

RFMNet: Robust Deep Functional Maps for Unsupervised Non-rigid Shape Correspondence

Ling Hu¹, Qinsong Li^{2,3}, Shengjun Liu^{*2}, Dong-Ming Yan⁴, Haojun Xu², Xinru Liu²

¹School of Mathematics and Statistics, Hunan First Normal University

²Institute of Engineering Modeling and Scientific Computing, Central South University

³Big Data Institute, Central South University

⁴National Laboratory of Pattern Recognition, Chinese Academy of Sciences

Abstract

In traditional deep functional maps for non-rigid shape correspondence, estimating a functional map including high-frequency information requires enough linearly independent features via the least square method, which is prone to be violated in practice, especially at an early stage of training, or costly post-processing, e.g. ZoomOut. In this paper, we introduce a novel strategy to compute the functional map in the deep functional map framework, which jointly considers training stability and more geometric shape features than previous works. We directly first produce a pointwise map by resorting to optimal transport and then convert it to an initial functional map. Such a mechanism mitigates the requirements for the descriptor and avoids the training instabilities resulting from the least square solver. Benefitting from the novel strategy, we successfully integrate a state-of-the-art geometric regularization for further optimizing the functional map, which substantially filters the initial functional map. We show our novel computing functional map module brings more stable training even under encoding the functional map with high-frequency information and faster convergence speed. Considering the pointwise and functional maps, an unsupervised loss is presented for penalizing the correspondence distortion of Delta functions between shapes. To catch discretization-resistant and orientation-aware shape features with our network, we utilize DiffusionNet as a feature extractor. Experimental results demonstrate our apparent superiority in correspondence quality and generalization across various shape discretizations and different datasets compared to the state-of-the-art learning methods.

Keywords: *shape correspondence, functional maps, unsupervised learning, optimal transport*

*Corresponding authors: shjliu.cg@csu.edu.cn (S.Liu).

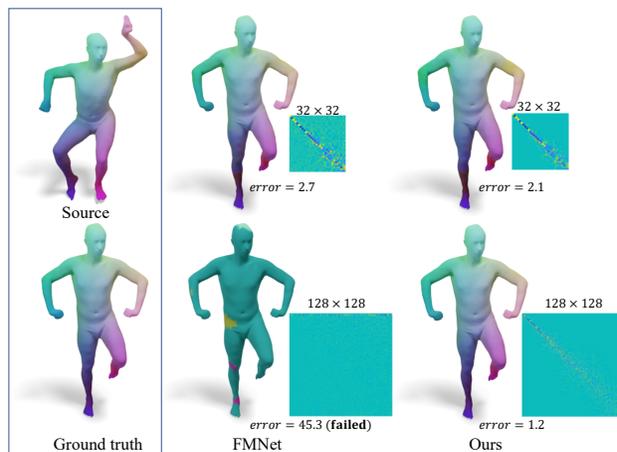


Figure 1. Our approach avoids the training instabilities resulting from the least squares solver compared with traditional deep functional maps (FMNet) [21] that enable our approach to directly recover high-frequency information. Meanwhile, integrating a state-of-the-art geometric regularization makes our approach more accurate than FMNet in the same settings.

1. Introduction

Computing shape correspondence is a fundamental problem in computer graphics, vision, and pattern recognition, with extensive applications in shape comparison, texture transfer, and shape interpolation, among many others [42]. It is also a considerably challenging problem, especially when addressing the cases of non-rigid motion, scanning noise (e.g., partiality and topological noise), different resolutions and connectivity between source and target shapes.

In recent years, various approaches have been proposed for computing correspondence [14, 12, 40]. One influential framework is called the functional map [28, 32, 31, 25, 17, 30], which aims to determine a functional map operator that maps square-integrable functions defined on shapes rather than mapping points directly. A desirable pointwise correspondence finally can be recovered from the mapping of

Delta functions. It makes computing correspondence, originally a quite complicated problem boils down to a simple and efficient linear algebraic problem. However, these axiomatic functional maps heavily rely on the quality of hand-crafted shape features and most of them make restrictive assumptions about the discretization, topology, or morphology of the considered shapes.

To these problems, motivated by great capabilities of deep learning in various fields such as image analysis, a recent line of researches aim at integrating deep learning techniques with axiomatic functional maps. The pioneering work FMNet [21] learned optimal descriptor transformations to produce desired maps as close as possible to ground truth correspondence. It has significantly enhanced the learning process as elegant geometric priors were considered. A series of follow-up works generally utilized the main structure proposed by FMNet [21] while focusing on improving its certain blocks. Some efforts were paid to improve shape feature learning or functional map computing schemes, like [10]. Another part of the works aims to design appropriate loss functions to penalize computed maps, permitting learning from supervised [21, 10] to unsupervised [16, 33] manner. More recent researches are devoted to addressing particularly challenging shapes such as with large partialities [2] or symmetry issues [9]. Although these approaches have achieved significant improvements, they still suffer from severe drawbacks w.r.t. correspondence quality and efficiency. Computation instabilities are prone to appear when solving systems of linear equations to estimate the functional map. Moreover, as integrating differential geometric constraints into the learning is considerably challenging, most recent computations of functional maps just depend on descriptor preservation. Such constraints still insufficiently encode the properties of the underlying map. This lacking of geometric regularization greatly affects the performance of the network.

In this paper, we propose a novel unsupervised deep functional map for shape correspondence, jointly considering training stability and more informative geometric features from three major components of the neural network. First, we use the DiffusionNet [37], a recent geometrically informative convolutional neural network to learn discretization-robust and orientation-aware shape features. It benefits the robustness of our network in terms of shape discretizations and disambiguating bilateral symmetry. Second, instead of using the learned descriptors to estimate the functional maps in previous approaches, we use them to directly produce an initial pointwise map, by resorting to an optimal transport problem with an entropic regularization term [8] and then get a functional map using a simple relation between these two maps. Such a mechanism efficiently avoids the instabilities resulting from solving systems of linear equations in earlier works. More importantly, with

these maps, we successfully integrate a simple but powerful wavelet regularization to optimize the computed functional map, motivated from a current state-of-the-art axiomatic functional map [17], where multiscale spectral manifold wavelets should be correspondingly preserved at each scale by the functional map. Compared to earlier works mainly dependent on descriptor preservation, much more geometrical features of the maps are considered in our computation by this wavelet-preserving constraint. Remarkably, the optimization can be solved efficiently only by referring to matrix multiplications. Finally, we are permitted to design a quite simple unsupervised loss, only penalizing the correspondence distortion of Delta functions between shapes. This yields great efficiency improvements to existing works, where either complex structural penalties on maps or expensive computations such as estimating geodesic distance are employed to make the maps satisfy certain criteria. A wide variety of experiments in challenging datasets demonstrate our superiorities to the state-of-the-arts, in correspondence quality and computing efficiency, as well as generalization across different datasets and shape discretizations.

The main contributions of this work are:

- In the deep functional map framework, we propose a novel strategy to compute the functional map and integrate a state-of-the-art geometric regularization to further optimize it.
- We show that, in comparison to existing techniques, our solution avoids the training instabilities resulting from the least square solver, which makes the network compute more stable and converge faster during the training stage and can directly recover high-frequency information without any post-processing.
- Using a simple unsupervised loss and DiffusionNet as a feature extractor, we achieve significant improvements in correspondence quality and generalization power across shape discretizations and datasets on the full shape matching application. Additionally, we can address partial shape matching by replacing DiffusionNet with a suitable feature extractor.

2. Related Work

We refer readers to the survey [34] for an in-depth view of shape correspondence. Below we review the methods most related to ours.

Axiomatic Functional Maps. Functional maps were originally introduced by *Ovsjanikov et.al.* [28] and have been extended significantly in the follow-up works [32, 31, 25, 17, 30] (see an overview in [29]). These methods are based on the notion that it is often easier to obtain correspondences between functions, rather than points, by first

using a reduced functional basis and second by formulating many linear constraints that allow recovering the functional map by solving a least-squares system. Finally, a good pointwise correspondence can be extracted from the functional map. Compared to point-based methods, the functional map framework has the particular advantage of being flexible, as it allows to easily incorporating of constraints such as the preservation of geometric quantities (descriptors) into the computation. To this end, several works try to formulate more powerful constraints in functional maps to achieve desirable properties, including the partial functional map [32], more powerful descriptor constraints via commutativity [27], continuous and orientation-preserving correspondences [31], among many others. Differing from these works commonly relying on a large set of high-quality descriptor functions whose obtaining is often challenging, the work [17] instead uses the multiscale spectral manifold wavelet preservation as new descriptor constraints to optimize the functional map. Such work could encode multiscale geometric features into the mapping and keep high computation efficiency, leading the maps closer to an isometric one. This greatly motivates us to integrate wavelet preservation into deep learning.

