DOI: 10.1111/cgf.14656
Pacific Graphics 2022
N. Umetani, E. Vouga, and C. Wojtan
(Guest Editors)

Volume 41 (2022), Number 7

WTFM Layer: An Effective Map Extractor for Unsupervised Shape Correspondence

Shengjun Liu[1], Haojun Xu[1], Dong-Ming Yan[2], Ling Hu[3], Xinru Liu[1], Qinsong Li[†][1,4]

[1]Institute of Engineering Modeling and Scientific Computing, Central South University, Changsha, Hunan 410083, China
[2]National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China
[3]School of Mathematics and Statistics, Hunan First Normal University, Changsha, Hunan 410205, China
[4]Big Data Institute, Central South University, Changsha, Hunan 410083, China

Abstract

We propose a novel unsupervised learning approach for computing correspondences between non-rigid 3D shapes. The core idea is that we integrate a novel structural constraint into the deep functional map pipeline, a recently dominant learning framework for shape correspondence, via a powerful spectral manifold wavelet transform (SMWT). As SMWT is isometrically invariant operator and can analyze features from multiple frequency bands, we use the multiscale SMWT results of the learned features as function preservation constraints to optimize the functional map by assuming each frequency-band information of the descriptors should be correspondingly preserved by the functional map. Such a strategy allows extracting significantly more deep feature information than existing approaches which only use the learned descriptors to estimate the functional map. And our formula strongly ensure the isometric properties of the underlying map. We also prove that our computation of the functional map amounts to filtering processes only referring to matrix multiplication. Then, we leverage the alignment errors of intrinsic embedding between shapes as a loss function and solve it in an unsupervised way using the Sinkhorn algorithm. Finally, we utilize DiffusionNet as a feature extractor to ensure that discretization-resistant and directional shape features are produced. Experiments on multiple challenging datasets prove that our method can achieve state-of-the-art correspondence quality. Furthermore, our method yields significant improvements in robustness to shape discretization and generalization across the different datasets. The source code and trained models will be available at `https://github.com/HJ-Xu/WTFM-Layer`.

CCS Concepts
• ***Computing methodologies*** → ***Shape analysis; Matching;***

1. Introduction

Shape correspondence is the core of many applications in computer vision and graphics (augmented/virtual reality). Many different solutions for this problem have been proposed over the years. Classical methods mostly rely on handcrafted characteristics and concentrate on using the geometric information of the input surfaces to build the correspondence [STDS14, SOG09, ASC11]. However, for some complex tasks, finding handcrafted geometric features with high quality is not easy, as they require more professional knowledge.

With the great successes of machine learning, it is a trend to apply data-driven neural networks to obtain more generalized features for shape correspondence. The most impactive learning approaches proposed in recent years are deep functional maps, pioneered by FMNet [LRR*17] and with plenty of follow-ups works [DSO20, RSO19, HLR*19, ALC20, ETLTC20, SO20, DCO22]. These works combine the learnable local feature extractors with the axiomatic functional map framework [OBCS*12], which is an important breakthrough in learning correspondence. More recently, new approaches have focused on unsupervised learning because high-quality labeled data is difficult to obtain. The representative of such approaches include weakly supervised [SO20] and unsupervised works [RSO19, HLR*19, ALC20, ETLTC20, DCO22].

Although unsupervised deep functional maps discussed above have achieved remarkable successes, they still have shortcomings when dealing with isometrically deformed shapes. The estimation of the functional maps mainly relies on the descriptor preservation constraints (e.g. [HLR*19, ALC20, RSO19, ETLTC20]), the lack of regularization makes the solutions of the functional map very sensitive to inconsistencies in the computed descriptors, and insufficiently encode the structural properties of the functional map, which leads to an overall loss of robustness and accuracy.

[†] Corresponding author: qinsli.cg@foxmail.com (Q.Li).

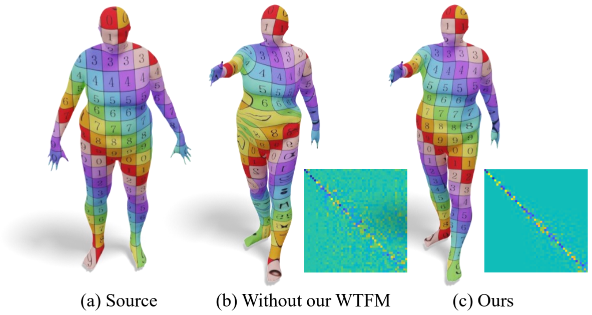

Figure 1: *Our approach extracts richer shape features via the spectral manifold wavelet transform, which strongly enforces the isometric constraint in the deep functional map framework. This leads to practical improvements on point-wise maps and functional maps compared with the previous commonly used module, which only considers descriptor preservation. We show the point-wise map by texture transferring. All the experiment settings for (b) and (c) are the same, except the computing functional map layer.*

Contribution: We build a novel map extraction layer, called the wavelet transform functional map (WTFM) layer, based on the spectral manifold wavelet transform (SMWT). Rather than previous methods only using the learned descriptors as function preservation constraints for the functional map, our WTFM layer demands multiscale SMWT results of the learned descriptor should be correspondingly preserved by the functional map. As SMWT is isometrically invariant and able to analyze features from multiple frequency bands, this strategy allows the exploit of more underlying information of the descriptors to strongly guarantee the isometrics of the underlying pointwise mapping, and significantly improved network generalization performance (see Sec.5.2). Note that since no additional terms are used to regulate the overall structural properties of the map and the WTFM layer is computationally concise, this leads to no additional time loss from it (see Sec.5.2). Further, we embedded WTFM layers in previous work [LRR*17, HLR*19], which can significantly improve their performance (see Sec.5.4). Second, we propose a novel and efficient unsupervised deep functional map framework that is based on the WTFM layer. our pipeline only needs to use a simple unsupervised loss function. Third, [LQSX21] is a state-of-the-art axiomatic method based on functional map framework using spectral manifold wavelet functions. We observe [LQSX21] is a special case of our method in theory when the input features are aligned indicator functions of a pair of shapes. Finally, a variety of experiments demonstrate that we can obtain state-of-the-art results on multiple datasets in terms of correspondence quality and generalization across different datasets.

2. Related Work

The problem of shape correspondence have been studied extensively in recent years. In the following of this section, we review the approaches most related to our work, focusing on learning-based, and especially unsupervised techniques. A detailed introduction is referred to a recent survey [Sah20].

2.1. Axiomatic functional maps

The functional map was first proposed in [OBCS*12]. This framework determines a functional map operator that maps between the spaces of square-integrable functions on the respective shapes, then a high-quality pointwise correspondence can be recovered from mapping special functions. The most attractive property of the functional map is that, finding correspondence boils down to the determination of a small functional map matrix, which encodes relations between basis functions defined on the shapes. Therefore, many studies try to formulate powerful constraints such as preservation of geometric quantities (e.g., descriptors) combined with commutativity to optimize the structural properties of the functional maps [OBCS*12]. Later, follow-up research have been extended to partial shapes [RCB*17, LRB*16], refined pointwise maps [RMC17], direction-preserving maps [RPWO18], and iteratively spectral upsampling maps [MRR*19], as well as maps combined with the matrix scaling schemes from computational optimal transport [PRM*21] and with extrinsic shape alignment [ELC20]. Despite great successes achieved by these axiomatic methods, their performances still heavily depend on the quality of the inputting handcrafted descriptor and most of them make restrictive assumptions about the discretization, topology, or deformation of the considered shapes.

