

Spatially and Spectrally Consistent Deep Functional Maps

Mingze Sun¹ Shiwei Mao¹ Puhua Jiang^{1, 2} Maks Ovsjanikov³ Ruqi Huang^{1*}

¹Tsinghua Shenzhen International Graduate School, China

²Peng Cheng Laboratory, China

³LIX, 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/rqhuang88/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 [33], statistical shape analysis [2, 6], to name a few. In order to establish correspondences between shapes undergoing non-rigid deformations, early approaches [49, ?, 50] 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 accu-



Figure 1. We train unsupervised non-rigid shape matching pipelines on the challenging non-isometric dataset DT4D-H, which involves 8 categories of humanoid shapes undergoing significant distortions. Correspondences are visualized by texture transfer. Our method (bottom row) outperforms the state-of-the-art, AttentiveFMaps [29] (middle row) and UDMSM [9] (top row), by a large margin.

racy, 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 [32]. Functional maps [39], 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 [20, 45, 18, 13, 9, 29] 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 [23] and functional [22, 25] 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-

*Corresponding author: rqhuang@sz.tsinghua.edu.cn

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 [45] 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 [39] and then significantly extended in follow-up works (see, e.g., [40]). 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 [27, 22, 8, 44, 38].

Deep Functional Maps In contrast to axiomatic approaches that use hand-crafted features [49, ?], the deep functional maps approach, pioneered by FMNet [32], 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 [20, 45] demonstrate that it is sufficient to learn from geometric map priors. More recently, with the development of robust mesh feature extractors [47], more frameworks [9, 29, 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 [26, 56, 37, 55, 18], as well as associate cycle consistency with low-rank properties of matrices encoding map networks [23, 53, 28].

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 [52, 22, 24, 48]. Some recent learning-based approach [21] 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, UDMMSM [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 [39], 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 [36, 25, 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 [29], 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.

3. Cycle Consistency of Deep Functional Maps

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 [32] 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}_Θ , where Θ denotes the set of learnable parameters. By feeding forward shapes through \mathcal{F}_Θ , 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, \mathbf{C}^* , by solving the following linear system:

$$\mathbf{C}^* = \arg \min_{\mathbf{C}} E_{\text{desc}}(\mathbf{C}) + E_{\text{reg}}(\mathbf{C}), \quad (1)$$

where $E_{\text{desc}}(\mathbf{C}) = \|\mathbf{C}\mathbf{A}_1 - \mathbf{A}_2\|^2$, and $E_{\text{reg}}(\mathbf{C})$ is the regularization term promoting structural properties

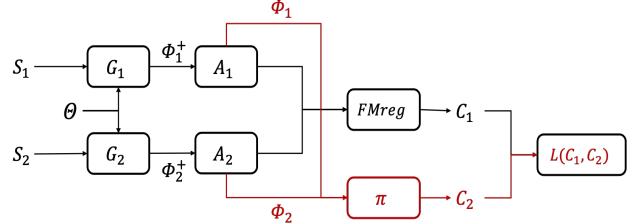


Figure 2. Overview of our two-branch DFM network. The part colored in black corresponds to a standard DFM, we introduce a novel branch, colored in red, that estimates maps from spatial perspective, our loss function is defined as Eqn.(6). Φ_i, Φ_i^\dagger indicate projection of the regarding features into the spatial and spectral domain, respectively. See more details in Sec. 4.2.

of \mathbf{C} , e.g., enforcing \mathbf{C} to be commutative with the Laplace-Beltrami operators and to be orthogonal. In particular, we let $E_{\text{total}}(\mathbf{C}) = E_{\text{desc}}(\mathbf{C}) + E_{\text{reg}}(\mathbf{C})$.

4. Convert the estimated functional map \mathbf{C} to a pointwise map by conducting nearest neighbor search between the rows of $\Phi_1 \mathbf{C}$ and that of Φ_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 $E_{\text{desc}}(\mathbf{C})$ [32], or by solving a sequence of linear systems involving Laplacian commutativity as well [11]. Both correspond to the $FMreg$ layer shown in Fig. 2. The rest of $E_{\text{reg}}(\mathbf{C})$ is then set as the structural loss, corresponding to $\mathcal{L}(\mathbf{C})$ 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 $\mathcal{S} = \{S_i\}_{i=1}^n$ and a set of maps $\mathcal{T} = \{T_{ij}\}_{i,j \in [1..n]}$. We call \mathcal{T} cycle consistent if for any shape S_i , and any closed path $(i, i_1, i_2, \dots, i_k, i)$, the map composition along this path $T_{ii} = T_{i_k i} \circ \dots \circ T_{i_1 i_2} \circ T_{i_1 i}$ 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 \mathcal{S} with respect to all possible pairs. Then the global energy is given by $E_{\text{total}}(\mathcal{C}) = \sum_{i,j} \|C_{ij} \mathbf{A}_i - \mathbf{A}_j\|^2 + \sum_{i,j} E_{\text{reg}}(C_{ij})$, where $\mathcal{C} = \{C_{ij}\}_{i,j \in [1..n]}$ is the set of functional maps among training shapes.

