

Anisotropic Multi-Scale Graph Convolutional Network for Dense Shape Correspondence

Mohammad Farazi, Wenhui Zhu, Zhangsihao Yang, and Yalin Wang
Arizona State University
Tempe, Arizona

{mfarazi, wzhu59, zyang195, ylwang}@asu.edu

Abstract

This paper studies 3D dense shape correspondence, a key shape analysis application in computer vision and graphics. We introduce a novel hybrid geometric deep learning-based model that learns geometrically meaningful and discretization-independent features. The proposed framework has a U-Net model as the primary node feature extractor, followed by a successive spectral-based graph convolutional network. To create a diverse set of filters, we use anisotropic wavelet basis filters, being sensitive to both different directions and band-passes. This filter set overcomes the common over-smoothing behavior of conventional graph neural networks. To further improve the model's performance, we add a function that perturbs the feature maps in the last layer ahead of fully connected layers, forcing the network to learn more discriminative features overall. The resulting correspondence maps show state-of-the-art performance on the benchmark datasets based on average geodesic errors and superior robustness to discretization in 3D meshes. Our approach provides new insights and practical solutions to the dense shape correspondence research.

1. Introduction

In recent years, learning-based non-rigid shape correspondence approaches have been revolutionized with the success of geometric deep learning on unstructured data [7]. These approaches come with various learning scenarios, from unsupervised to supervised learning methods on 3D mesh or point cloud representations [18, 14, 16, 25, 37]. While modeling dense shape correspondence problem can differ in terms of loss functions and problem formulations, the learned descriptors, ideally, have to be invariant to near-isometric deformations and discretization of 3D shapes. Unfortunately, many spatial-based graph neural network models, relying on neighbor feature aggregation, fail to gen-

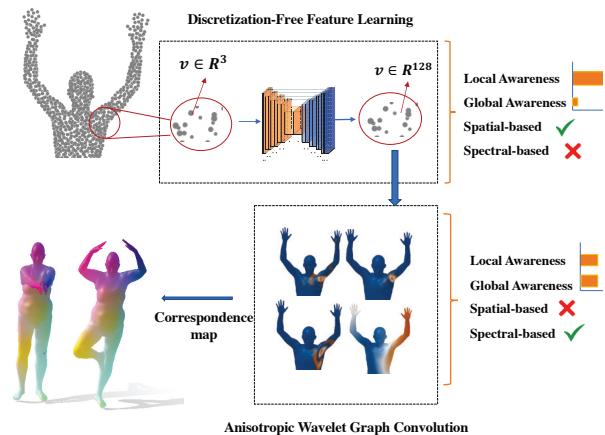


Figure 1. Our hybrid model learns discretization-free features using a U-Net structure based on a spatial-based point cloud model, KPconv. [29]. Then, to learn more globally and locally aware features, we use anisotropic wavelet based filters for our graph convolution layer to capture intrinsic information among different band-passes. Anisotropic wavelets also help our filters to be sensitive to direction.

eralize to meshes with different sizes and discretizations as they overfit to the mesh connectivity [28, 32, 13].

Lately, studies in [32, 12, 28, 13] have addressed the discretization invariance problem. The core to their approach is the idea of using the Laplace Beltrami Operator (LBO) eigen-basis and heat diffusion. The omnipresence of LBO can be used in many forms like the pre-computation of 3D shape wavelet filters [32], heat diffusion as a spatial communication among the features [28], or general functional map settings [26]. However, unlike state-of-the-art graph convolutional networks [22, 33, 11, 30, 16], spectral models usually have inferior performance in single-resolution mesh setting in terms of perfect matched points and average geodesic errors. This can be seen as a trade-off between lo-

cal and global feature learning awareness. To remedy this, introducing a more diverse set of filters in different band-passes and hybridization with spatial-based models can be a promising solution.

In this study, we propose a hybrid model (Figure 1) that incorporates the merits of both spatial and spectral-based approaches to overcome the limitations mentioned above. Our model learns in an end-to-end setting, starting with a point cloud U-Net block to learn geometric features and feed them to a spectral-based graph convolutional network to learn robust and discriminative features. To overcome the over-smoothing nature of graph convolutional networks, we adopt graph wavelet kernel as diverse filters to capture intrinsic information from different band-pass lenses. To make our filters even more diverse, we also propose to use the anisotropic LBO to make our kernels directionally sensitive. Finally, as the last layer of our network, we creatively use a layer to perturb the feature map to force the network to learn more discriminative features. As a result, it significantly improves our system performance.

Overall, our contributions are summarized as follows:

- (1) To our knowledge, it is the first geometric deep learning framework that learns both spatial-based (via U-net) and spectral-based (via anisotropic wavelet graph convolution network) geometric features in a data-driven fashion. The learned features are sensitive to subtle geometric changes and robust to 3D discretization differences.
- (2) We employ anisotropic wavelet filters to address the common over-smoothing drawback of conventional graph convolutional networks. Instead of only low-pass filtering, we design graph wavelet functions with different band-passes. Our work achieves a diverse set of filters sensitive to various directions and band-passes and effectively learns rich intrinsic geometric features.
- (3) We creatively apply a simple feature perturbation function in our last layer. It significantly boosts our model’s performance in both average geodesic errors and convergence speed. The remarkable result may enrich our understanding of geometric learning strategy designs and inspire new architectures in the geometric deep learning field.

