Learning Canonical Embeddings for Unsupervised Shape Correspondence With Locally Linear Transformations

Pan He ⓘ, *Member, IEEE*, Patrick Emami ⓘ, *Member, IEEE*, Sanjay Ranka ⓘ, *Fellow, IEEE*, and Anand Rangarajan ⓘ, *Member, IEEE*

Abstract—We present a new approach to unsupervised shape correspondence learning between pairs of point clouds. We make the first attempt to adapt the classical locally linear embedding algorithm (LLE)—originally designed for nonlinear dimensionality reduction—for shape correspondence. The key idea is to find dense correspondences between shapes by first obtaining high-dimensional neighborhood-preserving embeddings of low-dimensional point clouds and subsequently aligning the source and target embeddings using locally linear transformations. We demonstrate that learning the embedding using a new LLE-inspired point cloud reconstruction objective results in accurate shape correspondences. More specifically, the approach comprises an end-to-end learnable framework of extracting high-dimensional neighborhood-preserving embeddings, estimating locally linear transformations in the embedding space, and reconstructing shapes via divergence measure-based alignment of probability density functions built over reconstructed and target shapes. Our approach enforces embeddings of shapes in correspondence to lie in the same universal/canonical embedding space, which eventually helps regularize the learning process and leads to a simple nearest neighbors approach between shape embeddings for finding reliable correspondences. Comprehensive experiments show that the new method makes noticeable improvements over state-of-the-art approaches on standard shape correspondence benchmark datasets covering both human and nonhuman shapes.

Index Terms—Implicit correspondence learning, locally linear transformations, point cloud reconstruction, probability density functions, unsupervised shape correspondence.

I. INTRODUCTION

THE shape correspondence learning problem is fundamental to geometry processing and computer vision and has been used as a key component in many downstream applications such as deformation modeling [1], texture mapping [2], and medical imaging [3], to name a few.

Dense correspondences between a pair of shapes can be established by measuring the similarities of extracted feature descriptors. Traditional approaches have identified a set of geometric feature descriptors, including extrinsic and intrinsic descriptors [4], [5], [6], [7], [8], [9]. However, these handcrafted descriptors often lead to inaccurate and time-consuming solutions. More recently, we have seen the emergence of *data-driven* approaches built upon modern machine learning techniques that learn the optimal features directly from massive shape pair datasets [10], [11], [12], [13], [14], [15]. However, the major dissatisfaction here is a need for supervised learning, which relies on a sufficient number of labeled training pairs of high-quality ground truth correspondences, which are known to be scarce and difficult to obtain. By contrast, the unsupervised approaches [16], [17], [18], [19], [20] seek to remove the dependency on ground truth correspondence by employing autoencoder-inspired architectures, where they construct the deformation between a pair of shapes and leverage point reconstruction to learn suitable features for measuring the similarities between shapes. However, most of them suffer from the nontrivial optimization of the deformation and reconstruction, thus often requiring additional regularization or constraints, e.g., cycle consistency [21] and local smoothness [19], and usually achieving limited generalization performance.

In this article, we present a new unsupervised learning framework for shape correspondence between pairs of point clouds. Inspired by the classical locally linear embedding (LLE) algorithm [22] initially used for nonlinear dimensionality reduction, we make the first attempt to adapt this concept for shape correspondence. LLE succeeds in exploiting the local euclidean geometry of manifolds to approximately preserve the local euclidean geometry within neighborhoods. In a similar manner, because a point cloud shape is usually a sampled version of a smooth manifold, it is desirable to learn shape embeddings capable of capturing the underlying structure of the manifold. Our method achieves this by learning high-dimensional neighborhood-preserving embeddings of low-dimensional shapes such that nearby and corresponding points fall close to each other in both the high-dimensional embedding space *and* in the low-dimensional input space.
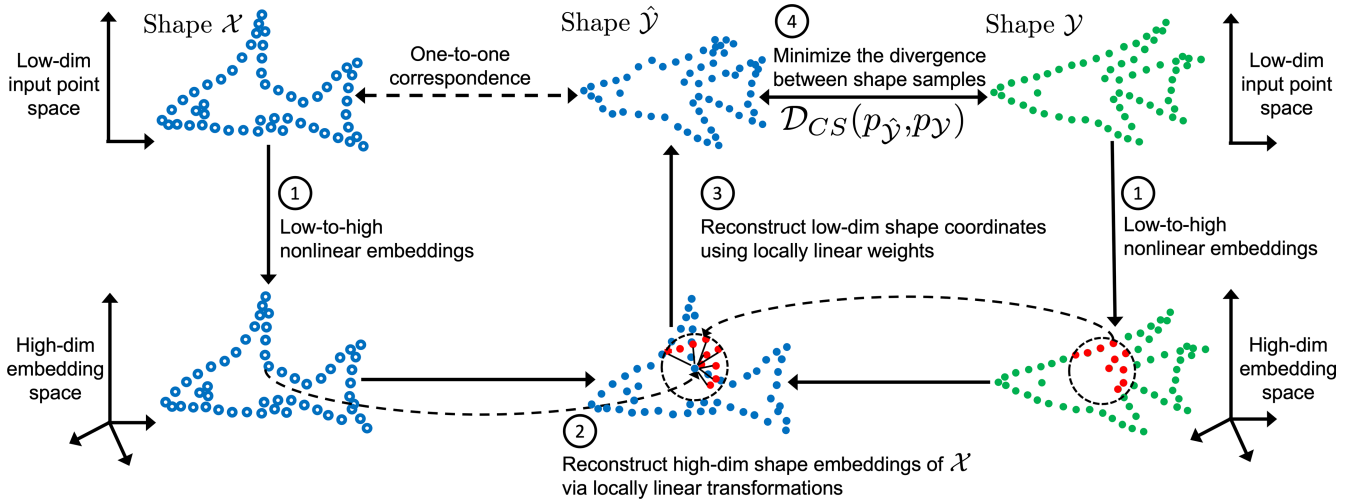
Fig. 1. Given the source and target shapes ($\mathcal{X}$ and $\mathcal{Y}$) as input: 1) We first project the low-dimensional point cloud coordinates into the high-dimensional embedding space; 2) we align feature embeddings of $\mathcal{X}$ and $\mathcal{Y}$ by computing the optimal locally linear transformation that can best cross-reconstruct each feature embedding of $\mathcal{X}$ using its top-K nearest neighbors (red color) in the embedding space of $\mathcal{Y}$; 3) we associate the embeddings of these neighbors to their original point cloud coordinates and further leverage the reconstruction weights of the optimal transformation to reconstruct a shape $\hat{\mathcal{Y}}$ sharing the same point indices to $\mathcal{X}$ (both in blue color); and 4) the embedding network can be optimized by minimizing the divergence between $\hat{\mathcal{Y}}$ and $\mathcal{Y}$.

Another point of departure is that in recent approaches like functional maps, non-rigid deformations between shapes are expected to become *linear* transformations once shapes are projected into a higher-dimensional embedding space. Essentially, point-to-point correspondences between shapes are generalized as a *linear map* between the corresponding function spaces [23]. It is worth emphasizing that our approach is fundamentally different from the functional map-based approaches [14], [23]. These approaches interpret the basis as the embedding for each shape and represent the mapping between a pair of shapes as a change of basis matrix, by applying a *global linear transformation* to every shape point. By contrast, our approach treats maps between shapes as *locally linear transformations* between embeddings, where each shape point has its own linear transformation computed from local neighboring regions. The locally linear transformations succeed in identifying the underlying structure of the shape manifold by enforcing embeddings of shapes in correspondence to lie in the same universal/canonical embedding space, which eventually helps regularize the learning process and leads to a simple nearest neighbors approach for finding reliable correspondences.

We achieve our goals through the following steps, all driven by the idea of marrying LLE and the construction of high-dimensional nonlinear embeddings of point cloud shapes (Fig. 1). Assume we have a nonlinear embedding structure taking a low-dimensional point cloud and returning a high-dimensional embedding, whose weights we seek to learn. Given two shapes whose correspondence we seek, in the first step, we attempt to cross-reconstruct each source point (in the embedding space) from its nearest neighbors in the target point cloud embedding. This step mirrors the first stage of LLE. Next, we take the obtained reconstruction weights and cross-reconstruct a point cloud shape (in the original low-dimensional space) from the same set of nearest neighbors in the target point cloud (again

in the original low-dimensional space). We thereby obtain a reconstructed point cloud in one-to-one correspondence with the source point cloud but based on the nearest neighbors in the target point cloud. We then minimize a suitable divergence measure between the cross-reconstructed and target point clouds with respect to the unknown weights of the nonlinear embedding. Minimization of the cross-reconstruction error minimizes the distance between the original and reconstructed point clouds, and minimization of the divergence measure brings the cross-reconstruction and target point clouds into register. In this way, we build an unsupervised shape correspondence engine capable of end-to-end learning of nonlinear universal embeddings of shape point clouds.

A. Contributions

In summary, our contributions are:

- A new perspective on finding dense correspondences between shapes as *locally linear* transformations in a high-dimensional embedding space, as a superior way to regularize the embeddings of shapes in correspondence by forcing them to lie in the same canonical embedding space

- An unsupervised shape correspondence learning framework for extracting nonlinear shape embeddings that preserve distances within local neighborhoods, estimating locally linear transformations in the embedding space, and reconstructing shapes via the alignment of probability density functions (PDFs) built over reconstructed and target shapes

- A divergence measure for bringing the cross-reconstructed and target shape PDFs into register, which shows improved performance over popular Chamfer distance (CD) and Earth mover's distance (EMD) measures

- A significant improvement compared to existing state-of-the-art methods on standard benchmarks covering both human and nonhuman shapes.

Comprehensive experiments show that the new method makes substantial improvements while showing strong model generalization across datasets with efficient training and inference. More importantly, the proposed idea could be useful for matching problems in other modalities such as images and meshes and cross-modality matching problems such as images to point clouds, and is potentially a promising approach for other tasks requiring the application of manifold learning concepts.

II. RELATED WORK

A. Shape Correspondence and Matching

Axiomatic Methods: Early efforts at representing correspondence (before the deep learning era) used the inexact weighted graph matching formulation with a permutation matrix and outliers for correspondence representation [24], [25], [26], which is further followed by simultaneous pose and correspondence estimation. Simpler (but not necessarily better) methods such as Iterative closest point (ICP) [27] and Chamfer matching (CM) [28] started seeing deployment, followed by the emergence of the Earth mover's distance (EMD) [29] and transportation-based distance measures. Simultaneously, soft correspondences via softassign [30], [31], [32] for both linear assignment and quadratic assignment alternatively estimated the transformations and updated the explicit point-to-point correspondence. Coherent Point Drift (CPD) [33] is similar to Robust Point Matching (RPM) [32] and used Gaussian radial basis functions (GRBF) instead of thin-plate splines (TPS) for non-rigid deformations. RPM L_2E [34], [35] leveraged the L_2E estimator for estimating transformations bootstrapped from shape contexts [36]. Later, point cloud density estimation approaches [37], [38], [39], [40] appeared coupled with distances between density functions optimized w.r.t. the unknown spatial transformation without establishing explicit point correspondence. Representative approaches include KC [38], GMMReg [39], [40], and CS [41], to name a few.

The functional map was introduced in the pioneering work of Ovsjanikov et al. [23] for solving non-rigid shape matching by avoiding the direct estimation of point-to-point correspondence and instead deploying linear transformations between the functional spaces of shapes, followed by subsequent extensions such as [17], [42], [43], [44], [45], [46]. Meanwhile, we have also witnessed progress in the *unsupervised* functional map approaches [17], [21], [47] that consider structural penalties on the inferred maps, e.g., bijectivity or orthogonality. Recently, Diff-FMaps [14] interprets the eigendecomposition based on the Laplace-Beltrami Operator (LBO) as higher-dimensional embeddings of shapes. More importantly, it demonstrated that learning a canonical embedding is a nontrivial problem, and splitting the correspondence learning into two parts (*invariant embedding + linear transformation*) is beneficial in regularizing embedding learning in challenging settings.

Machine Learning Methods: 3D-CODED [16] and Elementary [48] matched the deformable shapes by jointly encoding shapes and correspondences via deforming templates. Corr-Net3D [18] exploited DGCNN [49] to project shapes into a high-dimensional feature space. They enforced unsupervised feature learning by constructing a symmetric deformer for point cloud reconstruction. More recently, LoopReg [50] presented a novel end-to-end framework for fitting the parametric models to 3D scans of articulated humans via a self-supervised differentiable loop and demonstrated improved registration accuracy. Following [51], the authors proposed a novel optimization-based paradigm for 3D human model fitting where vertices are iteratively displaced towards the predicted body surface using standard gradient descent. Deep Shells [52] improved the popular smooth shells approach [53] by introducing an end-to-end trainable deep network and learning optimal local features based on entropy regularized optimal transport. Trappolini et al. [54] proposed a transformer-based framework to estimate the transformation between point cloud shapes efficiently. NeuroMorph [55] simultaneously addressed the problems of shape correspondence and interpolation. In [56], the authors proposed a novel framework for learning a reduced set of deformation parameters using mesh-free approximations and incorporating desirable first-order regularizations.