Deep Functional Maps. The combination of learnable feature extractors with traditional functional maps was pioneered by [21]. Their architecture used 7 residual multilayer perceptron (MLP) layers to optimize nonlinear transformations of the SHOT descriptor [35] to obtain a map as close as possible to the given ground truth. The follow-up work [10] still used the structure of FMNet [21] while extracting shape features directly from vertex coordinates of shapes and added regular term constraints to the functional maps to obtain more effective correspondence. However, these supervised methods spent expensive to get data with good labels. As a sequence, it is a natural way to explore unsupervised learning approaches, less depending on labelled data. Existing unsupervised methods [16, 33] have proposed different unsupervised loss functions. *Halimi et al.* [16] used unsupervised loss functions based on geodesic distances. Unfortunately, the geodesic distance matrix has costly storage and low calculation efficiency. [33] used a fully spectral-based unsupervised loss that aggregated several structural penalties on the functional map. However, these constraints still are insufficient to encode the properties of the maps, which leads to overall drops in robustness and performance. Utilizing the loss function of [33], [36] designed a weakly supervised learning framework that used manually aligned vertex coordinates instead of SHOT descriptors as inputs of the network. *Marvin et al.* [13] replaced the functional map layer with an iterative multiscale correspondence refinement layer based on optimal transport at cost of efficiency. *Attaiki et al.* [2] paid efforts to address

partial shape matching, a quite challenging correspondence task. *Donati et al.* [9] used complex functional maps to make their approach orientation-aware. However, the existing methods compute functional maps via solving systems of linear equations which is prone to error or computing instability during the training stage [10], especially at the beginning of training. Because the parameters of the network are randomly initialized and can't produce good descriptors to guarantee the unique solution to a system of linear equations. In addition, the existing unsupervised methods still have a large room for performance and efficiency improvements, as discussed in Section 4.1.

Convolutional Neural Networks (CNN). Except (deep) functional map based methods, there are plenty of works to design convolutional operator to extract shape features and build CNN to address shape correspondence [24, 4, 26, 20, 43, 15, 22]. They cast the shape correspondence into a pointwise classification problem and train the network using a cross-entropy loss function. Compared with the deep functional map based methods, the cross-entropy loss does not capture shape geometry resulting in inconsistent correspondence results. However, these convolutional neural networks can be used as the backbone for feature extraction in the deep functional map framework. We review some representative methods mainly for non-rigid shape correspondence. Please refer to [6, 44] for a comprehensive description of geometric deep learning.

The first intrinsic CNN was introduced by [24], which uses a local geodesic system of polar coordinates to extract patches. One important way to improve computational efficiency is to replace geodesic patches with anisotropic heat kernels [4]. ACSCNN [20] builds the anisotropic Chebyshev spectral CNN via anisotropic spectral filtering. MGCN [43] defines local patches via multiscale spectral manifold wavelets with a lack of orientation information. DiffusionNet [37] learns optimal diffusion times and introduces spatial gradient features to support directional filters.

3. Background & Notations

3.1. Functional Maps

Let $T : \mathcal{M} \rightarrow \mathcal{N}$ be a pointwise map between a pair of shapes (manifolds) \mathcal{M} and \mathcal{N} . This map T induces a functional map $T_F : \mathcal{L}^2(\mathcal{N}) \rightarrow \mathcal{L}^2(\mathcal{M})$, which maps the functions from the space of square-integrable functions defined on manifold \mathcal{N} to \mathcal{M} , with the image $f = T_F(g) = g \circ T, f \in \mathcal{L}^2(\mathcal{M}), g \in \mathcal{L}^2(\mathcal{N})$. If given two bases $\{\phi_i^{\mathcal{M}}\}_{i \geq 1}$ and $\{\phi_j^{\mathcal{N}}\}_{j \geq 1}$ of $\mathcal{L}^2(\mathcal{M})$ and $\mathcal{L}^2(\mathcal{N})$, this functional map will admit a matrix representation $\mathbf{C} = (c_{ij}) = \left(\langle T_F(\phi_j^{\mathcal{N}}), \phi_i^{\mathcal{M}} \rangle_{\mathcal{M}} \right)$.

Generally, the eigenfunctions of the Laplace–Beltrami Operators (LBO) $\Delta_{\mathcal{M}}$ and $\Delta_{\mathcal{N}}$ serve as the bases of the

functional map, as any function $g \in \mathcal{L}^2(\mathcal{N})$ can be expressed as $g(x) = \sum_{j \geq 1} \langle g, \phi_j^{\mathcal{N}} \rangle_{\mathcal{N}} \phi_j^{\mathcal{N}}(x)$, where the product $\langle g, \phi_j^{\mathcal{N}} \rangle_{\mathcal{N}}$ we called as spectral coefficients. In practical, only truncated (i.e, the first k) eigenfunctions are used, since they can approximate the functional map T_F well and make the size of matrix $\mathbf{C} \in \mathbb{R}^{k \times k}$ small, greatly simplifying the optimization problem in correspondence.

The computation of the functional map \mathbf{C} often relies on solving an optimization problem, which supposes \mathbf{C} satisfies certain geometric constraints or possesses special overall structural properties. It is typically formulated as

$$\mathbf{C}_{\text{opt}} = \arg \min_{\mathbf{C}} E_{\text{desc}}(\mathbf{C}) + \alpha E_{\text{reg}}(\mathbf{C}), \quad (1)$$

where the first term aims at descriptor preservations and the second term often penalizes the failure of the unknown functional map to commute with the LBOs.

In discretized settings, let shapes \mathcal{M} and \mathcal{N} be represented by triangular meshes with m and n vertices respectively. We calculate their discretized LBOs and let matrices $\Phi_{\mathcal{M}} \in \mathbb{R}^{m \times k}$ and $\Phi_{\mathcal{N}} \in \mathbb{R}^{n \times k}$ respectively contain their first k eigenvectors. Given q -dimensional descriptor functions $\mathbf{D}_{\mathcal{M}} \in \mathbb{R}^{m \times q}$ and $\mathbf{D}_{\mathcal{N}} \in \mathbb{R}^{n \times q}$ on shapes \mathcal{M} and \mathcal{N} , we store their spectral coefficients as the columns of the matrices $\hat{\mathbf{D}}_{\mathcal{M}}$ and $\hat{\mathbf{D}}_{\mathcal{N}}$, where $\hat{\mathbf{D}}_{\mathcal{M}} = \Phi_{\mathcal{M}}^{\dagger} \mathbf{D}_{\mathcal{M}}$, $\hat{\mathbf{D}}_{\mathcal{N}} = \Phi_{\mathcal{N}}^{\dagger} \mathbf{D}_{\mathcal{N}}$, here \dagger denotes the Moore-Penrose pseudo-inverse. Then the descriptor preservation on \mathbf{C} can be computed by a linear system of equations $\mathbf{C} \hat{\mathbf{D}}_{\mathcal{N}} = \hat{\mathbf{D}}_{\mathcal{M}}$, which is estimated as the solution to the following least-squares problem:

$$\mathbf{C} = \arg \min_{\mathbf{C}} \left\| \hat{\mathbf{D}}_{\mathcal{M}} - \mathbf{C} \hat{\mathbf{D}}_{\mathcal{N}} \right\|_{\text{F}}^2,$$

leading to:

$$\mathbf{C} = \hat{\mathbf{D}}_{\mathcal{M}} \hat{\mathbf{D}}_{\mathcal{N}}^{\dagger}. \quad (2)$$

Finally, the underlying pointwise map T can be recovered from the matching of Delta functions or with more advanced techniques [30, 14].

3.2. Deep Functional Maps

To alleviate the dependence of axiomatic functional maps on handcrafted features, several approaches have been proposed to learn optimal transformations of initial descriptors from data so that the optimal computed functional map satisfies some desired criteria during training. Typical structures of deep functional maps can be summarized as follows.

(1) *Learning features.* Input a shape pair \mathcal{M} and \mathcal{N} to a trainable Siamese feature network to produce q -dimensional learned features $\mathbf{D}_{\mathcal{M}}$ and $\mathbf{D}_{\mathcal{N}}$.

(2) *Computing functional maps.* Compute the spectral coefficient matrices of the learned descriptors and let them as the inputs of the functional map (FM) layer to calculate the functional map matrix $\mathbf{C} \in \mathbb{R}^{k \times k}$ according to Eq.(2) or adding commutation with LBO as regularization in Eq.(1).

(3) *Designing loss functions.* One strategy of designing loss functions is to directly penalize the function map \mathbf{C} . Like GeomFMNet [10] compared the optimized functional map \mathbf{C} with groundtruth \mathbf{C}_{gt} in a supervised manner, SURFMNet [33] designed an unsupervised loss that enforced the desired structural properties on the optimized functional map \mathbf{C} , such as its bijectivity, orthonormality, etc. Another strategy is converting \mathbf{C} to a soft correspondence matrix $\mathbf{P} \in \mathbb{R}^{n \times m}$, where $\mathbf{P} = \left| \Phi_{\mathcal{M}} \mathbf{C} \Phi_{\mathcal{N}}^{\dagger} \right|_{\|\cdot\|}$, then penalizing the distortion based on it. FMNet [21] computed a probability-weighted geodesic distance from the groundtruth, while UnsupFMNet [16] designed an unsupervised loss via geodesic distance distortion of predicted \mathbf{P} .

