf{g}_i + \mathbf{t}_i) - (\mathbf{g}_j + \mathbf{t}_j)\|, \quad (2)$$

where $\mathbf{g}_i, \mathbf{g}_j$ are adjacent nodes connected with an edge in \mathcal{E} . The $\mathcal{L}_{\text{ARAP}}$ term indicates that the local deformations of adjacent nodes should agree with one another.

Given a source mesh M_s and a target mesh M_t , as shown in Figure 2, the goal of our unsupervised dense deformation embedding network (UD²E-Net) is to output a deformed M'_s to align with M_t for matching, by predicting the local transformation $T = \{(\mathbf{R}_i, \mathbf{t}_i)\}_{i=1}^K$ of the deformation graph \mathcal{G} . To learn these transformations, we propose a novel Extrinsic-Intrinsic Autoencoder to correlate the extracted local geometric information from input shapes and fuse them into dense deformation embeddings, which is introduced as follows.

3.2. Extrinsic-Intrinsic Autoencoder

As shown in Figure 2, we utilize a Siamese mesh encoder E_g to encode source shape M_s and target shape M_t into dense geometric embeddings for each node in deformation graph. Due to incompatible resolution between input meshes and deformation graphs, we utilize graph pooling layers to gradually downsample M_s and M_t into deformation graphs \mathcal{G}_s and \mathcal{G}_t , which forms a mesh hierarchy. The outputs of Siamese mesh encoder are two sets of c -dimensional features $\mathbf{X}_s = [\mathbf{x}_1^s, \dots, \mathbf{x}_{K_s}^s] \in \mathbb{R}^{K_s \times c}$ and $\mathbf{X}_t = [\mathbf{x}_1^t, \dots, \mathbf{x}_{K_t}^t] \in \mathbb{R}^{K_t \times c}$, as shown in Figure 2, where each feature $\mathbf{x}_i^s \in \mathbb{R}^c$ ($i = 1, \dots, K_s$) corresponds to a node \mathbf{g}_i^s in the node set \mathcal{N}_s of \mathcal{G}_s and same for \mathbf{x}_j^t . Then, we utilize a PointNet-like [28] encoder E_p to aggregate \mathbf{X}_s and \mathbf{X}_t into d -dimensional global feature vectors $\mathbf{h}_s, \mathbf{h}_t \in \mathbb{R}^d$, respectively.

Typically, most previous methods [13, 5, 39] employ the global feature for deformation reasoning, which may lose local geometric information thus degrades the performance of deformation prediction. To tackle this issue, we utilize local features from source and target shapes (\mathbf{X}_s and \mathbf{X}_t) to embed deformation information for nodes in \mathcal{N}_s . However, it is non-trivial to match these deformation-variant local features. Former methods leverage projection [38], or find nearest points in Euclidean space [25] to directly match features from source and target shapes. Nevertheless, they can lead to faulty correspondences due to large articulated deformation between input shapes. A seemingly promising solution is to learn deformation-invariant features embedded with intrinsic property [40] for matching, which, however, forces them to abandon essential extrinsic geometric information for deformation prediction.

Therefore, we propose a novel *Extrinsic-Intrinsic Autoencoder* (EI-AE) to learn both intrinsic properties for matching and extrinsic properties for deformation prediction. As shown in Fig. 2, it constructs a shared canonical

shape $\mathcal{C} = [\mathbf{c}_1, \dots, \mathbf{c}_{K_s}] \in \mathbb{R}^{K_s \times e}$, where e is the dimension of the canonical shape and \mathbf{c}_i is the intrinsic coordinate shared by corresponding vertices on all shapes. Given the local feature \mathbf{x}_i^s and the global feature of source mesh \mathbf{h}_s , EI-AE learns to parameterize each node \mathbf{g}_i^s in \mathcal{G}_s with its intrinsic coordinate \mathbf{c}_i via an *Intrinsic Encoder* \mathcal{I} :

$$\mathcal{C} = \mathcal{I}(\mathbf{X}_s \oplus \mathbf{h}_s), \quad (3)$$

where \oplus represents feature concatenation operation. Due to coarseness of \mathcal{G}_s and \mathcal{G}_t , there are often no exact matches between \mathbf{X}_s and \mathbf{X}_t . Thus, we utilize the intrinsic coordinates in \mathcal{C} to synthesize the corresponding extrinsic target features $\mathbf{Y}_t = [\mathbf{y}_1^t, \dots, \mathbf{y}_{K_s}^t] \in \mathbb{R}^{K_s \times c}$ from the global feature of target shape \mathbf{h}_t via the *Extrinsic Decoder* \mathcal{I}^{-1} :

$$\mathbf{Y}_t = \mathcal{I}^{-1}(\mathcal{C} \oplus \mathbf{h}_t). \quad (4)$$

Note that, in this way, EI-AE can provide ultimate flexibility as it can correlate features from two deformation graphs \mathcal{G}_s , \mathcal{G}_t with different number of nodes. Since the synthesized features \mathbf{y}_i^t in \mathbf{Y}_t and \mathbf{x}_i^s in \mathbf{X}_s are expected to be corresponded, the final output of EI-AE is the concatenation of \mathbf{X}_s and \mathbf{Y}_t , *i.e.*, $[\mathbf{X}_s, \mathbf{Y}_t] \in \mathbb{R}^{K_s \times 2c}$, which are then sent into the deformation decoder to predict deformation parameters $T = \{(\mathbf{R}_i, \mathbf{t}_i)\}_{i=1}^{K_s}$ for \mathcal{G}_s .

