Spatially and Spectrally Consistent Deep Functional Maps
Mingze Sun, Shiwei Mao, Puhua Jiang, Maks Ovsjanikov, Ruqi Huang

To cite this version:
Mingze Sun, Shiwei Mao, Puhua Jiang, Maks Ovsjanikov, Ruqi Huang. Spatially and Spectrally Consistent Deep Functional Maps. ICCV 2023 - International Conference on Computer Vision, Oct 2023, Paris, France. hal-04352370

HAL Id: hal-04352370
https://hal.science/hal-04352370
Submitted on 19 Dec 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Spatially and Spectrally Consistent Deep Functional Maps

Mingze Sun\(^1\), Shiwei Mao\(^1\), Puhua Jiang\(^1,2\), Maks Ovsjanikov\(^3\), Ruqi Huang\(^1\)*

\(^1\) Tsinghua Shenzhen International Graduate School, China
\(^2\) Peng Cheng Laboratory, China
\(^3\) LIX, École polytechnique, IP Paris, France

Abstract

Cycle consistency has long been exploited as a powerful prior for jointly optimizing maps within a collection of shapes. In this paper, we investigate its utility in the approaches of Deep Functional Maps, which are considered state-of-the-art in non-rigid shape matching. We first justify that under certain conditions, the learned maps, when represented in the spectral domain, are already cycle consistent. Furthermore, we identify the discrepancy that spectrally consistent maps are not necessarily spatially, or point-wise, consistent. In light of this, we present a novel design of unsupervised Deep Functional Maps, which effectively enforces the harmony of learned maps under the spectral and the point-wise representation. By taking advantage of cycle consistency, our framework produces state-of-the-art results in mapping shapes even under significant distortions. Beyond that, by independently estimating maps in both spectral and spatial domains, our method naturally alleviates over-fitting in network training, yielding superior generalization performance and accuracy within an array of challenging tests for both near-isometric and non-isometric datasets. Codes are available at https://github.com/rqhuang888/Spatially-and-Spectrally-Consistent-Deep-Functional-Maps.

1. Introduction

Non-rigid shape matching is a fundamental task in shape analysis, playing a pivotal role in a wide spectrum of applications including texture transfer [1], volume parameterization [31], statistical shape analysis [2, 6], to name a few. In order to establish correspondences between shapes undergoing non-rigid deformations, early approaches [46, 4, 47] focus on designing hand-crafted features by exploiting geometric invariance. Nowadays, data-driven techniques have been widely adopted to learn features in a more flexible way, leading to significant improvements regarding accuracy, efficiency, and robustness [10].

A noticeable trend among the learning-based shape matching approaches is based on the formalism of Deep Functional Maps (DFM), pioneered by the FMNet [30]. Functional maps [37], as a spectral map representation, allows to encode maps into compact matrices and to express desirable map priors (e.g., area-preservation, isometry, bijectivity) in simple algebraic forms. In particular, by learning features that optimize functional map priors, several unsupervised DFM frameworks [19, 43, 18, 13, 9, 28] have been proposed and, remarkably, achieve even comparable results with respect to the supervised counterparts. On the other hand, while the majority of DFM works focus on exploring local, pairwise-level map priors, there is relatively less attention (with the exception of [9]) paid to the problem of injecting global map priors into DFM pipelines.

As a global map prior, cycle consistency has long been used as a powerful regularizer for jointly optimizing maps among shapes, both under point-wise [22] and functional [21, 24] map representation. In this paper, we exploit the utility of cycle consistency within DFM frameworks and propose a novel two-branch unsupervised DFM framework, which promotes cycle consistency in both spectral and spatial domains. We first re-examine the generic DFM frame-
work and outline a theoretical condition, based on the residuals of a linear solver used in functional map estimation, that guarantees spectrally cycle consistency in DFM over training shapes. Then we identify the discrepancy that spectrally consistent maps are not necessarily spatially, or point-wise, consistent. In light of this, we leverage our theoretical insight to construct a latent shape in DFM and employ this construction to formulate a novel two-branch design of unsupervised DFM. We align each shape’s spectral embedding, represented as the eigenbasis of the Laplace-Beltrami operator, to a canonical space given by the constructed universal latent shape. We then compute point-wise maps in this canonical embedding domain, which promotes consistency across computed point-wise maps. In the end, the point-wise maps are converted to functional maps, which are expected to agree with the ones from the original spectral branch. Overall, our two-branch DFM takes advantage of the justified spectrally cycle consistency and further enhances it to spatially cycle consistent.

We conduct extensive experiments on several non-rigid shape matching benchmarks, and demonstrate that our method achieves superior performance over existing methods, especially in the presence of significant distortions among shapes (see, e.g., Fig. 1). We also observe that our method, by enforcing the harmony of spatial and spectral map representations, reduces over-fitting during training, which leads to remarkable generalization performance within an array of challenging tests. Finally, our two-branch design can be easily plugged into any existing DFM framework, and we show evidence that it achieves significant performance gains upon both recent and early DFM approaches [43] with marginal computational burden.