Proposition 1 *If $E_{\text{total}}(\mathcal{C}) = 0$, then for any shape S_i , and any path $(i, i_1, i_2, \dots, i_k, i)$, 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. 2, when $\mathbf{A}_i \in \mathbb{R}^{k \times d}$ is of full row rank, then $\mathcal{C} = \{\mathbf{C}_{ij}\}_{i,j \in [1..n]}$ 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 Sec. 4.2.

We defer the proof of Prop. 2 to Supp. Material. In fact, a similar claim has been formulated and proven in [52] (see Sec. 3.1 therein), but in the context of map refinement via promoting cycle consistency.

Though being technically similar, the theoretical argument of [52] 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 spectral 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}_Θ , which returns the respective eigenbasis transformed by a universal \mathbf{A}_0 . That is, $\mathbf{G}_i = \Phi_i \mathbf{A}_0, \forall i$, which implies $\mathbf{A}_i = \Phi_i^\dagger \mathbf{G}_i = \mathbf{A}_0, \forall i$. Then we have $\mathbf{C}_{ij} \equiv I_k$, 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 [39] 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 [36], 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 [36] 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 $T : S_1 \rightarrow S_2$ 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 [42].

Finally, we remark that some recent DFM advances also promote the properness of the resulting spectral maps.

For instance, AttentiveFMaps [29] follows the spirit of ZoomOut [36] and explicitly performs a conversion between spectral and spatial map representations across different dimensions of eigenbasis; UDMMSM [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. 2 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 [52, 24, 25], 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 $\Phi_j \mathbf{C}_{ij}$ and that of Φ_i . From this point of view, denoting the virtual spectral embedding of the latent shape by Φ_0 , $\Phi_i \mathbf{A}_i$ 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 Φ_0 , we can align the spectral embedding of S_i to Φ_0 by computing $\Phi_i \mathbf{A}_i$. Once all the spectral embeddings are aligned to the canonical embedding domain regarding Φ_0 , we can compute the soft point-wise map between S_i and S_j by nearest neighbor searching between the rows of $\Phi_i \mathbf{A}_i$ and those of $\Phi_j \mathbf{A}_j$.

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:

$$\delta_{qp} = \|\Phi_i[p] \mathbf{A}_i - \Phi_j[q] \mathbf{A}_j\|_2, \quad (2)$$

where $\Phi_i[p]$ denotes the p -th row of Φ_i , and similarly we define $\Phi_j[q]$. The soft point-wise map $\Pi \in \mathbb{R}^{n_j \times n_i}$ is then given by:

$$\Pi(q, p) = \frac{\exp(-\alpha \delta_{qp})}{\sum_{p'} \exp(-\alpha \delta_{qp'})}. \quad (3)$$

Note that by construction, each row of Π is non-negative and sums up to 1, forming a probability distribution. The parameter α controls the entropy of each distribution – the smaller/larger α is, the fuzzier/sharper the distribution is.

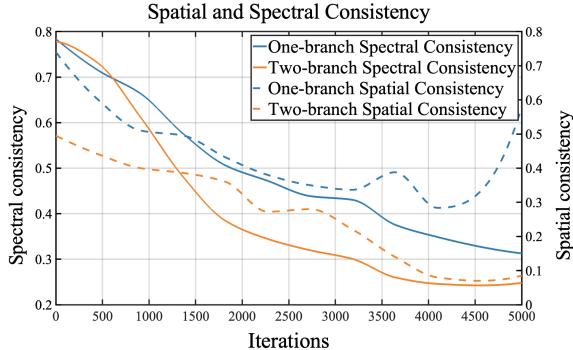


Figure 3. We train our two-branch DFM and a vanilla single-branch version on **DT4D-H** and monitor spectrally and spatially cycle consistency along the training.

Instead of manually tuning the optimal α , we propose a learning scheme that dynamically controls α over training, which is inspired by curriculum learning [5]. We defer the respective details to Sec. 4.3.

We convert the soft point-wise map to a functional map by

$$\mathbf{C}_2 = \Phi_j^\dagger \Pi \Phi_i. \quad (4)$$

In the end, we enforce \mathbf{C}_2 to be consistent with \mathbf{C}_1 , the intermediate output from the *FMreg* layer.

To summarize, thanks to the spectral cycle consistency we identify in Sec. 3.2, we are allowed to construct a spectral latent shape and induce spatial maps via it. By enforcing the spatial estimations to be consistent with the spectral ones, we obtain a spatially and spectrally consistent deep functional maps framework.