Our extensive experimental results verify the effectiveness of our method. We hope this work contributes to shape correspondence research and sheds new light on general geometric deep learning mechanisms to maximize their learning power.

2. Related Works

Shape correspondence computation methods generally fall into three main categories: (1) traditional hand-crafted descriptors defined in spatial and spectral domains, (2) optimization-based methods rooted in the spectrum of shapes with functional maps being the pillar, and (3), more recent geometric deep learning techniques, notably graph

neural networks.

Starting with hand-crafted descriptors, they mainly fall into spatial and spectral methods. The spatial-based descriptors are usually based on statistics of locally defined features [27]. These methods, mostly, suffer from generalizability regarding changes in surface discretization and capturing global information of the shape [32]. This behavior is similar to spatial-based graph neural network frameworks that rely on learning small receptive fields around each vertex, making feature learning less globally aware. Spectral-based methods heavily rely on the LBO spectrum. Importantly, these intrinsic descriptors are isometry-invariant, making them robust to arbitrary spatial transformations. The methods proposed in [3, 8, 10] belong to the spectral-based category. Most recently, wavelet-based spectral-based descriptors have been proposed in [32, 21], exploiting a multi-scale setting to diversify the set of learned filters. However, our study does not rely on hand-crafted features as they are sub-optimal in learning-based models.

The second category of shape correspondence techniques is based on optimizing a dense map between shapes. Particularly, the vanguard of such methods is the successful functional map framework [26]. More recently, the authors in [13, 12, 23] proposed unsupervised and supervised learning schemes on top of the functional map for a more robust and accurate correspondence. In [13], authors proposed a U-Net structure to learn point-wise features to circumvent the pre-computation of hand-crafted features resulting in geometrically meaningful features independent of meshing structure. In [23], authors used a deep residual network model to enhance the functional maps in creating a soft correspondence map. One of the most recent works on using a functional map based on learned features is the work in [36]. They employed a continuous geodesic convolution in an end-to-end fashion. In another study [12] with functional map setting, authors introduced an unsupervised framework for learning orientation-preserving features for functional map computation, which is also robust to discretization changes in the mesh. The last category belongs to the state-of-the-art methods rooted in geometric deep learning, a new realm of deep learning in unstructured data like point clouds and meshes. In [5], authors proposed windowed Fourier transform in a supervised-learning setting to learn local shape descriptors for shape matching problems. Later, they used anisotropic LBO to learn multi-kernel filters to further improve the accuracy and diversity of learned features [21]. Though mainly used for node and graph classification settings, graph neural network models have also been used for 3D dense shape correspondence. Frameworks in [16, 25, 6] are among methods that hold state-of-the-art performance in graph classification, 3D shape segmentation, and shape matching. Another model based on anisotropic LBO is proposed in [22] and their experiment

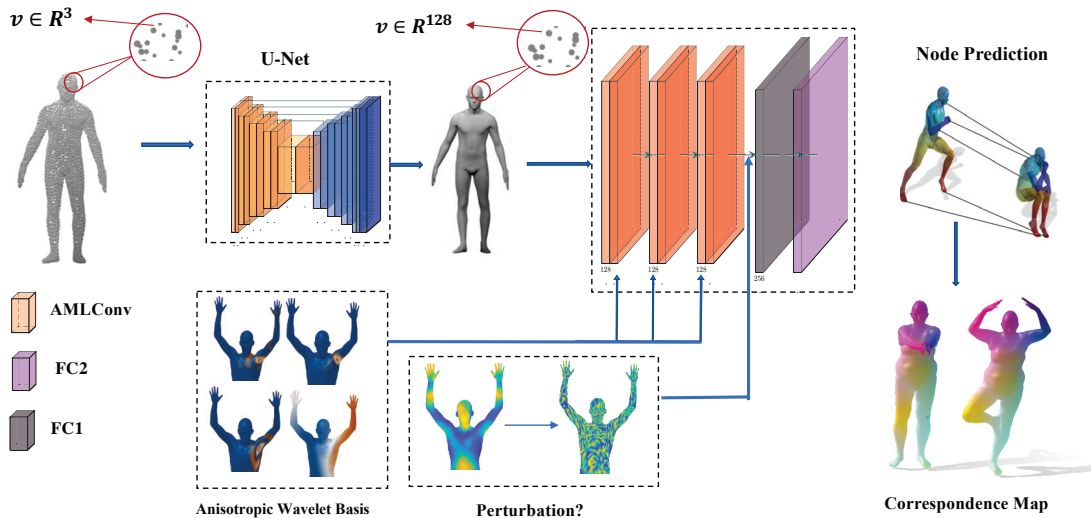


Figure 2. Our hybrid model at a glance. We use a U-Net feature extractor to learn geometric features from the raw point cloud; then we feed the geometric learned features and pre-computed wavelets $\Psi_{t,\alpha,\theta,j}$ to our proposed anisotropic multi-scale graph convolution layers. Our network, therefore, learns local and global features, which are also robust to discretization. In the last layer, after we add a perturbation layer, the network forces to learn more discriminative features, as illustrated qualitatively in Figure 3.

shows a boost in performance among other graph neural network-based models. Also, they showed with an explicit ablation study how the different number of anisotropic kernels, i.e., diffusion kernels in different angles, affect the model’s performance. In a different approach, the authors in [17] used a deep learning model to create local geometry images to learn descriptors based on local patches. Recently, [28] proposed a geometric deep learning model using the concept of heat diffusion for spatial communication among the mesh equipped with spatial gradient features for better robustness. This method was applied to many shape analysis applications like segmentation, shape correspondence, and classification. However, an integration of spatial and spectral-based feature learning in the deep learning framework has not been fully studied. Our work aims to fill the gap by exploring this direction.