To summarize, our main contributions are as follows: (1) We perform theoretical analysis on spectrally cycle consistency of DFM frameworks; (2) We formulate a simple yet effective two-branch design of unsupervised DFM based on our theoretical justification, which introduces spatially cycle consistency. (3) We prove the effectiveness of our method through an array of challenging non-rigid shape matching tasks, demonstrating improvements upon existing methods in terms of accuracy, consistency, and generalization performance.

2. Related Work

Functional Maps Our method is built upon functional map representation, introduced in [37] and then significantly extended in follow-up works (see, e.g., [38]). The key idea is to encode shape correspondences as transformations between the respective spectral embeddings, which are represented by compact matrices by using reduced eigenbasis. The functional maps pipeline has been further improved in accuracy, efficiency, and robustness by many recent works including [26, 21, 8, 42, 36].

Deep Functional Maps In contrast to axiomatic approaches that use hand-crafted features [46, 4], the deep functional maps approach, pioneered by FMNet [30], aims to learn the optimal features from data. FMNet contributes several key designs of DFM: (1) it leverages Siamese network to conduct learning in a set of shape pairs; (2) it advocates refining the input descriptors with non-linear transformations. FMNet is then supervised by labeled maps to learn optimal features.

Instead of learning from labeled maps, unsupervised approaches [19, 43] demonstrate that it is sufficient to learn from geometric map priors. More recently, with the development of robust mesh feature extractors [44], more frameworks [9, 28, 11, 3] are proposed to learn directly from geometry, yielding state-of-the-art performance.

Cycle Consistency Cycle consistency has long been used as a strong prior for joint map optimization among a collection of shapes. Axiomatic approaches detect and eliminate inconsistent cycles using consistency constraints [25, 52, 35, 51, 18], as well as associate cycle consistency with low-rank properties of matrices encoding map networks [22, 50, 27]. Related to the latter, the matrix nature of the functional maps enables convenient access to map composition, which naturally bridges the functional map framework and consistent map refinement techniques [49, 21, 23, 45]. Some recent learning-based approach [20] also incorporate cycle consistency in the pipeline. It is worth noting, though, the above works all utilize cycle consistency as a prior for test-time optimization, which depends on test shape collection and initial maps. Contrastingly, our approach exploits cycle consistency during training to boost feature learning and poses no constraint on a test.

From this viewpoint, UDMMS [9] is perhaps the most relevant work to ours, as both construct a latent shape during training to inject cycle consistency. The key difference between their work and ours, though, is how the latent shape is constructed. In [9], the authors propose to construct a universal shape in the learned feature space and establish point-wise maps from real shapes to the universal one by training a classifier. As shown in Sec.5, though this construction leads to great performance on matching near-isometric shapes, it suffers from large shape variability in mapping non-isometric ones. On the other hand, we leverage spectral information in estimating point-wise maps. As a result, our approach benefits from more direct usage of intrinsic geometric information encoded in the spectral embedding, yielding better generalization performance.

Dual Map Representations Thanks to the inherent connection and efficient conversion between point-wise and functional maps, it has long been observed that jointly estimating both map representations can improve the mapping quality. For instance, in the original work [37], the authors have already proposed to apply an ICP-like technique on
functional maps. More recent advances take advantage of the multi-scale properties of the eigenbasis of the Laplace-Beltrami operator. In the works [34, 24, 13, 15], conversions are done between spatial domain and a series of spectral domains spanned by eigenfunctions of increasing dimensions. This idea has also been incorporated into DFM. In AttentiveFMaps [28], the authors propose to fuse functional maps of different dimensions by converting them into the common spatial domain. This fusion technique in turn allows training an attention network to dynamically choose the optimal spectral resolution.

The above methods enforce consistency across map representations by iterative projections, which is computationally heavy. By exploiting our theoretical insight on the spectral cycle consistency of DFM, we only introduce a marginal computational overhead to a standard DFM trained on a single spectral resolution. Our method is much lighter and simpler but also shows superior performance in accuracy and generalization.


In this section, we first briefly review the general deep functional maps pipeline. Then we present a theoretical analysis of its cycle consistency.

3.1. Deep Functional Maps

We assume to be given a pair of deformable shapes \( S_1 \) and \( S_2 \), which are discretized as triangular meshes of \( n_1 \) and \( n_2 \) vertices, respectively. The generic deep functional map pipeline introduced in [30] learns a map that spectrally aligns the shapes through the following four steps:

1. Compute the leading \( k \) eigenfunctions of the Laplace-Beltrami operator on each shape, which can be treated as a high-dimensional spectral embedding of the respective shape. The eigenfunctions are stored as matrices \( \Phi_i \in \mathbb{R}^{n_i \times k} \), \( i = 1, 2 \).

2. Instantiate a feature extractor network, \( \mathcal{F}_\theta \), where \( \Theta \) denotes the set of learnable parameters. By feeding forward shapes through \( \mathcal{F}_\theta \), descriptors are obtained and expected to be approximately preserved by the underlying map. In general, we denote \( \mathbf{G}_i = \mathcal{F}_\theta(S_i) \in \mathbb{R}^{n_i \times d} \), \( i = 1, 2 \), where \( d \) is the predefined number of descriptors. They are then projected onto the eigenbasis above, resulting in a couple of coefficient matrices \( \mathbf{A}_i = \Phi_i^\dagger \mathbf{G}_i \in \mathbb{R}^{k \times d} \), \( i = 1, 2 \).