2.2. Learning correspondence

The difficulties of the axiomatic methods mentioned above perhaps could be solved by deep learning technology, motivated by its great successes in the field of image analysis and many others. Therefore, lots of efforts have been paid to combing deep learning technology with axiomatic functional maps. They roughly fall into two categories, supervised and unsupervised learning.

2.2.1. Supervised learning

The combination of learnable feature extractor with traditional functional maps method was pioneered by [LRR*17]. Their architecture used seven residual multi-layer perceptron (MLP) layers to optimize the nonlinear transformation of the SHOT descriptor [STDS14] to obtain a map as close as possible to the given groundtruth. [DSO20] extracted shape features directly from vertex coordinates of shapes through the network and added regular term constraints to the functional maps to obtain more effective corresponding. We also note that there also exist other supervised learning methods that learn correspondences without functional map representation. [GFK*18] proposed a scheme of generating models to calculate the correspondence of specific types of shapes. The main idea is to learn how to deform a template of a specific class of objects, such as a template of a person. However, such methods tend to require a large amount of training data and may not be robust to join relationships.

There are other researches to design local convolutional operators to extract shape features and build convolutional neural networks to compute shape correspondence [MBBV15, BMRB16, MBM*17, FELWM18, LLHL20, WRY*20, GWC*20, LLC*22]. They cast the shape correspondence into a classification problem and train the network using a cross-entropy loss function. But cross-entropy loss can not capture shape geometry compared with deep functional maps. And some methods, such as [FELWM18, LLHL20] significantly overfit to a particular resolution or even a particular triangulation.

In addition, these methods discussed above are supervised neural networks, which require high-quality labeled data. In fact, acquiring these labeled data is often quite challenging. Therefore, it is a natural way to explore unsupervised learning approaches which use little labels or no labels at all during the training stage.

2.2.2. Weakly\Unsupervised learning

Unsupervised learning seems to be more practical because of its training without using data labels. Existing unsupervised methods [RSO19, HLR*19, ALC20] all followed the pipeline proposed in [LRR*17], while proposing different unsupervised loss functions. [HLR*19] and [ALC20] used unsupervised loss functions based on geodesic distance and heat kernels respectively. Unfortunately, the geodesic distance matrix has costly storage and low calculation efficiency. [RSO19] used what they called a spectral unsupervised loss function which aggregated several structural penalties on the functional map. However, these constraints are still insufficient to encode the properties of the maps, which leads to overall drops in robustness and performance. Utilizing the loss function of [RSO19], [SO20] designed a weakly supervised learning framework that used manually aligned vertex coordinates instead of SHOT descriptor [STDS14] as the input of the network, obtained by PointNet++ [QYSG17]. More recently, [ETLTC20] proposed to replace the functional maps layer with a multi-scale correspondence refinement layer based on optimal transport at the cost of efficiency. [APO21] presented an approach for addressing partial-to-partial non-rigid shape matching, using the DiffusionNet [SACO22] as the feature extraction module. [DCO22] used complex functional maps [DCMO22] with functional maps to construct an unsupervised network framework that can make their approach orientation-aware.

However, despite great efforts, networks' robustness to shape discretization and generalization to cross datasets is still a problem that needs to be solved. [DCMO22] tries to do this, but the method requires that the input mesh is manifold.

2.3. Multiscale spectral manifold wavelets preservation

Recently, a state-of-the-art axiomatic approach [LQSX21] has introduced spectral manifold wavelet into shape correspondence tasks. Spectral manifold wavelets inherit most of the powerful properties of the classical wavelets, such as space-frequency locality, multiscale characteristics, etc, and can be computed efficiently. [LQSX21] aims to directly align the multiscale spectral manifold wavelets (SMW) on the shape pair and achieve significant improvements in terms of accuracy and computational efficiency on the shape correspondence task. But this approach still heavily relies on axiomatic descriptors, and an iterative approach is needed to further refine the results in its pipeline. Therefore, it is unclear how to incorporate SMW into a learning framework while maintaining accuracy and efficiency.

3. Foundation

We intend to build an accurate and efficient unsupervised learning approach for shape correspondence, which integrates the powerful tool SMWT into the pipeline of the deep functional map framework. To make the paper self-contained, in this section, we briefly review the general framework of the learning-based functional maps and the basic knowledge of the SMWT and the wavelets.

3.1. Deep functional maps

We first review the basic pipeline of the deep functional map. Given the source and target shapes $\mathcal{M}$ and $\mathcal{N}$ with n_1 and n_2 vertices respectively, then

1. Compute the discrete LBOs [MDSB03] of the shapes $\mathcal{M}$ and $\mathcal{N}$ and their eigendecomposition. Formulate their first k eigenvectors as the matrices $\Phi_{\mathcal{M}}$ and $\Phi_{\mathcal{N}}$ respectively.
2. Given a pair of shape features $\mathbf{F}_{\mathcal{M}} \in \mathbb{R}^{n_1 \times p}$ and $\mathbf{G}_{\mathcal{N}} \in \mathbb{R}^{n_2 \times p}$ as input, existing frameworks typically train a Siamese networks $\mathcal{F}$ that share parameters to get descriptors, where $\mathcal{F}(\mathbf{F}_{\mathcal{M}}) : \mathbb{R}^{n_1 \times p} \longrightarrow \mathbb{R}^{n_1 \times q}$, $\mathcal{F}(\mathbf{G}_{\mathcal{N}}) : \mathbb{R}^{n_2 \times p} \longrightarrow \mathbb{R}^{n_2 \times q}$. Then the learned descriptors are further projected onto the bases $\Phi_{\mathcal{M}}$ and $\Phi_{\mathcal{N}}$. Store their coefficients as the columns of the matrices $\hat{\mathbf{F}}$ and $\hat{\mathbf{G}}$. The optimal functional map $\mathbf{C} \in \mathbb{R}^{k \times k}$ is computed as:
$$\min_{\mathbf{C}} \left\| \mathbf{C}\hat{\mathbf{F}} - \hat{\mathbf{G}} \right\| + \alpha E_{reg}(\mathbf{C}), \tag{1}$$
The first part of Eq. (1) is for the preservation of descriptor functions, and the second part acts as a regularizer that adds geometric constraints to it. Then, the pointwise maps matrix $\mathbf{P}$ is restored by the obtained $\mathbf{C}$.
3. Finally, the parameters of network $\mathcal{F}$ are optimized by minimizing the training loss.

The above process is the general process of learning-based methods. Because the solution process is usually non-differentiable after adding the regular term, existing work [HLR*19, RSO19] generally uses only the first part of Eq. (1). [DSO20] tries to integrate the second part of Eq. (1) and construct a differentiable method to optimize it, but this brings additional computational complexity.

3.2. Multiscale spectral manifold wavelets (SMW)

SMW was first introduced in [HVG11], which is a successful extension of wavelet theory from regular (Euclidean) to irregular (graph or manifold) spaces. SMW can effectively characterize the multiscale neighborhood topology of the points and equip them with various excellent geometric properties [HVG11, LVDV13, LHL*21]. And SMW exhibits good localization properties both in the spatial and spectral domain and inherits LBO's isometric invariant properties. Please refer to [HVG11] for details.