Particularly, DPC [19] demonstrated a self- and cross-reconstruction framework to learn the latent affinity via a simplified point reconstruction, which is completely different from existing encoder-decoder frameworks [16], [18] regressing ordered point clouds to determine matching points. DPC normalized the similarity of each feature embedding's K-nearest neighbors via softmax and cross-reconstructed a shape using the corresponding input points and similarity scores. An additional mapping loss and self-reconstruction have to be included to impose smooth constraints, otherwise, the performance of DPC drops significantly (and please see their ablation study on design choices). In this paper, we focus on learning LLEs capable of capturing the underlying structure of the shape manifold, with the main goal to find a proper design of the embedding via leveraging local neighborhood relations. Our method simultaneously learns the embedding and optimal locally linear transformation using the LLE-inspired first-stage method and point reconstruction. Consequently, the cross-reconstruction of the source shape using target points gets an implicit regularization via the close relationship to the reconstruction of the source's high-dimensional embedding counterparts. We conjecture that this is exactly what is missing in DPC and hence explains our superior performance using just cross-reconstruction.

B. Shape Descriptor and Feature Learning

The shape analysis community has actively investigated extracting descriptors and feature maps from shapes to capture geometric properties around the neighborhood of points of interest. A detailed discussion of classic *hand-crafted* descriptors can be found in [57], [58].

Hand-crafted Descriptors: Early attempts focused on invariance under a global spatial transformation, e.g., a rigid motion, as shown in shape contexts [36], spin image [4], and multiscale local feature [59]. The community has then extended

these methods to the nonrigid case by considering geodesic distances and conformal factors [60], [61]. Later, *diffusion geometry* [62] established invariant metrics based on eigenvalues and eigenvectors of the Laplace-Beltrami operator obtained from shapes, showing significantly more robustness compared to the geodesic counterparts [63], [64]. Follow-ups include global point signature (GPS) [7], heat kernel signature (HKS) [8], wave kernel signature (WKS) [9] and the complex wave representation (CWR) [65].

Data-driven Descriptors: The *data-driven* feature descriptors have shown their advantages over the *hand-crafted* features in robustness and efficiency. The bag-of-features (BoF) descriptor extracted the frequency histograms of geometric words from shapes [66]. In [67], a robust and invariant point signature is learned from the contextual 3D neighborhood information of salient points for shape matching. Recently, the community has started to extract deep features from shapes in a *data-driven* fashion. For example, GCNN captured invariant shape features from triangular meshes [68]. PointNet [69] showed that the learned features could be used to compute shape correspondences. FMNet [44] leveraged a Siamese residual network [70] for descriptor learning. SplineCNN [71] introduced a novel convolution operator based on B-splines to filter the geometric input efficiently. DeepGFM [72] used KPConv [73], a classic point cloud convolutional filter, to extract robust shape features.

Our approach belongs to the class of *data-driven* approaches. To the best of our knowledge, our approach is the first attempt to adapt the classic LLE algorithm for unsupervised shape correspondence learning into an end-to-end learnable framework. Correspondence is determined via nearest neighbor searches once we learn such discriminative feature representations.

III. LOCALLY LINEAR EMBEDDING

Before describing the proposed approach, we provide a background on the classic LLE framework [22]. Given input point features, LLE has three steps: first, it identifies K nearest neighbors for each point; second, it uses least-squares to compute the weights for reconstructing each point from its nearest neighbors; and third, it reuses the same set of weights and computes the embedding of each point in a low-dimensional space. Specifically, let $P = \{\boldsymbol{p}_i \in \mathbb{R}^D | i = 1, \ldots, N\}$ denote the input point set. LLE builds a K-nearest neighbors (KNN) graph over P by measuring the pairwise euclidean distance and removing self-loops from the graph. Denoting $\boldsymbol{p}_{ik} \in \mathbb{R}^D$ as the k-th nearest neighbor of point $\boldsymbol{p}_i$, we obtain the LLE reconstruction weights by solving

$$\underset{\boldsymbol{W}}{\text{minimize}} \quad E(\boldsymbol{W}) = \sum_{i=1}^{N} \left\| \boldsymbol{p}_i - \sum_{l=1}^{K} w_{ik} \boldsymbol{p}_{ik} \right\|_2^2$$

$$\text{subject to} \quad \sum_{k=1}^{K} w_{ik} = 1, \forall i \in \{1, \ldots, N\}, \quad (1)$$

where $\mathbb{R}^{N \times K} \ni \boldsymbol{W} := [\boldsymbol{w}_1, \ldots, \boldsymbol{w}_N]^T$ denotes the reconstruction weights, and $\mathbb{R}^K \ni \boldsymbol{w}_i := [w_{i1}, \ldots, w_{iK}]^T$ denotes the weights associated to the KNN neighbors $\{\boldsymbol{p}_{ik}\}_{k=1}^K$ for

reconstructing point $\boldsymbol{p}_i$. The sum-to-one weight constraint $\sum_{k=1}^{K} w_{ik} = 1$ leads to the specific properties—each point $\boldsymbol{p}_i$ is invariant to rotations, translations, and rescalings of itself and its nearest neighbors [22]. The optimal $\boldsymbol{W}$ can be found by solving a constrained least squares problem as detailed in Appendix A, available online. Given the optimal $\boldsymbol{W}$, LLE further finds the lower-dimensional embeddings of input points denoted as $Q = \{\boldsymbol{q}_i \in \mathbb{R}^{D_2} | i = 1, \ldots, N\}, D_2 \ll D$, by solving a sparse eigenvalue problem [22].

IV. LTENET

Having briefly summarized LLE, we introduce our novel approach to unsupervised shape correspondence learning called *LTENet*—*L*ocally *L*inear *T*ransformation based *E*mbedding *Net*works. To the best of our knowledge, this is the first attempt at introducing an LLE-inspired algorithm that represents maps between pairs of shapes as locally linear transformations while simultaneously deploying an LLE shape reconstruction objective to optimize nonlinear embeddings towards the same universal/canonical space. In this section, we first define the shape correspondence problem. We then introduce the dovetailing of locally linear transformations with novel LLE point cloud reconstructions for learning both the optimal transformation and embedding. Finally, we describe the divergence measure between the cross-reconstructed and target point clouds. The pipeline of LTENet is summarized in Fig. 2.

A. Problem Definition and Objectives

Let point clouds $\mathbb{R}^{N \times 3} \ni \mathcal{X} := [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N]^T$ and $\mathbb{R}^{N \times 3} \ni \mathcal{Y} := [\boldsymbol{y}_1, \ldots, \boldsymbol{y}_N]^T$ denote the source and target shapes, respectively, where $\boldsymbol{x}_i, \boldsymbol{y}_j \in \mathbb{R}^3$ and N is the number of points. Our goal is to find a point-to-point correspondence or map defined as $T_{\mathcal{X}\mathcal{Y}} : \mathcal{X} \to \mathcal{Y}$ such that every point $\boldsymbol{x}_i$ in $\mathcal{X}$ has its corresponding point $\boldsymbol{y}_{j^*} := T_{\mathcal{X}\mathcal{Y}}(\boldsymbol{x}_i)$ in $\mathcal{Y}$, where $1 \leq i, j^* \leq N$. Inspired by LLE [22], the proposed LTENet leads to suitable embeddings for shape correspondence that preserve the local configurations of nearest neighbors. From a high-level perspective, it shares a similar spirit with the LBO operator which relies on preserving distances between nearby points. While the LBO operator is usually constructed over point cloud coordinates, LTENet operates on nonlinear feature embeddings obtained from deep neural networks, which are more robust and efficient to compute. It is worth mentioning that the proposed approach directly takes raw point cloud coordinates as the input without requiring any point connectivity information.

B. Optimal Locally Linear Transformations

Given $\mathcal{X}$ and $\mathcal{Y}$, we extract their nonlinear feature embeddings $\mathcal{F}^{\mathcal{X}}, \mathcal{F}^{\mathcal{Y}} \in \mathbb{R}^{N \times D}$, respectively, via a neural network $\mathcal{F}$. This allows us to transition from the point cloud coordinates ($\mathbb{R}^{N \times 3}$) to a higher dimensional embedding space ($\mathbb{R}^{N \times D}$) where finding shape correspondence is more likely to be successful as demonstrated by the functional map paradigm [23].

To align $\mathcal{F}^{\mathcal{X}}$ and $\mathcal{F}^{\mathcal{Y}}$, we must find a transformation between them while also jointly optimizing $\mathcal{F}$ to obtain suitable
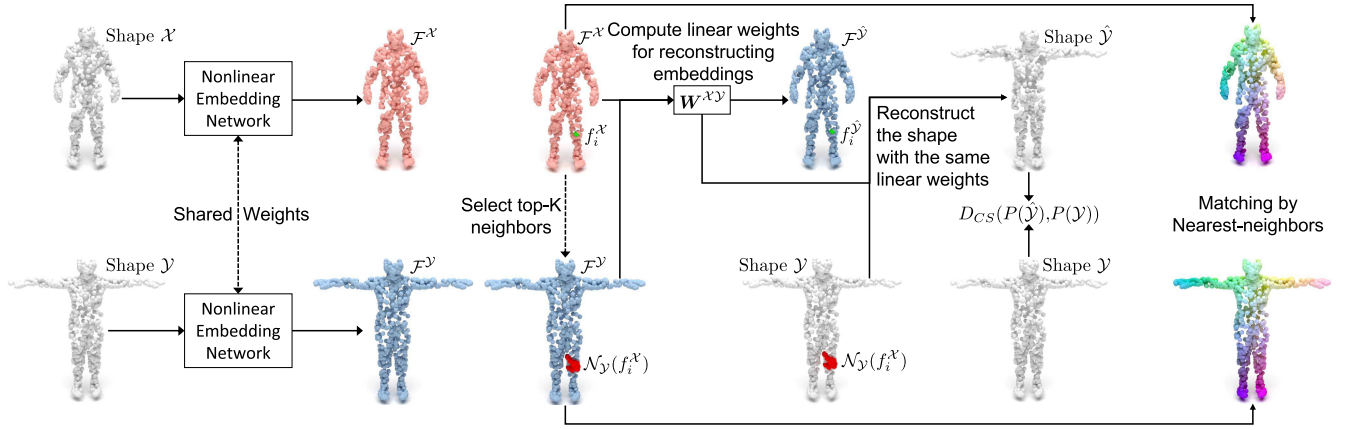
Fig. 2.   Pipeline overview of LTENet: 1) Extract nonlinear shape embeddings $\mathcal{F}^{\mathcal{X}}$ and $\mathcal{F}^{\mathcal{Y}}$, given $\mathcal{X}$ and $\mathcal{Y}$; 2) select top-K neighbors for each feature embedding $f_i^{\mathcal{X}}$ based on the cosine similarity between $\mathcal{F}^{\mathcal{X}}$ and $\mathcal{F}^{\mathcal{Y}}$; 3) estimate the locally linear transformations following (2) and (3) to best reconstruct $\mathcal{F}^{\mathcal{X}}$ using $\mathcal{F}^{\mathcal{Y}}$ and denoted as $\mathcal{F}^{\hat{\mathcal{Y}}}$; 4) reconstruct a shape $\hat{\mathcal{Y}}$ following (6); 5) learn the embedding network $\mathcal{F}$ via minimizing the divergence $\mathcal{D}_{CS}(P(\hat{\mathcal{Y}}), P(\mathcal{Y}))$; and 6) determine the correspondence using nearest neighbors between embeddings.

embeddings for estimating shape correspondence. Recall that (1) in LLE is able to approximate the original input points using the LLE reconstruction weights and nearest neighbors. Denote $\mathcal{F}^{\hat{\mathcal{Y}}} \in \mathbb{R}^{N \times D}$ as the feature embeddings obtained by applying the transformation to $\mathcal{F}^{\mathcal{Y}}$ to achieve the alignment, i.e., $\mathcal{F}^{\mathcal{X}} \approx \mathcal{F}^{\hat{\mathcal{Y}}}$. Given all $f_i^{\mathcal{X}} \in \mathcal{F}^{\mathcal{X}}$ and $f_j^{\mathcal{Y}} \in \mathcal{F}^{\mathcal{Y}}$, we implement the transformation by first considering

$$\underset{\boldsymbol{W}^{\mathcal{X}\mathcal{Y}}}{\text{minimize}} \quad E(\boldsymbol{W}^{\mathcal{X}\mathcal{Y}}) = \sum_{i=1}^{N} \left\| f_i^{\mathcal{X}} - \sum_{l \in \mathcal{N}_{\mathcal{Y}}(f_i^{\mathcal{X}})} w_{i,l}^{\mathcal{X}\mathcal{Y}} f_l^{\mathcal{Y}} \right\|_2^2$$

$$\text{subject to} \quad \sum_{l \in \mathcal{N}_{\mathcal{Y}}(f_i^{\mathcal{X}})} w_{i,l}^{\mathcal{X}\mathcal{Y}} = 1, \forall i \in \{1, \dots, N\}, \quad (2)$$

where $\mathcal{N}_{\mathcal{Y}}(f_i^{\mathcal{X}})$ are all K indices of $f_i^{\mathcal{X}}$'s nearest neighbors in $\mathcal{F}^{\mathcal{Y}}$ based on the cosine similarity. $\mathbb{R}^{N \times K} \ni \boldsymbol{W}^{\mathcal{X}\mathcal{Y}} := [\boldsymbol{w}_1^{\mathcal{X}\mathcal{Y}}, \dots, \boldsymbol{w}_N^{\mathcal{X}\mathcal{Y}}]^T$ denotes the reconstruction weights, and $\boldsymbol{w}_i^{\mathcal{X}\mathcal{Y}} \in \mathbb{R}^K$ stacks all the reconstruction weights $w_{i,*}^{\mathcal{X}\mathcal{Y}}$ associated to $f_i^{\mathcal{X}}$ and $\mathcal{N}_{\mathcal{Y}}(f_i^{\mathcal{X}})$. We then express each transformed $f_i^{\hat{\mathcal{Y}}} \in \mathcal{F}^{\hat{\mathcal{Y}}}$ as

$$f_i^{\hat{\mathcal{Y}}} = \sum_{l \in \mathcal{N}_{\mathcal{Y}}(f_i^{\mathcal{X}})} w_{i,l}^{\mathcal{X}\mathcal{Y}} f_l^{\mathcal{Y}} \approx f_i^{\mathcal{X}}. \quad (3)$$

As observed from (2) and (3), we make two unique modifications compared to (1) of the original LLE paper: 1) We directly operate on high-dimensional nonlinear feature embeddings rather than raw input data, and 2) unlike LLE which reconstructs the input data by picking nearest neighbors from itself, we conduct a cross-reconstruction such that the feature embeddings $\mathcal{F}^{\mathcal{X}}$ of the source shape will select nearest neighbors from $\mathcal{F}^{\mathcal{Y}}$ in the target shape, which enforces embeddings of shapes in correspondence to lie in the same universal/canonical embedding space.