4.2. Network Design

As shown in Fig. 2, our two-branch network is built upon a standard DFM framework. In the following, we denote by \mathbf{C}_1 and \mathbf{C}_2 the estimated functional maps from the original *FMreg* layer and our novel branch, respectively.

Specifically, we use DiffusionNet [47] as our feature extractor. And WKS [?] descriptors are fed into it as initialization of learned features. We borrow the *FMreg* layer from [12]. It takes both $E_{\text{desc}}(\mathbf{C})$ and commutativity with the Laplace-Beltrami operator into consideration, where the latter is given as:

$$L_{\text{lap}} = \|\mathbf{C}_1 \Lambda_1 - \Lambda_2 \mathbf{C}_1\|^2, \quad (5)$$

where Λ_1 and Λ_2 are diagonal matrices of the Laplace-Beltrami eigenvalues on the two shapes.

The estimation of \mathbf{C}_2 has been described in detail in Sec. 4.1. In the end, we formulate the training loss as:

$$\mathcal{L}(\mathbf{C}_1, \mathbf{C}_2) = \|\mathbf{C}_1^T \mathbf{C}_1 - \mathbf{I}\|^2 + \|\mathbf{C}_1 - \mathbf{C}_2\|^2, \quad (6)$$

where the first term promotes the orthogonality of \mathbf{C}_1 , while the second term promotes the consistency between the functional maps estimated from different branches.

Finally, we remark that by combining the *FMreg* layer and $\mathcal{L}(\mathbf{C}_1, \mathbf{C}_2)$, 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 Sec. 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., $\frac{1}{M} \sum_{(i,j,k)} \|\mathbf{C}_{ki} \mathbf{C}_{jk} \mathbf{C}_{ij} - \mathbf{I}\|^2 / \|\mathbf{I}\|^2$; (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 α in Eqn. (3)

The soft point-map conversion (Eqn. 3) has been applied in several prior works [32, 29], which all set α to be a manually selected constant. Ideally, we expect Π 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 α . 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 α can alleviate such difficulty but falls short of fully pushing functional maps to be proper. As demonstrated in Sec. 5.5, neither small nor large α 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 α at the beginning of network training and increase it by a constant step size for every fixed number of epochs. As shown in Tab. 1, 2, 3, our scheme does not rely on hyperparameter tuning but also achieves state-of-the-art results.

4.4. Implementation Details

We implement our network with PyTorch [41]. We use four DiffusionNet blocks [47] 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 [?] 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. Following the learning

strategy in Sec. 4.3, we initialize α to 1 and increase it by 5 per epoch. As indicated in Eqn. (6), We weigh equally the orthogonality loss with respect to \mathbf{C}_1 and the residual between \mathbf{C}_1 and \mathbf{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 [43] of FAUST dataset[6] contains 100 human shapes. Following [45], it is split into 80/20 for training and testing.

SCAPE_r: The remeshed version [43] of SCAPE dataset[2] contains 71 human shapes. Following [45], it is split into 51/20 for training and testing.

SHREC19_r: The remeshed version of SHREC19 dataset [35] 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 [34]: The remeshed subset of the large scale animation dataset DeformingThings4D [31]. In particular, DT4D-H includes 10 categories of humanoid shapes undergoing significant pose and style variances, forming a challenging benchmark.

SMAL_r: The remeshed SMAL dataset [57] contains 49 animal shapes with 8 species. We follow the setting from [29], 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 [36], BCICP [43], IsoMuSh [17], Smooth Shells [13], CZO [25]; (2) Supervised learning methods including TransMatch [51], GeomFMaps [12], and supervised version of AttentiveFMaps [29]; (3) Unsupervised learning methods including NeuroMorph [14], SyNoRiM [16], Deep Shell [15], AttentiveFMaps [29], UDMSM [9], DUO-FM [11].

For all learning-based methods, we train models on **FAUST_r** and **SCAPE_r** respectively. Tab. 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 UDMSM) tend to overfit, and therefore suffer poor generalization (especially to **SREHC19_r**). Meanwhile, our default setting, denoted by **Ours**, achieves reasonable performance in the standard tests but also outperforms the external baselines in 3 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, 25, 16] in Tab. 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 UDMSM), which is beneficial to near-isometric matching [29]. 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 [29], UDMSM [9] and DUO-FM [11].

SMAL_r: We follow the split and input descriptors from [29] (more details are provided in the Supp. Material). Tab. 2 reports results on **SMAL_r**, our method

Table 1. Mean geodesic errors ($\times 100$) on FAUST_r, SCAPE_r, and SHREC19_r. The **best** and the **second best** are highlighted.