3. Estimate the optimal functional map, \( C \), by solving the following linear system:

\[
C^* = \arg \min_C E_{\text{desc}}(C) + E_{\text{reg}}(C),
\]

where \( E_{\text{desc}}(C) = \| \mathbf{CA}_1 - \mathbf{A}_2 \|^2 \), and \( E_{\text{reg}}(C) \) is the regularization term promoting structural properties of \( C \), e.g., enforcing \( C \) to be commutative with the Laplace-Beltrami operators and to be orthogonal. In particular, we let \( E_{\text{total}}(C) = E_{\text{desc}}(C) + E_{\text{reg}}(C) \).

4. Convert the estimated functional map \( C \) to a point-wise map by conducting nearest neighbor search between the rows of \( \Phi_1 C \) and that of \( \Phi_2 \).

The feature extractor is then trained on a set of training shape pairs, which typically enumerates all pairs from some given training set. The sub-optimization problem in Step (3) is generally decomposed into two parts to circumvent using an iterative solver. Namely, a closed-form solution is obtained either by a least-square estimation w.r.t \( E_{\text{desc}}(C) \) [30], or by solving a sequence of linear systems involving Laplacian commutativity as well [11]. Both correspond to the \( \mathcal{F}_{\text{reg}} \) layer shown in Fig. 2. The rest of \( E_{\text{reg}}(C) \) is then set as the structural loss, corresponding to \( \mathcal{L}(C) \) in Fig. 2.

3.2. Cycle Consistency

We start by giving a formal definition of cycle consistency. We assume to be given a collection of shapes \( S = \{S_i\}_{i=1}^n \) and a set of maps \( T = \{T_{ij}\}_{i,j \in [1..n]} \). We call \( T \) cycle consistent if for any shape \( S_i \) and any closed path \( (i, i_1, i_2, \ldots, i_k, i) \), the map composition along this path \( T_{ij} = T_{ik} \circ \cdots \circ T_{i_1} \circ T_{i_2} \circ \cdots \circ T_{i_k} \) is an identity map on \( S_i \). We can similarly define cycle consistency on functional maps, the only difference is that map composition is given by matrix multiplication and we require the final functional map \( C_{ii} \) to be an identity matrix.

We suppose that deep functional maps are trained on \( S \) with respect to all possible pairs. Then the global energy is given by \( E_{\text{total}}(C) = \sum_{i,j} \| \mathbf{C}_{ij} \mathbf{A}_i - \mathbf{A}_j \|^2 + \sum_{i,j} E_{\text{reg}}(\mathbf{C}_{ij}) \), where \( C = \{ \mathbf{C}_{ij} \}_{i,j \in [1..n]} \) is the set of functional maps among training shapes.

**Proposition 1** If \( E_{\text{total}}(C) = 0 \), then for any shape \( S_i \) and any path \( (i, i_1, i_2, \ldots, i_k, i) \), the map composition \( C_{ii} \) is cycle consistent within the functional space spanned by the columns of \( \mathbf{A}_i \), i.e., \( C_{ii} \mathbf{A}_i = \mathbf{A}_i \).
As a consequence of Prop. 1, when \( \mathbf{A}_i \in \mathbb{R}^{k \times d} \) is of full row rank, then \( \mathbf{C} = \{ \mathbf{C}_{ij} \}_{i,j \in [1..n]} \) is cycle consistent. In practice, we generally set \( d > k \). Moreover, during network training, \( \mathbf{C}_{ij} \) is computed via pseudo-inverse, which implicitly assumes the full-rankness of \( \mathbf{A}_i, \mathbf{A}_j \). We refer readers to the empirical validation in Section 4.2.

We defer the proof of Prop. 1 to Supp. Material. In fact, a similar claim has been formulated and proven in [49] (see Sec.3.1 therein), but in the context of map refinement via promoting cycle consistency.

Though being technically similar, the theoretical argument of [49] and that of ours have fundamentally different implications. More specifically, the former justifies a test-time optimization algorithm, which is used to promote cycle consistency of maps among a fixed test shape collection. While the latter suggests that spectrally cycle consistency has been ensured and further leveraged to enhance the universal feature extractor (independent of the test data) during training in any DFM framework following the generic pipeline presented in Sec. 3.1.

4. Two-branch Deep Functional Maps

It has long been recognized, both theoretically and empirically, that optimizing purely in the spectral domain is not sufficient. As a toy example, a trivial solution attaining global optima can be constructed as follows: Suppose that we have learned a feature extractor \( \mathcal{F}_\Theta \), which returns the respective eigenbasis transformed by a universal \( \mathbf{A}_0 \). That is, \( \mathbf{G}_i = \mathbf{\Phi}_i \mathbf{A}_0, \forall i \), which implies \( \mathbf{A}_i = \mathbf{\Phi}_i \mathbf{G}_i = \mathbf{A}_0, \forall i \). Then we have \( \mathbf{C}_{ij} \equiv \mathbf{I}_k \), which exactly satisfies \( E_{\text{desc}}(\mathbf{C}_{ij}) = E_{\text{reg}}(\mathbf{C}_{ij}) = 0, \forall i, j \). However, it probably induces poor point-wise maps.