3.3. Bounded Maximum Mean Discrepancy

Though the shared canonical shape \mathcal{C} solves the problem of correlating extrinsic features, it is difficult for EI-AE to figure out how to form such a canonical shape without any self-supervision, which would result in severe overfitting. The reason for this is the domain gap between input features $\mathbf{X}_t \sim \mathcal{X}$ and the synthesized features $\mathbf{Y}_t \sim \mathcal{Y}$, where \mathcal{X} and \mathcal{Y} are the domains of input and synthesized features, respectively. Inspired by domain adaptation techniques [26, 9, 8, 7], we propose to bridge the domain gap between \mathcal{X} and \mathcal{Y} by minimizing a Maximum Mean Discrepancy (MMD) [12] loss:

$$\begin{aligned} M(\mathbf{X}_t, \mathbf{Y}_t) = & \mathbb{E}_{\mathbf{x}_i, \mathbf{x}_j \in \mathbf{X}}[\kappa(\mathbf{x}_i, \mathbf{x}_j)] + \mathbb{E}_{\mathbf{y}_i, \mathbf{y}_j \in \mathbf{Y}}[\kappa(\mathbf{y}_i, \mathbf{y}_j)] \\ & - 2\mathbb{E}_{\mathbf{x}_i \in \mathbf{X}, \mathbf{y}_j \in \mathbf{Y}}[\kappa(\mathbf{x}_i, \mathbf{y}_j)], \end{aligned} \quad (5)$$

where we omit the superscripts t , and κ is the Radial Basis Function (RBF) kernel with $\kappa_\sigma(x, y) = \exp(-\frac{1}{2\sigma^2}\|x - y\|^2)$. We follow previous methods [20] to apply a linear combination of kernels with five scales $\kappa(x, y) = \sum_i^5 \kappa_{\sigma_i}(x, y)$, where $\sigma_i \in \{1, \sqrt{2}, 2, 2\sqrt{2}, 4\}$.

The motivation of this loss is to encourage the feature distributions inside original feature domain \mathcal{X} and synthesized feature domain \mathcal{Y} to be more compact (*i.e.*, the first two terms of Eq. (5)), and meanwhile drives the synthesized feature domain \mathcal{Y} to align with the original feature

domain \mathcal{X} by minimizing the distribution divergence across these two domains (*i.e.*, the last term of Eq. (5)). It has been proven that $M(\mathcal{X}, \mathcal{Y}) \geq 0$ and the equality holds if and only if $\mathcal{X} = \mathcal{Y}$ [12]. However, with a limit number of samples from \mathcal{X} and \mathcal{Y} , optimizing MMD to be 0 may corrupt the deformation information in \mathbf{X}_t and \mathbf{Y}_t . Thus, inspired by hinge loss, we introduce a bounded maximum mean discrepancy loss, with the bound $\beta = 0.01$ in all experiments:

$$\mathcal{L}_{\text{Feat}} = \max(0, M(\mathbf{X}_t, \mathbf{Y}_t) - \beta). \quad (6)$$

3.4. Trace and propagation

In Embedded Deformation (ED), a vertex v_i^s on the source mesh M_s is controlled by its k nearest nodes $\mathcal{A}(v_i^s) = \{\mathbf{g}_j^s\}_{j=1}^k$ in its deformation graph \mathcal{G}_s . Different from former methods with a fixed template [35], our method seeks to deform from source meshes with different topologies. Thus, the control nodes $\mathcal{A}(v_i^s)$ for each vertex v_i^s need to be redefined for each source mesh. Previous methods find $\mathcal{A}(v_i^s)$ by searching k nearest neighbour (*knn*) of v_i^s based on either geodesic or Euclidean distance. However, the geodesic distance is too heavy to compute, and the Euclidean distance may find wrong neighbouring nodes. For example, as shown in Fig. 3, in M_s , the right arm is too close to the rib. Control nodes in arm are wrongly assigned to vertices in rib based on Euclidean distance, thus bring the ribs up with the lift of the arm. To this end, we propose a trace and propagation algorithm to find the control nodes $\mathcal{A}(v_i^s)$, which is very efficient without any *knn* search but can also improve the accuracy of the neighbours.

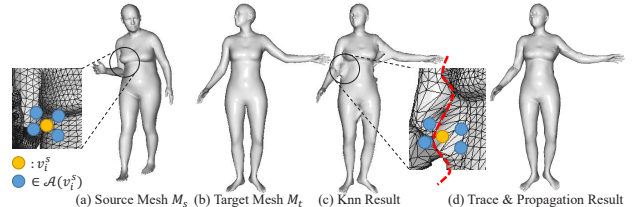


Figure 3. Comparison of using *knn* in Euclidean space and our trace and propagation for node-to-vertex assignment.