Method	Train Test	FAUST_r			SCAPE_r		
		FAUST_r	SCAPE_r	SHREC19_r	SCAPE_r	FAUST_r	SHREC19_r
ZM[36]		6.1	\	\	7.5	\	\
BCICP[43]		6.4	\	\	11.0	\	\
IsoMuSh[17]		4.4	\	\	5.6	\	\
Smooth Shell[13]		2.5	\	\	4.7	\	\
CZO[25]		2.2	\	\	2.5	\	\
TransMatch[51]		2.7	33.6	21.0	18.3	18.6	38.8
GeomFMaps[12]	supervised	2.6	3.3	9.9	3.0	3.0	12.2
AttentiveFMaps[29]		1.4	2.2	9.4	1.7	1.8	12.2
NeuroMorph[14]		8.5	28.5	26.3	29.9	18.2	27.6
SyNoRiM[16]		7.9	21.7	25.5	9.5	24.6	26.8
Deep Shell[15]		1.7	5.4	27.4	2.5	2.7	23.4
AttentiveFMaps[29]	unsupervised	1.9	2.6	6.4	2.2	2.2	9.9
UDMSM[9]		1.5	7.3	21.5	2.0	8.6	30.7
DUO-FM[11]		2.5	4.2	6.4	2.7	2.8	8.4
Ours		2.3	2.6	3.8	2.4	2.5	4.5
Ours (80 dim)		1.7	2.6	5.5	2.2	2.0	5.8

Table 2. Mean geodesic errors ($\times 100$) on SMAL_r. The **best** and the **second best** are highlighted correspondingly.

Method	Train Test	SMAL_r	
		SMAL_r	TOSCA_r
DeepShell[15]		29.3	8.7
GeomFMaps[12]		7.6	24.5
AttentiveFMaps[29]		5.4	20.9
UDMSM[9]		24.6	21.7
DUO-FM[11]		32.8	15.3
Ours		5.4	7.9

achieves the best performance, which is on-par with AttentiveFMaps [29]. 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 ($\times 3.8$ and $\times 3.2$ 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 [29], 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 [29], 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 [29] in Supp. Material

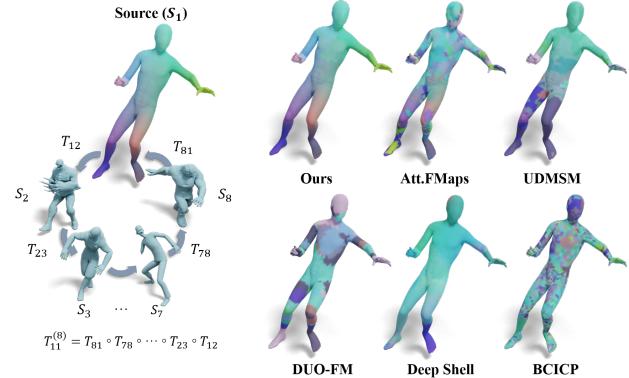


Figure 4. Qualitative evaluation of spatial cycle consistency of different methods. Even composed along a path of 8 highly deformed shapes, our resulting map remains close to identity, while all the baselines fail significantly.

and our method outperforms the baselines in both intra- and inter-category evaluation by a notable margin.

Tab. 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 (Tab. 3) and generalizing to unseen shapes (Tab. 1) to our effort to promote both spec-

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

Method	Train	DT4D-H (168)			DT4D-H (80)		
	Test	DT4D-H	FAUST_r	SCAPE_r	DT4D-H	FAUST_r	SCAPE_r
DeepShell[15]		27.0	4.9	6.5	29.3	4.7	7.0
AttentiveFMaps[29]		25.7	3.4	6.4	28.9	2.7	6.3
UDMSM[9]		46.8	43.3	47.9	49.7	42.5	40.0
DUO-FM[11]		22.4	10.0	12.2	24.7	8.0	9.2
Ours		7.7	3.1	6.1	9.0	2.6	6.2

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

Method	SURFMNet	SURFMNet + Ours
FAUST_r	6.0	3.5
SCAPE_r	6.8	3.4
SMAL_r	20.4	13.3
DT4D-H	18.3	15.0

tral and spatial cycle consistency. Especially, the isometry assumption is likely violated in the former case, thus cycle consistency, as a generic prior, plays an important role of regularizing maps.

As an illustration, we present a qualitative evaluation on the point-wise cycle consistency in Fig. 4. We sample 8 shapes from the test set of DT4D-H (one from each category) and compose the maps along the path ($S_1 \rightarrow S_2 \rightarrow \dots \rightarrow S_8 \rightarrow S_1$) with respect to different approaches. It is evident that due to the significant distortion undergoing among the shapes, all but our method fail to preserve cycle consistency in this demanding test, while our composing map approximates the identity map on S_1 . It also aligns nicely with the quantitative results reported in Tab. 3.

5.4. Integration with SURFMNet [45]

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 [45], one of the earliest approaches of unsupervised DFM, by adding our new branch. Tab. 4 shows the matching accuracy on the four main benchmarks. It is evident that in every case, incorporating our design leads to significant error reduction ranging from 18% to 50%. Especially, in the near-isometric cases, we obtain 41.6% and 50% error reduction respectively. Note that the absolute scores, 0.035, 0.034, are reasonable even compared to the state-of-the-art results reported in Tab. 1.