In fact, in [37] the authors have proposed to use an ICP-like technique to encourage the estimated functional maps to be induced by some point-wise maps. In [34], the authors propose a spectral upsampling method for map refinement, which essentially converts maps back and forth between spectral and spatial domains. Moreover, the following lemma from [34] sheds light on the necessity of taking both spectral and spatial representations into consideration.

**Lemma 1** Given a pair of shapes \( S_1, S_2 \) each having non-repeating Laplacian eigenvalues, which are the same. A point-wise map \( T : S_1 \to S_2 \) is an isometry if and only if the corresponding functional map \( \mathbf{C} \) in the complete Laplacian basis is both diagonal and orthonormal.

The above lemma suggests that apart from promoting the structural properties of functional maps, it is also critical to enforce them to be associated with certain point-wise maps, or termed as *properness* of functional maps in [40].

Finally, we remark that some recent DFM advances also promote the properness of the resulting spectral maps. For instance, AttentiveFMaps [28] follows the spirit of ZoomOut [34] and explicitly performs a conversion between spectral and spatial map representations across different dimensions of eigenbasis; UDMSM [9] constructs explicitly a universal shape in the feature space, and enforce the spectral map estimation to be consistent with the spatial maps induced via the universal shape.

4.1. Two-branch Map Estimation

In this part, we leverage our observation made in Prop. 1 and propose a novel, simple yet effective design of unsupervised deep functional maps, which introduces a new branch that independently estimates maps from spatial perspective.

Our key insight is that, once cycle consistency is valid and \( \mathbf{A}_i \) is of full row rank, \( \mathbf{A}_i \) can be seen as a functional map from a universal latent shape, \( S_0 \), to \( S_i \). This perspective has been explored in several prior works [49, 23, 24], we provide the following details to be self-contained. The above assumption implies \( \mathbf{C}_{ij} = \mathbf{A}_j \mathbf{A}_i^\dagger \). Then \( \mathbf{C}_{ij} \) can be interpreted as a functional map composition from \( S_i \) to \( S_0 \), followed by a map from \( S_0 \) to \( S_j \). On the other hand, one can align the spectral embeddings of \( S_j \) to that of \( S_i \) by simply transforming the former by \( \mathbf{C}_{ij} \). Indeed, we convert \( \mathbf{C}_{ij} \) into the point-wise map by the nearest neighbor searching between the rows of \( \mathbf{\Phi}_j \mathbf{C}_{ij} \) and that of \( \mathbf{\Phi}_i \). From this point of view, denoting the virtual spectral embedding of the latent shape by \( \mathbf{\Phi}_0 \), \( \mathbf{\Phi}_i \mathbf{A}_i \) can be then treated as the spectral embedding of \( S_i \) aligned to that of \( S_0 \). Therefore, given a pair of shapes \( S_i, S_j \), since we have aligned their eigenbasis to the canonical frame defined by the virtual spectral embedding \( \mathbf{\Phi}_0 \), we can align the spectral embedding of \( S_i \) to \( \mathbf{\Phi}_0 \) by computing \( \mathbf{\Phi}_i \mathbf{A}_i \). Once all the spectral embeddings are aligned to the canonical embedding domain regarding \( \mathbf{\Phi}_0 \), we can compute the soft point-wise map between \( S_i \) and \( S_j \) by nearest neighbor searching between the rows of \( \mathbf{\Phi}_i \mathbf{A}_i \) and those of \( \mathbf{\Phi}_j \mathbf{A}_j \).

Based on the above derivation, given the learned features projected in the spectral domain, \( \mathbf{A}_i, \mathbf{A}_j \), and a pair of indices \( p \in [1..n_i], q \in [1..n_j] \), we can compute point-wise maps. Firstly we compute residual:

\[
\delta_{qp} = \| \mathbf{\Phi}_i[p] \mathbf{A}_i - \mathbf{\Phi}_j[q] \mathbf{A}_j \|_2,
\]

where \( \mathbf{\Phi}_{i}[p] \) denotes the \( p \)-th row of \( \mathbf{\Phi}_i \), and similarly we define \( \mathbf{\Phi}_{j}[q] \). The soft point-wise map \( \Pi \in \mathbb{R}^{n_i \times n_j} \) is then given by:

\[
\Pi(q, p) = \frac{\exp(-\alpha \delta_{qp})}{\sum_{q'} \exp(-\alpha \delta_{qp'})}.
\]

Note that by construction, each row of \( \Pi \) is non-negative and sums up to 1, forming a probability distribution. The parameter \( \alpha \) controls the entropy of each distribution – the smaller/larger \( \alpha \) is, the fuzzier/sharper the distribution is.
Finally, we remark that by combining the \( FMreg \) layer and \( \mathcal{L}(C_1, C_2) \), we have incorporated every factor in Lemma 1 into our design.