Given a spectral filter $g(\cdot)$, a smooth and compactly supported real-valued function, the SMW at point y with scale s is defined as:
$$\psi_{s,y}(x) = \sum_{i \geq 1} g(s\lambda_i)\phi_i(y)\phi_i(x). \tag{2}$$

Here, s is a hyperparameter, $\{\lambda_i\}_{i\geq 1}$ and $\{\phi_i\}_{i\geq 1}$ are the eigenvalues and the eigenvectors of the LBOs. The SMW with smaller

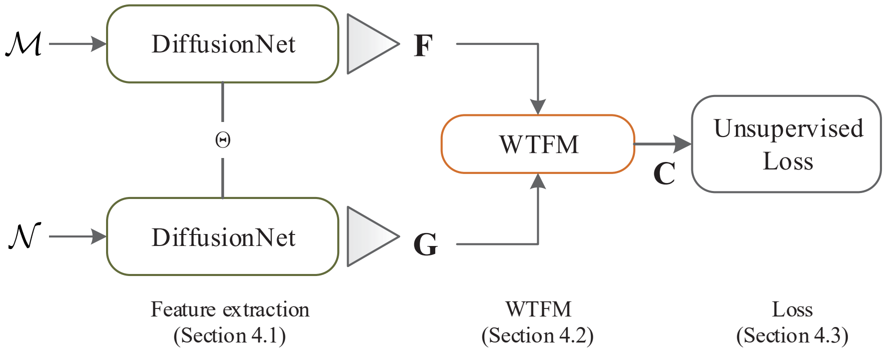

Figure 2: *Overview of our network. Given two shapes on $\mathcal{M}$ and $\mathcal{N}$, the descriptors* **F** *and* **G** *are obtained by the feature extraction module, and then an optimal functional maps* **C** *is obtained using our WTFM block and converted into a pointwise matrix* **P**. *Finally, we use an unsupervised loss function to train the network.*

scale s can capture the local geometric features around their located points, while a larger parameter s allows the SMW to spread farther on the shape that encode overall information of a neighborhood with a larger radius. More importantly, as constructed based on the eigenvalues and eigenfunctions of the LBOs (see Eq.(2)) , the SMW naturally inherits the property of isometric deformation invariance. Such property plays an important role in the field of non-rigid shape analysis. For example, [LQSX21] utilizes the isometric invariant property of SMW and uses the aligned multiscale SMW as functional conservation constraints on the functional map framework to recover an isometric map.

3.3. Multiscale spectral manifold wavelets transform (SMWT)

Given a function f defined on $\mathcal{M}$, its SMWT is defined by the inner product of f and the corresponding SMW, i.e.,

$$W_f(s,y)=\langle f,\psi_{s,y}\rangle_{\mathcal{M}}=\sum_{i\geq 1} g(s\lambda_i)\hat{f}(i)\phi_i(y), \tag{3}$$

where $\hat{f}(i)=\langle f,\phi_i\rangle_{\mathcal{M}}$. According to spectral analysis, parameter s controls the frequency bands passed.

Due to the isometric invariance of SMW, we claim the SMWT results $W_f(s,y)$ of a given function f is isometric invariant if f is isometric invariant. In essence, the multiscale SMWT of function f is to extract different frequency-bands information of function f.

We have totally different policies for wavelet usage compared with [LQSX21]. We apply the spectral manifold wavelet transform (SMWT) to the learned descriptors and use the SMWT results to regularize the structural properties of the functional map. The SMWT is a powerful tool that can analyze signals defined on shapes from multi frequency bands and allows to capture of more deep signal features, which obtain satisfactory results without complex regularization.

4. Methodology

In this section, we describe our approach in detail. Specifically, like Fig. 2, our approach consists of three main components. The first component is the feature extraction module in Sec.4.1. The second component, we call it as WTFM layer in Sec.4.2, and the final component is the loss function for unsupervised learning in Sec.4.3. Next, we describe the design of each component in the follwing.

4.1. Feature extractor

As mentioned above and shown in Fig. 2, our first module is the feature extraction module. Its structure is a shared weight Siamese network used to extract the features of source and target shapes. We hope that the features extracted by this module can be insensitive to deformation and robust to shape discretization.

Based on the above requirements, we use DiffusionNet [SACO22] as the feature extraction module rather than relying on the SHOT descriptor as before [ETLTC20]. Based on its oriented gradient blocks, DiffusionNet can adaptively extract features from shapes with different deformations or different resolutions. The effectiveness of this module for learning shape correspondence has been verified in DUO-FMNet [DCMO22], but not sufficient(see Sec. 5.2).

To further improve performance, we introduce SMWT into the deep functional framework and construct a differentiable pipeline to improve shape correspondence performance in terms of correspondence quality and generalizatoin across different datasets.

4.2. The wavelet transform functional map (WTFM) layer

Most of existing works estimate the functional maps by solving an optimization problem $\min_{\mathbf{C}}\left\|\mathbf{C}\hat{\mathbf{F}}-\hat{\mathbf{G}}\right\|$, added with certain regular terms or structural penalties in loss function. However, recent constraints on the structure of **C** still seem to be over completed, such as multi conditions including orthogonality and commutativity, etc., or perhaps fail to guarantee the isometric invariance of the underlying maps.

For this problem, we try to add constraints to descriptors to optimize the functional map matrix **C**. Inspired by the powerful abilities of the SMWT that can extract more deep signal information via filtering and their isometric invariance property, we intend to demand that the multiscale SMWT results of the input descriptor functions should be correspondingly preserved by the functional maps, rather than only using the learned descriptors as preservation constraints in previous methods. Note that, significantly different to [LQSX21] which directly aligns the multi-scale wavelet functions on the two shapes, our process is filtering to the learned feature descriptors (as mentioned in Sec.3.2) by SMWT and using the results to estimate the functional map.

Given the learned feature matrices **F** and **G** of shapes $\mathcal{M}$ and $\mathcal{N}$ respectively. Each column of **F** and **G** are corresponding functions between shapes $\mathcal{M}$ and $\mathcal{N}$. We first perform SMWT to learned features. According to Eq.(3), the SMWT results of learned features can be represented as following matrices,

$$\mathbf{W}_{\mathcal{M},s} = \Phi_{\mathcal{M}} g\left(s\Lambda_{\mathcal{M}}\right) \Phi_{\mathcal{M}}^{\dagger} \mathbf{F}, \forall s,$$

$$\mathbf{W}_{\mathcal{N},s} = \Phi_{\mathcal{N}} g\left(s\Lambda_{\mathcal{N}}\right) \Phi_{\mathcal{N}}^{\dagger} \mathbf{G}, \forall s.$$