4. Method

4.1. Motivation & Overview

Despite significant progress achieved, recent deep functional maps still suffer several drawbacks and have a large room for improvement. Firstly, their feature extractor networks recently mainly depend on a MLP or a point cloud convolutional network. They either still insufficiently extract shape features due to spatial structures of the surface unconsidered, or are prone to make the learning overfitted on shape discretization or connectivity. Secondly, most unsupervised deep functional maps only utilize the transformed features to compute the functional map \mathbf{C} via solving a system of linear equations (see Eq.(2)). This is due to integrating other differentiable regularizations into the learning is challenging when \mathbf{C} is computed via an iterative solver. As claimed in [10], such a strategy results in lots of limitations. On the one hand, only at least k linearly independent feature functions could make the linear system invertible, but this condition is prone to be violated in practice, especially in the early stages of learning, potentially resulting in a fatal error. On the other hand, heavily relying on descriptor preservation especially makes the solved functional map very sensitive to inconsistencies in the computed descriptors, which leads to an overall loss of robustness.

To solve these issues, we propose RFMNet (**R**obust **D**eep **F**unctional **M**ap **N**ets), a more robust and geometrically informative architecture of unsupervised deep functional maps. Firstly, RFMNet utilizes DiffusionNet as a feature extraction backbone to produce discretization-resistant

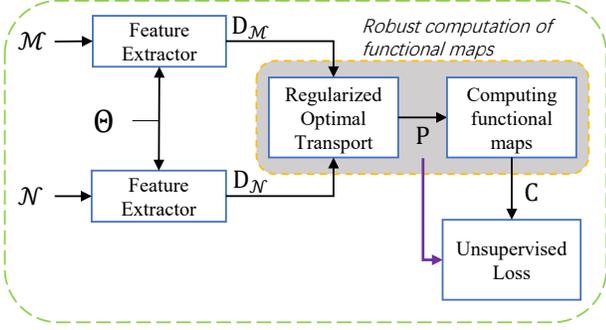


Figure 2. An overview of our RFMNet. (1) Input a pair of shapes \mathcal{M} and \mathcal{N} to a trainable Siamese feature network to produce learned features $\mathbf{D}_{\mathcal{M}}$ and $\mathbf{D}_{\mathcal{N}}$. (2) Use learned features to compute an initial pointwise map \mathbf{P} by resorting to regularized optimal transport [8] and compute the functional map \mathbf{C} using the pointwise map afterwards. (3) Use the pointwise and functional map computed in the previous block to calculate training loss.

and orientation-aware learned shape features. Then, we propose a novel strategy to compute the functional map and integrate the state-of-the-art geometric constraints to further optimize it, which avoids the training instabilities resulting from solving systems of linear equations commonly used in previous works. Finally, using the pointwise and functional map computed in the previous block, we introduce a simple but efficient unsupervised loss by penalizing the correspondence distortion of Delta functions between shapes.

The overview of our pipeline is introduced in Figure 2. It includes three parts: the DiffusionNet [37] for learning shape features, the proposed novel block for computing functional maps, and a simple but efficient unsupervised loss function for training our network.

4.2. The Feature Extractor

We integrate the DiffusionNet proposed by [37] into our pipeline to learn the transformed descriptors. This convolutional neural network depends on a simple diffusion layer equipped with spatial gradients for directional filters. It automatically generalizes across different samplings and resolutions of a shape, thus allowing our network to be applied on various geometric representations such as triangle meshes or point clouds, and even be trained on one representation and applied on another, just using suitable discretized Laplace-Beltrami operator. Moreover, DiffusionNet can produce orientation-aware shape features to disambiguate bilateral symmetry even in a purely intrinsic representation.

4.3. The Functional Maps Module

We propose an efficient method to compute the functional map \mathbf{C} by resorting to optimal transport. It is independent of the dimension of input descriptor functions and

could avoid the errors derived from solving linear systems of equations. More specially, this block could be elegantly integrated with powerful geometric regularizations on \mathbf{C} , thus permitting the production of more desirable functional maps and promoting learning efficiency.

Given the input shapes \mathcal{M} and \mathcal{N} , represented by triangular meshes with m and n vertices respectively, we denote the pointwise map $T : \mathcal{M} \rightarrow \mathcal{N}$ between them as a matrix $\mathbf{P} \in \mathbb{R}^{m \times n}$, s.t. $\mathbf{P}(i, j) = 1$ if $T(i) = j$ and 0 otherwise, where i and j are the vertex indices of \mathcal{M} and \mathcal{N} respectively. First, we build a rough pointwise map directly from the learned descriptors on two shapes, i.e, to solve the following optimization problem

$$\mathbf{P} = \arg \min_{\mathbf{P}} \|\mathbf{D}_{\mathcal{M}} - \mathbf{P}\mathbf{D}_{\mathcal{N}}\|_{\mathbb{F}}^2. \quad (3)$$

Perhaps nearest neighbor searching is the simplest way to achieve this goal. However, it is not differentiable and thus prohibitive in the training stage of neural networks. As a sequence, we resort to the optimal transport problem with an entropic regularization term [8], which can be efficiently solved on GPU with the Sinkhorn algorithm. It is differentiable, classically used for bipartite matching, that consists of iteratively normalizing distance matrix along rows and columns.

As mentioned above, we demonstrate how to create a pointwise map using learned features. However, this map heavily relies on shape features and ignores the overall structure of shape geometry. Considering the advantages of functional map representations, we convert the pointwise map to a functional map representation by

$$\mathbf{C} = \Phi_{\mathcal{M}}^{\dagger} \mathbf{P} \Phi_{\mathcal{N}}. \quad (4)$$

We intend to add more geometric structural regularization to the functional map. Although there are plenty of axiomatic functional map methods that propose various constraints on the functional map representation, the solvers for most of these optimization problems are complex and are inapplicable to the deep functional map framework.

Fortunately, we find that the constraints provided by a recent axiomatic functional map framework [17] can be conveniently incorporated into our deep learning pipeline. They require multiscale spectral manifold wavelets (SMWs) should be preserved at each scale correspondingly by the functional map \mathbf{C} , via solving the optimization problem

$$\min_{\mathbf{C}} \sum_{l=1}^L \left\| \mathbf{C}g(s_l \Lambda_{\mathcal{N}}) \Phi_{\mathcal{N}}^{\dagger} - g(s_l \Lambda_{\mathcal{M}}) \Phi_{\mathcal{M}}^{\dagger} \mathbf{P} \right\|_{\mathbb{F}}^2, \quad (5)$$

here the diagonal matrix $\Lambda_{\mathcal{M}} = \text{diag}(\lambda_1^{\mathcal{M}}, \lambda_2^{\mathcal{M}}, \dots, \lambda_k^{\mathcal{M}})$ contains the first k eigenvalues of the LBO (similar to $\Lambda_{\mathcal{N}}$), and $\{s_l\}_{l=1}^L$ is the set of discretized scales of the wavelets.

$g(\cdot)$ is a spectral filter, a smooth and compactly supported real-valued function.

As SMWs possess plenty of attractive properties, such as intrinsic invariance under shape isometric deformations, and efficiently encoding multiscale shape features, their work is considerably efficient in encoding features of isometric maps. They also demonstrate that using Parseval tight wavelet frames leads the computation of the functional map matrix boils down to a simple iterative filtering procedure by various band-pass filters. If given a filter $g(\lambda)$ for the wavelets, a smooth real-valued and positive function, and specifically its generated wavelets could form a Parseval tight frame, then the functional map \mathbf{C} under multiscale wavelet preservation constraints can be computed via simple algebraic operations, shown as following

$$\mathbf{C} = \sum_{l=1}^L g(s_l \Lambda_{\mathcal{M}}) \Phi_{\mathcal{M}}^\dagger \mathbf{P} \Phi_{\mathcal{N}} g(s_l \Lambda_{\mathcal{N}}). \quad (6)$$

Note that, the computation of \mathbf{C} in Eq.(6) can be treated as filtering processes to the rough functional map in Eq.(4). This result greatly benefits us by simply integrating multiscale spectral manifold wavelets preservation constraints into our computation of the functional map block. We just need to compute the functional map via Eq.(6), after getting the pointwise map \mathbf{P} in Eq.(3) and the initial functional map in Eq.(4).

4.4. The Unsupervised Loss

As we have successfully incorporated powerful wavelet preservation to regularize the functional map \mathbf{C} , we are permitted to lower the structural requirements of \mathbf{C} in the loss. We intend to use a simple and basic relation between the functional map and the pointwise map as our loss.

According to [28], the pointwise map T indeed can be recovered from the correspondences of Delta functions. For each Delta function δ_x centered at the point x on the shape, its spectral coefficient $\langle \delta_x, \phi_i \rangle = \phi_i(x)$. Thus, in discrete settings, we have each column of the matrices $\Phi_{\mathcal{N}}^T$ and $\Phi_{\mathcal{M}}^T$ corresponds to the spectral coefficient vector of each Delta function located at the point with the same index on respective shapes. Based on these observations, theoretically, the pointwise map \mathbf{P} and the functional map \mathbf{C} should satisfy $\mathbf{C} \Phi_{\mathcal{N}}^T \mathbf{P}^T = \Phi_{\mathcal{M}}^T$. With a simple transpose to this relation, we finally formulate our unsupervised loss as

$$loss(\mathcal{M}, \mathcal{N}) = \|\Phi_{\mathcal{M}} - \mathbf{P} \Phi_{\mathcal{N}} \mathbf{C}^T\|_F^2, \quad (7)$$

which penalizes the correspondence distortion of the Delta functions between shapes. Note that, all matrices in Eq.(7) have been computed in the previous steps of the network. Compared with some existing works, we don't need to compute additional variables, which makes our method more efficient.