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 [57].

First of all, instead of using the updating scheme in

Sec. 4.3, we test the performance of our pipeline using two fixed values of α in Eqn. (3): $\alpha = 1$ and $\alpha = 50$. Compared to our proposed model, the two variants yield a noticeable performance drop. Especially in the case $\alpha = 50$, the network fails to deliver reasonable matching results. We believe it is because a large α 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 *FMreg* 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 *FMreg* layer). The accuracy drop reported in the third and fourth row of Tab. 6 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 [36, 29]: 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 $\mathbf{A}_1, \mathbf{A}_2$, we perform SVD on \mathbf{A}_1 , i.e., $\mathbf{A}_1 = U_1 \Sigma_1 V_1^T$, then we set $\hat{\mathbf{A}}_1 = \mathbf{A}_1 \hat{V}_1$, and set $\hat{\mathbf{A}}_2 = \mathbf{A}_2 \hat{V}_1$, where \hat{V}_1 is the first m columns of V_1 . We set $m = 30$ by replacing \mathbf{A}_i with $\hat{\mathbf{A}}_i$ 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.

6. Conclusion

To conclude, we provide a theoretical justification for spectral cycle consistency in DFM. To compensate for the discrepancy of purely spectral 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-

Table 5. Mean geodesic errors ($\times 100$) on different ablated settings, the models are all train on **SMAL_r**.

Ours	5.4
$\alpha = 1$	6.6
$\alpha = 50$	35.2
remove spatial branch	33.4
remove spectral branch	14.3
Vertex down-sampling	5.7
Feature down-sampling	5.9

tral and spatial domains within DFM. 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

- [1] Noam Aigerman, Roi Poranne, and Yaron Lipman. Seamless Surface Mappings. *ACM Transactions on Graphics (TOG)*, 34(4):72, 2015. [1](#)
- [2] Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis. SCAPE: Shape Completion and Animation of People. In *ACM Transactions on Graphics (TOG)*, volume 24, pages 408–416. ACM, 2005. [1, 6](#)
- [3] Souhaib Attaiki, Gautam Pai, and Maks Ovsjanikov. Dpfm: Deep partial functional maps. In *2021 International Conference on 3D Vision (3DV)*, pages 175–185. IEEE, 2021. [2](#)
- [4] Mathieu Aubry, Ulrich Schlickewei, and Daniel Cremers. The Wave Kernel Signature: A Quantum Mechanical Approach to Shape Analysis. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 1626–1633. IEEE, 2011.
- [5] Yoshua Bengio, Jerome Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *International Conference on Machine Learning*, 2009. [5](#)
- [6] Federica Bogo, Javier Romero, Matthew Loper, and Michael J. Black. FAUST: Dataset and evaluation for 3D mesh registration. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Piscataway, NJ, USA, June 2014. IEEE. [1, 6](#)
- [7] Alexander M Bronstein, Michael M Bronstein, and Ron Kimmel. *Numerical geometry of non-rigid shapes*. Springer Science & Business Media, 2008. [6](#)
- [8] Oliver Burghard, Alexander Dieckmann, and Reinhard Klein. Embedding shapes with Green’s functions for global shape matching. *Computers & Graphics*, 68:1–10, 2017. [2](#)
- [9] D. Cao and F. Bernard. Unsupervised deep multi-shape matching. In *European Conference on Computer Vision (ECCV)*, 2022. [1, 2, 4, 6, 7, 8, 13, 14](#)
- [10] Bailind Deng, Yuxin Yao, Roberto M. Dyke, and Juyong Zhang. A survey of non-rigid 3d registration. 2022. [1](#)
- [11] Nicolas Donati, Etienne Corman, and Maks Ovsjanikov. Deep orientation-aware functional maps: Tackling symmetry issues in shape matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 742–751, 2022. [2, 3, 6, 7, 8, 14](#)
- [12] Nicolas Donati, Abhishek Sharma, and Maks Ovsjanikov. Deep geometric functional maps: Robust feature learning for shape correspondence. In *CVPR*, 2020. [5, 6, 7, 13, 14](#)
- [13] Marvin Eisenberger, Zorah Lahner, and Daniel Cremers. Smooth shells: Multi-scale shape registration with functional maps. In *CVPR*, 2020. [1, 3, 6, 7, 14](#)
- [14] Marvin Eisenberger, David Novotny, Gael Kerchenbaum, Patrick Labatut, Natalia Neverova, Daniel Cremers, and Andrea Vedaldi. Neuromorph: Unsupervised shape interpolation and correspondence in one go. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7473–7483, 2021. [6, 7, 14](#)
- [15] Marvin Eisenberger, Aysim Toker, Laura Leal-Taixé, and Daniel Cremers. Deep shells: Unsupervised shape correspondence with optimal transport. *Advances in Neural Information Processing Systems*, 33:10491–10502, 2020. [3, 6, 7, 8, 13, 14](#)
- [16] Jiahui Huang et. al. Multiway non-rigid point cloud registration via learned functional map synchronization, 2022. [6, 7, 14](#)
- [17] Maolin Gao, Zorah Lahner, Johan Thunberg, Daniel Cremers, and Florian Bernard. Isometric multi-shape matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14183–14193, 2021. [6, 7, 14](#)
- [18] Dvir Ginzburg and D. Raviv. Cyclic functional mapping: Self-supervised correspondence between non-isometric deformable shapes. In *ECCV*, 2020. [1, 2](#)
- [19] Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. 3d-coded: 3d correspondences by deep deformation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 230–246, 2018. [14](#)
- [20] Oshri Halimi, Or Litany, Emanuele Rodol'a, Alex Bronstein, and Ron Kimmel. Unsupervised learning of dense shape correspondence. In *CVPR*, 2019. [1, 2, 14](#)
- [21] Jiahui Huang, Tolga Birdal, Zan Gojcic, Leonidas J. Guibas, and Shi-Min Hu. Multiway Non-rigid Point Cloud Registration via Learned Functional Map Synchronization. *IEEE*

- Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2022. 2
- [22] Qixing Huang, Fan Wang, and Leonidas Guibas. Functional map networks for analyzing and exploring large shape collections. *ACM Trans. Gr.*, 33(4), 2014. 1, 2
- [23] Qi-Xing Huang and Leonidas Guibas. Consistent shape maps via semidefinite programming. In *Computer Graphics Forum*, volume 32, pages 177–186. Wiley Online Library, 2013. 1, 2
- [24] Ruqi Huang, Panos Achlioptas, Leonidas Guibas, and Maks Ovsjanikov. Limit shapes – a tool for understanding shape differences and variability in 3d model collections. *Computer Graphics Forum*, 38(5):187–202, 2019. 2, 4
- [25] Ruqi Huang, Jing Ren, Peter Wonka, and Maks Ovsjanikov. Consistent zoomout: Efficient spectral map synchronization. In *Computer Graphics Forum*, volume 39, pages 265–278. Wiley Online Library, 2020. 1, 3, 4, 6, 7, 14
- [26] Daniel F Huber. *Automatic three-dimensional modeling from reality*. PhD thesis, Citeseer, 2002. 2
- [27] Artiom Kovnatsky, Michael M Bronstein, Alexander M Bronstein, Klaus Glashoff, and Ron Kimmel. Coupled quasi-harmonic bases. In *Computer Graphics Forum*, volume 32, pages 439–448, 2013. 2
- [28] Spyridon Leonardos, Xiaowei Zhou, and Kostas Daniilidis. Distributed consistent data association via permutation synchronization. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2645–2652. IEEE, 2017. 2
- [29] Lei Li, Nicolas Donati, and Maks Ovsjanikov. Learning multi-resolution functional maps with spectral attention for robust shape matching. In *Advances in Neural Information Processing Systems*, 2022. 1, 2, 3, 4, 5, 6, 7, 8, 12, 13, 14
- [30] Qinsong Li, Shengjun Liu, Ling Hu, and Xinru Liu. Shape correspondence using anisotropic chebyshev spectral cnns. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14658–14667, 2020. 14
- [31] Yang Li, Hikari Takehara, Takafumi Taketomi, Bo Zheng, and Matthias Nießner. 4dcomplete: Non-rigid motion estimation beyond the observable surface. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12706–12716, 2021. 6
- [32] Or Litany, Tal Remez, Emanuele Rodolà, Alexander M. Bronstein, and Michael M. Bronstein. Deep functional maps: Structured prediction for dense shape correspondence. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 5660–5668, 2017. 1, 2, 3, 5, 14
- [33] Rabinovich M., Poranne R., Panozzo D., and Sorkine-Hornung O. Scalable locally injective mappings. In *ACM Transactions on Graphics*, 2017. 1
- [34] Robin Magnet, Jing Ren, Olga Sorkine-Hornung, and Maks Ovsjanikov. Smooth non-rigid shape matching via effective dirichlet energy optimization. In *International Conference on 3D Vision*, 2022. 6
- [35] Simone Melzi, Riccardo Marin, Emanuele Rodolà, Umberto Castellani, Jing Ren, Adrien Poulenard, Peter Wonka, and Maks Ovsjanikov. Shrec 2019: Matching humans with different connectivity. In *Eurographics Workshop on 3D Object Retrieval*, volume 7, page 3, 2019. 6
- [36] 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, 14
- [37] Andy Nguyen, Mirela Ben-Chen, Katarzyna Welnicka, Yinyu Ye, and Leonidas Guibas. An optimization approach to improving collections of shape maps. In *Computer Graphics Forum*, volume 30, pages 1481–1491. Wiley Online Library, 2011. 2
- [38] Dorian Nogneng and Maks Ovsjanikov. Informative descriptor preservation via commutativity for shape matching. *Computer Graphics Forum*, 36(2):259–267, 2017. 2
- [39] Maks Ovsjanikov, Mirela Ben-Chen, Justin Solomon, Adrian Butscher, and Leonidas Guibas. Functional Maps: A Flexible Representation of Maps Between Shapes. *ACM Transactions on Graphics (TOG)*, 31(4):30, 2012. 1, 2, 4
- [40] Maks Ovsjanikov, Etienne Corman, Michael Bronstein, Emanuele Rodolà, Mirela Ben-Chen, Leonidas Guibas, Frederic Chazal, and Alex Bronstein. Computing and processing correspondences with functional maps. In *ACM SIGGRAPH 2017 Courses*, page 5. ACM, 2017. 2
- [41] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Linand N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. Pytorch: An imperative style, high-performance deep learning library, 2019. 5, 14
- [42] Jing Ren, Simone Melzi, Peter Wonka, and Maks Ovsjanikov. Discrete optimization for shape matching, 2021. 4
- [43] Jing Ren, Adrien Poulenard, Peter Wonka, and Maks Ovsjanikov. Continuous and orientation-preserving correspondences via functional maps. *ACM Trans. Graph.*, 37(6):248:1–248:16, Dec. 2018. 6, 7, 14
- [44] Emanuele Rodolà, Luca Cosmo, Michael M Bronstein, Andrea Torsello, and Daniel Cremers. Partial functional correspondence. In *Computer Graphics Forum*, volume 36, pages 222–236. Wiley Online Library, 2017. 2
- [45] Jean-Michel Roufosse, Abhishek Sharma, and Maks Ovsjanikov. Unsupervised deep learning for structured shape matching. *ICCV*, 2019. 1, 2, 6, 8, 14, 15
- [46] Abhishek Sharma and Maks Ovsjanikov. Weakly supervised deep functional maps for shape matching. In *NeurIPS*, 2020. 14
- [47] Nicholas Sharp, Souhaib Attaiki, Keenan Crane, and Maks Ovsjanikov. Diffusionnet: Discretization agnostic learning on surfaces. *ACM Transactions on Graphics*, 2022. 2, 5, 15
- [48] Meged Shoham, Amir Vaxman, and Mirela Ben-Chen. Hierarchical functional maps between subdivision surfaces. In *Computer Graphics Forum*, volume 38, pages 55–73. Wiley Online Library, 2019. 2
- [49] Jian Sun, Maks Ovsjanikov, and Leonidas Guibas. A Concise and Provably Informative Multi-Scale Signature Based on Heat Diffusion. In *Computer graphics forum*, volume 28, pages 1383–1392, 2009. 1, 2