Here, the matrices $\Lambda_{\mathcal{M}}$=diag$\left(\lambda_1^{\mathcal{M}}, \lambda_2^{\mathcal{M}}, ..., \lambda_k^{\mathcal{M}}\right)$ and $\Lambda_{\mathcal{N}}$=diag$\left(\lambda_1^{\mathcal{N}}, \lambda_2^{\mathcal{N}}, ..., \lambda_k^{\mathcal{N}}\right)$ contain the first k eigenvalues of the LBOs of $\mathcal{M}$ and $\mathcal{N}$ respectively, and every column of the matrices $\Phi_{\mathcal{M}}$ and $\Phi_{\mathcal{N}}$ represent the corresponding eigenvectors of the LBOs. † denotes the Moore-Penrose pseudo-inverse and satisfy $\Phi^{\dagger}\Phi = \mathbf{I}$, where **I** is identity matrix. As discussed in Sec.3.2, the isometric properties of the LBOs can be perfectly inherited by the SMW. Thus, the matrix $\mathbf{W}_{\mathcal{M},s}$ which can be treated as the filtering results of the original features, will be preserved well on the shape $\mathcal{N}$ by an isometric functional map. Inspired by this conclusion, we intend to use $\mathbf{W}_{\mathcal{M},s}$ and $\mathbf{W}_{\mathcal{N},s}$ as the new features for the subsequent operations, which will effectively guarantee the isometric invariance of the maps. To this end, we first get their coefficients matrices under the bases $\Phi_{\mathcal{M}}$ and $\Phi_{\mathcal{N}}$ by following projection

$$\hat{\mathbf{W}}_{\mathcal{M},s} = \Phi_{\mathcal{M}}^{\dagger} \mathbf{W}_{\mathcal{M},s} = g\left(s\Lambda_{\mathcal{M}}\right) \Phi_{\mathcal{M}}^{\dagger} \mathbf{F}, \forall s,$$

$$\hat{\mathbf{W}}_{\mathcal{N},s} = \Phi_{\mathcal{N}}^{\dagger} \mathbf{W}_{\mathcal{N},s} = g\left(s\Lambda_{\mathcal{N}}\right) \Phi_{\mathcal{N}}^{\dagger} \mathbf{G}, \forall s. \tag{4}$$

Next, we hope to find a functional map **C** that satisfies the following optimization problem

$$\min_{\mathbf{C}} \sum_{s} \left\| \mathbf{C} \hat{\mathbf{W}}_{\mathcal{M},s} - \hat{\mathbf{W}}_{\mathcal{N},s} \right\|. \tag{5}$$

Note that, we use multi-discrete values of the scale parameter $s \in S = \{s_1, s_2, ...\}$ in our objective function. This means each frequency-band information of the inputs is required to be preserved by the functional map, which allows to sufficiently guarantee the isometric deformation invariance of the map.

Then, we discuss the solution of Eq.(5). Substituting Eq.(4) into Eq.(5) and letting $\hat{\mathbf{F}} = \Phi_{\mathcal{M}}^{\dagger}\mathbf{F}$ and $\hat{\mathbf{G}} = \Phi_{\mathcal{N}}^{\dagger}\mathbf{G}$, the objective function in Eq.(5) can be represented as

$$\min_{\mathbf{C}} \sum_{s_i} \left\| \mathbf{C} g\left(s_i\Lambda_{\mathcal{M}}\right) \hat{\mathbf{F}} - g\left(s_i\Lambda_{\mathcal{N}}\right) \hat{\mathbf{G}} \right\|. \tag{6}$$

Eq.(6) is still linear and actually has a analytical solution, if we let

$$\mathbf{C} g\left(s_i\Lambda_{\mathcal{M}}\right) \hat{\mathbf{F}} = g\left(s_i\Lambda_{\mathcal{N}}\right) \hat{\mathbf{G}}, \forall s_i \in S. \tag{7}$$

However, solving the system of linear equations of Eq.(7) directly is unstable and costly, we intend to find a more efficient computing strategy.

To this end, we multiply both sides of Eq.(7) with $\hat{\mathbf{F}}^{\mathrm{T}}$ and $\left(\hat{\mathbf{F}}\hat{\mathbf{F}}^{\mathrm{T}}\right)^{-1}$, then Eq.(7) becomes

$$\mathbf{C} g(s_i\Lambda_{\mathcal{M}}) = g(s_i\Lambda_{\mathcal{N}}) \hat{\mathbf{G}}\hat{\mathbf{F}}^{\mathrm{T}} \left(\hat{\mathbf{F}}\hat{\mathbf{F}}^{\mathrm{T}}\right)^{-1}, \forall s_i \in S$$

Finally, we multiply both sides of above equation by $g(s_i\Lambda_{\mathcal{M}})$ again, and we get

$$\mathbf{C} g(s_i\Lambda_{\mathcal{M}})^2 = g(s_i\Lambda_{\mathcal{N}}) \hat{\mathbf{G}}\hat{\mathbf{F}}^{\mathrm{T}} \left(\hat{\mathbf{F}}\hat{\mathbf{F}}^{\mathrm{T}}\right)^{-1} g(s_i\Lambda_{\mathcal{M}}), \forall s_i \in S$$

Note that, the above derivation just refers to one single scale. In order to get enough structural information of the map, we aggregate the features of all scales, by performing the same operation for all scales and summing up the results. Particularly, when using a set of filters satisfy Parseval tight frame [LVDV13], i.e., $g(\cdot)$ satisfy $\sum_{s_i} g\left(s_i\Lambda_{\mathcal{M}}\right)^2 = \mathbf{I}$, we will get

$$\mathbf{C} = \sum_{s_i} g\left(s_i\Lambda_{\mathcal{N}}\right) \hat{\mathbf{G}}\hat{\mathbf{F}}^{\mathrm{T}} \left(\hat{\mathbf{F}}\hat{\mathbf{F}}^{\mathrm{T}}\right)^{-1} g\left(s_i\Lambda_{\mathcal{M}}\right). \tag{8}$$

Eq.(8) can be implemented in Pytorch. It allows us to estimate the isometric functional maps robustly with differentiability. We call this whole process WTFM Layer.

Our work is closely related to the approach of [LQSX21], a current state-of-the-art axiomatic correspondence method resorting to the SMW instead of SMWT in our WTFM Layer. These two works are built upon the spectral graph(manifold) theory first introduced in [HVG11]. From Sec.3.2 and Sec.3.3, we know SMW and SMWT are different concepts. And we observe [LQSX21] is a special case of our method, when learned features **F** and **G** in Eq.(6) are aligned indicator functions. The derivations are as follows.

Unlike our use of imposing SMWT on features, [LQSX21] aims to directly align the multiscale SMW on the shape pair. Their strategy can be expressed as the following optimization problem:

$$\min_{\mathbf{C}} \sum_{s_i} \left\| \mathbf{C} g\left(s_i\Lambda_{\mathcal{M}}\right) \Phi_{\mathcal{M}}{}^{\dagger} - g\left(s_i\Lambda_{\mathcal{N}}\right) \Phi_{\mathcal{N}}^{\dagger} \mathbf{P}^{\mathrm{T}} \right\|, \tag{9}$$

where **P** is a giving permutation matrix to encode the point-wise correspondence from shape $\mathcal{M}$ to $\mathcal{N}$. The solution to the above equation is:

$$\mathbf{C} = \sum_{s_i} g\left(s_i\Lambda_{\mathcal{N}}\right) \Phi_{\mathcal{N}}^{\dagger} \mathbf{P}^{\mathrm{T}} \Phi_{\mathcal{M}} g\left(s_i\Lambda_{\mathcal{M}}\right).$$

We consider the special case of our objective function in Eq.(6). Supposing the learned features **F** of shape $\mathcal{M}$ are indicator functions. For each column of **F**, the function value is 1 in the column index, otherwise 0. So in this special case, the learned features **F** can be represented as an identity matrix $\mathbf{I}$. Then giving the point-wise correspondence matrix $\mathbf{P}$, the learned features $\mathbf{G}$ in shape $\mathcal{N}$ satisfy $\mathbf{G} = \mathbf{I}\mathbf{P}^{\mathrm{T}}$ as correspondence functions. Then we project the learned features onto the bases $\Phi_{\mathcal{M}}$ and $\Phi_{\mathcal{N}}$ and computed the coefficients $\hat{\mathbf{F}} = \Phi_{\mathcal{M}}^{\dagger}\mathbf{F} = \Phi_{\mathcal{M}}^{\dagger}$ and $\hat{\mathbf{G}} = \Phi_{\mathcal{N}}^{\dagger}\mathbf{G} = \Phi_{\mathcal{N}}^{\dagger}\mathbf{P}^{\mathrm{T}}$. If we substitute the coefficients into our objective functions in Eq.(6), the result formula is equivalent to Eq.(9).