5. Experimental Results

5.1. Implementation

All experiments are tested on a PC with Intel(R) Core i7-9700K CPU at 3.6GHz, 32G RAM, and Nvidia GeForce RTX 2080 Ti (11G). We implement our method in PyTorch v1.11 by adapting the open-source implementation of DiffusionNet [37]. We use the DiffusionNet as a feature extractor with its default settings which uses 16-dimensional HKS [39] as input features and produces 128-dimensional learned features. The Adam [18] optimizer with an initial learning rate of 0.001 is used for all training. For triangle meshes and point clouds, the first k eigenvalues and eigenfunctions of the discretized LBOs are precomputed according to [38]. We choose the Meyer wavelet filters [19] which satisfies Parseval tight frame to generate our wavelets.

Note that, We use the *average geodesic error* to evaluate the shape correspondence accuracy which is computed over all pairs and points in the dataset and normalized by the geodesic diameter of the source shape. All results are multiplied by 100 for the sake of readability.

5.2. Results

In this section, we show the comparisons with several state-of-the-art shape correspondence methods, including:

- *Axiomatic methods:* BCICP [31], ZoomOut [25], SmoothShells [11], MWP [17] and GCPD [14].
- *Supervised learning methods:* FMNet [21] and GeomFMNet [10, 37].
- *Unsupervised learning methods:* UnsupFMNet [16], SURFMNet [33], DeepShells [13], DUO-FMNet [9], WTFMNet [22].

Note that, axiomatic methods are slower than a test pass of our method which could directly recover high-frequency information and does not require any post-processing. GeomFMNet was first introduced in [10] which use KPConv [41] to learned features from 3D vertices coordinates. We use the improved version of GeomFMNet provided in [37] which replaces KPConv with DiffusionNet [37] as the feature extractor and uses 16-dimensional HKS [39] as input features, the same as our method. In addition, DUO-FMNet and WTFMNet also use DiffusionNet [37] as the feature extractor and 100-dimensional WKS as input features that require higher-order input descriptors.

Benchmark Tests. We first evaluate our method on the remeshed versions of the standard benchmarks FAUST [3] and SCAPE [1] from [10]. These datasets contain incompatible mesh structures, and thus are more challenging and realistic than the original ones which template with the same

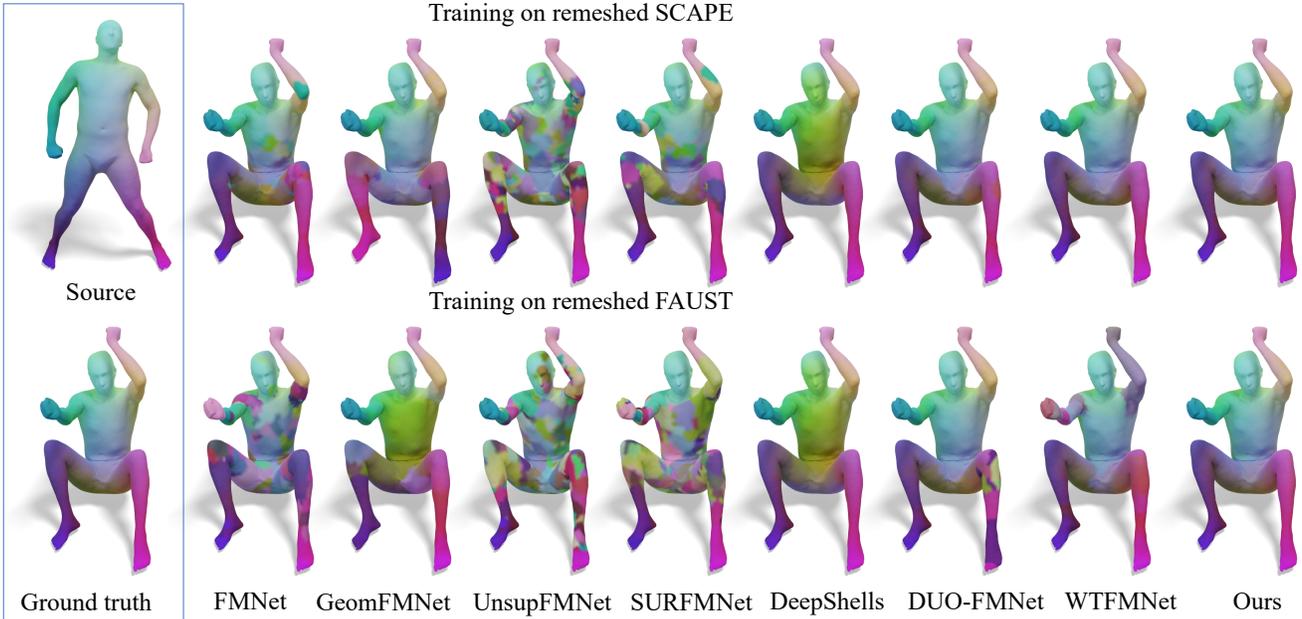


Figure 3. Generalization. We test the correspondence accuracy of several supervised and unsupervised learning methods on a shape pair from the remeshed SCAPE, where the correspondences are visualized via color transfer. All methods are trained on the remeshed SCAPE and remeshed FAUST respectively, shown in the top and bottom rows. Our results have the least color distortion and outperform other state-of-the-art methods.

Method / Dataset	FAUST	SCAPE	F on S	S on F
BCICP [31]	4.0	6.9	–	–
ZoomOut [25]	5.6	8.2	–	–
SmoothShells [11]	2.0	3.1	–	–
MWP [17]	1.8	2.4	–	–
GCPD [14]	2.0	3.6	–	–
FMNet [21]	3.2	4.2	14.4	17.2
GeomFMNet [10]	2.0	3.4	7.0	3.9
UnsupFMNet [16]	10.8	13.5	17.2	15.6
SURFMNet [33]	10.1	6.6	21.6	17.4
DeepShells [13]	1.1	1.9	4.4	2.7
DUO-FMNet [9]	1.5	1.7	4.8	1.7
WTFMNet [22]	1.5	1.8	2.4	1.7
Ours	1.0	1.2	1.4	1.0

Table 1. Benchmark tests. The numbers in the table are average geodesic errors. From top to bottom, this table is subdivided into three categories, including axiomatic, supervised and unsupervised learning methods. F on S shows the results on the test set of remeshed SCAPE trained on the remeshed FAUST, while vice versa.

number of vertices and connectivity. FAUST contains interclass pairs of 10 different people in 10 different poses, with significant variability existing between different human subjects. The 71 SCAPE shapes all show the same person in different poses. We split both datasets into training sets of 80 and 51 shapes respectively and the remain-

ing 20 shapes for testing. The results of these benchmarks are demonstrated in Table 1, where our method is compared with current state-of-the-art axiomatic and learning approaches.

Generalization. In addition, to evaluate the matching accuracy on the individual benchmarks, we also show generalization results across different datasets, which are shown in the last two columns of Table 1. To that end, we apply the network parameters learned on the remeshed FAUST to the test set of remeshed SCAPE and vice versa. These results show that our network can extract robust local features for previously unseen data, even when the local geometry of the inputs varies significantly, thus possessing much better generalization than existing learning approaches. This advantage is confirmed again via qualitative comparisons in Figure 3, where the correspondences are visualized by color transfer. Our method has little color distortion no matter trained on the same dataset or a different dataset.

Robustness to shape discretizations. Ideally, we expect correspondence methods to be agnostic to shape discretizations because scanning of real-world objects typically leads to incompatible meshing or representations. We use the benchmark of the discretization robust correspondence provided by [37] to test for the generalization of the new or totally different discretizations of shapes. This benchmark contains test shapes of human bodies, derived from the

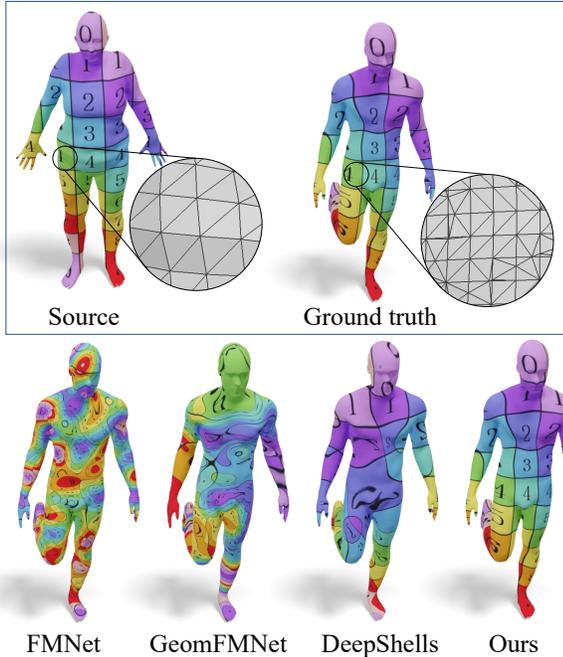


Figure 4. Qualitative comparisons of the robustness of shape discretizations. The correspondences between two shapes in the FAUST dataset are visualized via texture transfer. The target shape is from the *mc* dataset [37], with a poor quality of triangles while the source from the *orig* dataset. All methods are pre-trained on *orig* dataset. Apparently, little texture distortion happens in our method, achieving better results than other approaches.