3. Overview

The overview of our hybrid system is illustrated in Figure 2. We first introduce the geometrical foundations of building our anisotropic wavelet kernels to construct our graph convolution layers named **Anisotropic Multi-Scale Graph Convolution (AMLCONV)**. Then, we will delve into the initial feature extractor block using Kernel Point Convolution (KPCConv) [29], which is responsible for capturing geometry-aware features from the raw geometry input. Lastly, we will introduce a perturbation trick to improve the learning power in identifying discriminative features.

3.1. Geometrical background

We model a 3D shape surface as a compact two-Riemannian manifold X with set of vertices and edges as (V, E) , where $V = \{v_i | i = 1, \dots, N\}$ and $E = \{e_{ij} | i, j = 1, \dots, N\}$. We use $T_x X$ for each $x \in X$ to show the tangent plane of an arbitrary point on our shape X . The Riemannian metric is an inner product on this plane with the notation $\langle \cdot, \cdot \rangle_{T_x X} : T_x X \times T_x X \rightarrow \mathbb{R}$, denoting how manifold locally deviates from the plane. To calculate this deviation and bending from the plane, the second fundamental form, represented by a 2×2 matrix, is used, whose eigenvalues are called principal curvatures k_m and k_M . The corresponding eigenvectors form an orthogonal basis for T_X on x , denoting the directions of curvatures. Many shape analysis applications rely on the LBO, on account of the local geometry based on the tangential plane, alleviating the problem by converting the calculations from 3D to 2D. Specifically, in this study, as introduced in [1], we focus on the anisotropic version of LBO, namely, ALBO.

3.2. Anisotropic Laplace Beltrami Operator

Before delving into ALBO, we define the omnipresent LBO. From a signal processing perspective, the eigen-system of LBO acts as Fourier bases on signals. However, on a 2-Manifold, the bases are not periodic functions like sine and cosine. Although orthogonal functions like spherical harmonics have the same property as pseudo-spherical shapes, the LBO eigen-system is generally used for geometry processing applications on deformed shapes. The defi-

inition of LBO on shape X is

$$\Delta_X f(x) = -div_X(\nabla_X f(x)) \quad (1)$$

where Δ_X and $div_X(\cdot)$ are, respectively, intrinsic gradient and divergence of $f(x) \in L^2(x)$. Due to the positive semi-definiteness of the LBO operator, we have the real eigen-decomposition

$$\Delta_X \phi_k(x) = \lambda_k \phi_k(x), \quad (2)$$

with non-negative eigen-values $\{\lambda_i \leq \lambda_j | i < j\}$ and eigen-functions $\{\phi_i(x)\}$ forming an orthonormal basis for $L^2(x)$. Due to the highly expensive eigen decomposition of LBO, as it has been used in numerous applications [26, 23, 32], we use the first k eigen-values and eigen-functions to circumvent the curse of costly computation.

The LBO is both intrinsic and isotropic, meaning it is invariant under isometric deformation and is not sensitive to directions. However, by making it dependent on direction θ , we can exploit its sensitiveness among different deformations in different directions. In other words, we try to embed extrinsic geometry into intrinsic by introducing ALBO. Now, the new equation is

$$\Delta_X f(x) = -div_X(D(x)\nabla_X f(x)), \quad (3)$$

where D is a 2*2 matrix named anisotropic tensor or thermal conductivity tensor acting on the intrinsic gradient. By introducing D the diffusion can be controlled in both direction and magnitude but to have a standard direction among all points, researchers suggested anisotropy along maximum curvature resulting in a matrix

$$D_\alpha(x) = \begin{pmatrix} \frac{1}{1+\alpha} & \\ & 1 \end{pmatrix}, \quad (4)$$

where α controls the anisotropy level, instead of considering one direction, we can consider multiple directions for our kernel [1]. The definition of new anisotropic tensor D is as follows

$$D_{\alpha\theta}(x) = R_\theta \begin{pmatrix} \frac{1}{1+\alpha} & \\ & 1 \end{pmatrix} R_\theta^T, \quad (5)$$

By setting the reference angle $\theta = 0$ to principal curvature direction, we can form a set of rotations in $[0, 2\pi]$. Now, the eigen decomposition of the new LBO in Eq. 2 becomes

$$\Delta_{\alpha\theta} \phi_{\alpha\theta,k}(x) = \lambda_{\alpha\theta,k} \phi_{\alpha\theta,k}(x), \quad (6)$$

where our ALBO still satisfies the semi-positive definiteness and forms a set of orthogonal eigen-functions $\phi_{\alpha\theta,k}(x)$. As mentioned for the LBO, here for ALBO we compute the first k eigen-values and eigen-functions.