Intuitively, (2) represents the spatial transformation between shapes, e.g., a non-rigid transformation, in the input point space as an equivalent locally linear transformation between $\mathcal{F}_{\mathcal{X}}$ and $\mathcal{F}_{\mathcal{Y}}$. An optimal $\boldsymbol{W}^{\mathcal{X}\mathcal{Y}}$ implies that $\mathcal{F}^{\mathcal{X}}$ has been properly

reconstructed by $\mathcal{F}^{\hat{\mathcal{Y}}}$ using $\mathcal{F}^{\mathcal{Y}}$. Following LLE [22], [74], the optimal $\boldsymbol{W}^{\mathcal{X}\mathcal{Y}}$ can be found by solving a constrained least squares problem:

$$\boldsymbol{w}_i^{\mathcal{X}\mathcal{Y}} = \frac{(\boldsymbol{G}_i^{\mathcal{X}\mathcal{Y}} + \gamma \boldsymbol{I})^{-1} \mathbf{1}}{\mathbf{1}^T (\boldsymbol{G}_i^{\mathcal{X}\mathcal{Y}} + \gamma \boldsymbol{I})^{-1} \mathbf{1}}, \quad (4)$$

where $\boldsymbol{I}$ is the identity matrix and $\mathbf{1} \in \mathbb{R}^{K \times 1}$ is the matrix filling all elements with one. $\boldsymbol{G}_i^{\mathcal{X}\mathcal{Y}}$ denotes the Gram matrix defined as

$$\mathbb{R}^{K \times K} \ni \boldsymbol{G}_i^{\mathcal{X}\mathcal{Y}} := (f_i^{\mathcal{X}} \mathbf{1}^T - \boldsymbol{\eta}_i^{\mathcal{Y}})^T (f_i^{\mathcal{X}} \mathbf{1}^T - \boldsymbol{\eta}_i^{\mathcal{Y}}), \quad (5)$$

where $\boldsymbol{\eta}_i^{\mathcal{Y}} \in \mathbb{R}^{D \times K}$ stack all feature embeddings of the K neighbors of $f_i^{\mathcal{X}}$ found in $\mathcal{F}^{\mathcal{Y}}$. Adding $\gamma \boldsymbol{I}$ to (4) leads to numerically stable solutions by avoiding the possible singularity of $\boldsymbol{G}_i^{\mathcal{X}\mathcal{Y}}$ [74], [75]. This also links our work to Robust LLE with an ℓ_2 norm-based regularization (see Appendix A, available online for details).

C. Learning Canonical Embeddings

The cross-reconstruction weights $\boldsymbol{W}^{\mathcal{X}\mathcal{Y}}$ allow us to represent $\mathcal{F}_{\mathcal{X}}$ in terms of $\mathcal{F}_{\mathcal{Y}}$. However, this does not necessarily imply $\boldsymbol{W}^{\mathcal{X}\mathcal{Y}}$ will lead to a better embedding network $\mathcal{F}$ suitable for shape correspondence. To show that, we observe that the closed-form expression of $\boldsymbol{W}^{\mathcal{X}\mathcal{Y}}$ only depends on the Gram matrix $\boldsymbol{G}^{\mathcal{X}\mathcal{Y}}$, where each $\boldsymbol{G}_i^{\mathcal{X}\mathcal{Y}}$ is constructed based on the feature difference between $f_i^{\mathcal{X}}$ and $\boldsymbol{\eta}_i^{\mathcal{Y}}$. Therefore, the optimal $\boldsymbol{W}^{\mathcal{X}\mathcal{Y}}$ essentially relies on $\mathcal{F}^{\mathcal{X}}$ and $\mathcal{F}^{\mathcal{Y}}$. $\boldsymbol{W}^{\mathcal{X}\mathcal{Y}}$ is optimal in terms of the reconstruction of $\mathcal{F}^{\mathcal{X}}$ using $\mathcal{F}^{\mathcal{Y}}$ but is not optimal for shape correspondence if no additional optimization step is applied.

To properly train the embedding network, we associate the high-dimensional embeddings to the original low-dimensional point cloud coordinates. Specifically, observing $f_i^{\hat{\mathcal{Y}}} = \sum_{l \in \mathcal{N}_{\mathcal{Y}}(f_i^{\mathcal{X}})} w_{i,l}^{\mathcal{X}\mathcal{Y}} f_l^{\mathcal{Y}} \approx f_i^{\mathcal{X}}$, we can interpret each feature embedding $f_i^{\mathcal{X}}$ or $f_i^{\hat{\mathcal{Y}}}$ as a linear combination of a few basis elements $\{f_l^{\hat{\mathcal{Y}}} | l \in \mathcal{N}_{\mathcal{Y}}(f_i^{\mathcal{X}})\}$ with the associated coefficients
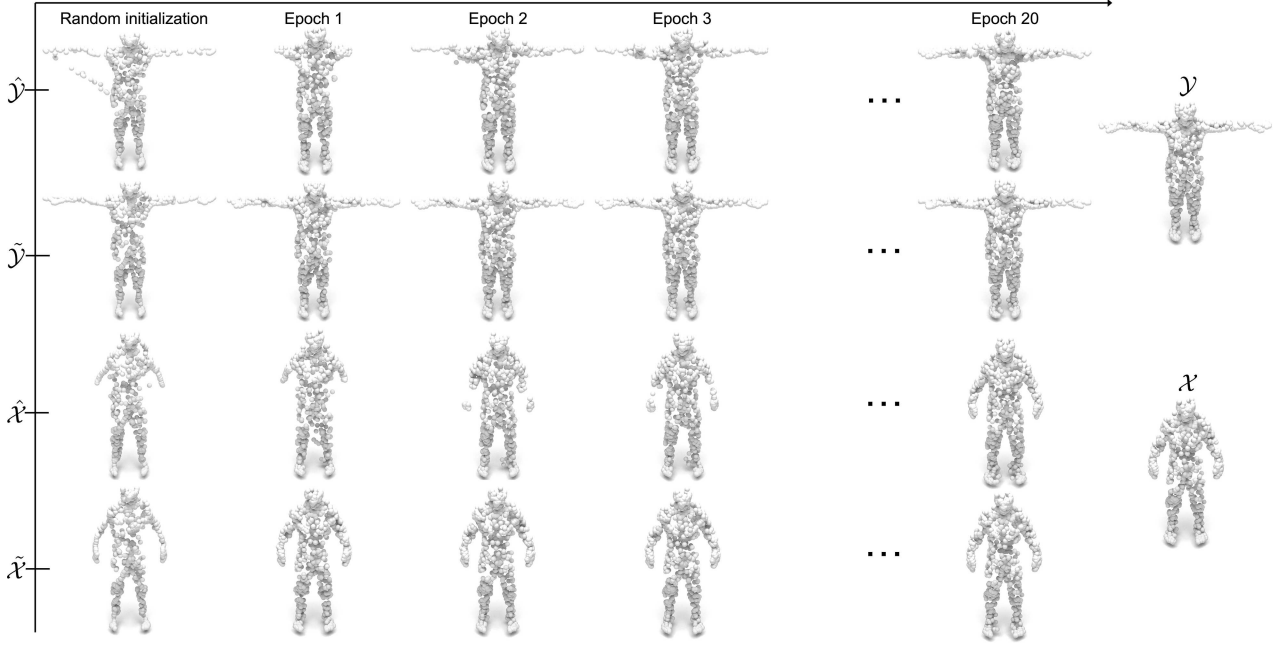
Fig. 3. Visualization of reconstructed point clouds of the proposed LTENet. We obtain the reconstructed shapes in cross-reconstruction ($\hat{\mathcal{X}}$ and $\hat{\mathcal{Y}}$) and self-reconstruction ($\tilde{\mathcal{X}}$ and $\tilde{\mathcal{Y}}$) from models at different training epochs. Starting from random initialization on the embedding network, it is clear that the reconstructed shapes are getting closer to the source and target shapes $\mathcal{X}$ and $\mathcal{Y}$ as the training progresses.

$\{w_{i,l}^{\mathcal{X}\mathcal{Y}}|l \in \mathcal{N}_{\mathcal{Y}}(f_i^{\mathcal{X}})\}$. Using the same coefficients, we could reconstruct the low-dimensional point $\hat{\boldsymbol{y}}_i \in \mathbb{R}^3$ for each $f_i^{\hat{\mathcal{Y}}}$, which gives

$$\hat{\boldsymbol{y}}_i = \sum_{l \in \mathcal{N}_{\mathcal{Y}}(f_i^{\mathcal{X}})} w_{i,l}^{\mathcal{X}\mathcal{Y}} \boldsymbol{y}_l \qquad (6)$$

where $\boldsymbol{y}_l \in \mathcal{Y}$ is the point associated to $f_l^{\mathcal{Y}}$. We then obtain $\mathbb{R}^{N \times 3} \ni \hat{\mathcal{Y}} := [\hat{\boldsymbol{y}}_1, \ldots, \hat{\boldsymbol{y}}_n]^T$, interpreted as the linearly reconstructed shape for $\mathcal{F}^{\hat{\mathcal{Y}}}$ in the basis elements (point coordinates) of $\mathcal{Y}$. The indices of $\hat{\mathcal{Y}}$ are in *exact* one-to-one correspondence with the indices of $\mathcal{X}$. Also, $\hat{\boldsymbol{y}}_i$ can be understood as a *soft* correspondence of $\boldsymbol{x}_i$ because the indices of $\{\boldsymbol{y}_l|l \in \mathcal{N}_{\mathcal{Y}}(f_i^{\mathcal{X}})\}$ are the same indices of the top-K nearest neighbors of $f_i^{\mathcal{X}}$. Because these neighbors are selected from $\mathcal{F}^{\mathcal{Y}}$ with the highest similarity to $f_i^{\mathcal{X}}$, the point $\boldsymbol{y}_l$ associated to each neighbor could be a candidate matching point of $\boldsymbol{x}_i$. If each point $\boldsymbol{x}_i$ finds its approximate matching point $\hat{\boldsymbol{y}}_i$, we would expect $\hat{\mathcal{Y}}$ to be similar to $\mathcal{Y}$ as the training progresses (Fig. 3). To this end, we can train $\mathcal{F}$ by solving

$$\underset{\mathcal{F}}{\text{minimize}} \quad E(\mathcal{F}) = \mathcal{D}(\hat{\mathcal{Y}}, \mathcal{Y}) \qquad (7)$$

where $\mathcal{D}(\cdot, \cdot)$ defines a dissimilarity measure. Because $\hat{\mathcal{Y}}$ and $\mathcal{X}$ share the same point indices while being different from those of $\mathcal{Y}$, we do not have an obvious one-to-one correspondence between $\hat{\mathcal{Y}}$ and $\mathcal{Y}$, e.g., $\hat{\boldsymbol{y}}_1$ probably does not correspond to $\boldsymbol{y}_1$, and so on.

LLE Versus LTENet: LLE finds the embedded vector for each input point by solving an expensive eigenvalue problem (a projection from the high-dimensional input space to the low-dimensional embedding space). In contrast, our approach

approximates *source* points via shape reconstruction using linear combinations of nearest neighbor *target* point coordinates $\mathcal{Y}$ and the (closed-form) weights $\boldsymbol{W}^{\mathcal{X}\mathcal{Y}}$ obtained from first stage LLE. Unlike LLE, which is focused on identifying suitable low-dimensional embeddings for high-dimensional input data, LTENet establishes dense correspondences between shapes by forcing a pair of shapes to lie on the same manifold. This is achieved by our fully differentiable LTENet framework, which pushes their embeddings towards a locally linearly invariant space via maximizing the similarity between a shape and its reconstructed counterpart. LTENet demonstrates a principled approach by adopting the central concept of classic LLE for shape correspondence. Next, we introduce a suitable distance measure $\mathcal{D}(\cdot, \cdot)$ for end-to-end training.

D. Implicit Correspondence Learning Via the Alignment of PDFs

Most unsupervised approaches [19] adopt popular CD and EMD measures to reconstruct point clouds, which are sensitive to outliers or are computationally intensive. Point clouds are quite often nothing but discrete samples of the underlying continuous shapes and surfaces. Therefore, we instead represent point clouds as probability density functions and seek to minimize a divergence between reconstructed and original shapes. Formally, given the shape $\mathcal{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N]^T$, we represent an arbitrary point $\boldsymbol{x}$ by its kernel (Parzen) density estimation (KDE) of the PDF using an arbitrary kernel function $\mathsf{K}(\cdot)$:

$$p_{\mathcal{X}}(x) = \frac{1}{N} \sum_{i=1}^{N} \mathsf{K}\left(\frac{\boldsymbol{x} - \boldsymbol{x}_i}{\sigma}\right), \qquad (8)$$

where σ is the bandwidth parameter. We choose the Gaussian kernel $G_\sigma(\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{(2\pi\sigma^2)^{3/2}} \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{y}\|_2^2}{2\sigma^2}\right)$ in 3D as the kernel function due to its nice properties: it is symmetric, positive definite, and its value approaches zero when the point $\boldsymbol{x}$ moves away from the center $\boldsymbol{y}$ while being controlled by a decay factor determined by σ.