**Conceptual Validation**  In this part, we train a network on DT4D-H dataset (see Section 5.1 for details) with our two-branch network, and a single-branch variant without our spatial map estimation branch. We monitor and plot the following quantities along training: (1) Average spectral cycle consistency over sampled triplets, i.e., \( \frac{1}{M} \sum_{(i,j,k)} \left\| C_{ij} C_{jk} C_{ij} - I \right\|^2 / \| I \|^2 \); (2) Average spatial cycle consistency over sampled triplets, i.e., the mean Euclidean deviation from composed maps \( T_{ki} \circ T_{jk} \circ T_{ij} \) to identity map on \( S_i \). Here \( n = 80 \) is the number of training shapes, and \( M = 1000 \) is the number of sampled triplets. The behavior of the blue curves after 4500 iterations verifies our argument that spectrally cycle consistency does not imply spatially cycle consistency. On the other hand, by introducing our two-branch design, the discrepancy is well compensated and evidently better cycle consistencies in both spatial and spectral domains are achieved.

4.3. Updating Scheme of \( \alpha \) in Eqn. (3)

The soft point-map conversion (Eqn. 3) has been applied in several prior works [30, 28], which all set \( \alpha \) to be a manually selected constant. Ideally, we expect \( \Pi \) in Eqn. (3) to be close to a permutation matrix, i.e., each row forms a binary vector. This seems to suggest a preference for a large \( \alpha \). Unfortunately, it would severely hinder network training, since the learned features and maps are of low quality in the early stage. On the other hand, a small \( \alpha \) can alleviate such difficulty but falls short of fully pushing functional maps to be proper. As demonstrated in Section 5.5, neither small nor large \( \alpha \) produces satisfying results.

Based upon the above analysis, we propose a novel updating scheme, which is inspired by curriculum learning [5]. Namely, we initiate a small \( \alpha \) at the beginning of network training and increase it by a constant step size for every fixed number of epochs. As shown in Table 1, 2, 3, our scheme does not rely on hyper-parameter tuning but also achieves state-of-the-art results.

4.4. Implementation Details

We implement our network with PyTorch [39]. We use four DiffusionNet blocks [44] as feature backbone and borrow the functional map block with Laplacian regularizer from [12]. The dimension of the Laplace-Beltrami eigenbasis is set to 50, WKS [4] descriptors are used as the input signal to our network. The dimensions of the input and the output descriptors are both set to 128. During training, the value of the learning rate is set to 2e-4 with ADAM optimizer. In all experiments, we train our method for 10,000 iterations with a batch size of 1.
strategy in Section 4.3, we initialize $\alpha$ to 1 and increase it by 5 per epoch. As indicated in Eqn. (6), We weigh equally the orthogonality loss with respect to $C_1$ and the residual between $C_1$ and $C_2$. More implementation details are provided in the Supp. Material.

5. Experimental Results

In this section, we conduct an extensive set of experiments of non-rigid shape matching on various datasets including humanoids and animals. We test on both near-isometric and non-isometric shape pairs. Our method is compared to a set of competitive baselines including axiomatic, supervised, weakly-supervised, and unsupervised learning methods. We emphasize that in this section, all the maps from the learning-based pipelines are directly inferred from the trained models, without any post-processing procedure. We evaluate the matching results in terms of a mean geodesic error on shapes normalized to unit area. Finally, our point-wise maps are all inferred by converting the output functional maps, as all the other DFM frameworks.

5.1. Datasets

**FAUST_r**: The remeshed version [41] of FAUST dataset [6] contains 100 human shapes. Following [43], it is split into 80/20 for training and testing.

**SCAPE_r**: The remeshed version [41] of SCAPE dataset [2] contains 71 human shapes. Following [43], it is split into 51/20 for training and testing.

**SHREC19_r**: The remeshed version of SHREC19 dataset [33] collects 44 human shapes from 11 independent datasets with distinctive poses and styles. We abandon shape 40 due to its partiality, we test on 407 pairs among the rest 43 shapes, which come with ground-truth.

**DT4D-H** [32]: The remeshed subset of the large scale animation dataset DeformingThings4D [29]. In particular, DT4D-H includes 10 categories of humanoid shapes undergoing significant pose and style variations, forming a challenging benchmark.

**SMAL_r**: The remeshed SMAL dataset [53] contains 49 animal shapes with 8 species. We follow the setting from [28], which splits 29 (5 species) and 20 (3 species) shapes for training and testing.

**TOSCA_r**: The remeshed TOSCA dataset [7] contains multiple shape categories. We choose 4 animal categories, including cat, dog, horse, and wolf to verify the generalization performance of networks trained on SMAL_r. Note that we only infer the intra-category maps, due to the absence of ground-truth inter-category maps.

We refer readers to Supp. Material for visualizations illustrating the variability of the above datasets.