3.3. 3D Mesh Wavelets

A graph wavelet function is locally defined on vertex v denoted as $\Psi_{t,v}$ per scale t . Wavelet Graph Neural Network models employed in [9, 34] use just one scaling function instead of considering all the scales for t , which negates the spirit of multi-scale analysis [34, 9]. We need a kernel function g to sweep the frequencies in the spectral domain for different band-passes. Mexican Hat and Meyer wavelets are among the many wavelet functions that frequently appear in the literature. Unlike graphs, we deal with a discretized two-manifold with a metric, and Graph Laplacian no longer captures the geometry of the mesh. We rather use the introduced ALBO in the previous section. Using the area matrix A for X , we define the Voronoi area $a(v)$ at vertex v , $\phi_j(v)$ the eigen-function corresponding to vertex v to define the localized graph wavelet at vertex v

$$\Psi_{\alpha\theta,t,v} = \sum_{j=0}^{N-1} a(v)g(t\lambda_{\alpha\theta,j})\phi_{\alpha\theta,j}(v)\phi_{\alpha\theta,j}. \quad (7)$$

The low pass filter representing the wavelet basis is called the scaling function and is defined based on filter $h(x)$.

$$\xi_{\alpha\theta,v} = \sum_{j=0}^{N-1} a(v)h(\lambda_{\alpha\theta,j})\phi_{\alpha\theta,j}(v)\phi_{\alpha\theta,j}. \quad (8)$$

To find the wavelet coefficients for both scaling and wavelet functions, we have to project the given function f onto the wavelet basis. Coefficients are for spectral wavelet, and its scaling function is derived by

$$W_f(\alpha\theta, t, v) = \sum_{j=0}^{N-1} a(v)g(t\lambda_{\alpha\theta,j})\sigma_{\alpha\theta,j}\phi_{\alpha\theta,j}(v), \quad (9)$$

$$S_f(\alpha\theta, v) = \sum_{j=0}^{N-1} a(v)h(\lambda_{\alpha\theta,j})\sigma_{\alpha\theta,j}\phi_{\alpha\theta,j}(v). \quad (10)$$

If h and g are chosen to be a Parseval frame, we can reconstruct the signal using the equation below [32]:

$$f = \sum_{j=0}^K \sum_v a(v)^{-1} W_f(\alpha\theta, t_j, v) \Psi_{\alpha\theta,t_j,v}. \quad (11)$$

3.4. Feature Extractor

This block is responsible for learning the geometric characteristics of the mesh as the input for the AMLCONV layers. We use the same point cloud learning method KPConv [29] with a U-net architecture that was primarily used for the segmentation task in the main study. Using the learned features helps us with more informative and data-centric features to build our multi-scale network on top of it.

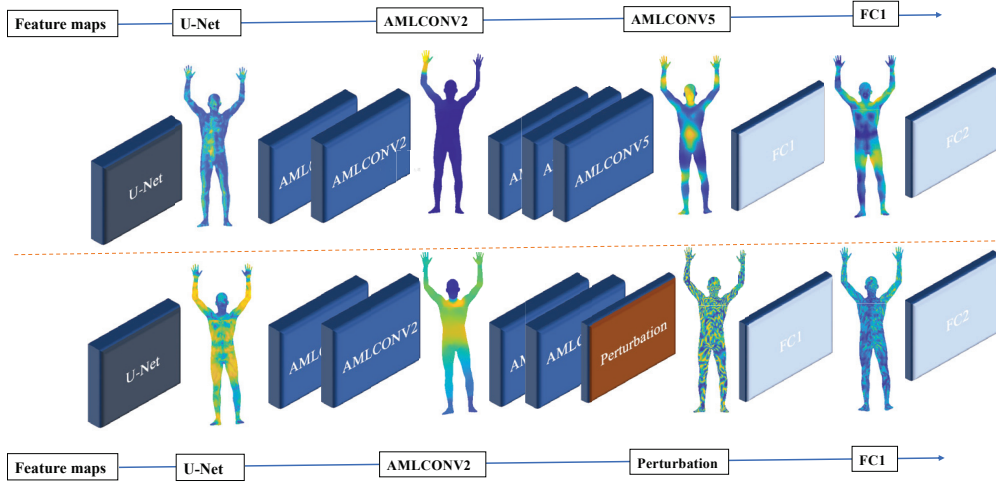


Figure 3. Intermediate feature maps comparison between our vanilla and perturbed version model. Top: Network with no perturbation layer. Bottom: Network with a perturbed feature map as the last layer before fully connected layers. As shown in the perturbed version, the feature maps of previous layers become more semantically meaningful and distinctive.

Unlike the previous works [32, 26] using hand-crafted features, the input to the feature extractor network is a point cloud with just the 3D coordinates as input. The U-Net architecture is composed of 4 down-sampling and four up-sampling layers equipped with convolutional layers, i.e., KPConv. For more information on the kernel used in KPConv, we refer readers to [29]. As for pooling, grid sampling is used for flexibility over density control at each pooling layer.

3.5. Convolutional Layer Setup

The feature extractor block outputs D' dimensional learned features to the main multi-scale layer, which is a graph convolutional neural network architecture. Therefore, considering the input signal x to our multi-scale layer $\in R^{N \times D'}$ indicating N vertices and D' dimensional signal on each point. The multi-scale layer, as witnessed in [11], convolves the signal x with filter g in the spectral domain as follows

$$x * g = \Phi((\Phi^T g) \odot (\Phi^T x)) = \Phi W \Phi^T x, \quad (12)$$

where \odot is the element-wise product and Φ is the k first eigen vector matrix $\Phi \in R^{N \times k}$, and w is the filter represented in the frequency domain. Mainly the focus of studies is on how to design the filter w . In ChebyNet [11], weighted sum of powers of eigen-value matrix is used, circumventing the use of eigen decomposition. Specifically, they used polynomial powers of eigen-value matrix Λ which can be computed recursively and induce k hop receptive field kernel in spatial domain with approximation order k

$$w = \sum_{j=0}^{k-1} \theta_j (\Lambda)^j, \quad (13)$$

where the diagonal matrix Λ is of size $k * k$, and the convolution equation in Eq. 2 becomes

$$x_{out} = g * x = \sum_{j=0}^{k-1} \theta_j T_j(L)x, \quad (14)$$

where θ_j are the set of learnable parameters in the network and $T_j(L)$ is the j order Chebyshev Polynomials. As a result, we have an m-localized receptive field around each vertex v . There are two drawbacks to such a filter. One is that this filter is an all-pass filter, and secondly, it is not resolution independent as it overfits to the mesh connectivity.