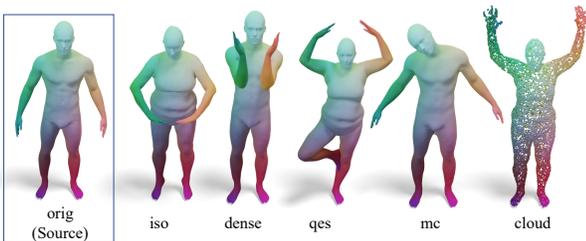


Figure 5. Qualitative demonstration of our method's robustness across shape discretizations. Our method is trained on the *orig* dataset and predicts the correspondence between other remeshed/resampled variants and the original shape of FAUST [3]. The correspondence results are visualized by color transfer. The results show our method could generalize well across different shape discretizations.

FAUST [3] and remeshed/resampled with several strategies: *orig* is the original test mesh, *iso* is a uniform isotropic remeshing, *dense* refines the mesh in randomly sampled regions, *qes* first refines the meshes and then applies quadric error simplification, *mc* is extracted via the marching cubes algorithm. *cloud* is a point cloud with normals sampled from the surface. All methods are trained on the *orig* type discretization dataset and evaluate the correspondence be-

Method	<i>orig</i>	remeshed/resampled variants				
		<i>iso</i>	<i>dense</i>	<i>qes</i>	<i>mc</i>	<i>cloud</i>
FMNet [21]	1.9	20.9	11.9	38.3	33.3	–
DeepShells [13]	0.6	2.1	4.9	12.9	20.4	–
GeomFMNet [10]	1.4	1.5	1.4	1.5	12.1	2.1
Ours	0.6	0.8	0.7	0.9	2.2	1.9

Table 2. Quantitative comparisons of the robustness of shape discretizations. We use the discretization robust correspondence benchmark provided by [37] to test the generalization ability of the methods across different discretizations. We train on the *orig* dataset, and then test the correspondences using the shapes of *orig*, *iso*, *dense*, *qes*, *mc*, and *cloud* datasets respectively as the targets and the shapes of *orig* as the sources.

tween other types and *orig* type discretization shapes. All quantitative results are shown in Table 2. Moreover, we also qualitatively demonstrate the results on these challenging datasets in Figure 4 and Figure 5. Our work announces desirable robustness across incompatible mesh structures and shape representations, even on the point cloud.

Robustness to shape noises. To further demonstrate the robustness of our method to noisy data, we evaluate the correspondence quality of shapes with noises. First, we train RFMNet on the noise-free remeshed FAUST dataset and compute shape correspondences of the noisy shapes in the test stage. The correspondence results are visualized in Figure 6. The first is a source shape, and the others are target shapes. The two middle shapes have varying degrees of Gaussian geometric noise, and the rightmost shape has a different topology structure due to legs sticking together compared with the source shape. The results demonstrate our method is robust to geometric and topological noises. It should be noted that we cannot handle high levels of topological noise because this is no longer an approximately isometric deformation.

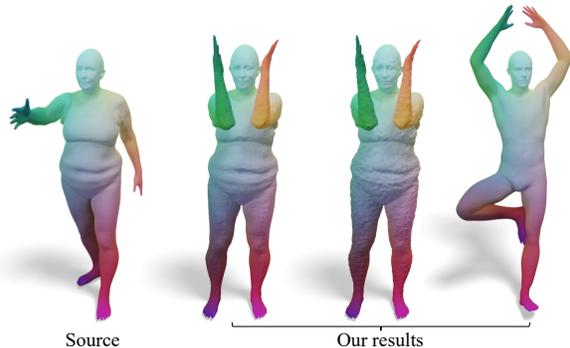


Figure 6. Qualitative demonstration of our method's robustness to shape noises. The first is a source shape, and the others are targets. The two middle shapes have varying degrees of Gaussian geometric noise, and the rightmost shape has a different topology structure due to legs sticking together compared with the source.

Method / Dataset	TOSCA_r
BCICP [31]	6.1
ZoomOut [25]	6.6
MWP [17]	3.0
UnsupFMNet [16]	26.0
Deepshell [13]	8.1
DUO-FMNet [9]	19.0
WTFMNet [22]	2.4
Ours	1.2

Table 3. Shape matching on remeshed TOSCA [5] dataset from [31]. Numbers in the table are the average geodesic errors ($\times 10^2$).

Evaluation on TOSCA Dataset. The remeshed TOSCA dataset constitutes shapes from 8 categories (cats, dogs, wolves, horses, centaurs, gorillas, and male and female humans). This dataset is also a challenging one and we compare it with the existing state-of-the-art axiomatic methods and unsupervised learning methods. As shown in Table 3, we obtain the best results that comparing both unsupervised learning-based and axiom-based methods. In particular, DUO-FMNet requires that the input mesh must be manifold without borders. Direct training on the TOSCA dataset using the publicly official code will fail ($loss = NaN$), we set $n_cfmap = 0$ as the authors’ suggestion.

Method / Dataset	SMAL_r
BCICP [31]	16.3
ZoomOut [25]	28.6
SmoothShells [11]	20.5
MWP [17]	12.3
UnsupFMNet [16]	35.0
Deepshell [13]	20.6
DUO-FMNet [9]	4.8
WTFMNet [22]	4.6
Ours	3.0

Table 4. Non-isometric shape matching on remeshed SMAL dataset. The numbers in the table are average geodesic errors ($\times 10^2$).

Non-isometric Matching. We also test our performance on SMAL remeshed shapes [45] from [9], where the connectivity of every mesh is different. This dataset is considerably challenging since it contains 49 animal shapes of different species which are strongly non-isometric. Note that, for this non-isometric test, we slightly modify our pipeline, where we replace Eq.(6) with simple Eq.(4) to compute the functional map to relax strong isometric conditions imposed by the multiscale spectral manifold wavelets preservations as the geometric regularization. We split it into 32 training shapes and 17 test shapes. The results are shown in Table

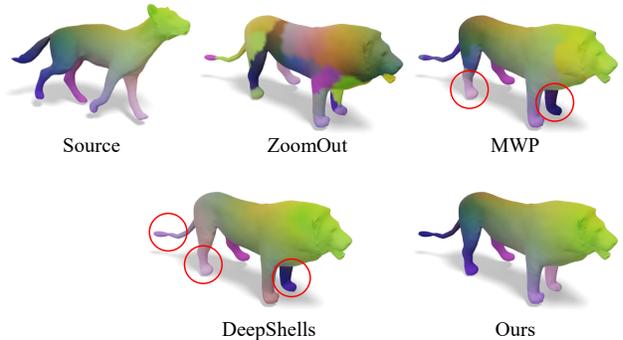


Figure 7. A qualitative comparison on a challenging non-isometric pair from the remeshed SMAL dataset.

4 and Figure 7. It highlights our method still achieves the best performance compared to the state-of-the-arts even in this quite challenging case.

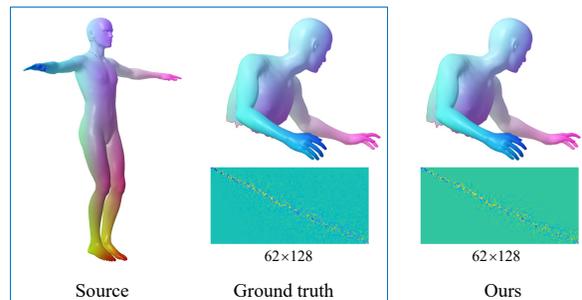


Figure 8. A qualitative demonstration of our method on a challenging part-to-full shape correspondence, where shapes from SHREC’16 Partiality benchmark [7]. The leftmost full shape is the source shape. We show the pointwise map and functional map of ground truth and ours on the target shape, respectively. The pointwise maps are visualized by color transfer.

Method / Dataset	Cuts	Holes
PFM [32]	5.6	11.4
ZoomOut [25]	4.7	8.2
MWP [17]	2.2	4.7
FMNet [21]	14.3	15.4
DeepShells [13]	32.4	22.5
Ours	1.7	3.6

Table 5. Correspondence accuracy on SHREC’16 Partiality benchmark [7]. Here our method uses 7 residual multilayer perceptron layers as feature extractors and SHOT [35] descriptors as input features, instead of DiffusionNet suggested in the paper. It aims to make our network more suitable to address partial shapes. We compare with the state-of-the-art axiomatic and learning methods, respectively demonstrated in the first and second sections of the table. All results show we achieve the best accuracy even on this considerably challenging dataset.