5.2. Near-isometric Shape Matching

In this part, we perform comparisons with an array of non-rigid shape matching methods: (1) Axiomatic methods including ZoomOut [34], BCICP [41], IsoMuSh [17], Smooth Shells [13], CZO [24]; (2) Supervised learning methods including TransMatch [48], GeomFMaps [12], and supervised version of AttentiveFMaps [28]; (3) Un-supervised learning methods including NeuroMorph [14], SyNoRiM [16], Deep Shell [15], AttentiveFMaps [28], UDM SM [9], DUO-FM [11].

For all learning-based methods, we train models on **FAUST_r** and **SCAPE_r** respectively. Table 1 reports results on both standard tests and more challenging generalizations. We observe a trade-off between the two tasks, methods that performs the best in the former (e.g., supervised AttentiveFMaps and UDM SM) tend to over-fit, and therefore suffer poor generalization (especially to SHREC19_r). Meanwhile, our default setting, denoted by **Ours**, achieves reasonable performance in the standard tests but also outperforms the external baselines in 4 out of 4 generalization tests. Especially, in generalizing to SHREC19_r, **Ours** outperforms the external baselines by a large margin, resulting in 41% (3.8 vs. 6.4) and 46% (4.5 vs. 8.4) error reduction upon the second best.

We highlight that post-processing with cycle consistency generally depends on the initialized map quality and the size of the test set (e.g., ≥ 3 shapes). In contrast, we leverage cycle consistency to improve the feature extractor during training. We also report the results from post-processing techniques based on cycle consistency [17, 24, 16] in Table 1. They are significantly outperformed by our method, which is inferred per-pair and without any post-processing.

We further augment the dimension of functional maps in network training to 80 (same as UDM SM), which is beneficial to near-isometric matching [28]. It is evident that **Ours (80 dim)** achieves on-par performance with the regarding state-of-the-art methods in standard tests. On the other hand, due to the significant variability between SHREC19_r and the training sets (see Supp. Material), augmenting dimension leads to worse generalization than before (5.5 vs. 3.8, 5.8 vs. 4.5). Nevertheless, even in this case, our method outperforms the external baselines in all generalization tests by a notable margin.

5.3. Non-isometric Shape Matching

We also train our network on non-isometric datasets, **SMAL_r** and **DT4D-H**, and compare it with the state-of-the-art baselines including DeepShells [15], AttentiveFMaps [28], UDM SM [9] and DUO-FM [11].

**SMAL_r**: We follow the split and input descriptors from [28] (more details are provided in the Supp. Material). Table 2 reports results on **SMAL_r**, our method
Table 1. Mean geodesic errors (×100) on FAUST$_r$, SCAPE$_r$, and SHREC19$_r$. The best and the second best are highlighted.

<table>
<thead>
<tr>
<th>Method</th>
<th>Train</th>
<th>FAUST$_r$</th>
<th>FAUST$_r$</th>
<th>SCAPE$_r$</th>
<th>SCAPE$_r$</th>
<th>SHREC19$_r$</th>
<th>SHREC19$_r$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Test</td>
<td>FAUST$_r$</td>
<td>SCAPE$_r$</td>
<td>SHREC19$_r$</td>
<td>SCAPE$_r$</td>
<td>FAUST$_r$</td>
<td>SHREC19$_r$</td>
</tr>
<tr>
<td>ZM[41]</td>
<td>6.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BCICP[41]</td>
<td>6.4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IsoMuSh[17]</td>
<td>4.4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Smooth Shell[13]</td>
<td>2.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CZO[24]</td>
<td>2.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TransMatch[48]</td>
<td>2.7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GeomFMaps[12]</td>
<td>supervised</td>
<td>2.6</td>
<td>3.3</td>
<td>9.9</td>
<td>3.0</td>
<td>3.0</td>
<td>12.2</td>
</tr>
<tr>
<td>AttentiveFMaps[28]</td>
<td>1.4</td>
<td>2.2</td>
<td>9.4</td>
<td>1.7</td>
<td>1.8</td>
<td>12.2</td>
<td></td>
</tr>
<tr>
<td>NeuroMorph[12]</td>
<td>8.5</td>
<td>28.5</td>
<td>26.3</td>
<td>29.9</td>
<td>18.2</td>
<td>27.6</td>
<td></td>
</tr>
<tr>
<td>SyNoRIM[16]</td>
<td>7.9</td>
<td>21.7</td>
<td>25.5</td>
<td>9.5</td>
<td>24.6</td>
<td>26.8</td>
<td></td>
</tr>
<tr>
<td>Deep Shell[15]</td>
<td>1.7</td>
<td>5.4</td>
<td>27.4</td>
<td>2.5</td>
<td>2.7</td>
<td>23.4</td>
<td></td>
</tr>
<tr>
<td>AttentiveFMaps[28] unsupervised</td>
<td>1.9</td>
<td>2.6</td>
<td>6.4</td>
<td>2.2</td>
<td>2.2</td>
<td>9.9</td>
<td></td>
</tr>
<tr>
<td>UDMSM[9]</td>
<td>1.5</td>
<td>7.3</td>
<td>21.5</td>
<td>2.0</td>
<td>8.6</td>
<td>30.7</td>
<td></td>
</tr>
<tr>
<td>DUO-FM[11]</td>
<td>2.5</td>
<td>4.2</td>
<td>6.4</td>
<td>2.7</td>
<td>2.8</td>
<td>8.4</td>
<td></td>
</tr>
<tr>
<td>Ours</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ours (80 dim)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2. Mean geodesic errors (×100) on SMAL$_r$. The best and the second best are highlighted correspondingly.