4.3. Loss function

The final module of the deep functional map framework is to design the loss function for training the neural network. One strategy of designing loss functions is to directly penalize the function map $\mathbf{C}$. Like GeomFMNet [DSO20] compared the optimized functional map $\mathbf{C}$ with groundtruth $\mathbf{C}_{gt}$ in a supervised manner, SURFMNet [RSO19] 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 $\mathbf{P} \in \mathbb{R}^{n_1 \times n_2}$, where $\mathbf{P} = \left|\Phi_{\mathcal{M}}\mathbf{C}\Phi_{\mathcal{N}}^{\dagger}\right|_{\|\cdot\|}$, then penalizing the distortion based on it. FMNet [LRR*17] computed a probability-weighted geodesic distance from the groundtruth, while UnsupFMNet [HLR*19] designed an unsupervised loss via geodesic distance distortion of predicted $\mathbf{P}$.

For simplicity, we leverage the alignment errors of intrinsic embedding between shapes as a loss function,

$$loss = \min_{\mathbf{P}} \left\| \Phi_{\mathcal{M}}\mathbf{C}^{\mathrm{T}} - \mathbf{P}\Phi_{\mathcal{N}} \right\|_{\mathrm{F}}^{2}. \tag{10}$$

The objective function in Eq.(10) is often used to convert a functional map to a point-wise map and solve it efficiently with the nearest neighbor search algorithm. However, it is not differentiable and thus prohibitive in the training stage of neural networks. Therefore, we reformulate Eq.(10) as an entropy regularized optimal transport [Cut13], which can be solved by the Sinkhorn algorithm in an unsupervised way [ETLTC20]. The Sinkhorn algorithm is differential, so the network can be trained without any labeled data.

5. Experiments and Results

Implementation details. Our method is implemented with Pytorch v1.11 and Python 3.8. Offline training is run on an i7-6800K CPU + NVIDIA GeForce GTX 1080Ti GPU.

In our experiments, the same as [DCMO22], we use 120-dimension WKS [ASC11] descriptors as inputs, and the Adam [KB15] optimizer with an initial learning rate of 0.001 for all training. For each shape, the first 50 eigenvalues and eigenfunctions of the LBOs are precomputed. See the 5.2 for more discussion about the parameter settings. We choose the Meyer wavelet satisfying Parseval tigh frame with adaptive bandwidth [LVDV13] as our filters, and the number of the discrete scale parameter s is set to 5. Correspondence quality is evaluated by the mean geodesic error [KLF11].

Datasets. We tested on several datasets to fully validate the effectiveness of our approach: the remeshed versions of FAUST(F_r) and SCAPE(S_r) [DSO20], the remeshed versions of TOSCA(T_r), the anisotropic remeshed versions of FAUST(F_a) and SCAPE(S_a) [DCMO22], as-well-as non-isometric datasets SMAL [DCMO22]. Existing methods can deal with F_r and S_r datasets, and fitting experiments on these two datasets are not enough to test the performance of the method. Therefore, we only test the generalization on F_r and S_r. F_a and S_a are anisotropic remeshed datasets in which the shapes have completely different numbers of vertices and connections on the left and right sides. The tests on anisotropic remeshed datasets can verify the robustness of the approach to mesh discretization.

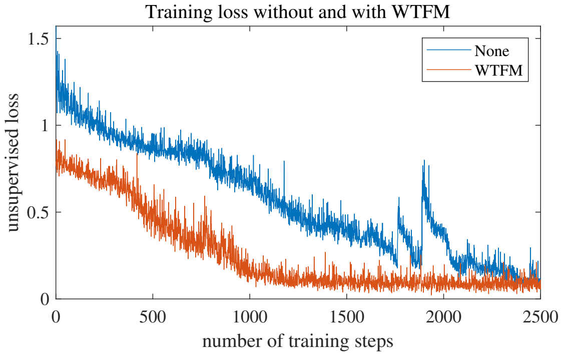

Figure 3: *Comparison of convergence speed without and with WTFM in the computing functional map block. The training loss evolution shows that our WTFM layer helps drastically with the convergence speed and is more stable during training stage.*

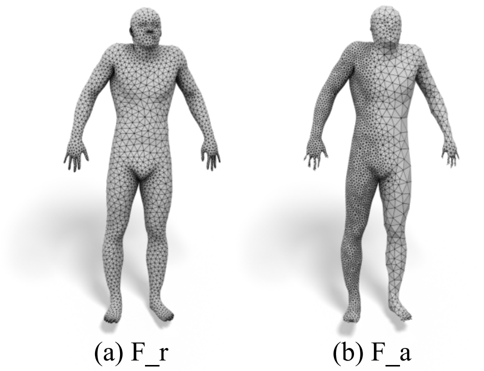

Figure 4: *Demonstration of shapes from remeshed FAUST (F_r) and anisotropic remeshed FAUST (F_a), respectively. The vertices of the shape from F_r are evenly distributed over the surface. But the shape from F_a has a denser number of vertices on the left than on the right.*

					Train on-Test on	
Method	F_r-F_a	S_r-S_a	F_r-S_a	S_r-F_a	F_r-S_r	S_r-F_r
BCICP [RPWO18]	8.5	14.0	-	-	-	-
ZoomOut [MRR*19]	8.7	15.0	-	-	-	-
MWP [LQSX21]	6.8	6.5	-	-	-	-
FMNet [LRR*17]	32.0	18.0	35.0	39.0	21.0	27.0
GeomFMNet(WKS) [DSO20]	2.6	**2.3**	3.8	8.4	3.8	9.9
UnsupFMNet [HLR*19]	28.0	29.0	33.0	36.0	22.0	29.0
DeepShells [ETLTC20]	12.0	10.0	5.7	15.0	5.7	2.7
DUO-FMNet [DCO22]	3.0	2.7	4.2	3.1	4.2	2.7
Ours	**2.5**	**2.5**	**2.7**	**2.8**	**2.7**	**2.2**

Table 1: *Results on remeshed FAUST and SCAPE. The matching error is evalutated by mean geodesic errors ($\times 10^2$). F-S means that the network parameters are trained on the FAUST while tested on the SCAPE, and vice versa.*

5.1. Results

5.1.1. Evaluation on human dataset

Benchmark test. We tested different versions of the FAUST and SCAPE datasets. As mentioned above, the scheme of training and testing in remeshed FAUST datasets or remeshed SCAPE datasets at the same time has been unable to effectively distinguish the performance of the method. Therefor, We set up training on the remeshed version of the dataset and testing on the anisotropic version of the dataset, i.e., F_r to F_a and S_r to S_a.

We compare our method to several state-of-the-art methods, where the first, second and third category respectively includes various axiomatic methods, supervised and unsupervised deep functional maps. From the results reported in Table 1, For discrete structurally inconsistent datasets F_a and S_a, [ETLTC20, HLR*19] cannot obtain satisfactory results due to the limitation of the input descriptor SHOT. Our proposed WTFM layer can encode the isometric deformation invariant properties of the underlying map and effectively characterize the shape geometric information, even though our approach has the same feature extraction modules and inputs as DUO-FMNet, our approach can still achieve better correspondence results than DUO-FMNet.