Our main idea is to first trace back through the mesh hierarchy that is pre-constructed in the mesh encoder, and then propagate on the deformation graph to define more accurate neighbours. The mesh hierarchy (M_1, M_2, \dots, M_L) , starting with the input mesh M_1 and ending with the coarsest deformation graph M_L , is generated by a series of poolings $(\mathcal{T}_1, \mathcal{T}_2, \dots, \mathcal{T}_{L-1})$. As shown in Figure 4(a), a pooling operation $\mathcal{T}_l : \mathcal{V}^l \rightarrow \mathcal{V}^{l+1}$ maps a small partition of neighbouring vertices $\mathcal{O} = \{\mathbf{v}_i^l\} \subset \mathcal{V}^l$, $\mathbf{v}_i^l \in \mathbb{R}^3$ of mesh M_l to one vertex \mathbf{v}_j^{l+1} in M_{l+1} by

$$\mathbf{v}_j^{l+1} = \sum_{\mathbf{v}_i^l \in \mathcal{O}} w_i^l \mathbf{v}_i^l, \quad (7)$$

with the weight $w_i^l \geq 0$ and $\sum w_i^l = 1$. We further define a tracing operation \mathcal{T}_l^{-1} as $\mathcal{T}_l^{-1}(v_j^{l+1}) = \mathcal{O}$, which restores the vertices in \mathcal{O} from v_j^{l+1} . Then, a node g_i^s in M_L can trace back to M_1 layer by layer (i.e., $C_i = \mathcal{T}_1^{-1} \circ \mathcal{T}_2^{-1} \circ \dots \circ \mathcal{T}_{L-1}^{-1}(g_i^s)$) to form a tree, as shown in Figure 4(b), where C_i is the leaf set of the tree and contains all the original vertices in M_1 that are merged into g_i^s . As shown in Figure 4(c), C_i can be regarded as a cluster of neighbouring vertices. According to Eq. (7), since g_i^s is derived by a series of interpolation of vertices in C_i , g_i^s should lie near the center of C_i . For instance, if all w_i are equal then g_i^s is the center of C_i . Thus, we regard g_i^s as the first control node for vertices in C_i , and define the rest of the control nodes for vertices in C_i as the neighbours of g_i^s in deformation graph \mathcal{G}_s (i.e., $\mathcal{A}(v_i^s) = \{g_i^s\} \cup \{g_j^s | (i, j) \in \mathcal{E}_s\}$).

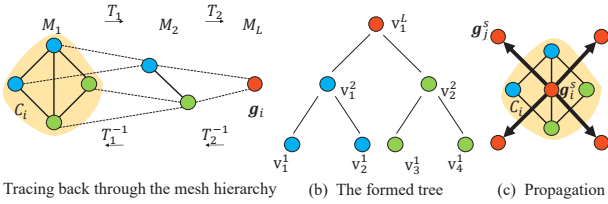


Figure 4. Example of the proposed *trace and propagation* algorithm on a 3-level mesh hierarchy.

The trace operation can be easily implemented as a look-up operation with computation complexity $O(N)$, where N is the number of vertices in M_1 . As shown in Figure 3(d), with our proposed trace and propagation algorithm, the deformed mesh is smoother without any artifacts.

3.5. Formulation

We train our proposed UD²E-Net without any supervision of ground-truth correspondences. Formally, the formulation objective consists of three main terms:

$$\mathcal{L} = \mathcal{L}_{\text{Align}} + \mathcal{L}_{\text{Smooth}} + \lambda_f \mathcal{L}_{\text{Feat}}, \quad (8)$$

where $\mathcal{L}_{\text{Align}}$ drives the source shape to align with the target, $\mathcal{L}_{\text{Smooth}}$ regularizes the network to learn reasonable deformation based on smoothness and local rigidity. The third term is designed for EI-AE to align the feature space. The balanced weight λ_f is set as 0.008 for all experiments.

For $\mathcal{L}_{\text{Align}}$, we minimize the symmetric *Chamfer Distance* \mathcal{L}_{Ch} between the vertices set V'_s, V_t of deformed source M'_s and the target mesh M_t . However, for large deformation of articulated human bodies, \mathcal{L}_{Ch} may provide incorrect correspondences and lead to a bad local minima. Inspired by [15], we define a cycle-consistent loss \mathcal{L}_{Cyc} to enforce cycle-consistent correspondences between M_s and M_t , which indicates that a cycle of deformation $M_s \rightarrow M_t \rightarrow M_s$ should map a point $p \in V_s$ back to itself. Note that, to calculate this loss, [15] forwards M_s and M_t twice for the deformations $M_s \rightarrow M_t$ and

$M_t \rightarrow M_s$, which could be a large computational cost. While our Siamese pair of mesh encoders only forward the inputs once since they share the same parameters. Therefore, the objective $\mathcal{L}_{\text{Align}}$ is:

$$\mathcal{L}_{\text{Align}} = \mathcal{L}_{\text{Cyc}} + \mathcal{L}_{\text{Ch}}. \quad (9)$$

To regularize the deformation, motivated by [13], we introduce an edge length term $\mathcal{L}_{\text{Edge}}$ to penalize edge length changes and a Laplacian term \mathcal{L}_{Lap} to preserve local geometric details. With the as-rigid-as-possible term $\mathcal{L}_{\text{ARAP}}$ in Eq. (2) for surface smoothness, $\mathcal{L}_{\text{Smooth}}$ is formulated as:

$$\mathcal{L}_{\text{Smooth}} = \lambda_{\text{Edge}} \mathcal{L}_{\text{Edge}} + \lambda_{\text{Lap}} \mathcal{L}_{\text{Lap}} + \lambda_{\text{ARAP}} \mathcal{L}_{\text{ARAP}}, \quad (10)$$

where the balanced weights $\lambda_{\text{Edge}}, \lambda_{\text{Lap}}, \lambda_{\text{ARAP}}$ are set as 0.005.