- [50] Federico Tombari, Samuele Salti, and Luigi Di Stefano. Unique signatures of histograms for local surface description. In *International Conference on Computer Vision (ICCV)*, pages 356–369, 2010. [1](#), [14](#)
- [51] Giovanni Trappolini, Luca Cosmo, Luca Moschella, Riccardo Marin, Simone Melzi, and Emanuele Rodolà. Shape registration in the time of transformers. *Advances in Neural Information Processing Systems*, 34:5731–5744, 2021. [6](#), [7](#), [14](#)
- [52] Fan Wang, Qixing Huang, and Leonidas J. Guibas. Image co-segmentation via consistent functional maps. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 849–856, 2013. [2](#), [4](#)
- [53] Lanhai Wang and Amit Singer. Exact and stable recovery of rotations for robust synchronization. *Information and Inference: A Journal of the IMA*, 2(2):145–193, 2013. [2](#)
- [54] Ruben Wiersma, Elmar Eisemann, and Klaus Hildebrandt. Cnns on surfaces using rotation-equivariant features. *ACM Transactions on Graphics (ToG)*, 39(4):92–1, 2020. [14](#)
- [55] Lei Yang, Wenxi Liu, Zhiming Cui, Nenglun Chen, and Wenping Wang. Mapping in a cycle: Sinkhorn regularized unsupervised learning for point cloud shapes. In *ECCV*, 2020. [2](#)
- [56] Christopher Zach, Manfred Klopschitz, and Marc Pollefeys. Disambiguating visual relations using loop constraints. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1426–1433. IEEE, 2010. [2](#)
- [57] 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](#)

In this supplementary material, we start by proving Proposition 1 in the main submission in Sec. 7. We then show data variability in our experiments in Sec. 8. Sec. 9 clarifies the annotation preparation regarding **DT4D-H**. Finally, in Sec. 10, more experimental results and implementation details are provided.