Here we use w in terms of wavelet filter basis in the spectral domain rather than any polynomials mentioned above. This is beneficial to track different frequency band-passes and not just use a low-pass filter resulting in over-smoothness of the signals on the vertices

$$W = \sum_{j=0}^k \theta_j f_{t_j}(\Lambda), \quad (15)$$

and the corresponding convolution is

$$x_{out} = g * x = \sum_{j=0}^k \theta_j \Psi_{t_j}^T x, \quad (16)$$

where the wavelet basis matrices at each scale t_j , $\Psi_{t_j} \in R^{N \times N}$, are used for the convolution. Since the magnitude of different scales of Ψ varies, we use normalization for a better learning process. We perform L_1 normalization on the wavelet functions. We show the normalized wavelet ba-

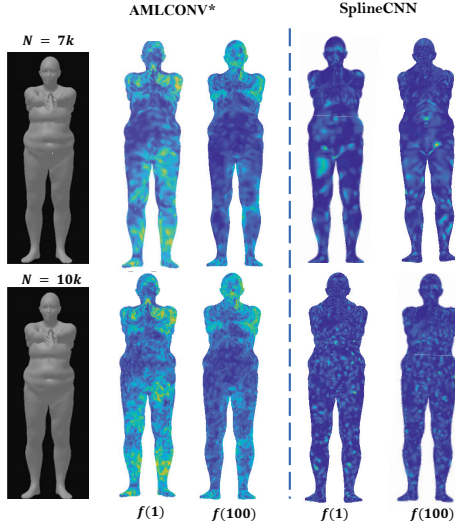


Figure 4. Feature map (f) consistency among two different resolutions of the same shape. $f(x)$ shows the x -dimension of feature map. As illustrated, features learned by our model are semantically and visually more consistent in both resolutions, whereas in splineCNN, there is no consistency and smoothness

sis functions as $\bar{\Psi}$, and the new convolution becomes [32]

$$x_{out} = g * x = \sum_{j=0}^k \theta_j \bar{\Psi}_{t_j}^T x. \quad (17)$$

Consequently, the multi-scale layer, including a custom batch normalization “Norm” and activation function “SELU” with isotropic LBO, is as follows

$$\text{Norm} \left(\text{SELU} \left(\sum_{j=0}^k \theta_j \bar{\Psi}_{t_j}^T x \right) \right). \quad (18)$$

Finally, the anisotropic multi-scale convolution layer based on the definition of anisotropic LBO in 1.5 and 1.6 by setting α as a hyper-parameter of our network is

$$\text{Norm} \left(\text{SELU} \left(\sum_{\theta} \sum_{j=0}^k \theta_{\alpha\theta,j} \bar{\Psi}_{t_{\alpha\theta,j}}^T x \right) \right). \quad (19)$$

3.6. A Perturbation Layer

In this study, we discovered that using a perturbation layer at the end of the network can accelerate training convergence and performance by a large margin. For the last layer, instead of using Eq. 20, we can use a fixed shuffling of features learned in the previous layer x_{n-1} . This permutation of features acts as a perturbation of the whole feature map. Surprisingly, this forces the entire network, including

Table 1. Average geodesic error on all pairs of (15×14) shapes in FAUST test dataset. The results, for the sake of fair comparison, excluding ours, are derived from [32, 28]. All average geodesic errors are reported $\times 100$. In the table, “Ours” refer to our vanilla model, and “ours*” refer to our model using perturbation layer. Terms WEDS and XYZ refer to “Wavelet Energy Decomposition Signature” [32], and “3D xyz” coordinate of vertices.

Network	Descriptor	Resolution	
		7k	10k
OSD [24]	1	39.8	45.7
SplineCNN [16]	1	27.6	52.4
ChebyGCN [32]	WEDS	0.6	55.1
Geo-Based [31]	WEDS	14.7	23.9
DiffusionNet [28]	XYZ	0.33	0.83
ACSCNN [28, 22]	1	0.05	41.1
MGCN [32]	WEDS	0.8	2.6
U-Net (KPconv [29])	XYZ	2.8	5.7
Ours	XYZ	0.8	2.4
Ours*	XYZ	0.12	0.85

Table 2. Average geodesic error on all test pairs (10×9) of shapes in SCAPE test dataset. The results, for the sake of fair comparison, excluding ours are derived from [32, 28]. All average geodesic errors are reported $\times 100$. In the table, “Ours” refer to our vanilla model, and “ours*” refer to our model using perturbation layer.