4. Experiments

4.1. Dataset

To validate the effectiveness of our model in both shape correspondence and shape reconstruction tasks, we evaluate the proposed UD²E-Net on several challenging datasets, i.e., watertight meshes (DFAUST [2], SURREAL [36], CoMA [30]) and real scans from FAUST [1].

FAUST [1] consists of 100 watertight meshes with ground-truth correspondences for training and 200 meshes for testing. The test data are real scans with noise and holes. DFAUST [2] consists of sequences of meshes. There are 41220 watertight meshes in total. We follow the train-test split in [35] for fair comparison. SURREAL [36] is a synthetic dataset based on SMPL [27] model. We follow [13] to generate 230k meshes including 200k human poses from [36] and 30k bent poses that can be unrealistic. We also sample 23k samples with 20k from [36] and 3k bent shapes to form a smaller training set. The testing set contains 200 meshes sampled from [36]. CoMA [30] contains 17794 watertight meshes of human faces spanning 12 subjects in 12 different expressions. The train-test split is also defined as in [35].

For all human body datasets, the meshes share the same topology with 6890 vertices. We decimate these meshes into 2757 vertices using Quadratic Error Metrics [10] for training. When testing, the network also takes paired decimated meshes as input but output a high-resolution mesh by directly applying the predicted deformation parameters on the high-resolution source mesh. By doing this, we try to highlight that UD²E-Net is capable of decoupling the number of deformation parameters from mesh resolution. Moreover, this also greatly accelerates the training process.

4.2. Implementation Details

Siamese Mesh Encoder. Our Siamese mesh encoder consists of graph convolution layers and graph pooling layers

defined on a mesh hierarchy, which is constructed with Graculus [6] algorithm. We apply three downsampling layers to explore four mesh levels. The vertex number of each mesh level is approximately half of the preceding level. In each mesh level, we apply two *EdgeConvs* [41] with a residual connection for feature extraction.

Deformation Decoder. The architecture of deformation decoder D_d follows mesh encoder but with only one mesh level to further build connections between neighbour nodes g_i^s . It encourages the predicted transformations T_i to be closely similar for smoothness. We use two multi-layer perceptrons that end with linear layers to regress rotation R_i and translation $t_i \in \mathbb{R}^3$, respectively. For rotation, we employ a 6D over-parameterized representation [45], given by the first two rows of the rotational matrix.

Training Strategy. Our UD²E-Net is trained end-to-end from scratch. We train it via the formulation introduced in Sec. 3.5 and utilize the Adam optimizer with a fixed learning rate of 0.001. When the cycle-consistent loss stops to decrease after a few epochs, we train UD²E-Net via $\mathcal{L}_{\text{Align}} = \mathcal{L}_{\text{Ch}}$ for the rest of epochs. Specifically, we first train UD²E-Net on SURREAL dataset (230k) for 10 epochs, followed by 20 epochs without \mathcal{L}_{Cyc} . On SURREAL dataset (23k), CoMA and DFAUST, UD²E-Net is trained with \mathcal{L}_{Cyc} for 20 epochs followed by 180 epochs without it. For the evaluation on FAUST scans, we follow [13, 5] to train UD²E-Net on SURREAL dataset (230k) without finetuning on the FAUST training set.

4.3. Baseline and Evaluation Criterion

Baseline. To demonstrate the advantages of the fine-grained fusion enabled by EI-AE over the common global feature, we build a strong baseline, *global embedding* (GE), which directly concatenates source local features \mathbf{X}_s with the target global feature \mathbf{h}_t as deformation embeddings. See the supplemental for detailed architecture.

Evaluation Criterion. We evaluate our method on shape reconstruction and shape correspondence tasks. The reconstruction error measures the distance between the input target shape and reconstructed shape by computing Chamfer distance (R_{Ch}) or mean Euclidean distance between ground-truth point pairs (R_{Gt}). Correspondence error compares mean Euclidean distance between deformed points (C_{De}) or their projections on target (C_{Pr}) and the ground-truth correspondences on target. For template-based methods, correspondences are built following [13].

4.4. Comparison Results

Experiments on Watertight Meshes. As shown in Table 1, we first compare our method with state-of-the-art methods on synthetic SURREAL dataset. The comparing methods apply a template-based [13, 5, 35] or an autoencoder framework [3, 30]. For shape reconstruction, we also employ

a fixed template as the source mesh M_s for testing. Different from [13, 5] that apply the same template shape in both training and testing, UD²E-Net is trained to reconstruct from random source shapes. Thus the comparison is not entirely fair. However, on SURREAL dataset (230k), our UD²E-Net still outperforms all supervised methods 3D-CODED [13] and Elementary [5]. With less sufficient training data (SURREAL 23k), the performance of 3D-CODED [13] and Elementary [5] drops significantly up to 0.36cm. In contrast, neither UD²E-Net nor UD²E-Net (GE) experience notable accuracy drop ($\leq 0.11\text{cm}$) and they outperform all supervised methods, which proves the effectiveness of ED to alleviate strong dependence on data volume. For shape correspondence task, our UD²E-Net outperforms the methods with a fixed template [13, 5] by a large margin (0.25~0.72cm for 230k and 1.19~1.32cm for 23k), which shows the natural advantage of our template-free framework on shape correspondence task.