Inspired by [38], [39], [41], [76], we adopt the Cauchy-Schwarz (CS) divergence [77], denoted as $\mathcal{D}_{CS}(q, p)$, to measure the similarity between two density functions, which is defined as

$$
\begin{aligned}
\mathcal{D}_{CS}(q, p) = & -\log\left(\frac{\int q(x)p(x)dx}{\sqrt{\int q^2(x)dx \int p^2(x)dx}}\right) \\
= & -\log\int q(x)p(x)dx + 0.5\log\int q^2(x)dx \\
& + 0.5\log\int p^2(x)dx,
\end{aligned} \tag{9}
$$

which is symmetric for any two PDFs q and p such that $0 \leq \mathcal{D}_{CS} < \infty$ where the minimum is obtained iff $q(x) = p(x)$. We substitute Gaussian kernel PDF estimators for $\mathcal{Y}$ and $\hat{\mathcal{Y}}$ into $q(x), p(x)$, and make straightforward manipulations based on the convolution theorem for Gaussian functions (see the detailed derivation in Appendix B, available online), which gives

$$
\begin{aligned}
\mathcal{D}_{CS}(p_{\hat{y}}, p_y) = & -\log\sum_{j=1}^{N}\sum_{i=1}^{N} G_{\sqrt{2}\sigma}(\hat{\boldsymbol{y}}_j, \boldsymbol{y}_i) \\
& + 0.5\log\sum_{j'=1}^{N}\sum_{j=1}^{N} G_{\sqrt{2}\sigma}(\hat{\boldsymbol{y}}_{j'}, \hat{\boldsymbol{y}}_j) \\
& + 0.5\log\sum_{i'=1}^{N}\sum_{i=1}^{N} G_{\sqrt{2}\sigma}(\boldsymbol{y}_{i'}, \boldsymbol{y}_i).
\end{aligned} \tag{10}
$$

Later, we will show that CS leads to better performance compared to CD and EMD objectives by handling outliers using the Gaussian kernels [78]. Specifically, the Gaussian kernel G is able to mitigate the oversensitivity to outliers by suppressing large distances between reference and reconstructed shape points. In CD and EMD, these large distances due to outliers negatively impact model training, leading to degraded performance. It is worth mentioning that CS is closely related to graph cuts and Mercer kernel theory [78].

Implementation of the CS loss: The CS divergence loss can be implemented in PyTorch with a few lines of code. To handle numerical issues, we leverage the Log-Sum-Exp trick as shown in Algorithm 1.

E. The Training Objective

Similar to (2) and (6) using $\boldsymbol{W}^{\mathcal{X}\mathcal{Y}}$ to reconstruct $\mathcal{F}^{\mathcal{Y}}$ and $\hat{\mathcal{Y}}$, we compute the reconstruction weights $\boldsymbol{W}^{\mathcal{Y}\mathcal{X}}$ to approximate the original input shape $\mathcal{X}$, which results in the reconstructed shape $\mathbb{R}^{N\times 3} \ni \hat{\mathcal{X}} := [\hat{\boldsymbol{x}}_1, \ldots, \hat{\boldsymbol{x}}_N]^T$. In addition, we approximate the original $\mathcal{Y}$ using $\boldsymbol{W}^{\mathcal{Y}\mathcal{Y}}$ and $\mathcal{Y}$, which is a self-construction process to obtain the approximate shape

---

Algorithm 1: The CS Divergence Implemented in PyTorch.

```
# reconstructed target and target point
    coordinates <- - B X N x 3 and B X N x 3
# bandwidth, the kernel bandwith <- - scalar
def gmm(rec_target, target, bandwidth):
    # B X N x N x 3 <- B X N X 1 x C - B X N X 1 X N X C
diff_ij = (rec_target.unsqueeze(2) - target.
    unsqueeze(1))
    # B X N x N
    factor = 2*bandwidth*bandwidth
    # B X N x N
diff_ij = (diff_ij**2).sum(-1).div(factor).
    mul(-0.5) - 0.5*math.log(2*math.pi) -
    math.log(math.sqrt(2)*bandwidth)

dist = torch.logsumexp((diff_ij).reshape
    (diff_ij.shape[0], -1),dim=1).mean()

    return dist

def cs_divergence(rec_target, target, bandwidth):
r_t_dist = -1 * gmm(rec_target, target,
    bandwidth)
r_r_dist = 0.5 * gmm(rec_target, rec_target,
    bandwidth)
t_t_dist = 0.5 * gmm(target, target, bandwidth)

    return r_t_dist + r_r_dist + t_t_dist
```

$\mathbb{R}^{N\times 3} \ni \tilde{\mathcal{Y}} := [\tilde{\boldsymbol{y}}_1, \ldots, \tilde{\boldsymbol{y}}_N]^T$. The approximation of $\mathcal{X}$ is similarly expressed as $\mathbb{R}^{N\times 3} \ni \tilde{\mathcal{X}} := [\tilde{\boldsymbol{x}}_1, \ldots, \tilde{\boldsymbol{x}}_N]^T$. The final training objective is defined as

$$
\begin{aligned}
\underset{\mathcal{F}}{\text{minimize}} \quad & \lambda_{\text{cross}}(\mathcal{D}(\hat{\mathcal{X}}, \mathcal{X}) + \mathcal{D}(\hat{\mathcal{Y}}, \mathcal{Y})) \\
& + \lambda_{\text{self}}(\mathcal{D}(\tilde{\mathcal{X}}, \mathcal{X}) + \mathcal{D}(\tilde{\mathcal{Y}}, \mathcal{Y})) \\
& + \lambda_{\text{reg}}(E_r(\mathcal{X}, \hat{\mathcal{Y}}) + E_r(\mathcal{Y}, \hat{\mathcal{X}}))
\end{aligned} \tag{11}
$$

where $\lambda_{\text{cross}}, \lambda_{\text{self}}$, and λ_{reg} are the hyperparameters to balance different losses and $\mathcal{D}(\cdot, \cdot)$ is the CS objective in (10). $E_r(\cdot, \cdot)$ is the optional smoothness term defined as the mapping loss [19], which encourages points in $\hat{\mathcal{Y}}$ (or $\hat{\mathcal{X}}$) to remain close if their one-to-one corresponding points in $\mathcal{X}$ (or $\mathcal{Y}$) are close to each other. $E_r(\mathcal{X}, \hat{\mathcal{Y}})$ is defined as

$$
E_r(\mathcal{X}, \hat{\mathcal{Y}}) = \frac{1}{NK}\sum_{i=1}^{N}\sum_{l \in \mathcal{N}_{\mathcal{X}}(\boldsymbol{x}_i)} v_{i,l}^{\mathcal{X}\hat{\mathcal{Y}}}\|\hat{\boldsymbol{y}}_i - \hat{\boldsymbol{y}}_l\|_2^2 \tag{12}
$$

where $\mathcal{N}_{\mathcal{X}}(\boldsymbol{x}_i)$ is the euclidean neighborhood of $\boldsymbol{x}_i$ in $\mathcal{X}$ of size K, $v_{i,l}^{\mathcal{X}\hat{\mathcal{Y}}} = \exp\frac{-\|\boldsymbol{x}_i-\boldsymbol{x}_l\|_2^2}{\alpha}$ where α is a hyperparameter configured by following [19]. $E_r(\mathcal{Y}, \hat{\mathcal{X}})$ is similarly defined.

F. Test Phase

In the test phase, we obtain the correspondence for each source point $\boldsymbol{x}_i$ by selecting a point from the target shape whose embedding is the nearest neighbor of $\boldsymbol{x}_i$'s embedding based on the cosine similarity. This gives

$$
T_{\mathcal{X}\mathcal{Y}}(\boldsymbol{x}_i) = \boldsymbol{y}_{j^*}, \quad j^* = \underset{j\in\{1,\ldots,N\}}{\arg\max} \frac{(f_i^{\mathcal{X}}) \cdot (f_j^{\mathcal{Y}})^T}{\|f_i^{\mathcal{X}}\|_2\|f_j^{\mathcal{Y}}\|_2}. \tag{13}
$$

Summary: We have presented the LTENet framework for unsupervised shape correspondence learning, which unifies nonlinear embeddings, LLE transformations in the embedding space, point cloud reconstruction, and implicit correspondence learning with the CS divergence. We consider the following analogy for LTENet: *CS divergences and top LLE transformations are to shape correspondence as Kullback–Leibler (KL) divergences and top linear classifiers are to classification. By doing so, we are able to learn universal feature embeddings where correspondences are directly obtained using nearest neighbors. This is also analogous to the open-set classification problems where we handle samples of unseen classes by comparing feature distances between these samples and nearest neighbor trained examples of seen classes.*

V. EXPERIMENTS

In this section, we compare LTENet against recent state-of-the-art approaches on several well-established datasets for shape correspondence, and we conduct ablation studies.

A. Experimental Setup

Datasets: Following [18], [19], we conduct experiments on standard datasets covering both human and nonhuman shapes. For human shapes, we use the large-scale SURREAL dataset [16] prepared by 3D-CODED [16], which leverages SMPL [79] to generate a total of 230,000 samples. We select arbitrary shapes as training pairs from SURREAL. We then evaluate on the challenging SHREC-19 [80] containing 430 non-rigid shape pairs generated from 44 real human scans. For non-human shapes, we adopt SMAL [81] and TOSCA [82] for training and evaluation, respectively. SMAL provides the 3D articulated parametric model for animals. We create a training set of 10,000 shapes by generating 2,000 samples under each animal category. We pair arbitrary shapes of the same category. Similarly, we consider 41 animal figures out of the total 80 objects in TOSCA to match species in SMAL and generate 286 test shape pairs from the SMAL dataset accordingly.

Evaluation metrics: A common evaluation metric is the geodesic distance error assuming a known point adjacency matrix, which is unavailable in point clouds. Instead, we follow [19] to calculate the correspondence error as

$$err = \frac{1}{N} \sum_{i=1}^{N} \|T_{\mathcal{XY}}(\boldsymbol{x}_i) - T_{\mathcal{XY}}^{gt}(\boldsymbol{x}_i)\|_2. \quad (14)$$

where $T_{\mathcal{XY}}(\boldsymbol{x}_i)$, $T_{\mathcal{XY}}^{gt}(\boldsymbol{x}_i)$ denote the predicted and ground truth correspondence of point $\boldsymbol{x}_i$ w.r.t. $\mathcal{Y}$ and $\|\cdot\|_2$ is the ℓ_2 norm of a vector. Additionally, we use the error tolerance $\epsilon = r/dist_{\max}$ coupled with a tolerant radius r, where $dist_{\max} = \max\{\|\boldsymbol{y}_i - \boldsymbol{y}_j\|_2, \forall i, \forall j\}$ denotes the maximal distance of all pairwise point distances in $\mathcal{Y}$. The correspondence accuracy under ϵ is defined as

$$acc(\epsilon) = \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}(\|T_{\mathcal{XY}}(\boldsymbol{x}_i) - T_{\mathcal{XY}}^{gt}(\boldsymbol{x}_i)\|_2 < \epsilon \, dist_{\max}), \quad (15)$$

Method	SURREAL/ SHREC		SHREC/ SHREC/		SMAL/ TOSCA	
	acc ↑	err ↓	acc ↑	err ↓	acc ↑	err ↓
SURFMNet [17]	4.3%	0.3	5.9%	0.2	*	*
GeoFMNet [72]	8.2%	0.2	*	*	*	*
Diff-FMaps [14]	4.0%	7.1	*	*	*	*
3D-CODED [16]	2.1%	8.1	*	*	0.5%	19.2
Elementary [48]	2.3%	7.6	*	*	0.5%	13.7
CorrNet3D [18]	6.0%	6.9	0.4%	33.8	5.3%	9.8
DPC [19]	17.7%	6.1	**15.3%**	5.6	33.2%	5.8
Released DPC [19]	17.5%	6.3	14.5%	**5.3**	33.5%	5.8
LTENet (Ours)	**20.7%**	**5.8**	14.3%	6.0	**38.1%**	**5.7**

The proposed LTENet achieves state-of-the-art shape correspondence performance, indicated by the correspondence accuracy at 1% tolerance (*Acc*, in percentage) and the average correspondence error (*Err*, in centimeters).

where $\mathbb{1}$ is the indicator function. We set different ϵ values between 0% to 20%.

Implementation details: The proposed LTENet is not limited to a specific model architecture for the embedding network $\mathcal{F}$. We followed DPC [19] to use the same variant of DGCNN [49] as $\mathcal{F}$, where its core component is the popular EdgeConv operator that builds a dynamic graph over points for learning the feature embeddings. We refer the reader to [49] for more details. Our models were implemented in Pytorch [83]. We used the AdamW optimizer [84] with an initial learning rate of 0.0003, momentum 0.9, and weight decay of 0.0005. We used a cosine decay learning rate scheduler for 300 epochs and 10 epochs of linear warm-up. We trained models with a batch size of 8 on a server equipped with AMD EPYC ROME microprocessors and NVIDIA A100 GPUs.