<table>
<thead>
<tr>
<th>Method</th>
<th>Train</th>
<th>SMAL$_r$</th>
<th>TOSCA$_r$</th>
</tr>
</thead>
<tbody>
<tr>
<td>GeomFMaps[12]</td>
<td>7.6</td>
<td>24.5</td>
<td></td>
</tr>
<tr>
<td>AttentiveFMaps[28]</td>
<td>5.4</td>
<td>20.9</td>
<td></td>
</tr>
<tr>
<td>DUO-FM[11]</td>
<td>32.8</td>
<td>15.3</td>
<td></td>
</tr>
<tr>
<td>Ours</td>
<td>5.4</td>
<td>7.9</td>
<td></td>
</tr>
</tbody>
</table>

achieves the best performance, which is on-par with AttentiveFMaps [28]. To evaluate generalization performance, we use the trained models to directly infer intra-category maps within TOSCA$_r$. It turns out that AttentiveFMaps and GeomFMaps both suffer from significant performance drops (×3.8 and ×3.2 larger geodesic error). It is also worth noting that DeepShells achieves the second-best generalization score in the relatively simpler task. However, it fails dramatically regarding the base task. In contrast, our method achieves the best balance between learning in difficult non-isometric pairs and generalizing to relatively easy near-isometric pairs.

**DT4D-H:** We follow the train/test (198/95) split of [28], but ignore the categories mousey and ortiz in both train and test, due to the lack of inter-category map labels regarding them, resulting a split of 168/80. We emphasize that we conduct training and test in a category-agnostic manner, i.e., no class label is used, and the training pairs can consist of shapes from arbitrary two categories. This is significantly different from [28], in which training pairs are selected according to clustering information. Obviously, our setting is more practical, but also more challenging. For completeness, we report results under the setting of [28] in Supp. Material and our method outperforms the baselines in both intra- and inter-category evaluation by a notable margin.

Table 3 reports results on **DT4D-H**, in which we preserve 80 shapes for test and train networks with 168 and 80 shapes, respectively. Note that we report mean geodesic errors over all possible test shape pairs, which may undergo significant distortions (see, e.g., Fig. 1). Our method obtains a 67.2% (7.7 vs. 22.4) geodesic error reduction with respect to the second-best baseline. On top of that, we also test the generalization of the trained model on near-isometric benchmarks – our method also generalizes the best in generalization to FAUST$_r$ and SCAPE$_r$. The same pattern is observed when the training set is reduced by more than half. Remarkably, our network trained on the reduced set still outperforms all the baselines trained on the full set.

Overall, we attribute our performance on matching challenging non-isometric shapes (Table 3) and generalizing to unseen shapes (Table 1) to our effort to promote both spec-
5.4. Integration with SURFMNet [43]

Our two-branch design can be easily incorporated into any existing DFM framework following the general design outlined in Sec. 3.1. To demonstrate this, we modify the SURFMNet [43], one of the earliest approaches of unsupervised DFM, by adding our new branch. Table 4 shows the matching accuracy on the four main benchmarks. It is evident that due to the significant distortion undergone among the shapes, all but our method fails to preserve cycle consistency in this demanding test, while our composing map approximates the identity map on meshes. Especially, in the near-isometric cases, we obtain noticeable performance drop. Especially in the case $\alpha = 50$, the network fails to deliver reasonable matching results. We believe it is because a large $\alpha$ amplifies the noise of maps learned at the early training stage.

Then we justify our two-branch network design. Removing spatial branch amounts to training a standard single-branch DFM. To remove the spectral branch, we remove the $FM_{\text{reg}}$ layer and instead use our new branch to compute point-wise maps and convert them to functional maps. In the end, we modify the training loss so that it covers descriptor preservation, commutativity with the Laplace-Beltrami operator, and orthogonality (the latter two compensate the removed $FM_{\text{reg}}$ layer). The accuracy drop reported in the third and fourth row of Table 5 clearly suggests the necessity of our two-branch design.

In our experiments, we always use the full resolution meshes ($\sim 5k$ vertices) and compute in Eqn. (2) with all of the 128 descriptors. We anticipate that efficiency can become an issue when the input mesh resolution is high, and/or we would like to increase the size of learned descriptors. Therefore, we test the robustness of our pipeline with respect to down-sampling, which is commonly used in functional maps-based frameworks [34, 28]: 1) We down-sample 3000 vertices on each shape via furthest point sampling; 2) In order to down-sample the feature dimension, we operate as the following during training: given a $A_{1}$, $A_{2}$, we perform SVD on $A_{1}$, i.e., $A_{1} = V_{1} \Sigma_{1} V_{1}^{T}$, then we set $\hat{A}_{1} = A_{1} \hat{V}_{1}$, and set $\hat{A}_{2} = A_{2} \hat{V}_{1}$, where $\hat{V}_{1}$ is the first $m$ columns of $V_{1}$. We set $m = 30$ by replacing $A_{i}$ with $\hat{A}_{i}$ in Eqn. (2). The results in the bottom two rows show that the above operation has a relatively minor effect on the performance, proving the robustness of our method.