Network	Descriptor	Resolution (5k)
OSD [24]	1	25.9
SplineCNN [32, 16]	1	29.7
ChebyGCN [32]	WEDS	6.8
Geo-Based [32]	WEDS	16.3
DiffusionNet [28]	XYZ	3.0
MGCN [32]	WEDS	4.8
Ours	XYZ	3.4
Ours*	XYZ	1.2

the primary U-Net block, to learn more discriminative and robust features. The formulation of one possible perturbation is as follows:

$$\text{Norm} (\text{SELU} (\theta * x_{perm})), \quad (20)$$

The perturbation term in the above equation must be fixed for all batches. For clarification, if this perturbation changes the node feature i and j for the first batch, it must be done for the remaining batches. Note that we still have learnable parameters θ which control this perturbation’s weight for each feature. We illustrate a shape’s intermediate feature maps after training on both training schemes, with and without perturbation, in Figure 3. Although the final feature map of the perturbed network is less smooth, it clearly shows how this perturbation results in a semantically discriminative feature map in previous layers. In our study, we simply use a matrix transpose of feature maps to shuffle the features among nodes.



Figure 5. Visualization of different anisotropic wavelet filters on the same mesh with the same selected node on the right hand. A sequence of results from the low-pass, band-pass, and high-pass filters with different anisotropy levels are illustrated. It is conceivable from these kernel maps that wavelet filters act both locally and globally for robust feature learning.

3.7. Network Details

The KPconv U-Net network is a residual-based point cloud convolutional network, with four convolutional layers with leaky linear activation functions and poolings and successive up-sampling layers to the final 128 feature dimension. The step-size of sub-sampling is set to 0.3 as defined in [13]. Next, we use four layers of AMLCONV with input and out feature dimension of 128, with a fully connected layer and a softmax at the end to predict the labels of each vertex for shape matching. We use scaled exponential linear units “SELU” activation function for our AMLCONV layers.

As for the wavelet function candidate, we use the Mexican Hat filter following the same setting in [32]. The function g is as follows:

$$x_{out} = g * x = \sum_{j=0}^k \theta_j \Psi_{t_j}^T x, \quad (21)$$

We use 16 different filters, including 4 different anisotropy directions, and for each, we compute 4 different wavelet functions. For the LBO eigen-basis, we computed the first 200 eigen-values and eigen-vectors to compute our wavelet filters accordingly. In Figure 5 different filters, from the low-pass to the high-pass, and different diffusivity directions are illustrated for a specific node on a shape.

3.8. Experimental Results

Datasets. To test our network, we apply it to two publicly available datasets, FAUST [4] and SCAPE, [2] to eval-

uate our descriptor learning performance for dense shape correspondence in *near-isometric* shapes. All the datasets are provided with ground-truth labels for shape correspondence and consist of deformable shapes of human poses. We also use another re-meshed version of FAUST [31, 32] to evaluate the robustness of our model in terms of re-meshing and variability in size. The re-meshed FAUST dataset maintains the coordinates of the original $7k$ resolution dataset. The resolution we choose is $10k$ for testing on the trained model with $7k$ resolution. A total of 15 shapes are used for the test set in FAUST dataset. For the SCAPE dataset, we only use the simplified re-meshed version based on the setting in [35] containing approximately $5k$ vertices. This re-meshing, however, changes the position and triangulation [32]. For the sake of fair comparison, we report the results based on the main benchmark study in [32]. In SCAPE dataset, we used 10 shapes for the test set.

Evaluation criteria. Per [21], we take the Princeton benchmark protocol [19] to evaluate the correspondence accuracy based on the percentage of matches that are at most r -distant from the ground-truth map. As commonly used in the literature [32], we call it Cumulative Geodesic Error (CGE). The other criterion we use to measure the performance is the average geodesic errors in all test pairs. To find the predicted point in each test pair of the source-target shapes, we use nearest-neighbor searching using the L2 distance in the feature space.

Experimental setup All experiments were done on a V100 Nvidia GPU, 32GB of RAM using Pytorch-Geometric packages [15] (Python 3.7). The training was done in 200 epochs for the vanilla network and 50 epochs with our network with the perturbation layer. In both settings, the training was done with the ADAM optimizer [20] with a learning rate of 0.001 and weight decay of 0.0001.

Results Quantitative results based on the average geodesic error on the FAUST dataset are tabulated in Table 1. For the sake of fair comparison, the results with the baseline selected in [32] and [28] are reported here. Graph neural network approaches that produce state-of-the-results in terms of minimum geodesic error like ACSCNN, ChebyNet, and SplineCNN show superior results in the same resolution. Our results using the perturbation layer beside ACSCNN have the best CGE. However, our model, unlike ACSCNN and SplineCNN, significantly outperforms other models, except DiffusionNet, after testing on the re-meshed dataset. DiffusionNet and our model are within the same margin of error when testing on different mesh sizes. In Figure 6 the CGE of different methods is depicted for the original FAUST dataset ($7k$). The results after testing on $10k$ meshes are illustrated in Figure 7.

To show the consistency of feature maps among different resolutions of shapes, in Figure 4, we compare our results with SplineCNN. There is a high level of consistency

among the feature maps in our model while tested on different shape resolutions.

As for the SCAPE dataset, the quantitative results based on average geodesic error are tabulated in Table 2. The results show the superior performance of our model over other benchmarks. Even our vanilla network improves upon MGCN, the closest benchmark to ours. The vanilla network is also within the same margin of error as DiffusionNet in functional map setup. The CGE plots are depicted in Figure 6, showing the high number of perfect matching points based on our model with the perturbation layer. The convergence in our model with the perturbation layer is reached within 50 epochs, making it 4 times faster than the vanilla network and MGCN [32] with 200 epochs with the same time complexity.