Baseline methods: We consider recent state-of-the-art unsupervised shape correspondence learning approaches (DPC [19] and CorrNet3D [18]) as competitive baselines. We compare against supervised approaches, including Diff-FMaps [14], 3D-CODED [16], and Elementary Structures [48], and mesh-based approaches, including the unsupervised SURFMNet [17] and the supervised GeoFMNet [72].

B. Results on Human Datasets

We explored two training and evaluation settings. For a fair comparison, we followed DPC [19] to train our models by selecting the first 2000 shapes out of the total 230,000 samples in SURREAL and evaluated on the official 430 SHREC pairs (SURREAL/SHREC). We also trained models on random pairs generated from SHREC and evaluated on the same test pairs (SHREC/SHREC).

Quantitative evaluation: Table I summarizes the *acc* at 1% error tolerance, indicating a near-perfect correspondence matching and average correspondence error *err*. On SHREC/SHREC, we achieved a competitive performance against DPC. On SURREAL/SHREC, we outperformed all baseline models. Specifically, SURFMNet [17] and GeoFMNet [72] achieved impressive performance. However, they require the expensive computation of the LBO basis and complex test-time post-processing [27],
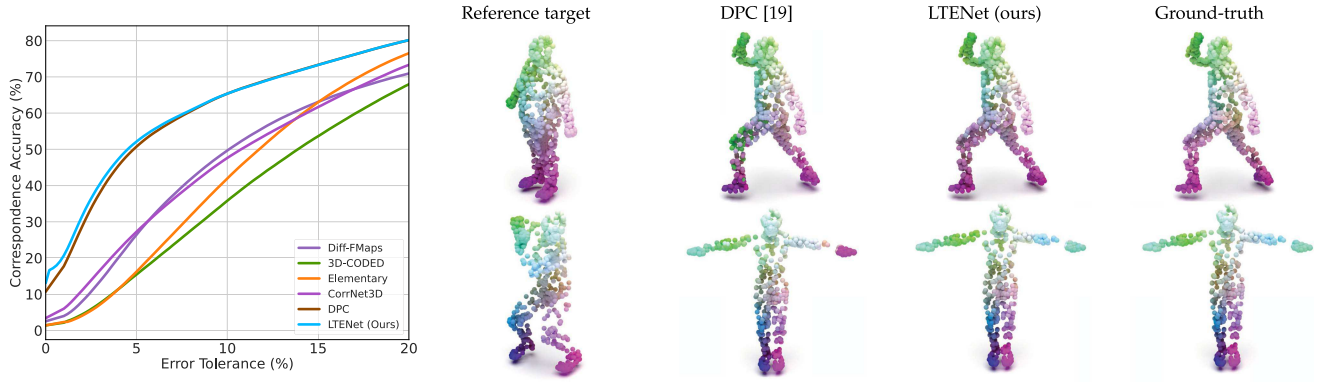
Fig. 4. (Left) The correspondence accuracies under different error tolerance values in the SURREAL/SHREC setting. Our method achieves better performance compared to the state-of-the-art DPC. (Right) Visual examples of SHREC test pairs. DPC contains outlier matches, e.g., wrongly matching hands to thighs or feet to hands. LTENet generates more accurate and smoother predictions.
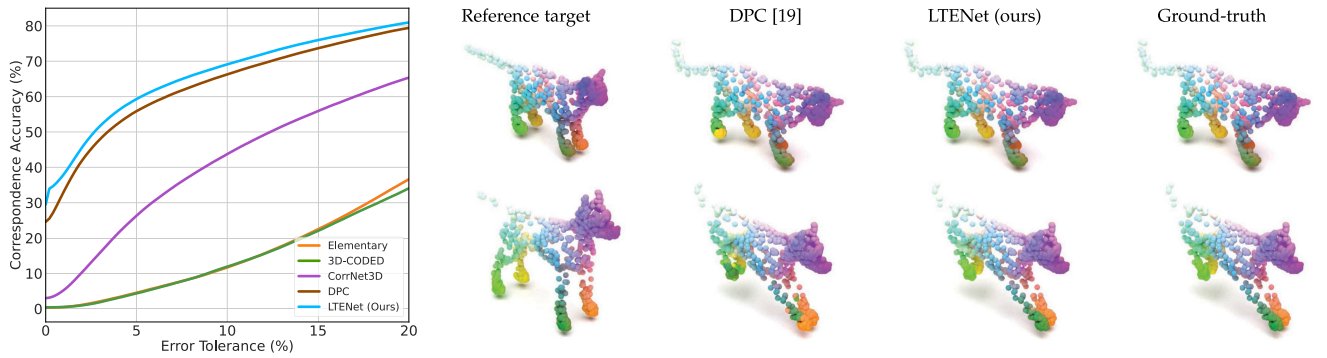

Fig. 5. (Left) The correspondence accuracies under different error tolerance values in the SMAL/TOSCA setting. Our method substantially improves the correspondence accuracies under all tolerance values. (Right) Visual examples of TOSCA test pairs. DPC suffers from prediction errors caused by the difficulty of distinguishing between left and right or rear and front legs. Our method generates more accurate correspondence predictions closer to the ground truth correspondence maps.

[85]. Our method achieved approximately $5\times$ and $2.5\times$ accuracies compared to SURFMNet and GeoFMNet, respectively, while showing a comparable run-time inference speed to DPC [19], which is about $100\times$ faster against SURFMNet and GeoFMNet (see the table in DPC [19] for more details). Diff-FMaps [14] suffers from over-fitting on training samples without exploiting shape priors, e.g., local smoothness. Corr-Net3D [18] shows improvements over 3D-CODED [16] and Elementary Structures [48] but requires nontrivial optimization in the Sinkhorn-inspired DeSmooth module and the decoder, which limits its generalization performance. DPC [19] is the current state-of-the-art method using learning the latent affinity via a simplified point reconstruction. Our LTENet achieved the best acc of 20.7% and the lowest err of 5.8, which significantly exceeds the accuracy of DPC by 17.0% and reduces the error by 4.9%. The accuracies in Fig. 4 (left) indicate that we achieved a clear improvement, especially for almost-perfect matching with $\epsilon < 5\%$.

Qualitative evaluation: We provide visual examples in Fig. 4 (right), showing the clear improvement made by LTENet (see more results in the appendix, available online). Additional visual examples using texture transfer can be found in Figs. 10 and 11.

C. Results on the Nonhuman Datasets

We trained models on the SMAL dataset and evaluated on the unseen TOSCA dataset that contains animal objects with diverse poses (SMAL/TOSCA).

Quantitative evaluation: As shown in Table I, LTENet achieved the best performance on SMAL/TOSCA in terms of the acc at 1% and err. The significant pose and shape differences between SMAL and TOSCA impact 3D-CODED and Elementary Structures relying on a single standard template, e.g., a standing cat. They struggle to handle shapes in different categories and various poses in the TOSCA test pairs. The proposed LTENet achieves an acc of 38.1% at 1% error, obtaining an increase of 4.3% in absolute accuracy compared to DPC's best acc of 33.8%. The detailed correspondence accuracy under different error tolerance values can be found in Fig. 5 (left), which shows that our method obtains a substantial improvement over other methods.

Qualitative evaluation: Fig. 5 (right) provides some visual results on the TOSCA test pairs, which verify that our method could generate more accurate correspondence predictions. More results can be found in the appendix, available online.
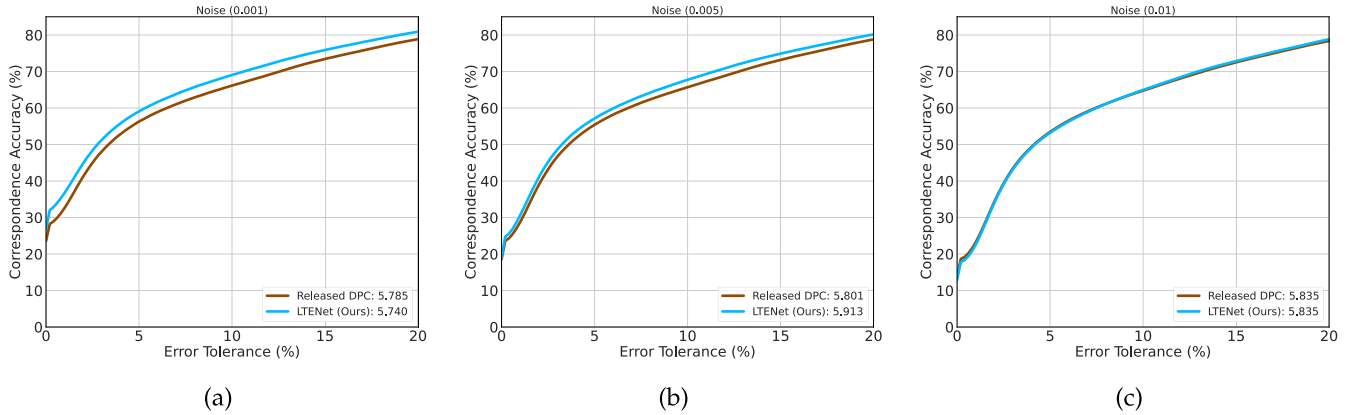
Fig. 6. Evaluation of correspondence prediction of TOSCA test point clouds in the SMAL/TOSCA setting with additional noise. From (a) to (c), we gradually add stronger Gaussian noise with zero means and larger standard deviations, i.e., $0.001, 0.005, 0.01$, to source shapes. Our method demonstrates its resilience against noise.

TABLE II
ABLATION STUDY ON ROTATIONS, SYMMETRIC (LEFT–RIGHT) FLIPS AND ROTATIONAL (FRONT–BACK) FLIPS

Rotations & Flips	acc ↑	err ↓
Without rotations	20.7%	5.8
rotations $\{\pm 10, \pm 10, \pm 10\}$	20.2% (-0.5%)	6.0 (+0.2)
rotations $\{\pm 20, \pm 20, \pm 20\}$	17.1% (-3.6%)	6.8 (+1.0)
rotations $\{\pm 30, \pm 30, \pm 30\}$	12.8% (-7.9%)	9.4 (+3.6)
rotational (front–back) flips	9.5% (-11.2%)	6.5 (+0.7)
symmetric (left–right) flips	6.3% (-14.4%)	10.0 (+4.2)

We conduct the ablation study under the SURREAL/SHREC setting.

D. Model Robustness Under Rotations, Symmetric and Rotational Flips

We test the model robustness of our LTENet against rotations, symmetric (left–right) flips, and rotational (front–back) flips. We conduct the ablation study under the SURREAL/SHREC setting. For the rotation experiment, given a pair of shapes, we rotate the target shape by different rotation matrices along three axes $\{x, y, z\}$, each with a fixed degree, e.g., $\pm 10, \pm 20$, or ± 30. For example, rotations $\{\pm 10, \pm 10, \pm 10\}$ in Table II indicate that there are eight rotation matrices as each axis can be rotated by $+10$ or -10. Regarding the symmetric (left–right) flips, or rotational (front–back) flips, we apply the flip transformation matrices that negate the coordinate in the corresponding axis for left–right or front–back and preserve the others.

As can be seen in Table II, our approach demonstrates some robustness against certain rotations. For example, when the rotations are $\{\pm 10, \pm 10, \pm 10\}$, the performance of our approach degrades by a small margin with an absolute accuracy reduction of 0.5% and an absolute error increase of 0.2. However, our model is not robust against symmetric (left–right) flips and rotational (front–back) flips, which we initially expected because we built our feature extractor based on a variant of DGCNN [49] that is not rotation-invariant and symmetry-aware. We highly encourage a future investigation on extending our approach along this direction [86], [87].

E. Model Robustness Under Presence of Noise

We investigate the robustness of the learned embeddings by perturbing the test dataset with Gaussian noise in the setting of SMAL/TOSCA, which is particularly challenging due to the presence of noise that ruins the underlying shape structure. Specifically, we select DPC as the competitive baseline. For the test samples from the TOSCA dataset, we add Gaussian noise with zero means and different standard deviations, i.e., $0.001, 0.005, 0.01$ to source shapes.

As can be seen in Fig. 6, our approach outperforms the state-of-the-art DPC in terms of correspondence accuracy and shows comparable performance in correspondence errors. Our method demonstrates moderate resilience against noise. Added noise heavily drops those correspondence accuracies under small error tolerance, e.g., less than 5%. In a similar manner, the additional experiment in the setting of SURREAL/SHREC can be found in the appendix, available online.

F. Comparisons Between DPC and LTENet

Our LTENet highlights *a novel approach to learning* locally linear *shape embeddings capable of capturing the underlying structure of the shape manifold*, which we achieve by marrying LLE with the construction of high-dimensional neighborhood-preserving shape embeddings. We built our architecture following the self- and cross-reconstruction framework in DPC [19]. Though both seek to learn good embeddings, our LTENet encourages the best *locally linear* alignment between shape embeddings without ambiguity via the closed-form expression of reconstruction weights.