On DFAUST dataset, variation of human poses and shapes are relatively limited, which results in poor performance for deformation-by-translation methods [13, 5] with $C_{\text{De}} \geq 6.46\text{cm}$ and $R_{\text{Gt}} \geq 4.77\text{cm}$, since they fail to form a latent deformation space. On the contrary, with ED, UD²E-Net holds high accuracy of 2.41cm and 2.17cm for C_{De} and R_{Gt} . When pretrained on 230k samples from SURREAL, UD²E-Net achieves 1.84cm for reconstruction and outperforms all competing methods even the autoencoder N. 3DMM [3] (1.99cm). Our method also suits for finer deformation of faces in CoMA dataset. As shown in Table 3, the proposed UD²E-Net (0.78mm) also outperforms the supervised DEMEA [35] (0.81mm). Moreover, we also show qualitative comparison with 3D-CODED [13] and Elementary [5] in Figure 5. The comparison methods fail to capture local geometric details with a global feature, thus results in local distortion, such as in hands, arms and especially faces where rich geometric details exist.

Experiments on Real Scans. We then test our UD²E-Net on more challenging scan dataset, where missing parts (*e.g.*, feet, hands, etc.) may exist due to self-occlusion. These meshes have approximately 170000 vertices. Instead of directly sending them into the network, we follow the pre-processing strategy for training to first downsample the scanned meshes to 2757 vertices. Since the testing meshes may have topological changes caused by self-contact, UD²E-Net may fail to deform parts with topological noise. Thus, given a pair of testing meshes, we also utilize a watertight mesh as a template to build correspondences as in [13, 5]. Though a template is employed, our method is still naturally superior to template-based methods since we can freely choose better templates. In practice, we choose the null pose of each test human as template. Similar to [13, 5], we optimize global feature of the target to obtain the final reconstruction. Some comparison

Data	Methods	R _{Gt}	R _{Ch}	C _{De}	C _{Pr}
230k	3D-CODED [13]	1.80	1.03	2.63	2.30
	Elementary [5]	1.67	0.89	2.19	1.87
	Unsup. 3D-CODED [13]	9.43	1.76	9.30	9.27
	Ours-GE	1.96	1.00	2.23	1.88
	Ours-w/oTP	3.25	1.15	2.85	3.00
	Ours-w/oMMD	2.11	1.06	2.47	2.12
	Ours-3D	1.71	0.86	2.02	1.96
	Ours	1.67	0.87	1.91	1.62
23k	3D-CODED [13]	2.10	1.18	3.36	3.01
	Elementary [5]	2.03	1.15	3.27	2.91
	Unsup. 3D-CODED [13]	9.32	1.86	9.59	9.43
	Ours-GE	2.22	1.07	2.61	2.28
	Ours-w/oTP	2.18	1.04	3.08	2.80
	Ours-w/oMMD	2.16	1.05	2.47	2.13
	Ours-3D	1.83	0.93	2.15	1.84
	Ours	1.78	0.911	2.04	1.72

Table 1. Performance comparison on synthetic SURREAL dataset [36]. ‘230k’ and ‘23k’ denote the amount of training data.

	Methods	C _{De} (cm)	R _{Gt} (cm)
Original	N. 3DMM [3]	-	1.99
	CoMA [30]	-	8.4
	DEMEA [35]	-	2.23
	3D-CODED [13]	6.46	4.81
	Elementary [5]	6.51	4.77
	Ours-GE	3.03	2.57
	Ours	2.41	2.17
230k	3D-CODED [13]	4.91	3.55
	Elementary [5]	4.58	3.36
	Ours-GE	2.24	1.95
	Ours	2.07	1.84

Table 2. Performance comparison on DFAUST dataset. ‘Original’ denotes only training on DFAUST[2]. ‘230k’ means finetuning on DFAUST with pretrained network on SURREAL (230k).

	DEMEA	N. 3DMM	Ours
R _{Gt} (mm)	0.81	0.71	0.78

Table 3. Performance comparison on CoMA dataset [30].

results are shown in Table 4. On the ‘Inter’ challenge, our method achieves 3.086cm, which outperforms unsupervised methods significantly by 24%~37% and is comparable to supervised methods. On the ‘Intra’ challenge, our method achieves **1.512**cm, which even outperforms state-of-the-art supervised method [5] by 13%.