Ablation study The important part of our model is the contribution of each part and parameters to the final learned descriptors. Since we use KPconv as the primary network block, we test a U-Net network followed by two fully connected layers to measure the performance of the U-Net feature extractor solely. Moreover, we test our model based on two training paradigms, with and without proposed perturbation, to see how feature maps would look like for better comparison. Lastly, since the number of wavelet filters would directly impact the learned descriptor, we also test our model based on a different number of anisotropic filters.

We show our ablation study results in Figure 8. As the number of filters increases, with different anisotropy levels and wavelet basis, the CGE improves. Using only the U-Net block for learning descriptor on FAUST dataset shows inferior results as opposed to our hybrid model. The average geodesic error is 2.8 which is worse than the hybrid model.

Limitations In our study, although the perturbation layer improves the model’s performance, it remains to be meticulously studied in the future for the potential theoretical reason behind this behavior. There can be more efficient perturbation other than the shuffling feature map that we did to reduce the geodesic error.

4. Conclusion and Future Works

In this study, we propose a hybrid end-to-end model taking advantage of the merits of spectral wavelet filters and geometry-aware spatial point cloud convolution model, KPconv, to overcome the overs-smoothing nature of most graph neural network models. Using the U-Net model, the learned geometric features are fed into our AMLCONV layers to learn robust descriptors using different band-pass filters. To further diversify the set of filters, we adopt the anisotropic LBO to capture directionally sensitive information. Later, we propose and depict how a simple perturbation of the feature map in the last layer of the network significantly improves the performance and convergence speed.

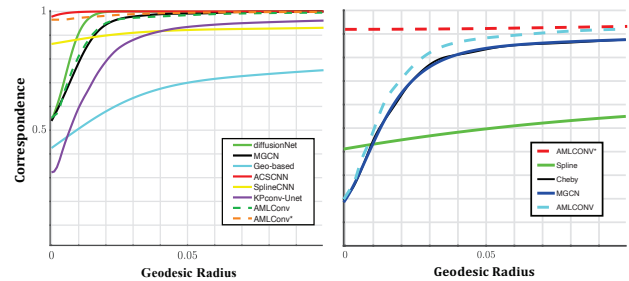


Figure 6. Left: CGE of FAUST dataset. Right: CGE of SCAPE dataset. Dashed lines indicate the resolution of the test set with our models using both vanilla and perturbed feature versions. As it is depicted, our perturbed version results in state-of-the-art outcomes in both datasets.

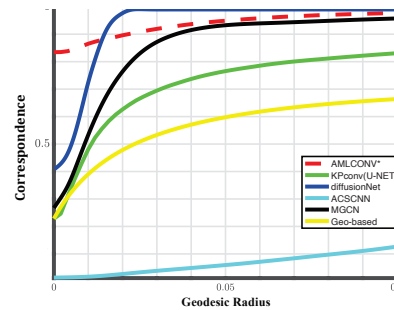


Figure 7. CGE of FAUST dataset after training on original meshes and testing 10k re-meshed shapes. Our model still results in state-of-the-art CGE besides DiffusionNet.

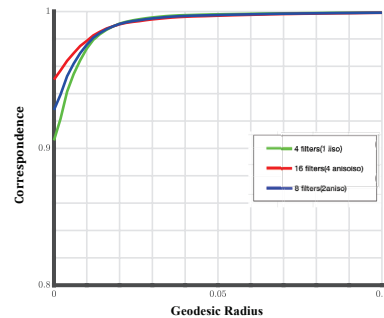


Figure 8. Direct CGE in the FAUST dataset with a different number of filters. We test three different settings, one with 4 wavelet filters using LBO. Second, with 8 different filters, with anisotropy of degree two. In the third setting, we use all 16 filters with 4 anisotropy levels. Results clearly show that more filters results in better performance. For more ablation results on number of filters please see [21, 32].

In the future, we will focus on the theoretical foundation of feature perturbation and explore how it can be generalized to, possibly, other existing models.

Acknowledgment: The research is partly supported by NIH (R21AG065942, R01EY032125, R01EB025032, and R01DE030286).