Table III demonstrates its benefit by summarizing our key results and additional results from the appendix of DPC, available online. We clarify that E_r (the smoothness term) is the same mapping loss as in DPC. Due to DPC's lack of a mechanism to enforce local linearity of embeddings, E_r is required in DPC for better performance. Without this term, DPC suffers a significant drop from an *acc* of 17.7% to 11.4%. Our LTENet significantly outperforms DPC by enforcing a suitable manifold learning on shape correspondence under the same setting.
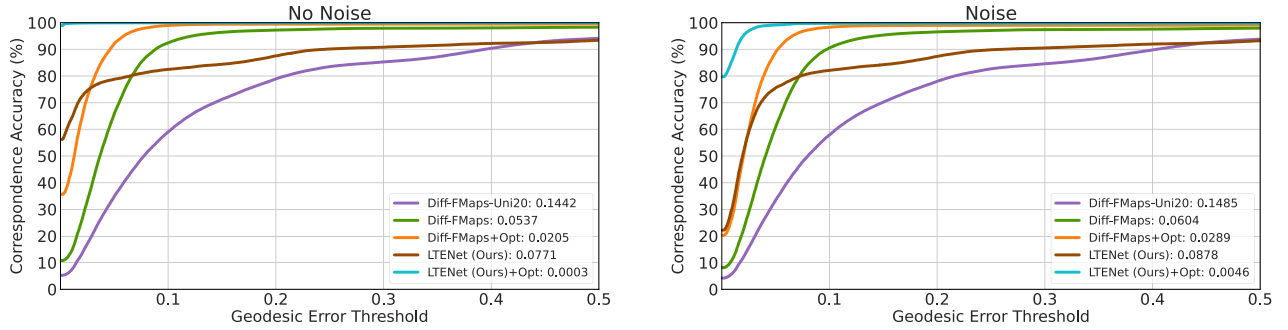
Fig. 7.   Evaluation of correspondence prediction of the FAUST test point clouds provided in Diff-FMaps [14] without or with additional noise. Cumulative curves with mean error in the legends. Without ground truth correspondence labels during the training, our model remains competitive or even outperforms some when compared against model variants of Diff-FMaps. It also verifies the generalization capability of our developed method.

<div style="display:flex; gap:2em;">

TABLE III
BENEFIT OF PURSUING LOCAL LINEARITY FOR EMBEDDING LEARNING

Method	$acc \uparrow$	$err \downarrow$
DPC (without E_r)	11.4%	6.7
DPC	17.7% (+6.3%)	6.1 (-0.6)
LTENet (without E_r)	20.6% (+9.2%)	5.9 (-0.8)
LTENet	20.7% (+9.3%)	5.8 (-0.9)

Our LTENet framework implicitly imposes a smooth constraint on embeddings such that it performs competitively without a regularization, while DPC [19] suffers a significant performance drop.
We conduct the ablation study under the SURREAL/SHREC setting.

TABLE IV
ABLATION STUDY ON MODEL CHOICES

Method	$\lambda_{\text{self}} = 1$	$\lambda_{\text{cross}} = 1$	$\lambda_{\text{reg}} = 10$	$acc \uparrow$	$err \downarrow$
Ours (Self)	✓			2.3%	8.9
Ours (Cross)		✓		20.6%	6.5
Ours (Self & Cross)	✓	✓		20.6%	5.9
Ours (CD)	✓	✓	✓	13.9%	6.3
Ours (EMD)	✓	✓	✓	19.6%	6.0
Ours	✓	✓	✓	20.7%	5.8

We conduct the ablation study under the SURREAL/SHREC setting.

</div>

G. Evaluation on the FAUST Dataset

We further validate our method and demonstrate their generalization capability on the popular FAUST Dataset, via following the training and evaluation of Diff-FMaps [14]. Specifically, we directly evaluated our model trained in the SURREAL/SHREC setting on the two test sets of point clouds sampled from the 100 shapes from the FAUST dataset with and without Gaussian noise. It is worth noting that our model was originally trained with shapes each with 1,024 points while the FAUST dataset is 1,000 points per test shape.

We compare our method against model variants of the fully supervised Diff-FMaps [14] trained with ground truth correspondence, while our method is unsupervised without ground truth correspondence. In Fig. 7, "Diff-FMaps-Uni20" denotes the competitive baseline using the universal embeddings in the original paper via enforcing the optimal linear transformation to be identity. "Diff-FMaps" is the complete baseline that predicts the correspondence after applying its predicted linear transformation to the universal embeddings via the learned descriptor network. "Diff-FMaps+Opt" is the model that applies the ground-truth linear transformation to the universal embeddings.

Without ground truth correspondence labels during the training, our model remains competitive or even outperforms some when compared against model variants of Diff-FMaps. It also verifies the generalization capability of our developed method.

H. Evaluation on Real-World Data

Inspired by [18], [19], we further examine our approach on model generalization and robustness by visualizing

correspondence predictions between point cloud pairs sampled from the behave dataset [88] captured in natural environments. Specifically, we use the mesh at the first frame in each mesh sequence to create the source point cloud by randomly sampling 1,024 vertices from its entire set of vertices. Similarly, we create ten target point clouds using meshes at the first frame's subsequent frames (i.e., frame 2 to frame 11).

As seen in Fig. 8, despite these shapes' significant pose differences, our approach shows its resilience by producing reliable correspondence results. It also verifies that our method is not robust against symmetric (left–right) flips and rotational (front–back) flips, which is also demonstrated previously in Table II. A future investigation is encouraged to handle these issues.

I. Ablation Study

We conduct the ablation study to evaluate the comparative effectiveness of the different components in LTENet under controlled experiments. All ablated versions were trained and evaluated following the SURREAL/SHREC setting.

Choices of the model designs: We first analyze LTENet with different design choices to control self- and cross-reconstruction of LTENet: 1) LTENet (Self) only uses the self-reconstruction of LTENet by setting $\lambda_{\text{self}} = 1, \lambda_{\text{cross}} = 0$; 2) LTENet (Cross) only uses the cross-reconstruction of LTENet by setting $\lambda_{\text{self}} = 0, \lambda_{\text{cross}} = 1$; 3) LTENet (Self & Cross) uses both self- and cross-reconstruction and removes the mapping loss for the regularization; and (4) LTENet (CD) and LTENet (EMD) are similar to the full model of LTENet in only replacing the CS objective with popular CD and EMD. All results are summarized in Table IV. It is clear that cross-reconstruction contributes most
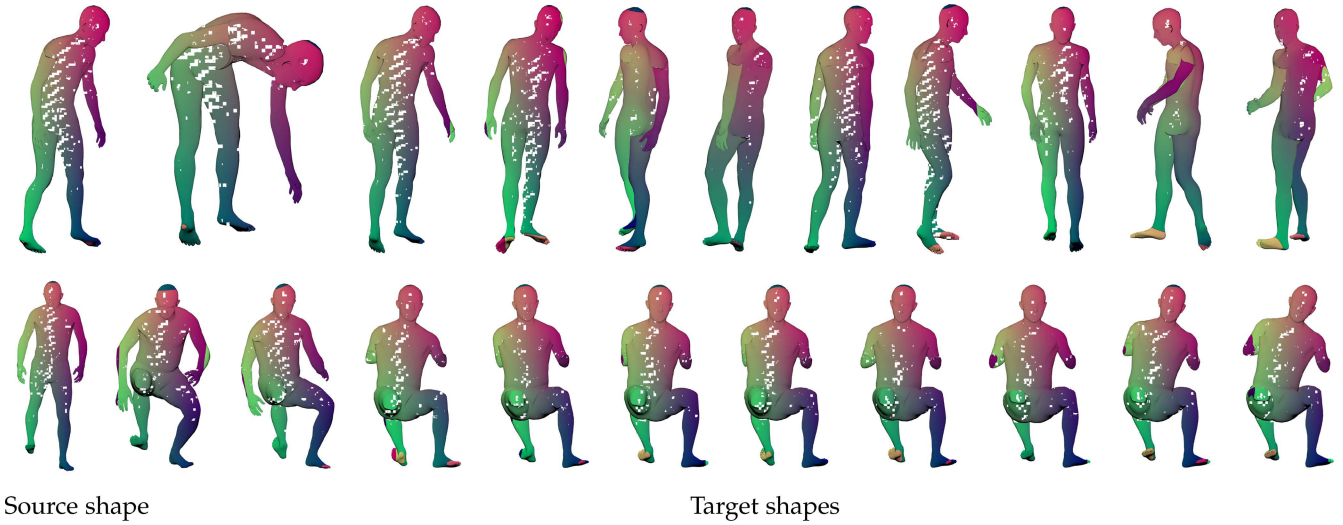
Fig. 8. Qualitative examples of correspondence predictions between point cloud pairs sampled from the behave dataset [88] captured in natural environments. For each row, the left-most shape is the source shape and the remaining shapes are target shapes. We visualize the correspondence between the source shape and its target shapes. Despite the significant differences in pose, our approach shows its resilience by producing reliable correspondence results. However, our approach is not robust against the rotational (front–back) flips.
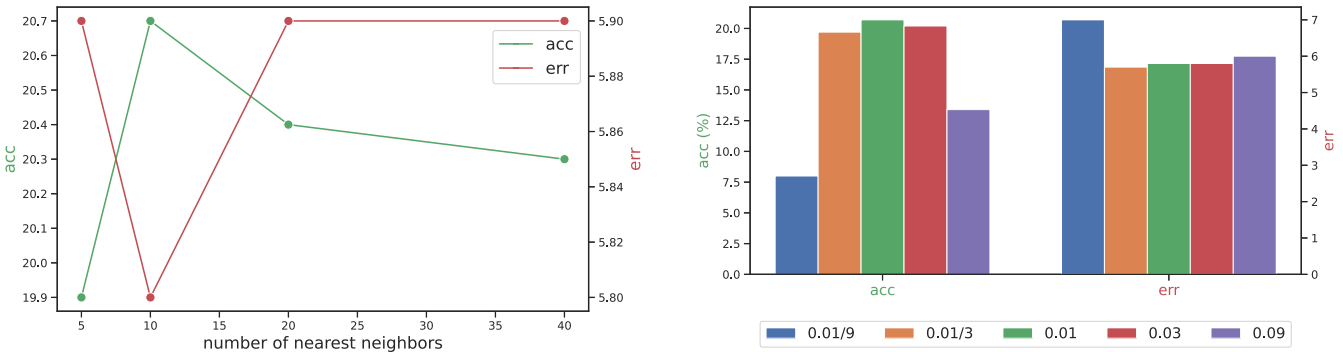

Fig. 9. (left) Model performance with different numbers of nearest neighbors; (right) Model performance with different kernel bandwidths. Choosing a suitable bandwidth or number of nearest neighbors leads to better performance.

to the final performance and that CS leads to better performance compared to CD and EMD.

Choices of the kernel bandwidth σ: As our CS objective is closely relevant to a fixed-bandwidth KDE with Gaussian kernels, it is important to choose a suitable bandwidth fitting the underlying data distribution of the training dataset — either too large or too small bandwidth values could lead to degraded performance. As shown in Fig. 9 (right), we chose different bandwidth values by setting $\sigma = 0.01/9, 0.01/3, 0.01, 0.03, 0.09$. The results suggest that 0.01 is a suitable bandwidth, which we used thereafter for LTENet. It is worth noting that we should adjust the bandwidth accordingly when moving to a new dataset.

Number of nearest neighbors: Similarly, we can set different numbers of nearest neighbors for the locally linear transformations, i.e., $K = 5, 10, 20, 40$. Fig. 9 (left) demonstrates that $K = 10$ leads to better performance.

The impact of training sample size: We trained models by increasing the training sample sizes from 2,000 to 230,000. Table V summarizes the experimental results of training models using different training samples. The experimental results show

TABLE V
ABLATION STUDY ON THE TRAINING SAMPLE SIZE

Method	$\lambda_{\text{self}} = 1$	$\lambda_{\text{cross}} = 1$	$\lambda_{\text{reg}} = 10$	size	acc ↑	err ↓
LTENet (Self)	✓			2K	2.3%	8.9
LTENet (Cross)		✓		2K	20.6%	6.5
LTENet (Self & Cross)	✓	✓		2K	20.6%	5.9
LTENet	✓	✓	✓	2K	20.7%	5.8
LTENet (Self)	✓			230K	2.2%	8.9
LTENet (Cross)		✓		230K	19.5%	7.2
LTENet (Self & Cross)	✓	✓		230K	20.5%	6.5
LTENet	✓	✓	✓	230K	20.9%	6.2

We conduct the ablation study under the SURREAL/SHREC setting.

that 2,000 samples are sufficient. A subtle difference is found in the slightly increased *err* when training models with 230,000 training samples, which we suspect is due to more training samples that are symmetric and rotated being included in training, thus creating noisy training signals. Handling symmetry of shapes remains an under-explored research area. For shape correspondence though, there exist some attempts [86], [89], which are promising.