5.5. Ablation Study

In this section, we present a set of ablation studies consisting of two parts. The first part verifies the rationality of our method, and the second part demonstrates the robustness of our method. We conduct all experiments on SMAL$_r$ dataset [53].

First of all, instead of using the updating scheme in Sec.

### Table 3. Mean geodesic errors ($\times 100$) on DT4D-H. The best and the second best are highlighted correspondingly.

<table>
<thead>
<tr>
<th>Method</th>
<th>Train</th>
<th>DT4D-H</th>
<th>FAUST$_r$</th>
<th>SCAPE$_r$</th>
<th>DT4D-H</th>
<th>FAUST$_r$</th>
<th>SCAPE$_r$</th>
</tr>
</thead>
<tbody>
<tr>
<td>SURFMNet</td>
<td>6.8</td>
<td>65.4</td>
<td>76.5</td>
<td>90.4</td>
<td>76.5</td>
<td>90.4</td>
<td>90.4</td>
</tr>
<tr>
<td>AttentiveFM [28]</td>
<td>6.8</td>
<td>65.4</td>
<td>76.5</td>
<td>90.4</td>
<td>76.5</td>
<td>90.4</td>
<td>90.4</td>
</tr>
<tr>
<td>UDMMSM [9]</td>
<td>6.8</td>
<td>65.4</td>
<td>76.5</td>
<td>90.4</td>
<td>76.5</td>
<td>90.4</td>
<td>90.4</td>
</tr>
<tr>
<td>DUO-FM [11]</td>
<td>6.8</td>
<td>65.4</td>
<td>76.5</td>
<td>90.4</td>
<td>76.5</td>
<td>90.4</td>
<td>90.4</td>
</tr>
<tr>
<td>Ours</td>
<td>6.8</td>
<td>65.4</td>
<td>76.5</td>
<td>90.4</td>
<td>76.5</td>
<td>90.4</td>
<td>90.4</td>
</tr>
</tbody>
</table>

### Table 4. Mean geodesic errors ($\times 100$) of SURFMNet and our variant trained on 4 datasets.

<table>
<thead>
<tr>
<th>Method</th>
<th>SURFMNet</th>
<th>SURFMNet + Ours</th>
</tr>
</thead>
<tbody>
<tr>
<td>FAUST$_r$</td>
<td>6.0</td>
<td>3.5</td>
</tr>
<tr>
<td>SCAPE$_r$</td>
<td>6.8</td>
<td>3.4</td>
</tr>
<tr>
<td>SMAL$_r$</td>
<td>20.4</td>
<td>13.3</td>
</tr>
<tr>
<td>DT4D-H</td>
<td>18.3</td>
<td>15.0</td>
</tr>
</tbody>
</table>

To conclude, we provide a theoretical justification for spectral cycle consistency in DFM. To compensate for the discrepancy of purely spectrally cycle consistency, we formulate a spectral latent shape that allows the alignment of the spectral embeddings of each shape to a canonical embedding domain. Based on this construction, we introduce a two-branch architecture for estimating maps in both spec-
The resulting network is simple, computationally efficient, and compatible with most existing DFM frameworks. We demonstrate the effectiveness of our framework through a comprehensive set of experiments, showing significant improvements upon state-of-the-art approaches in terms of accuracy, consistency, and generalization performance.

It is worth noting, though, our method is developed upon clean and complete meshes. It would be interesting for future work to investigate the utility of our approach in more general tasks, involving partial meshes, noisy point clouds, and other representations.

Acknowledgement This work was supported in part by the National Natural Science Foundation of China under contract No. 62171256 and Shenzhen Key Laboratory of next-generation interactive media innovative technology (No. ZDSYS20210623092001004), and in part by the ERC Starting Grant No. 758800 (EXPROTEA) and the ANR AI Chair AIGRETTE.

References


[34] Simone Melzi, Jing Ren, Emanuele Rodolà, Peter Wonka, and Maks Ovsjanikov. Zoomout: Spectral upsampling for efficient shape correspondence. Proc. SIGGRAPH Asia, 2019. 3, 4, 6, 7, 8


[40] Jing Ren, Simone Melzi, Peter Wonka, and Maks Ovsjanikov. Discrete optimization for shape matching, 2021. 4


[43] Jean-Michel Roufosse, Abhishek Sharma, and Maks Ovsjanikov. Unsupervised deep learning for structured shape matching. ICCV, 2019. 1, 2, 6, 8


[53] Silvia Zuffi, Angjoo Kanazawa, David W Jacobs, and Michael J Black. 3d menagerie: Modeling the 3d shape and pose of animals. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 6365–6373, 2017. 6, 8