Cross-dataset generalization. The performance of generalization across datasets has been the focus of attention, and here, we set up four sets of experiments to evaluate the performance of our method in generalization across datasets, i.e., training on F_r and testing on S_r and S_a, and vice versa. As shown in Table 1, similar to our approach, both GeomFMNet(WKS) and DUO-FMNet use WKS as input and DiffusionNet as a feature extractor, our approach can extract robust features, even for previously invisible poses and different mesh discretizations, Our approach is able to achieve better generalization than existing methods.

5.1.2. Evaluation on TOSCA dataset

The remeshed TOSCA dataset constitutes shapes from 8 categories (cats, dogs, wolves, horses, centaurs, gorillas, male and female humans). It is a challenging dataset As shown in Table 2, 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 authors' suggestion.

Method / Dataset	TOSCA_r
BCICP [RPWO18]	6.1
ZoomOut [MRR*19]	6.6
MWP [LQSX21]	3.0
Unsup-FMNet [HLR*19]	26.0
Deepshell [ETLTC20]	8.1
DUO-FMNet [DCO22]	19.0
Ours	**2.4**

Table 2: *Shape matching on remeshed TOSCA [BBK08] dataset from [RPWO18]. Numbers in the table are the mean geodesic errors ($\times 10^2$).*

Method / Dataset	SMAL_r
BCICP [RPWO18]	19.0
ZoomOut [MRR*19]	35.0
SmoothShells [ELC20]	28.0
MWP [LQSX21]	18.0
Unsup-FMNet [HLR*19]	35.0
DeepShells [ETLTC20]	25.0
DUO-FMNet [DCO22]	4.8
Ours	**4.6**

Table 3: *Non-isometric shape matching on remeshed SMAL [ZKJB17] dataset from [DCO22]. Numbers in the table are the mean geodesic errors* ($\times 10^2$).

5.1.3. Non-isometric shape matching

To further test the performance of our method, we test on the remeshed version of SMAL [ZKJB17] dataset from [DCO22].

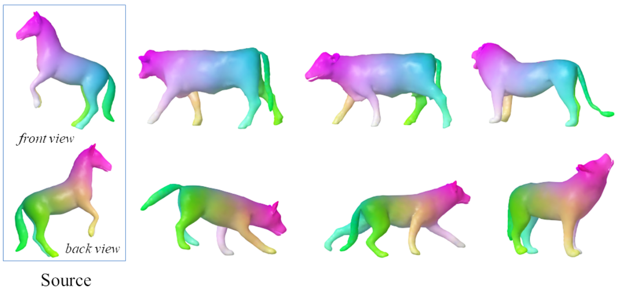

Figure 5: *Demonstration of shape correspondence on the remeshed SMAL dataset. The left-most column is the source shape in the front and back views, respectively. Others are the correspondence results of our methods visualized by color transferring.*

This dataset is quite challenging because it contains animal shapes from different species, most of which are non-isometric deformations. This dataset contains 49 shapes. According to the split of [DCMO22], we divided the data set into 32 training models and 17 test models. This dataset is mostly a non-isometric deformation shapes, therefore, this test sets the scale number $s = 1$.

The results are shown in Table 3 and Figure 5, where our method achieves the best correspondence results even for different class of shapes. This provides the possibility to further explore our method for non-isometric matching.

			Train on-Test on	
Method	F_r-F_a	S_r-S_a	F_r-S_a	S_r-F_a
None	8.8	16.0	19.0	20.0
WTFM	**2.5**	**2.5**	**2.7**	**2.8**

Table 4: *Ablation experiments for evaluating the effectiveness of WTFM layer. Where None means using WKS+DiffusionNet and WTFM means using WKS+DiffusionNet+WTFM. Numbers in the table are the mean geodesic errors* ($\times 10^2$).

5.2. Ablation Experiments

WTFM. In this section, the same parameters are used for all tests. In order to more fully verify the effectiveness of WTFM, we conducted comparative experiments on anisotropic data sets. As can be seen from Table 4, using only DiffusionNet in an unsupervised framework is not able to handle anisotropic datasets effectively, and adding the WTFM layer can significantly improve the performance. In addition, Fig. 3 shows the effect of adding WTFM or not on the convergence speed by visualization. Our method has converged when the number of iterations reaches 1000, i.e., the network passes 1000 pairs of shapes. The method without adding the WTFM layer does not approach convergence until 2500 iterations.

In addition, we evaluated the impact of WTFM layer on the computational efficiency, as shown in Table 5, since the operations in the WTFM layer are small matrix multiplications, there is almost no additional time consumption when using it.

Method	Times(minutes)
None	6.30
WTFM	6.31

Table 5: *Time(minutes) consumed to run an epoch under the same conditions.*

5.3. Parameter Analysis

The number of wavelet scales. This will discuss the setting of the number of the scale parameters s. In subsequent discussions, Nf is used to represent the number of the scale parameters for convenience. In Fig. 6, we show the corresponding performances with setting Nf from 1 to 8. Here we can see that with the increase of Nf, the corresponding accuracy is also improved, but this improvement has a supremum. When Nf achieves 5, if it continues to increase, the corresponding accuracy will oscillate within a certain range. This is a reasonable situation. Our method is equivalent to decomposing the signal to different frequency bands for filtering and combining the results in all the filtered frequency bands. When there are too many frequency bands, information redundancy will be generated. Thus we set the number of the scale parameters N to 5. It is worth noting that when $N = 1$, our method will degenerate to the general solving of least squares.

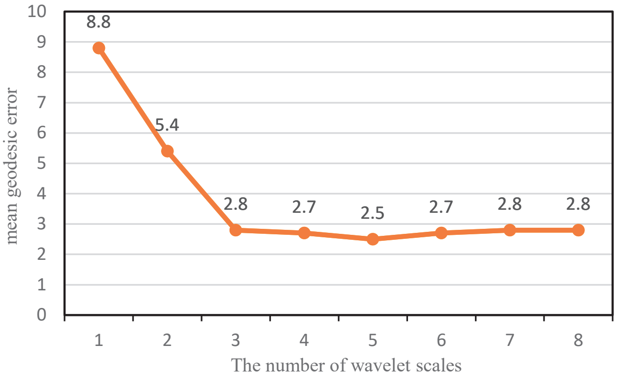

Figure 6: *The effect of using different numbers of discrete scale parameters Nf on the results is compared by visualization. The matching error is evaluated by mean geodesic errors ($\times 10^2$). All results in the figure are trained on F_r tested on F_a*

			Train on-Test on	
The num. of eig.fun.	F_r-F_a	S_r-S_a	F_r-S_a	S_r-F_a
30	3.4	3.5	5.1	4.5
50	2.5	**2.5**	**2.7**	**2.8**
70	**2.4**	2.7	4.1	3.9

Table 6: *Experiments of evaluating the rationality of the numbers of eigenfunctions we select. Numbers in the table are the mean geodesic errors ($\times 10^2$).*