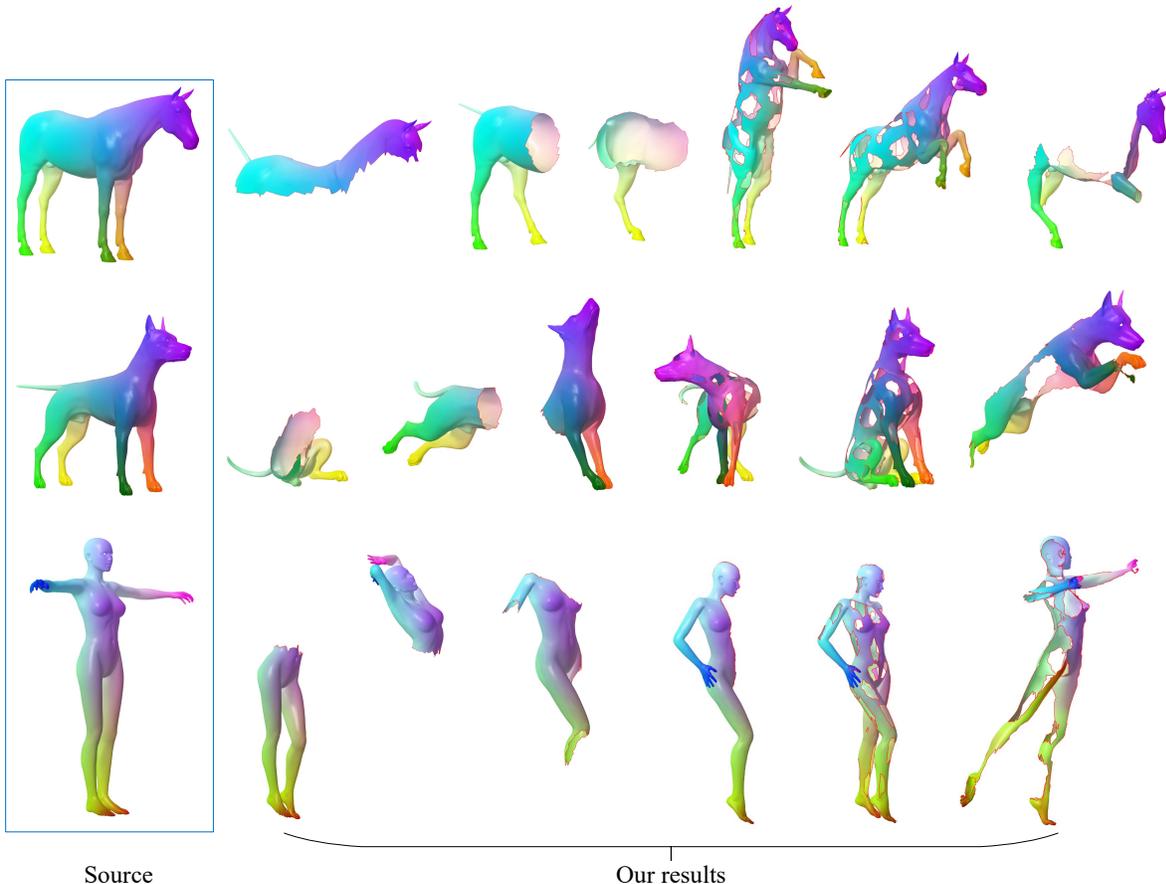


Figure 9. Visualized correspondence on a set of shapes with big cuts and holes from SHREC’16 Partiality benchmark [7] via color transfer. Here our method uses 7 residual multilayer perceptron (MLP) layers as a feature extractor and SHOT [35] descriptors as input features, rather than DiffusionNet [37] suggested in the paper. Such a process makes our network more suitable to address partial shapes. A little color distortion appears at this moment, demonstrating the advantages of our proposed computing functional map block and unsupervised loss function.

Part-to-full Shape Correspondence. A particularly challenging setting of shape correspondence occurs whenever one of the two shapes has missing geometry. As feature extractor DiffusionNet is not automatically robust to topological errors or outliers, resulting from diffusion does not allow any communication at all between distinct components of a surface, we fail to achieve desirable results on shapes with big cuts or holes. Fortunately, this limitation can be effectively mitigated by integrating other input descriptors and feature extractor networks like SHOT [35] and MLP which are less sensitive to missing parts. Firstly, we replace the DiffusionNet with 7 residual multilayer perceptron (MLP) layers as a feature extractor and use SHOT [35] descriptors as input features. In [32], it was shown that, in the case of partial isometries, the functional map matrix C has a “slanted diagonal” with a slope proportional to the area ratio A_M/A_N (here, M is a partial shape and N is a full shape.) Therefore, we truncate

the eigenvalues and the eigenvectors of partial and full shapes to different sizes for recovering the right isometry between them. We set $k_N = 128$ and determine k_M via the formula $k_M := \max_{i=1}^{k_M} \{i | \lambda_i^M \leq \max_{j=1}^{k_N} \lambda_j^N\}$, where λ_i^M and λ_j^N are eigenvalues of shape M and N , see Figure 8 for an illustration.

We test our method on the SHREC’16 Partiality benchmark [7], consisting of 8 shape classes (humans and animals) undergoing partiality transformations of two kinds: regular ‘cuts’ and irregular ‘holes’. Quantitative and qualitative results are reported in Table 5 and Figure 9, respectively. The results demonstrate our method still achieves superior accuracy to other methods even under challenging circumstances. Note that, the results also clarify again the superiorities of our proposed computing functional map block and loss function.

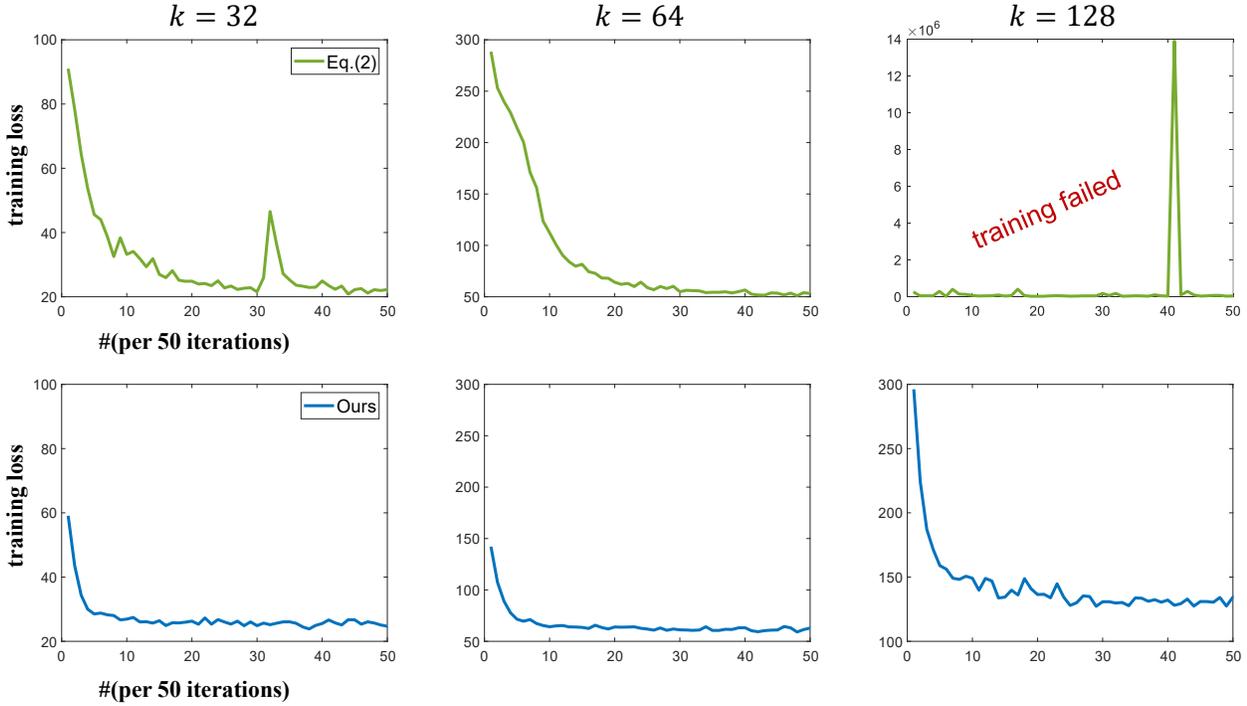


Figure 10. Comparisons in training loss between using Eq.(2) and ours as the network trained iteratively under different values of k . The results show our method is more stable under relatively large values of k when using Eq.(2) fails.