Ablation Study. As shown in Table 1, to demonstrate the effectiveness of each proposed component, we compare our final UD²E-Net with several variants including: 1) the baseline (Ours-GE), 2) UD²E-Net without MMD (Ours-w/oMMD), 3) UD²E-Net with *knn* instead of trace and propagation (Ours-w/oTP), and 4) UD²E-Net 3D that applies 3D canonical shape in the EI-AE (Ours-3D). Without

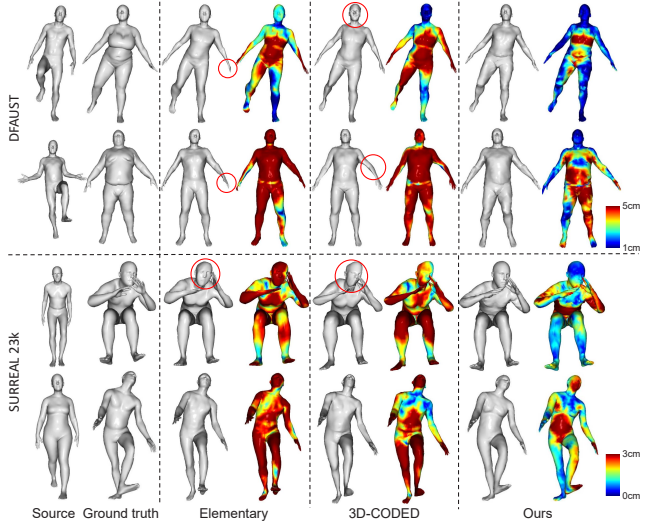


Figure 5. Qualitative comparison on DFAUST [2] and SURREAL 23k with 3D-CODED [13] and Elementary [5]. For each method, the left is the deformed source and right is its error map.

	Methods	Inter.	Intra.
Sup.	FMNet [24]	4.83	2.44
	3D-CODED [13]	2.878	1.985
	Elementary [5]	2.58	1.742
Unsup.	Unsup. 3D-CODED [13]	4.88	-
	LBS Autoencoder [21]	4.08	2.161
	Halimi <i>et al.</i> [17]	4.883	2.51
	Ginzburg <i>et al.</i> [11]	4.068	2.12
	Ours	3.086	1.512

Table 4. Performance comparison on Faust dataset. ‘Sup.’ and ‘Unsup.’ indicate the method is supervised or unsupervised.

the proposed EI-AE, the performance drops 13% in shape correspondence, which demonstrates the indispensable role of fine-grained feature fusion in deformation embedding. Moreover, the employed bounded MMD is critical to align the distributions between synthetic target features generated by EI-AE and the original features, and guide EI-AE to form the shared canonical shape. Without bounded MMD, the performance drops 23%~17%. Without accurate control nodes provided by the proposed trace and propagation algorithm, the performance drops 33%~51% on shape correspondence. With 3D canonical shape \mathcal{C} , UD²E-Net performs slightly worse than using a 10D shape, which shows that UD²E-Net is not sensitive to shape dimension.

4.5. Effect of the Extrinsic-Intrinsic Autoencoder

We study the effect of EI-AE on dense deformation embedding by exploring how the intrinsic coordinates distributed inside the shared canonical shape and the effect of shared coordinates across different bodies. We first visualize the distribution of 3D coordinates in Figure 6, which shows that the learned canonical shape presents the shape

of the skeleton of a body, whose limbs and head are consistent with the original human body. Though input bodies are highly different, the derived canonical shapes are nearly identical. Between these two shapes, we build correspondences by finding nearest neighbour based on coordinates c_i . As shown in Figure 7, the proposed EI-AE is capable of mapping corresponding vertices across different bodies into consistent coordinates in \mathcal{C} , though without any supervision of correspondences. Above results have proved our assumption that the EI-AE disentangles intrinsic location from geometric information, which enables the synthesized Y_t to focus on capturing geometric information, thus boosting the deformation quality. Moreover, since the canonical shape implies correspondences, it can help with unsupervised deformation learning by enabling the network to learn a deformation space that complies with these correspondences.

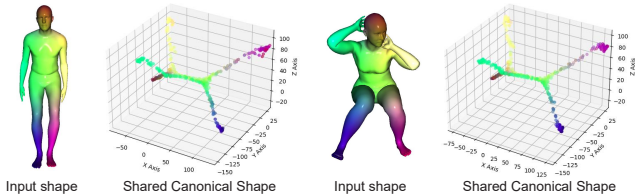


Figure 6. Visualization of the learned canonical shape in EI-AE. Same color indicates correspondence.

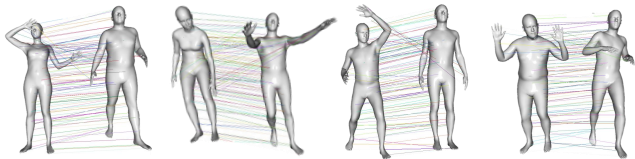


Figure 7. Results of point matches based on the coordinates c_i .

4.6. Applications

This subsection validates that our UD²E-Net could be successfully extended into several applications, *e.g.*, shape retrieval and human pose transfer. More results including shape interpolation are included in the supplemental.

Template Retrieval. For shape reconstruction problem, choosing a more similar template can obviously improve the performance. Given a target shape from the test set, we further try to retrieve the source shape from training set such that the cosine distance between their global embeddings are minimized. As shown in Table 5, where ‘Fixed’ and ‘Retrieved’ denote using a fixed or retrieved template, our model provides an additional improvement of 4.8% on DFAUST and a significant promotion of 37.1% on SURREAL. As shown in Figure 8, the retrieved shapes have similar poses with the query shapes.