References

- [1] Mathieu Andreux, Emanuele Rodola, Mathieu Aubry, and Daniel Cremers. Anisotropic laplace-beltrami operators for shape analysis. In *European Conference on Computer Vision*, pages 299–312. Springer, 2014.
- [2] Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis. Scape: shape completion and animation of people. In *ACM SIGGRAPH 2005 Papers*, pages 408–416. 2005.
- [3] Mathieu Aubry, Ulrich Schlickewei, and Daniel Cremers. The wave kernel signature: A quantum mechanical approach to shape analysis. In *2011 IEEE international conference on computer vision workshops (ICCV workshops)*, pages 1626–1633. IEEE, 2011.
- [4] Federica Bogo, Javier Romero, Matthew Loper, and Michael J Black. Faust: Dataset and evaluation for 3d mesh registration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3794–3801, 2014.
- [5] Davide Boscaini, Jonathan Masci, Simone Melzi, Michael M Bronstein, Umberto Castellani, and Pierre Vandergheynst. Learning class-specific descriptors for deformable shapes using localized spectral convolutional networks. In *Computer Graphics Forum*, volume 34, pages 13–23. Wiley Online Library, 2015.
- [6] 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.
- [7] Michael M Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. Geometric deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42, 2017.
- [8] 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.
- [9] Heng Chang, Yu Rong, Tingyang Xu, Wenbing Huang, Somayeh Sojoudi, Junzhou Huang, and Wenwu Zhu. Spectral graph attention network with fast eigen-approximation. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 2905–2909, 2021.
- [10] Abhijit J Chaudhari, Richard M Leahy, Barton L Wise, Nancy E Lane, Ramsey D Badawi, and Anand A Joshi. Global point signature for shape analysis of carpal bones. *Physics in Medicine & Biology*, 59(4):961, 2014.
- [11] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. *Advances in neural information processing systems*, 29:3844–3852, 2016.
- [12] Nicolas Donati, Etienne Corman, and Maks Ovsjanikov. Deep orientation-aware functional maps: Tackling symmetry issues in shape matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 742–751, 2022.
- [13] Nicolas Donati, Abhishek Sharma, and Maks Ovsjanikov. Deep geometric functional maps: Robust feature learning for shape correspondence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8592–8601, 2020.
- [14] Marvin Eisenberger, Aysim Toker, Laura Leal-Taixé, and Daniel Cremers. Deep shells: Unsupervised shape correspondence with optimal transport. *Advances in Neural Information Processing Systems*, 33:10491–10502, 2020.
- [15] Matthias Fey and Jan Eric Lenssen. Fast graph representation learning with pytorch geometric. *arXiv preprint arXiv:1903.02428*, 2019.
- [16] Matthias Fey, Jan Eric Lenssen, Frank Weichert, and Heinrich Müller. Splinecnn: Fast geometric deep learning with continuous b-spline kernels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 869–877, 2018.
- [17] Jianwei Guo, Hanyu Wang, Zhanglin Cheng, Xiaopeng Zhang, and Dong-Ming Yan. Learning local shape descriptors for computing non-rigid dense correspondence. *Computational Visual Media*, 6(1):95–112, 2020.
- [18] 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.
- [19] Vladimir G Kim, Yaron Lipman, and Thomas Funkhouser. Blended intrinsic maps. *ACM transactions on graphics (TOG)*, 30(4):1–12, 2011.
- [20] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [21] Qinsong Li, Ling Hu, Shengjun Liu, Dangfu Yang, and Xinru Liu. Anisotropic spectral manifold wavelet descriptor. In *Computer Graphics Forum*, volume 40, pages 81–96. Wiley Online Library, 2021.
- [22] Qinsong Li, Shengjun Liu, Ling Hu, and Xinru Liu. Shape correspondence using anisotropic chebyshev spectral cnns. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14658–14667, 2020.
- [23] 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.
- [24] Roei Litman and Alexander M Bronstein. Learning spectral descriptors for deformable shape correspondence. *IEEE transactions on pattern analysis and machine intelligence*, 36(1):171–180, 2013.
- [25] Federico Monti, Davide Boscaini, Jonathan Masci, Emanuele Rodola, Jan Svoboda, and Michael M Bronstein. Geometric deep learning on graphs and manifolds using mixture model cnns. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5115–5124, 2017.

- [26] Maks Ovsjanikov, Mirela Ben-Chen, Justin Solomon, Adrian Butscher, and Leonidas Guibas. Functional maps: a flexible representation of maps between shapes. *ACM Transactions on Graphics (TOG)*, 31(4):1–11, 2012.
- [27] Samuele Salti, Federico Tombari, and Luigi Di Stefano. Shot: Unique signatures of histograms for surface and texture description. *Computer Vision and Image Understanding*, 125:251–264, 2014.
- [28] Nicholas Sharp, Souhaib Attaiki, Keenan Crane, and Maks Ovsjanikov. Diffusionnet: Discretization agnostic learning on surfaces. *ACM Transactions on Graphics (TOG)*, 41(3):1–16, 2022.
- [29] Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J Guibas. Kpconv: Flexible and deformable convolution for point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6411–6420, 2019.
- [30] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.
- [31] Yiqun Wang, Jianwei Guo, Dong-Ming Yan, Kai Wang, and Xiaopeng Zhang. A robust local spectral descriptor for matching non-rigid shapes with incompatible shape structures. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6231–6240, 2019.
- [32] Yiqun Wang, Jing Ren, Dong-Ming Yan, Jianwei Guo, Xiaopeng Zhang, and Peter Wonka. Mgcn: descriptor learning using multiscale gcns. *ACM Transactions on Graphics (TOG)*, 39(4):122–1, 2020.
- [33] 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.
- [34] Bingbing Xu, Huawei Shen, Qi Cao, Yunqi Qiu, and Xueqi Cheng. Graph wavelet neural network. *arXiv preprint arXiv:1904.07785*, 2019.
- [35] Dong-Ming Yan, Guanbo Bao, Xiaopeng Zhang, and Peter Wonka. Low-resolution remeshing using the localized restricted voronoi diagram. *IEEE transactions on visualization and computer graphics*, 20(10):1418–1427, 2014.
- [36] Zhangsihao Yang, Or Litany, Tolga Birdal, Srinath Sridhar, and Leonidas Guibas. Continuous geodesic convolutions for learning on 3d shapes. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 134–144, 2021.
- [37] Yiming Zeng, Yue Qian, Zhiyu Zhu, Junhui Hou, Hui Yuan, and Ying He. Cornet3d: unsupervised end-to-end learning of dense correspondence for 3d point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6052–6061, 2021.