5.3. Discussion

As stated above, we propose a novel strategy to compute the functional map, following the deep functional map framework. This strategy avoids solving a system of linear equations leading to many excellent properties. We will discuss them theoretically and experimentally in the following. To verify our claims from an experimental perspective, we construct two deep function map methods with different strategies to compute the functional map representation. One of them is computed using Eq.(2) and other one is using our proposed Eq.(6). For a fair and simple comparison, all their other modules are the same, using 16-dimensional HKS [39] as input features, DiffusionNet [37] to produces 128-dimensional learned features and $\|C - C_{gt}\|_F^2$ as the loss function. We conduct experiments on remeshed SCAPE dataset from [10]. The first 51 shape are for training and the remaining 20 shapes for testing. We report the correspondence quality comparisons in Table 6. The training loss as the network iteratively trained is tracked in Figure 10 and 11. For the clarity of the presentation, we report the average of the training losses per 50 iterations.

Stability. To guarantee that Eq.(2) can be solved, the feature extractor must output learned shape features that adhere to certain requirements, e.g. $rank(\hat{D}_{\mathcal{N}}) > k$. As claimed in [10], this condition can be violated in practice, especially in the early stages of learning, potentially result-

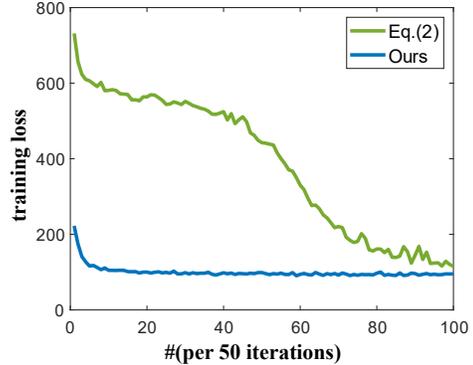


Figure 11. Comparison of convergence speed of networks using Eq.(2) or our proposed Eq.(6) to compute functional map representation. The results show our method significantly accelerates convergence speed. It gives optimal results within only 500 iterations.

ing in a fatal error. Figure 10 reports the average of the training losses per 50 iterations under setting different values of k . The network training will fail if relatively high values of k are used in the Eq.(2). Therefore, to ensure the stability of the computation, the majority of existing methods using Eq.(2) opt for relatively small values of k leading to a lack of high-frequency information. ZoomOut [25] is typically used as post-processing to recover high-frequency information from low-frequency spectral map representations by it-

erative upsampling in the spectral domain. However, our method successfully avoids this issue because it is independent of the dimension of learned features and just uses basic matrix operations. Therefore, we can produce functional maps representation with any values of k leading to the recovery of high-frequency information directly. And results in Table 6 demonstrate our superiority in comparisons with using Eq.(2) even with ZoomOut [25] as post-processing.

Method	Performance
Eq.(2)($k = 32$)	2.7
+ZoomOut _{32...128}	2.0
Eq.(2)($k = 64$)	3.1
+ZoomOut _{64...128}	2.1
Ours ($k = 128$)	1.2

Table 6. Comparison of map quality with different strategies to compute functional maps in the deep functional map framework. Our method allows us to directly recover high-frequency information and achieves quite excellent correspondence accuracy.

Convergence. Another limitation of using Eq.(2) to compute functional map representation in the deep functional map framework is that it lacks regularization on \mathbf{C} making the solved functional map \mathbf{C} overly dependent on learned features. This dramatically affects neural networks in correspondence quality and training convergence speed. [10] added commutation with LBO as regularization and derives a differentiable solution. But it still can not avoid solving a system of linear equations. We successfully integrate a state-of-the-art geometric regularization [17] into our method resorting to our novel strategy to compute the functional map representation. Figure 11 shows the comparisons between Eq.(2) and ours, which demonstrate our method significantly accelerates convergence speed. It gives optimal results within only 500 iterations.

5.4. Ablation Study

We assess the effects of proposed blocks in the ablation study in Table 7. Remarkably, our computation of functional maps based on wavelets brings significant performance improvements and generalization in different datasets, compared to the baseline structure (Compare the first row with the second row of the table). Integrating our unsupervised loss and DiffusionNet can further improve accuracy and generalization (Compare the second row with the last two rows of tables).

5.5. Parameter Settings

In addition to the architecture above, our method has some key hyper-parameters. We analyze the following two parameters of our method: the number of the first k eigenfunctions and the number of the scales of the wavelets L .

Network Architecture			F on F	F on S
Feat.Extr.	Comp.Fun.M.	Loss		
No	No	No	10.11	26.76
No	Yes	No	1.24	4.69
No	Yes	Yes	1.28	3.41
Yes	No	No	2.21	7.84
Yes	Yes	No	0.94	2.50
Yes	Yes	Yes	1.04	1.38

Table 7. Ablation study. We conduct ablation studies to test the effects of our three components. In this test, we alternate the supervised loss of FMNet [21] (as stated in Section 3.2) with the supervised spectral loss provided by GeomFMNet [10] to make the network more suitable for remeshed shapes. We use this modified FMNet as a baseline structure to test the performance of each proposed block, by successively replacing one or two or three blocks of this baseline with ours, where the related components are denoted as 'Yes', otherwise 'No'. We test their matching accuracy on benchmark datasets, with training on the remeshed FAUST and testing on the remeshed FASUT and SCAPE.

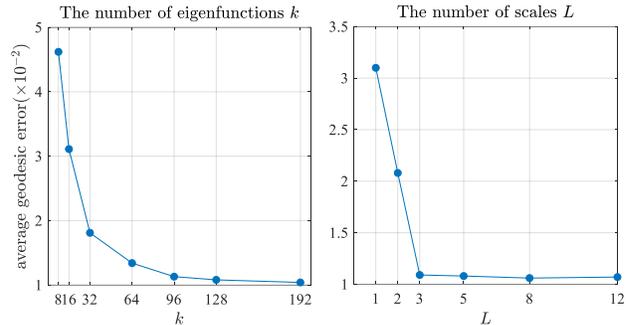


Figure 12. Parameter analysis. We test the correspondence accuracy on the remeshed FAUST dataset to find optimal values of the parameters for our method, including the number of the eigenfunctions k and the number of the scales of the wavelets L .

We set the optimal values of the parameters of our network according to the tests on the remeshed FAUST dataset. Experiment results are shown in Figure 12. To relatively small k and L , their increase brings apparent performance improvements. However, the results tend to be flat if further increasing them. Therefore, for the balance of computing efficiency and accuracy, we finally set $k = 128$, $L = 5$.

Runtime. We compute the typical timings for the remeshed FAUST dataset where each shape has approximately $5K$ vertices. It takes nearly 12 minutes for training (6340 pairs of shapes) per epoch and less than 1 minute for testing (380 pairs of shapes). Training 2 epoch is enough for our method.

6. Conclusions

We presented a novel functional map architecture for improving the generalization, correspondence quality and ef-

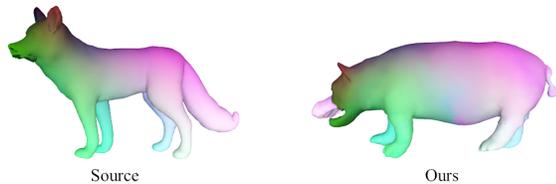


Figure 13. A failure case of our method. Our approach is still based on the bases of LBO, which can't address quite strong non-isometric shape correspondence.

efficiency of the learning. Building a novel strategy for computing functional maps benefits extracting more informative geometrical features of the map in the learning and improve the training stability. Remarkably we also design a quite simple but efficient unsupervised loss function for learning. Built upon the DiffusionNet for extracting shape features, our approach is demonstrated to be superior to the state-of-the-art methods, w.r.t. correspondence quality and computing efficiency, as well as generalization across different datasets and shape discretizations. Additionally, we can address partial shape matching by replacing DiffusionNet with a suitable feature extractor. Source codes to replicate our results are available at: <https://github.com/Qinsong-Li/RFMNet>.

Limitations. Although DiffusionNet can produce discretization-resistant and orientation-aware shape features, it is not automatically robust to topological errors or outliers. For partial matching, we must replace DiffusionNet with SHOT+MLP as the feature extractor in our architecture. We will find or design a robust feature extractor both for ideal and flawed shapes in the future. Another limitation of our method is that it's still based on the bases of LBO, which can't address quite strong non-isometric shape matching. A example of failure case is shown in Figure 13. This limitation might be mitigated by [23], which learns a spectral bases adapted to non-isometry from the datasets. We leave it for future work.

Acknowledgement

The authors wish to thank the anonymous reviewers for their valuable comments and helpful suggestions. This work was partially funded by the National Key Research and Development Program (2019YFB2204104), Hunan Provincial Natural Science Foundation of China (No. 2021JJ30172), the Natural Science Foundation of China (No. 62172447, 61876191, 62172415), and the Open Project Program of the National Laboratory of Pattern Recognition (NLPR) (No. 202200025).