Human pose transfer. The learned latent space also allows to transfer poses for human bodies. Given the meshes of human body A in two poses A_0, A_1 , we represent their embedded global features as h_0, h_1 . Then given another

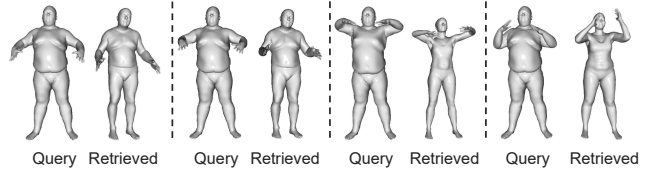


Figure 8. Results of shape retrieval on DFAUST dataset, where the query shapes are on the left and retrieved shapes are on the right.

	DFAUST		SURREAL (23K)	
	Ours (GE)	Ours	Ours (GE)	Ours
Fixed	1.95	1.84	2.22	1.78
Retrieved	1.87	1.752	1.67	1.12
Δ	-4.1%	-4.8%	-24.8%	-37.1%

Table 5. Performance improvement brought by template retrieval.

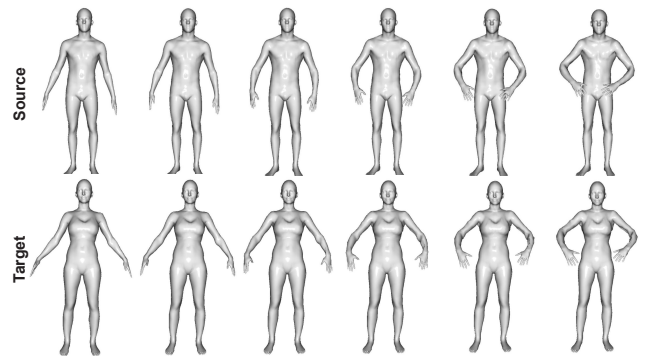


Figure 9. Deformation transfer results.

human body B_0 , which is of the same pose as A_0 with its embedding denoted as h'_0 , we seek a target shape B_1 performing same pose as A_1 . The deformation is transferred by computing the embedding of B_1 as $h'_1 = h_1 - h_0 + h'_0$. Then B_1 can be derived by decoding h'_1 . We evaluate the effectiveness of our model in a more challenging situation by transferring interpolated poses. As the qualitative results shown in Figure 9, the model trained on SURREAL dataset could be employed to transfer poses in FAUST dataset.

5. Conclusion

We have presented UD²E-Net, an end-to-end network that allows learning the deformation between arbitrary shape pairs from dense local features in an unsupervised manner. With the Extrinsic-Intrinsic Autoencoder guided by bounded MMD, the network can explore both extrinsic property for fine-grained deformation reasoning and intrinsic property for matching by forming a shared canonical shape. Moreover, the trace and propagation algorithm can improve both efficiency and quality of the deformation graph. UD²E-Net shows significant improvement over unsupervised methods and even outperforms supervised methods. We hope that our work inspires the field of neural deformation and can be extended to more challenging applications, *e.g.*, non-rigid reconstruction and tracking.