Source shape   DPC [19]   LTENet (ours)   Ground-truth   Source shape   DPC [19]   LTENet (ours)   Ground-truth
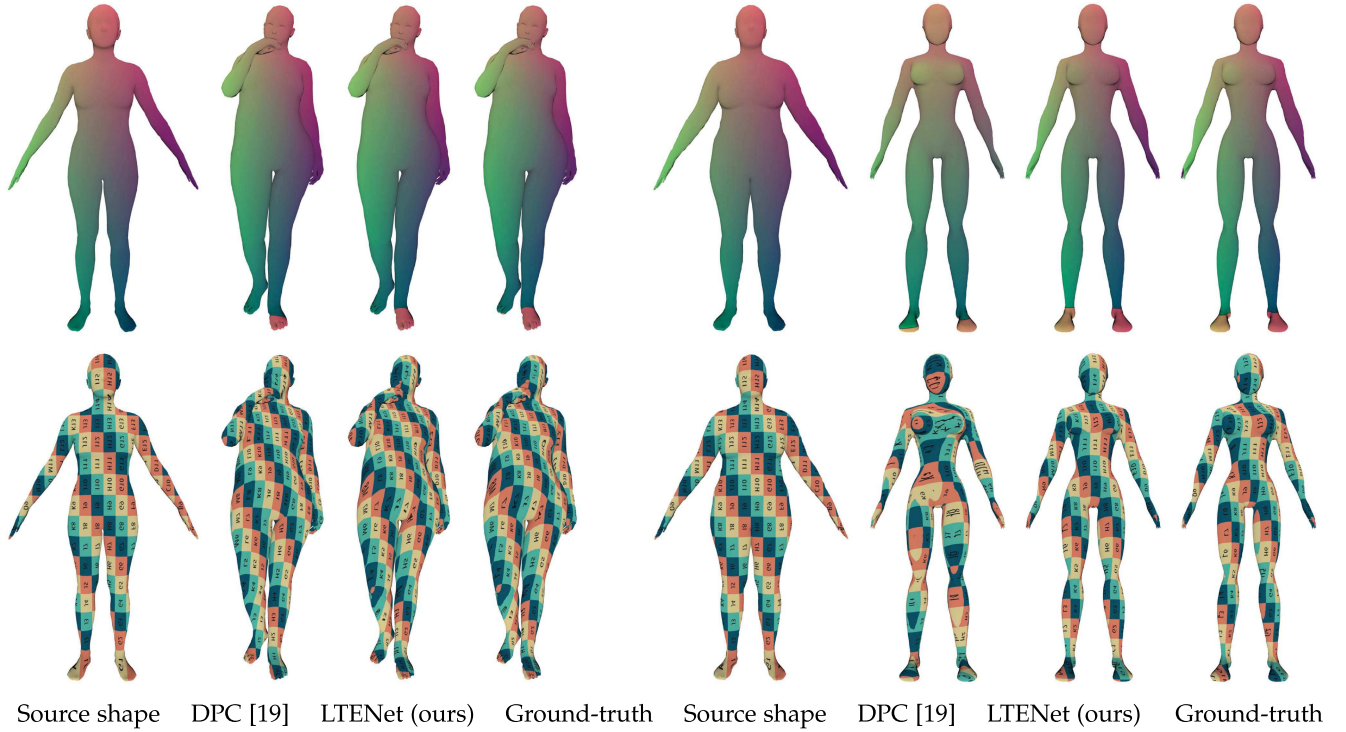
Fig. 10.   Visual texture transfer examples of SHREC test pairs using checker pattern (top row) and color code (bottom row) textures. The experiment follows the SURREAL/SHREC setting. The predicted maps from DPC [19] and our approach are used to pull a texture from the source shape to the target shapes and compare against that using the ground truth maps. The proposed LTENet generates more accurate correspondence predictions compared to DPC.

Source shape   DPC [19]   LTENet (ours)   Ground-truth   Source shape   DPC [19]   LTENet (ours)   Ground-truth
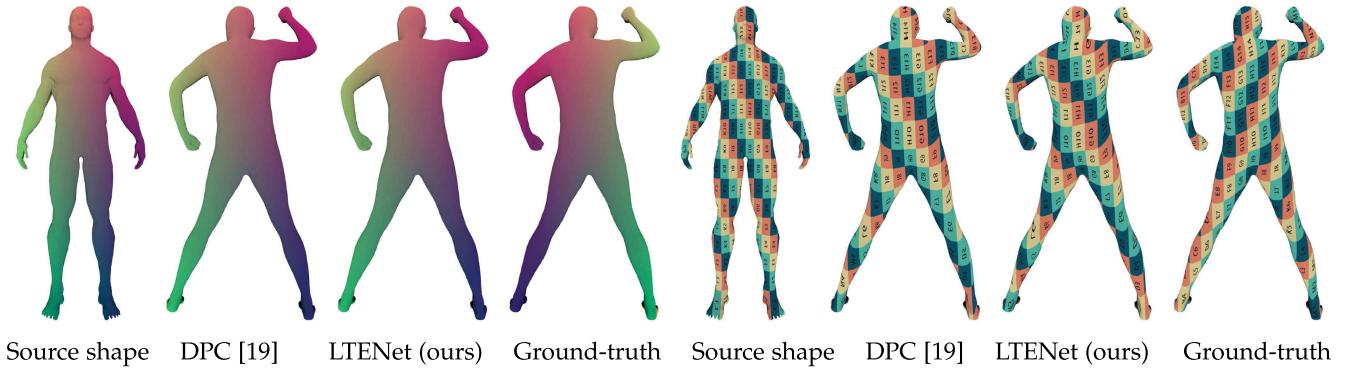
Fig. 11.   Typical failure example of SHREC test pairs using checker pattern (right) and color code (left) textures. The experiment follows the SURREAL/SHREC setting. It indicates that both DPC and our method can not handle the rotational (front–back) flip.

The effect of embedding dimension: Given $\mathcal{X}$ and $\mathcal{Y}$, we extract their nonlinear feature embeddings $\mathcal{F}^{\mathcal{X}}, \mathcal{F}^{\mathcal{Y}} \in \mathbb{R}^{N \times D}$, respectively, via a neural network $\mathcal{F}$. The embedding dimension D should be adjusted to achieve a balance between overfitting and underfitting and efficiency. Table VI demonstrates that the model of $D = 512$ gives a good generalization performance while being efficient, which we used thereafter for LTENet.

VI. DISCUSSION

Our experimental results showed that LTENet achieves superior performance compared to state-of-the-art unsupervised shape correspondence methods. We attribute this to the learning mechanism capable of capturing the underlying structure of the manifold and fully exploiting the local euclidean geometry of manifolds within local neighborhoods. Our approach encourages the best locally linear alignment between shape embeddings without ambiguity via the closed-form expression of reconstruction weights. The local linearity used in our approach leads to implicit regularization and universal/canonical embeddings between a pair of shapes in correspondence. We demonstrated the performance gap of our learned embeddings, with and without additional functional map-inspired *globally* linear transformations, and showed consistent improvements made by the additional linear transformations. We observed a mismatch in that embeddings learned from an unsupervised

TABLE VI
ABLATION STUDY ON THE EFFECTS OF EMBEDDING DIMENSION

Method	Feature dimension	*acc*	*err*
LTENet	32	19.1%	6.4
LTENet	64	19.4%	6.2
LTENet	128	19.7%	6.2
LTENet	256	20.1%	6.1
LTENet	512	20.7%	5.8
LTENet	1,024	20.8%	5.9

We conduct the study under the SURREAL/SHREC setting.

shape correspondence method are not necessarily suitable to be used as basis embeddings in the classic functional map framework.

A. Limitations and Future Work

It has been demonstrated that shape correspondence approaches are struggling in disambiguating shape symmetries [19], [86], [89], [90]. In our work, we also observed the symmetry issue—it leads to noisy training signals by wrongly matching components between shapes, e.g., associating the left hand in one human with the right hand in another human due to their opposite orientations. Future work is suggested to handle symmetry by exploiting priors on shapes to impose additional regularization or constraints on embedding learning.

Many extensions of LLE, such as modified locally linear embedding (MLLE) [91], LLE with geodesic distances [92], and LLE with penalty functions [75], have been proposed to further improve LLE. It is promising to incorporate these advanced designs and adapt them to shape correspondence for better performance. The discovered mismatch problem suggests that further exploration of learning embeddings suitable for use as bases in the functional map is a promising direction to establish a unified shape correspondence framework, particularly in the unsupervised learning setting.

In this work, we focus on the matching problem between point cloud shapes. In the future, we propose to extend our method to matching problems in other modalities, e.g., images and meshes, and cross-modality matching problems, e.g., images to point clouds.

VII. CONCLUSION

We have presented a novel approach to unsupervised shape correspondence learning between pairs of point clouds. LTENet is unique in that it introduces an LLE-inspired algorithm that represents maps between these shapes as locally linear transformations in the high-dimensional embedding spaces and leads to the learning of universal/canonical embeddings for shapes in correspondence. The embedding learning is driven by minimizing a suitable divergence measure between the LLE cross-reconstruction of source and target point clouds. Remarkably, LTENet achieves state-of-the-art performance on standard benchmark datasets while showing strong model generalization across datasets with efficient training and inference.

REFERENCES

[1] R. W. Sumner and J. Popović, "Deformation transfer for triangle meshes," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 399–405, 2004.

[2] K. Yin, J. Gao, M. Shugrina, S. Khamis, and S. Fidler, "3DStyleNet: Creating 3D shapes with geometric and texture style variations," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 12456–12465.

[3] D. Shen, E. H. Herskovits, and C. Davatzikos, "An adaptive-focus statistical shape model for segmentation and shape modeling of 3-D brain structures," *IEEE Trans. Med. Imag.*, vol. 20, no. 4, pp. 257–270, Apr. 2001.

[4] A. E. Johnson and M. Hebert, "Using spin images for efficient object recognition in cluttered 3D scenes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 5, pp. 433–449, May 1999.

[5] S. Belongie, G. Mori, and J. Malik, "Matching with shape contexts," in *Statistics and Analysis of Shapes*. Berlin, Germany: Springer, 2006, pp. 81–105.

[6] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (FPFH) for 3D registration," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2009, pp. 3212–3217.

[7] R. M. Rustamov, "Laplace-beltrami eigenfunctions for deformation invariant shape representation," in *Proc. 5th Eurographics Symp. Geometry Process.*, 2007, pp. 225–233.

[8] J. Sun, M. Ovsjanikov, and L. Guibas, "A concise and provably informative multi-scale signature based on heat diffusion," *Comput. Graph. Forum*, vol. 28, no. 5, pp. 1383–1392, 2009.

[9] M. Aubry, U. Schlickewei, and D. Cremers, "The wave kernel signature: A quantum mechanical approach to shape analysis," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, 2011, pp. 1626–1633.

[10] R. Litman and A. M. Bronstein, "Learning spectral descriptors for deformable shape correspondence," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 1, pp. 171–180, Jan. 2014.

[11] E. Rodola, S. Rota Bulo, T. Windheuser, M. Vestner, and D. Cremers, "Dense non-rigid shape correspondence using random forests," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 4177–4184.

[12] D. Boscaini, J. Masci, E. Rodolà, and M. Bronstein, "Learning shape correspondence with anisotropic convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 3189–3197.

[13] F. Monti, D. Boscaini, J. Masci, E. Rodola, J. Svoboda, and M. M. Bronstein, "Geometric deep learning on graphs and manifolds using mixture model CNNs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 5115–5124.

[14] R. Marin, M.-J. Rakotosaona, S. Melzi, and M. Ovsjanikov, "Correspondence learning via linearly-invariant embedding," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 1608–1620.

[15] L. Cosmo, G. Minello, M. Bronstein, L. Rossi, and A. Torsello, "The average mixing kernel signature," in *Proc. Eur. Conf. Comput. Vis.*, Springer, 2020, pp. 1–17.

[16] T. Groueix, M. Fisher, V. G. Kim, B. C. Russell, and M. Aubry, "3D-coded: 3D correspondences by deep deformation," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 230–246.

[17] J.-M. Roufosse, A. Sharma, and M. Ovsjanikov, "Unsupervised deep learning for structured shape matching," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 1617–1627.

[18] Y. Zeng, Y. Qian, Z. Zhu, J. Hou, H. Yuan, and Y. He, "CorrNet3D: Unsupervised end-to-end learning of dense correspondence for 3D point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 6052–6061.

[19] I. Lang, D. Ginzburg, S. Avidan, and D. Raviv, "DPC: Unsupervised deep point correspondence via cross and self construction," in *Proc. IEEE Int. Conf. 3D Vis.*, 2021, pp. 1442–1451.

[20] R. Chen, Y. Cong, and J. Dong, "Unsupervised dense deformation embedding network for template-free shape correspondence," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 8361–8370.

[21] D. Ginzburg and D. Raviv, "Cyclic functional mapping: Self-supervised correspondence between non-isometric deformable shapes," in *Proc. Eur. Conf. Comput. Vis.*, Springer, 2020, pp. 36–52.

[22] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[23] M. Ovsjanikov, M. Ben-Chen, J. Solomon, A. Butscher, and L. Guibas, "Functional maps: A flexible representation of maps between shapes," *ACM Trans. Graph.*, vol. 31, no. 4, pp. 1–11, 2012.

[24] M. Carcassoni and E. R. Hancock, "Spectral correspondence for point pattern matching," *Pattern Recognit.*, vol. 36, no. 1, pp. 193–204, 2003.

[25] B. Luo and E. R. Hancock, "A unified framework for alignment and correspondence," *Comput. Vis. Image Understanding*, vol. 92, no. 1, pp. 26–55, 2003.

[26] D. Knossow, A. Sharma, D. Mateus, and R. Horaud, "Inexact matching of large and sparse graphs using laplacian eigenvectors," in *Proc. Int. Workshop Graph-Based Representations Pattern Recognit.*, Springer, 2009, pp. 144–153.

[27] P. J. Besl and N. D. McKay, "Method for registration of 3-D shapes," in *Proc. Sensor Fusion IV: Control Paradigms Data Struct.*, International Society for Optics and Photonics, 1992, pp. 586–606.

[28] A. Thayananthan, B. Stenger, P. H. Torr, and R. Cipolla, "Shape context and chamfer matching in cluttered scenes," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2003, pp. I–I.

[29] E. Levina and P. Bickel, "The earth mover's distance is the mallows distance: Some insights from statistics," in *Proc. IEEE 8th Int. Conf. Comput. Vis.*, 2001, pp. 251–256.

[30] A. Rangarajan, A. Yuille, and E. Mjolsness, "Convergence properties of the softassign quadratic assignment algorithm," *Neural Comput.*, vol. 11, no. 6, pp. 1455–1474, 1999.