References

[1] D. Anguelov, P. Srinivasan, D. Koller, S. Thrun, J. Rodgers, and J. Davis. Scape: Shape completion and animation of

people. *ACM Trans. Graph.*, 24(3):408–416, July 2005. 6

[2] S. Attaiki, G. Pai, and M. Ovsjanikov. Dpfm: Deep partial functional maps. In *International Conference on 3D Vision (3DV)*, pages 175–185. IEEE, 2021. 2, 3

[3] F. Bogo, J. Romero, M. Loper, and M. J. Black. Faust: Dataset and evaluation for 3d mesh registration. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3794–3801, 2014. 6, 8

[4] D. Boscaini, J. Masci, E. Rodola, and M. M. Bronstein. Learning shape correspondence with anisotropic convolutional neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 3189–3197, 2016. 3

[5] A. M. Bronstein, M. M. Bronstein, and R. Kimmel. *Numerical geometry of non-rigid shapes*. Springer Science & Business Media, 2008. 9

[6] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst. Geometric deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42, 2017. 3

[7] L. Cosmo, E. Rodolà, M. M. Bronstein, A. Torsello, D. Cremers, and Y. Sahillioglu. Partial Matching of Deformable Shapes. In *Eurographics Workshop on 3D Object Retrieval*. The Eurographics Association, 2016. 9, 10

[8] M. Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 26, 2013. 2, 5

[9] N. Donati, E. Corman, and M. Ovsjanikov. Deep orientation-aware functional maps: Tackling symmetry issues in shape matching. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 732–741, 2022. 2, 3, 6, 7, 9

[10] N. Donati, A. Sharma, and M. Ovsjanikov. Deep geometric functional maps: Robust feature learning for shape correspondence. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8589–8598, 2020. 2, 3, 4, 6, 7, 8, 11, 12

[11] M. Eisenberger, Z. Löhner, and D. Cremers. Smooth shells: Multi-scale shape registration with functional maps. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12262–12271, 2020. 6, 7, 9

[12] M. Eisenberger, D. Novotny, G. Kerchenbaum, P. Labatut, N. Neverova, D. Cremers, and A. Vedaldi. Neuromorph: Unsupervised shape interpolation and correspondence in one go. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7469–7479, 2021. 1

[13] M. Eisenberger, A. Toker, L. Leal-Taixé, and D. Cremers. Deep shells: Unsupervised shape correspondence with optimal transport. In *Advances in Neural Information Processing (NeurIPS)*, 2020. 3, 6, 7, 8, 9

[14] A. Fan, J. Ma, X. Tian, X. , and W. Liu. Coherent point drift revisited for non-rigid shape matching and registration. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1424–1434, 2022. 1, 4, 6, 7

[15] J. Guo, H. Wang, Z. Cheng, X. Zhang, and D.-M. Yan. Learning local shape descriptors for computing non-rigid dense correspondence. *Computational Visual Media*, 6(1):95–112, 2020. 3

- [16] O. Halimi, O. Litany, E. R. Rodolà, A. M. Bronstein, and R. Kimmel. Unsupervised learning of dense shape correspondence. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4365–4374, 2019. [2](#), [3](#), [4](#), [6](#), [7](#), [9](#)
- [17] L. Hu, Q. Li, S. Liu, and X. Liu. Efficient deformable shape correspondence via multiscale spectral manifold wavelets preservation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14531–14540, 2021. [1](#), [2](#), [3](#), [5](#), [6](#), [7](#), [9](#), [12](#)
- [18] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *International Conference for Learning Representations (ICLR)*, 2015. [6](#)
- [19] N. Leonardi and D. Van De Ville. Tight wavelet frames on multislice graphs. *IEEE Transactions on Signal Processing*, 61(13):3357–3367, 2013. [6](#)
- [20] Q. Li, S. Liu, L. Hu, and X. Liu. Shape correspondence using anisotropic chebyshev spectral cnns. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14646–14655, 2020. [3](#)
- [21] O. Litany, T. Remez, E. Rodolà, A. Bronstein, and M. Bronstein. Deep functional maps: Structured prediction for dense shape correspondence. In *IEEE International Conference on Computer Vision (ICCV)*, pages 5660–5668, 2017. [1](#), [2](#), [3](#), [4](#), [6](#), [7](#), [8](#), [9](#), [12](#)
- [22] S. Liu, H. Xu, D. ming Yan, L. Hu, X. Liu, and Q. Li. Wtfn layer: An effective map extractor for unsupervised shape correspondence. *Computer Graphics Forum*, 2022. [3](#), [6](#), [7](#), [9](#)
- [23] R. Marin, M.-J. Rakotosaona, S. Melzi, and M. Ovsjanikov. Correspondence learning via linearly-invariant embedding. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 1608–1620, 2020. [13](#)
- [24] J. Masci, D. Boscaini, M. M. Bronstein, and P. Vndergheynst. Geodesic convolutional neural networks on riemannian manifolds. In *IEEE International Conference on Computer Vision Workshop*, pages 832–840, 2015. [3](#)
- [25] S. Melzi, J. Ren, E. Rodolà, A. Sharma, P. Wonka, and M. Ovsjanikov. Zoomout: Spectral upsampling for efficient shape correspondence. *ACM Trans. Graph.*, 38(6), nov 2019. [1](#), [2](#), [6](#), [7](#), [9](#), [11](#), [12](#)
- [26] F. Monti, D. Boscaini, J. Masci, E. Rodolà, J. Svoboda, and M. M. Bronstein. Geometric deep learning on graphs and manifolds using mixture model cnns. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5425–5434, 2017. [3](#)
- [27] D. Nogneng and M. Ovsjanikov. Informative descriptor preservation via commutativity for shape matching. *Computer Graphics Forum*, 36(2):259–267, 2017. [3](#)
- [28] M. Ovsjanikov, M. Ben-Chen, J. Solomon, A. Butscher, and L. Guibas. Functional maps: A flexible representation of maps between shapes. *ACM Trans. Graph.*, 31(4), July 2012. [1](#), [2](#), [6](#)
- [29] M. Ovsjanikov, E. Corman, M. Bronstein, E. Rodolà, M. Ben-Chen, L. Guibas, F. Chazal, and A. Bronstein. Computing and processing correspondences with functional maps. In *ACM SIGGRAPH 2017 Courses*, 2017. [2](#)
- [30] 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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 384–393, 2021. [1](#), [2](#), [4](#)
- [31] J. Ren, A. Poulénard, P. Wonka, and M. Ovsjanikov. Continuous and orientation-preserving correspondences via functional maps. *ACM Trans. Graph.*, 37(6), dec 2018. [1](#), [2](#), [3](#), [6](#), [7](#), [9](#)
- [32] E. Rodolà, L. Cosmo, M. M. Bronstein, A. Torsello, and D. Cremers. Partial Functional Correspondence. *Computer Graphics Forum*, 36(1):222–236, 2017. [1](#), [2](#), [3](#), [9](#), [10](#)
- [33] J.-M. Roufousse, A. Sharma, and M. Ovsjanikov. Unsupervised deep learning for structured shape matching. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1617–1627, 2019. [2](#), [3](#), [4](#), [6](#), [7](#)
- [34] Y. Sahillioglu. Recent advances in shape correspondence. *The Visual Computer*, 36(8):1705–1721, 2020. [2](#)
- [35] S. Salti, F. Tombari, and L. Di Stefano. Shot: Unique signatures of histograms for surface and texture description. *Computer Vision and Image Understanding*, 125:251–264, 2014. [3](#), [9](#), [10](#)
- [36] A. Sharma and M. Ovsjanikov. Weakly supervised deep functional maps for shape matching. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, 2020. [3](#)
- [37] N. Sharp, S. Attaike, K. Crane, and M. Ovsjanikov. DiffusionNet: Discretization Agnostic Learning on Surfaces. *ACM Trans. Graph.*, 2022. [2](#), [3](#), [5](#), [6](#), [7](#), [8](#), [10](#), [11](#)
- [38] N. Sharp and K. Crane. A laplacian for nonmanifold triangle meshes. *Computer Graphics Forum*, 39(5):69–80, 2020. [6](#)
- [39] J. Sun, M. Ovsjanikov, and L. Guibas. A concise and provably informative multi-scale signature based on heat diffusion. *Computer Graphics Forum*, 28(5):1383–1392, 2010. [6](#), [11](#)
- [40] R. Sundararaman, G. Pai, and M. Ovsjanikov. Implicit field supervision for robust non-rigid shape matching. In *European Conference on Computer Vision*, pages 344–362. Springer, 2022. [1](#)
- [41] H. Thomas, C. R. Qi, J.-E. Deschaud, B. Marcotegui, F. Goulette, and L. Guibas. Kpconv: Flexible and deformable convolution for point clouds. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6410–6419, 2019. [6](#)
- [42] O. van Kaick, H. Zhang, G. Hamarneh, and D. Cohen-Or. A survey on shape correspondence. *Computer Graphics Forum*, 30(6):1681–1707, 2011. [1](#)
- [43] Y. Wang, J. Ren, D.-M. Yan, J. Guo, X. Zhang, and P. Wonka. MgcN: Descriptor learning using multiscale gens. *ACM Trans. Graph.*, 39(4), July 2020. [3](#)
- [44] Y.-P. Xiao, Y.-K. Lai, F.-L. Zhang, C. Li, and L. Gao. A survey on deep geometry learning: From a representation perspective. *Computational Visual Media*, 6(2):113–133, 2020. [3](#)
- [45] S. Zuffi, A. Kanazawa, D. W. Jacobs, and M. J. Black. 3d menagerie: Modeling the 3d shape and pose of animals. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5524–5532, 2017. [9](#)