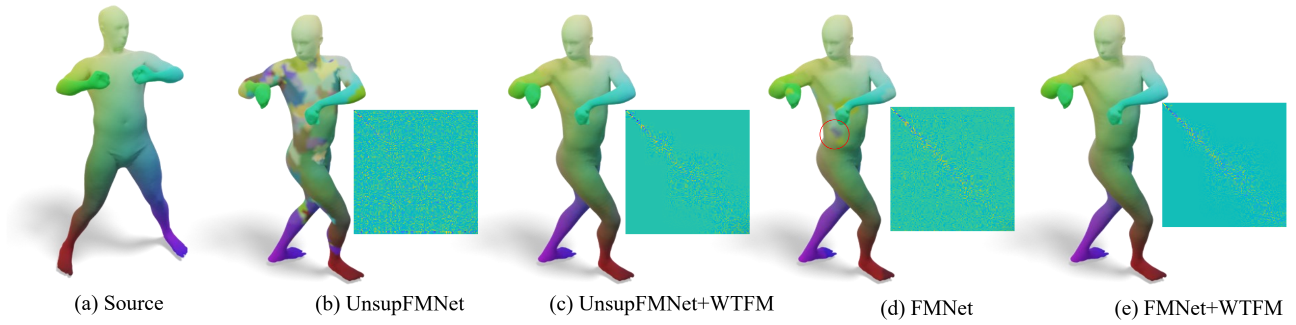
(a) Source (b) UnsupFMNet (c) UnsupFMNet+WTFM (d) FMNet (e) FMNet+WTFM

Figure 7: *Visualized correspondences between a pair of shapes from the remeshed SCAPE dataset via color transfer. We show the comparisons between FMNet [LRR*17] and UnsupFMNet [HLR*19] before and after replacing WTFM Layer, using WTFM layer can achieve less distortion.*

The number of eigenfunctions In this section, we discuss the selection of the numbers of the eigenfunctions used in this paper. We selected the first 30, the first 50 and the first 70 eigenfunctions respectively. Usually fewer LBOs eigenfunctions are favorable for generalization, but too few LBOs eigenfunctions are not favorable for feature retention. As shown in Table 6, the optimal results are achieved when we set the LBOs feature function to select the first 50.

			Train on-Test on	
Method	FAUST	SCAPE	F - S	S - F
FMNet [LRR*17]	5.1	6.2	21.0	27.0
FMNet+WTFM	2.9	2.6	9.8	18.8
UnsupFMNet [HLR*19]	10.0	16.0	22.0	29.0
UnsupFMNet+WTFM	4.3	4.2	10.0	12.0

Table 7: *The WTFM layer proposed in this paper can be used as a flexible plug-in, for FMNet or UnsupFMNet, giving rise to great performance improvements. Numbers in the table are the mean geodesic errors ($\times 10^{-2}$).*

5.4. WTFM Layer Universality

This section shows the corresponding precision improvements brought by embedding our proposed WTFM Layer as a "plug-in" into other works. As shown in Table 7, since our WTFM Layer can encode the isometric deformation invariant attributes of the underlying map, we can embed the WTFM Layer as a flexible module into the previous works like FMNet [LRR*17] and UnsupFMNet [HLR*19], making their performances improved. Fig. 7 visualizes the correspondences between a pair of models before and after using WTFM Layer via color transfer.

6. Conclusion and Future Work

We propose an unsupervised method to compute correspondences between deformable shapes. The key of this method is to use the multiscale SMWT results of the learned features as function preservation constraints to optimize the functional map. A variety of experiments show that our method outperforms the current state-of-the-art methods in terms of accuracy and generalization across different datasets.

Our approach still has limitations. We build it on the assumptions of isometric deformation invariance of the shapes. Although we can deal with shapes with non-isometric deformation to some extent, such as the shapes in SMAL, it still cannot address the large non-isometric deformation. For our method and the existing works (like [LRR*17, HLR*19, RSO19, ETLTC20, DSO20, SO20] etc), basis selection is fundamental to performance. However, due to the limitation of Laplacian eigenbasis, the existing works fail to deal with large non-isometric deformation well. In the future, we will try to construct a learnable Laplacian eigenbasis to solve this problem. Then, we also can try to construct a learnable wavelet filter to further improve our performance.

Acknowledgements

This work was supported by the Natural Science Foundation of China (No. 62172447, 61876191, 62172415), Hunan Provincial Natural Science Foundation of China (No. 2021JJ30172), the Open Project Program of the National Laboratory of Pattern Recognition (NLPR) (No. 202200025), and Fundamental Research Funds for the Central Universities of Central South University (No. 2022ZZTS0606).

References

[ALC20] Aygun M., Lahner Z., Cremers D.: Unsupervised Dense Shape Correspondence using Heat Kernels. In *International Conference on 3D Vision (3DV)* (2020), IEEE, pp. 573–582. doi:10.1109/3DV50981.2020.00067. 1, 3

[APO21] Attaiki S., Pai G., Ovsjanikov M.: Dpfm: Deep partial functional maps. In *2021 International Conference on 3D Vision (3DV)* (2021), pp. 175–185. doi:10.1109/3DV53792.2021.00040. 3

[ASC11] Aubry M., Schlickewei U., Cremers D.: The wave kernel signature: A quantum mechanical approach to shape analysis. In *International Conference on Computer Vision (ICCV)* (2011), pp. 1626–1633. doi:10.1109/ICCVW.2011.6130444. 1, 6

[BBK08] Bronstein A. M., Bronstein M. M., Kimmel R.: *Numerical geometry of non-rigid shapes*. Springer Science & Business Media, 2008. 7

[BMRB16] Boscaini D., Masci J., Rodolà E., Bronstein M.: Learning shape correspondence with anisotropic convolutional neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)* (2016), vol. 29, pp. 3189–3197. 2

[Cut13] Cuturi M.: Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems (NeurIPS)* (2013), vol. 26. 6

[DCMO22] Donati N., Corman E., Melzi S., Ovsjanikov M.: Complex functional maps : a conformal link between tangent bundles. *Computer Graphics Forum* (2022). 3, 4, 6, 8

[DCO22] Donati N., Corman E., Ovsjanikov M.: Deep orientation-aware functional maps: Tackling symmetry issues in shape matching. In *Conference on Computer Vision and Pattern Recognition (CVPR)* (2022). 1, 3, 7

[DSO20] Donati N., Sharma A., Ovsjanikov M.: Deep geometric functional maps: Robust feature learning for shape correspondence. In *Conference on Computer Vision and Pattern Recognition (CVPR)* (2020), IEEE, pp. 8592–8601. 1, 2, 3, 6, 7, 9

[ELC20] Eisenberger M., Lähner Z., Cremers D.: Smooth shells: Multi-scale shape registration with functional maps. In *Conference on Computer Vision and Pattern Recognition (CVPR)* (2020), pp. 12262–12271. arXiv:1905.12512, doi:10.1109/CVPR42600.2020.01228. 2, 7

[ETLTC20] Eisenberger M., Toker A., Leal-Taixé L., Cremers D.: Deep shells: Unsupervised shape correspondence with optimal transport. *Advances in Neural Information Processing Systems (NeurIPS) 34* (2020). 1, 3, 4, 6, 7, 9

[FELWM18] Fey M., Eric Lenssen J., Weichert F., Müller H.: Splinecnn: Fast geometric deep learning with continuous b-spline kernels. In *Conference on Computer Vision and Pattern Recognition (CVPR)* (2018), pp. 869–877. 2, 3

[GFK*18] Groueix T., Fisher M., Kim V. G., Russell B. C., Aubry M.: 3d-coded: 3d correspondences by deep deformation. In *European Conference on Computer Vision (ECCV)* (2018), Springer, pp. 230–246. 2