[31] A. Rangarajan, H. Chui, and F. L. Bookstein, "The softassign procrustes matching algorithm," in *Proc. Biennial Int. Conf. Inf. Process. Med. Imag.*, Springer, 1997, pp. 29–42.

[32] H. Chui and A. Rangarajan, "A new point matching algorithm for non-rigid registration," *Comput. Vis. Image Understanding*, vol. 89, no. 2/3, pp. 114–141, 2003.

[33] A. Myronenko and X. Song, "Point set registration: Coherent point drift," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 12, pp. 2262–2275, Dec. 2010.

[34] J. Ma, J. Zhao, J. Tian, Z. Tu, and A. L. Yuille, "Robust estimation of nonrigid transformation for point set registration," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 2147–2154.

[35] J. Ma, W. Qiu, J. Zhao, Y. Ma, A. L. Yuille, and Z. Tu, "Robust L2E estimation of transformation for non-rigid registration," *IEEE Trans. Signal Process.*, vol. 63, no. 5, pp. 1115–1129, Mar. 2015.

[36] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 4, pp. 509–522, Apr. 2002.

[37] A. W. Fitzgibbon, "Robust registration of 2D and 3D point sets," *Image Vis. Comput.*, vol. 21, no. 13/14, pp. 1145–1153, 2003.

[38] Y. Tsin and T. Kanade, "A correlation-based approach to robust point set registration," in *Proc. Eur. Conf. Comput. Vis.*, Springer, 2004, pp. 558–569.

[39] A. S. Roy, A. Gopinath, and A. Rangarajan, "Deformable density matching for 3D non-rigid registration of shapes," in *Proc. Int. Conf. Med. Image Comput. Comput.- Assist. Interv.*, Springer, 2007, pp. 942–949.

[40] B. Jian and B. C. Vemuri, "Robust point set registration using gaussian mixture models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 8, pp. 1633–1645, Aug. 2011.

[41] E. Hasanbelliu, L. S. Giraldo, and J. C. Príncipe, "A robust point matching algorithm for non-rigid registration using the cauchy-schwarz divergence," in *Proc. IEEE Int. Workshop Mach. Learn. Signal Process.*, 2011, pp. 1–6.

[42] A. Kovnatsky, M. M. Bronstein, X. Bresson, and P. Vandergheynst, "Functional correspondence by matrix completion," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 905–914.

[43] A. Kovnatsky, K. Glashoff, and M. M. Bronstein, "MADMM: A generic algorithm for non-smooth optimization on manifolds," in *Proc. Eur. Conf. Comput. Vis.*, Springer, 2016, pp. 680–696.

[44] O. Litany, T. Remez, E. Rodola, A. Bronstein, and M. Bronstein, "Deep functional maps: Structured prediction for dense shape correspondence," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 5659–5667.

[45] E. Rodolà, L. Cosmo, M. M. Bronstein, A. Torsello, and D. Cremers, "Partial functional correspondence," *Comput. Graph. Forum*, vol. 36, no. 1, pp. 222–236, 2017.

[46] G. Pai, J. Ren, S. Melzi, P. Wonka, and M. Ovsjanikov, "Fast sinkhorn filters: Using matrix scaling for non-rigid shape correspondence with functional maps," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 384–393.

[47] O. Halimi, O. Litany, E. Rodola, A. M. Bronstein, and R. Kimmel, "Unsupervised learning of dense shape correspondence," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 4370–4379.

[48] T. Deprelle, T. Groueix, M. Fisher, V. G. Kim, B. C. Russell, and M. Aubry, "Learning elementary structures for 3D shape generation and matching," 2019, *arXiv:1908.04725*.

[49] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph CNN for learning on point clouds," *ACM Trans. On Graph.*, vol. 38, no. 5, pp. 1–12, 2019.

[50] B. L. Bhatnagar, C. Sminchisescu, C. Theobalt, and G. Pons-Moll, "LoopReg: Self-supervised learning of implicit surface correspondences, pose and shape for 3D human mesh registration," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 12909–12922.

[51] E. Corona, G. Pons-Moll, G. Alenyà, and F. Moreno-Noguer, "Learned vertex descent: A new direction for 3D human model fitting," in *Proc. 17th Eur. Conf. Comput. Vis.*, Tel Aviv, Israel, Springer, 2022, pp. 146–165.

[52] M. Eisenberger, A. Toker, L. Leal-Taixé, and D. Cremers, "Deep shells: Unsupervised shape correspondence with optimal transport," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 10491–10502.

[53] M. Eisenberger, Z. Lahner, and D. Cremers, "Smooth shells: Multi-scale shape registration with functional maps," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 12265–12274.

[54] G. Trappolini, L. Cosmo, L. Moschella, R. Marin, S. Melzi, and E. Rodolà, "Shape registration in the time of transformers," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, pp. 5731–5744.

[55] M. Eisenberger et al., "NeuroMorph: Unsupervised shape interpolation and correspondence in one go," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 7473–7483.

[56] R. S. Sundararaman, R. Marin, E. Rodola, and M. Ovsjanikov, "Reduced representation of deformation fields for effective non-rigid shape matching," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, pp. 10405–10420.

[57] B. Bustos, D. A. Keim, D. Saupe, T. Schreck, and D. V. Vranić, "Feature-based similarity search in 3D object databases," *ACM Comput. Surv.*, vol. 37, no. 4, pp. 345–387, 2005.

[58] J. W. Tangelder and R. C. Veltkamp, "A survey of content based 3D shape retrieval methods," *Multimedia Tools Appl.*, vol. 39, no. 3, pp. 441–471, 2008.

[59] M. Pauly, R. Keiser, and M. Gross, "Multi-scale feature extraction on point-sampled surfaces," *Comput. Graph. Forum*, vol. 22, no. 3, pp. 281–289, 2003.

[60] A. B. Hamza and H. Krim, "Geodesic object representation and recognition," in *Proc. Int. Conf. Discrete Geometry Comput. Imagery*, Springer, 2003, pp. 378–387.

[61] Y. Lipman and T. Funkhouser, "Möbius voting for surface correspondence," *ACM Trans. Graph.*, vol. 28, no. 3, pp. 1–12, 2009.

[62] P. Bérard, G. Besson, and S. Gallot, "Embedding riemannian manifolds by their heat kernel," *Geometric Funct. Anal.*, vol. 4, no. 4, pp. 373–398, 1994.

[63] F. Mémoli, "Spectral gromov-wasserstein distances for shape matching," in *Proc. IEEE 12th Int. Conf. Comput. Vis. Workshops*, 2009, pp. 256–263.

[64] A. M. Bronstein, M. M. Bronstein, R. Kimmel, M. Mahmoudi, and G. Sapiro, "A gromov-hausdorff framework with diffusion geometry for topologically-robust non-rigid shape matching," *Int. J. Comput. Vis.*, vol. 89, no. 2/3, pp. 266–286, 2010.

[65] J. Corring and A. Rangarajan, "Shape from phase: An integrated level set and probability density shape representation," in *Proc. 22nd Int. Conf. Pattern Recognit.*, 2014, pp. 46–51.

[66] A. M. Bronstein, M. M. Bronstein, L. J. Guibas, and M. Ovsjanikov, "Shape google: Geometric words and expressions for invariant shape retrieval," *ACM Trans. Graph.*, vol. 30, no. 1, pp. 1–20, 2011.

[67] U. Castellani, M. Cristani, S. Fantoni, and V. Murino, "Sparse points matching by combining 3D mesh saliency with statistical descriptors," *Comput. Graph. Forum*, vol. 27, no. 2, pp. 643–652, 2008.

[68] J. Masci, D. Boscaini, M. Bronstein, and P. Vandergheynst, "Geodesic convolutional neural networks on riemannian manifolds," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, 2015, pp. 37–45.

[69] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 652–660.

[70] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[71] M. Fey, J. E. Lenssen, F. Weichert, and H. Müller, "SplineCNN: Fast geometric deep learning with continuous B-spline kernels," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 869–877.

[72] N. Donati, A. Sharma, and M. Ovsjanikov, "Deep geometric functional maps: Robust feature learning for shape correspondence," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 8592–8601.

[73] H. Thomas, C. R. Qi, J.-E. Deschaud, B. Marcotegui, F. Goulette, and L. J. Guibas, "KPConv: Flexible and deformable convolution for point clouds," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 6411–6420.

[74] B. Ghojogh, A. Ghodsi, F. Karray, and M. Crowley, "Locally linear embedding and its variants: Tutorial and survey," 2020, *arXiv:2011.10925*.

[75] M. Winlaw, L. S. Dehkordy, and A. Ghodsi, "Robust locally linear embedding using penalty functions," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, 2011, pp. 2305–2312.

[76] P. He, P. Emami, S. Ranka, and A. Rangarajan, "Self-supervised robust scene flow estimation via the alignment of probability density functions," in *Proc. 36th AAAI Conf. Artif. Intell.*, 2022, pp. 861–869.

[77] R. Jenssen, D. Erdogmus, K. E. Hild, J. C. Principe, and T. Eltoft, "Optimizing the Cauchy-Schwarz PDF distance for information theoretic, non-parametric clustering," in *Proc. Int. Workshop Energy Minimization Methods Comput. Vis. Pattern Recognit.*, 2005, pp. 34–45.

[78] R. Jenssen, J. C. Principe, D. Erdogmus, and T. Eltoft, "The Cauchy-Schwarz divergence and parzen windowing: Connections to graph theory and mercer kernels," *J. Franklin Inst.*, vol. 343, no. 6, pp. 614–629, 2006.

[79] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black, "SMPL: A skinned multi-person linear model," *ACM Trans. Graph.*, vol. 34, no. 6, pp. 1–16, 2015.

[80] S. Melzi et al., "SHREC 2019: Matching humans with different connectivity," in *Proc. Eurographics Workshop 3D Object Retrieval*, 2019, pp. 121–128.

[81] S. Zuffi, A. Kanazawa, D. W. Jacobs, and M. J. Black, "3D menagerie: Modeling the 3D shape and pose of animals," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 6365–6373.

[82] A. M. Bronstein, M. M. Bronstein, and R. Kimmel, *Numerical Geometry of Non-Rigid Shapes*. Berlin, Germany: Springer Science & Business Media, 2008.

[83] A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 8026–8037.

[84] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," 2017, *arXiv:1711.05101*.

[85] S. Melzi, J. Ren, E. Rodolà, A. Sharma, P. Wonka, and M. Ovsjanikov, "ZoomOut: Spectral upsampling for efficient shape correspondence," *ACM Trans. Graph.*, vol. 38, no. 6, Nov. 2019.

[86] Y. Yoshiyasu, E. Yoshida, and L. Guibas, "Symmetry aware embedding for shape correspondence," *Comput. Graph.*, vol. 60, pp. 9–22, 2016.

[87] A. Poulenard, M.-J. Rakotosaona, Y. Ponty, and M. Ovsjanikov, "Effective rotation-invariant point CNN with spherical harmonics kernels," in *Proc. Int. Conf. 3D Vis.*, 2019, pp. 47–56.

[88] B. L. Bhatnagar, X. Xie, I. A. Petrov, C. Sminchisescu, C. Theobalt, and G. Pons-Moll, "Behave: Dataset and method for tracking human object interactions," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 15935–15946.

[89] A. Sharma and M. Ovsjanikov, "Learning canonical embedding for non-rigid shape matching," 2021, *arXiv:2110.02994*.

[90] N. Donati, E. Corman, and M. Ovsjanikov, "Deep orientation-aware functional maps: Tackling symmetry issues in shape matching," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 742–751.

[91] Z. Zhang and J. Wang, "MLLE: Modified locally linear embedding using multiple weights," in *Proc. Adv. Neural Inf. Process. Syst.*, 2006, pp. 1593–1600.

[92] C. Varini, A. Degenhard, and T. Nattkemper, "ISOLLE: Locally linear embedding with geodesic distance," in *Proc. Eur. Conf. Princ. Data Mining Knowl. Discov.*, Springer, 2005, pp. 331–342.

Pan He (Student Member, IEEE) is currently working toward the PhD degree with the Department of Computer & Information Science & Engineering, University of Florida, Gainesville, FL. His research interests include deep learning and computer vision.

Patrick Emami (Member, IEEE) received the BSc degree in computer engineering from the University of Florida (UF), Gainesville, Florida, USA. in 2016, and the PhD degree in computer science from University of Florida, in 2021. His research is focused on developing neural algorithms for the analysis and generation of multi-object spatio-temporal data.

Sanjay Ranka (Fellow, IEEE) is a distinguished professor with the Department of Computer Information Science and Engineering, University of Florida, Gainesville, Florida, USA. His current research is on developing algorithms and software using machine learning, Internet of Things, GPU computing, and cloud computing for solving applications in transportation and health care. His work has received 14,500+ citations, with an h-index of 61 (Google Scholar). He is a fellow of the AAAS and AIAA (Asia-Pacific Artificial Intelligence Association) and a past member of IFIP Committee on System Modeling and Optimization. He was awarded the 2020 Research Impact Award from IEEE Technical Committee on Cloud Computing. His research is currently funded by NIH, NSF, USDOT, DOE, and FDOT.

Anand Rangarajan (Member, IEEE) is a professor with the Department of Computer and Information Science and Engineering, University of Florida, Gainesville, FL, USA. His research interests are computer vision, machine learning, intelligent transportation, medical and hyperspectral imaging and the science of consciousness.