References

- [1] Federica Bogo, Javier Romero, Matthew Loper, and Michael J. Black. FAUST: Dataset and evaluation for 3D mesh registration. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Piscataway, NJ, USA, June 2014. IEEE.
- [2] Federica Bogo, Javier Romero, Gerard Pons-Moll, and Michael J. Black. Dynamic FAUST: Registering human bodies in motion. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, July 2017.
- [3] 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.
- [4] Michael M Bronstein and Iasonas Kokkinos. Scale-invariant heat kernel signatures for non-rigid shape recognition. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1704–1711. IEEE, 2010.
- [5] Theo Deprelle, Thibault Groueix, Matthew Fisher, Vladimir Kim, Bryan Russell, and Mathieu Aubry. Learning elementary structures for 3d shape generation and matching. In *Advances in Neural Information Processing Systems*, pages 7435–7445, 2019.
- [6] Inderjit S Dhillon, Yuqiang Guan, and Brian Kulis. Weighted graph cuts without eigenvectors a multilevel approach. *IEEE transactions on pattern analysis and machine intelligence*, 29(11):1944–1957, 2007.
- [7] Jiahua Dong, Yang Cong, Gan Sun, and Dongdong Hou. Semantic-transferable weakly-supervised endoscopic lesions segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10711–10720, October 2019.
- [8] Jiahua Dong, Yang Cong, Gan Sun, Yuyang Liu, and Xiaowei Xu. Cscl: Critical semantic-consistent learning for unsupervised domain adaptation. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *European Conference on Computer Vision – ECCV 2020*, pages 745–762, 2020.
- [9] Jiahua Dong, Yang Cong, Gan Sun, Bineng Zhong, and Xiaowei Xu. What can be transferred: Unsupervised domain adaptation for endoscopic lesions segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4022–4031, June 2020.
- [10] Michael Garland and Paul S Heckbert. Surface simplification using quadric error metrics. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 209–216, 1997.
- [11] Dvir Ginzburg and Dan Raviv. Cyclic functional mapping: Self-supervised correspondence between non-isometric deformable shapes. In *European Conference on Computer Vision*, pages 36–52. Springer, 2020.
- [12] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012.
- [13] Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. 3d-coded: 3d correspondences by deep deformation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 230–246, 2018.
- [14] Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. A papier-mâché approach to learning 3d surface generation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 216–224, 2018.
- [15] Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. Unsupervised cycle-consistent deformation for shape matching. In *Computer Graphics Forum*, volume 38, pages 123–133. Wiley Online Library, 2019.
- [16] Oshri Halimi, Ido Imanuel, Or Litany, Giovanni Trappolini, Emanuele Rodolà, Leonidas Guibas, and Ron Kimmel. Towards precise completion of deformable shapes. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 359–377. Cham, 2020. Springer International Publishing.
- [17] Oshri Halimi, Or Litany, Emanuele Rodola, Alex M Bronstein, and Ron Kimmel. Unsupervised learning of dense shape correspondence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4370–4379, 2019.
- [18] Dominic Jack, Jhony K Pontes, Sridha Sridharan, Clinton Fookes, Sareh Shirazi, Frederic Maire, and Anders Eriksson. Learning free-form deformations for 3d object reconstruction. In *Asian Conference on Computer Vision*, pages 317–333. Springer, 2018.
- [19] Tao Ju, Scott Schaefer, and Joe Warren. Mean value coordinates for closed triangular meshes. In *ACM Siggraph 2005 Papers*, pages 561–566. 2005.
- [20] Chun-Liang Li, Wei-Cheng Chang, Yu Cheng, Yiming Yang, and Barnabás Póczos. Mmd gan: Towards deeper understanding of moment matching network. *arXiv preprint arXiv:1705.08584*, 2017.
- [21] Chun-Liang Li, Tomas Simon, Jason Saragih, Barnabás Póczos, and Yaser Sheikh. Lbs autoencoder: Self-supervised fitting of articulated meshes to point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11967–11976, 2019.
- [22] Hao Li, Bart Adams, Leonidas J Guibas, and Mark Pauly. Robust single-view geometry and motion reconstruction. *ACM Transactions on Graphics (ToG)*, 28(5):1–10, 2009.
- [23] Hao Li, Robert W Sumner, and Mark Pauly. Global correspondence optimization for non-rigid registration of depth scans. In *Computer graphics forum*, volume 27, pages 1421–1430. Wiley Online Library, 2008.
- [24] Or Litany, Tal Remez, Emanuele Rodola, Alex Bronstein, and Michael Bronstein. Deep functional maps: Structured prediction for dense shape correspondence. In *Proceedings of the IEEE international conference on computer vision*, pages 5659–5667, 2017.

- [25] Xingyu Liu, Charles R Qi, and Leonidas J Guibas. FlowNet3d: Learning scene flow in 3d point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 529–537, 2019.
- [26] Mingsheng Long, Jianmin Wang, Guiguang Ding, Jiaguang Sun, and Philip S Yu. Transfer feature learning with joint distribution adaptation. In *Proceedings of the IEEE international conference on computer vision*, pages 2200–2207, 2013.
- [27] 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.
- [28] 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*, pages 652–660, 2017.
- [29] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in neural information processing systems*, pages 5099–5108, 2017.
- [30] 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.
- [31] Thomas W Sederberg and Scott R Parry. Free-form deformation of solid geometric models. In *Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, pages 151–160, 1986.
- [32] Olga Sorkine and Marc Alexa. As-rigid-as-possible surface modeling. In *Symposium on Geometry processing*, volume 4, pages 109–116, 2007.
- [33] Robert W Sumner, Johannes Schmid, and Mark Pauly. Embedded deformation for shape manipulation. In *ACM SIGGRAPH 2007 papers*, pages 80–es. 2007.
- [34] Jian Sun, Maks Ovsjanikov, and Leonidas Guibas. A concise and provably informative multi-scale signature based on heat diffusion. In *Computer graphics forum*, volume 28, pages 1383–1392. Wiley Online Library, 2009.
- [35] 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.
- [36] Gul Varol, Javier Romero, Xavier Martin, Naureen Mahmood, Michael J Black, Ivan Laptev, and Cordelia Schmid. Learning from synthetic humans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 109–117, 2017.
- [37] Hanyu Wang, Jianwei Guo, Dong-Ming Yan, Weize Quan, and Xiaopeng Zhang. Learning 3d keypoint descriptors for non-rigid shape matching. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 3–19, 2018.
- [38] Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. Pixel2mesh: Generating 3d mesh models from single rgb images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 52–67, 2018.
- [39] Weiyue Wang, Duygu Ceylan, Radomir Mech, and Ulrich Neumann. 3dn: 3d deformation network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1038–1046, 2019.
- [40] Yue Wang and Justin M Solomon. Pnnet: Self-supervised learning for partial-to-partial registration. In *Advances in neural information processing systems*, pages 8814–8826, 2019.
- [41] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *Acm Transactions On Graphics (tog)*, 38(5):1–12, 2019.
- [42] Chao Wen, Yinda Zhang, Zhuwen Li, and Yanwei Fu. Pixel2mesh++: Multi-view 3d mesh generation via deformation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1042–1051, 2019.
- [43] Yaoqing Yang, Chen Feng, Yiru Shen, and Dong Tian. Foldingnet: Point cloud auto-encoder via deep grid deformation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 206–215, 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*, pages 75–83, 2020.
- [45] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5745–5753, 2019.