[GWC*20] Guo J., Wang H., Cheng Z., Zhang X., Yan D.-M.: Learning local shape descriptors for computing non-rigid dense correspondence. *Computational Visual Media 6*, 1 (2020), 95–112. doi:10.1007/s41095-020-0163-y. 2

[HLR*19] Halimi O., Litany O., Rodola E., Bronstein A. M., Kimmel R.: Unsupervised learning of dense shape correspondence. In *Conference on Computer Vision and Pattern Recognition (CVPR)* (2019), IEEE, pp. 4370–4379. 1, 2, 3, 6, 7, 9

[HVG11] Hammond D. K., Vandergheynst P., Gribonval R.: Wavelets on graphs via spectral graph theory. *Applied and Computational Harmonic Analysis 30*, 2 (2011), 129–150. 3, 5

[KB15] Kingma D. P., Ba J.: Adam: A method for stochastic optimization. In *International Conference for Learning Representations (ICLR)* (2015). 6

[KLF11] Kim V. G., Lipman Y., Funkhouser T.: Blended intrinsic maps. *ACM transactions on graphics 30*, 4 (2011), 1–12. 6

[LHL*21] Li Q., Hu L., Liu S., Yang D., Liu X.: Anisotropic spectral manifold wavelet descriptor. *Computer Graphics Forum 40*, 1 (2021), 81–96. doi:https://doi.org/10.1111/cgf.14120. 3

[LLC*22] Liu S., Liu H., Chen W., Yan D.-M., Hu L., Liu X., Li Q.: An Anisotropic Chebyshev Descriptor and its Optimization for Deformable Shape Correspondence. *Computational Visual Media* (2022). doi:10.1007/s41095-022-0290-8. 2

[LLHL20] Li Q., Liu S., Hu L., Liu X.: Shape correspondence using anisotropic chebyshev spectral cnns. In *Conference on Computer Vision and Pattern Recognition (CVPR)* (2020), pp. 14646–14655. doi:10.1109/CVPR42600.2020.01467. 2, 3

[LQSX21] Ling H., Qinsong L., Shengjun L., Xinru L.: Efficient deformable shape correspondence via multiscale spectral manifold wavelets preservation. In *Conference on Computer Vision and Pattern Recognition (CVPR)* (2021), IEEE, pp. 14536–14545. 2, 3, 4, 5, 7

[LRB*16] Litany O., Rodolà E., Bronstein A. M., Bronstein M. M., Cremers D.: Non-rigid puzzles. *Computer Graphics Forum 35*, 5 (2016), 135–143. 2

[LRR*17] Litany O., Remez T., Rodola E., Bronstein A., Bronstein M.: Deep functional maps: Structured prediction for dense shape correspondence. In *International conference on computer vision (ICCV)* (2017), IEEE, pp. 5659–5667. 1, 2, 3, 6, 7, 9

[LVDV13] Leonardi N., Van De Ville D.: Tight wavelet frames on multislice graphs. *IEEE Transactions on Signal Processing 61*, 13 (2013), 3357–3367. doi:10.1109/TSP.2013.2259825. 3, 5, 6

[MBBV15] Masci J., Boscaini D., Bronstein M. M., Vandergheynst P.: Geodesic convolutional neural networks on riemannian manifolds. In *International Conference on Computer Vision (ICCV)* (2015), pp. 832–840. doi:10.1109/ICCVW.2015.112. 2

[MBM*17] Monti F., Boscaini D., Masci J., Rodolà E., Svoboda J., Bronstein M. M.: Geometric deep learning on graphs and manifolds using mixture model cnns. In *Conference on Computer Vision and Pattern Recognition (CVPR)* (2017), pp. 5425–5434. 2

[MDSB03] Meyer M., Desbrun M., Schröder P., Barr A. H.: Discrete differential-geometry operators for triangulated 2-manifolds. In *Visualization and mathematics III*. Springer, 2003, pp. 35–57. 3

[MRR*19] Melzi S., Ren J., Rodolà E., Sharma A., Wonka P., Ovsjanikov M.: Zoomout: Spectral upsampling for efficient shape correspondence. *ACM Transactions on Graphics 38*, 6 (2019). 2, 7

[OBCS*12] Ovsjanikov M., Ben-Chen M., Solomon J., Butscher A., Guibas L.: Functional maps: a flexible representation of maps between shapes. *ACM Transactions on Graphics 31*, 4 (2012), 1–11. 1, 2

[PRM*21] Pai G., Ren J., Melzi S., Wonka P., Ovsjanikov M.: Fast sinkhorn filters: Using matrix scaling for non-rigid shape correspondence with functional maps. In *Conference on Computer Vision and Pattern Recognition (CVPR)* (2021), IEEE, pp. 384–393. 2

[QYSG17] Qi C. R., Yi L., Su H., Guibas L. J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems (NeurIPS)* (2017), Curran Associates Inc., pp. 5105–5114. 3

[RCB*17] Rodolà E., Cosmo L., Bronstein M. M., Torsello A., Cremers D.: Partial functional correspondence. *Computer Graphics Forum 36*, 1 (2017), 222–236. 2

[RMC17] Rodolà E., Moeller M., Cremers D.: Regularized pointwise map recovery from functional correspondence. *Computer Graphics Forum 36*, 8 (2017), 700–711. 2

[RPWO18] Ren J., Poulenard A., Wonka P., Ovsjanikov M.: Continuous and orientation-preserving correspondences via functional maps. *ACM Transactions on Graphics 37*, 6 (2018), 1–16. 2, 7

[RSO19] Roufosse J.-M., Sharma A., Ovsjanikov M.: Unsupervised deep learning for structured shape matching. In *International conference on computer vision (ICCV)* (2019), IEEE, pp. 1617–1627. 1, 3, 6, 9

[SACO22] Sharp N., Attaiki S., Crane K., Ovsjanikov M.: Diffusionnet: Discretization agnostic learning on surfaces. *ACM Trans. Graph. 41*, 3 (2022). doi:10.1145/3507905. 3, 4

[Sah20] Sahillioglu Y.: Recent advances in shape correspondence. *Vis. Comput. 36*, 8 (2020), 1705–1721. doi:10.1007/s00371-019-01760-0. 2

[SO20] Sharma A., Ovsjanikov M.: Weakly supervised deep functional maps for shape matching. In *Advances in Neural Information*

Processing Systems (NeurIPS) (2020), vol. 33, Curran Associates, Inc., pp. 19264–19275. 1, 3, 9

[SOG09] SUN J., OVSJANIKOV M., GUIBAS L.: A concise and provably informative multi-scale signature based on heat diffusion. *Computer graphics forum 28*, 5 (2009), 1383–1392. 1

[STDS14] SALTI S., TOMBARI F., DI STEFANO L.: Shot: Unique signatures of histograms for surface and texture description. *Computer Vision and Image Understanding 125* (2014), 251–264. 1, 2, 3

[WRY*20] WANG Y., REN J., YAN D.-M., GUO J., ZHANG X., WONKA P.: Mgcn: Descriptor learning using multiscale gcns. *ACM Trans. Graph. 39*, 4 (July 2020). doi:10.1145/3386569.3392443. 2

[ZKJB17] ZUFFI S., KANAZAWA A., JACOBS D. W., BLACK M. J.: 3d menagerie: Modeling the 3d shape and pose of animals. In *Conference on Computer Vision and Pattern Recognition (CVPR)* (2017), pp. 5524–5532. doi:10.1109/CVPR.2017.586. 7