7. Proof of Proposition 1

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

Proposition 2 *If $E_{\text{total}}(\mathcal{C}) = 0$, then for any shape S_i , and any path $(i, i_1, i_2, \dots, i_k, i)$, the map composition \mathbf{C}_{ii} is cycle consistent within the functional space spanned by the columns of \mathbf{A}_i , i.e., $\mathbf{C}_{ii}\mathbf{A}_i = \mathbf{A}_i$.*

Proof 1 It is obvious that $E_{\text{total}}(\mathcal{C}) = 0$ implies $E_{\text{desc}}(\mathcal{C}) = 0$. In the following, we show the case of the path of length 3 – (i, j, k, i) . The general case follows easily. Setting $\mathbf{C}_{ii} = \mathbf{C}_{ki}\mathbf{C}_{jk}\mathbf{C}_{ij}$, we get:

$$\mathbf{C}_{ii}\mathbf{A}_i = \mathbf{C}_{ki}\mathbf{C}_{jk}\mathbf{C}_{ij}\mathbf{A}_i = \mathbf{C}_{ki}\mathbf{C}_{jk}\mathbf{A}_j = \mathbf{C}_{ki}\mathbf{A}_k = \mathbf{A}_i. \quad (7)$$

The equities in Eqn. (7) follow from the fact $\|\mathbf{C}_{ij}\mathbf{A}_i - \mathbf{A}_j\| = 0, \forall i, j$, since $E_{\text{desc}}(\mathcal{C}) = 0$.

8. Data Variability

In the main submission, we highlight our generalization performance. To give a hint of the distinctiveness among the involved datasets, we visualize a subset of each of them in Fig. 5. The first four rows show shapes from the humanoid datasets. **FAUST_r** (a) consists of 10 different people with 10 fixed poses. **SCAPE_r** (b) shows more significant pose variability but is of the same character. It is worth noting that, **SHREC19_r** (c) manifests larger variability in both styles and poses when compared to the above two. Furthermore, **DT4D-H** (d) is a new challenging dataset consisting of distinctive humanoid categories, in which the inter-class maps are highly non-isometric, especially when compared to the aforementioned datasets.

There are 8 species of animals in **SMAL_r**. Following [29], we use 5 of them during training and the rest for testing. As shown in Fig. 5 (e) and (f), we observe obvious differences between them, rendering the difficulty of the task. In addition, the 31 animal shapes from **TOSCA_r** (g) fall into 4 categories and also demonstrate noticeable differences from the training set of **SMAL_r**.

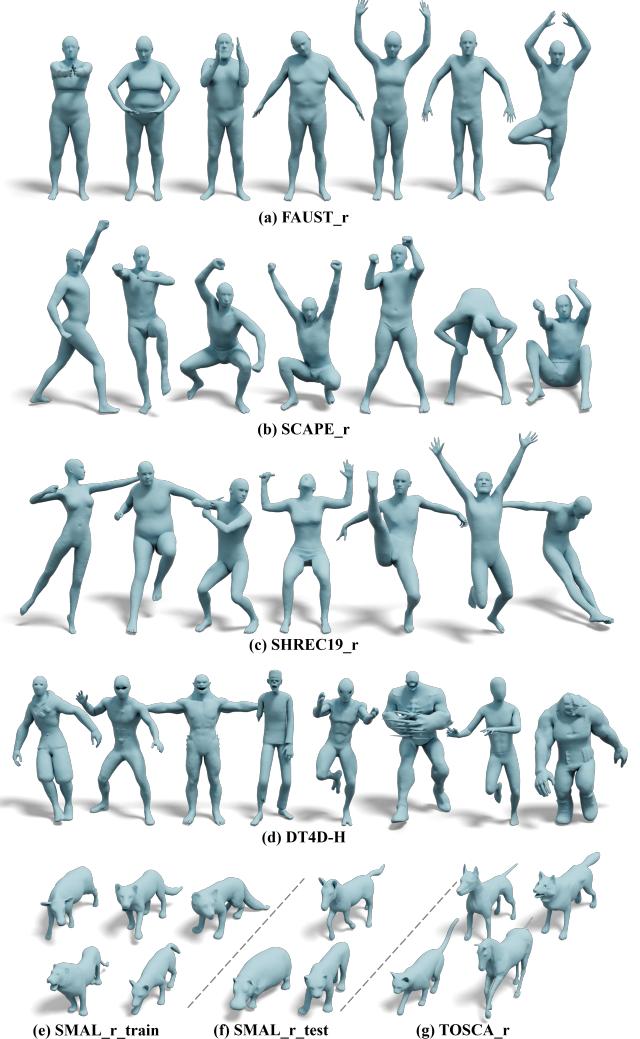


Figure 5. (a) part of the fixed poses from different individuals in **FAUST_r**; (b) part of the different poses in **SCAPE_r**; (c) shapes in **SHREC19_r**; (d) 8 categories of humanoid shapes in **DT4D-H**; (e) 5 categories of animals used in training; (f) 3 categories of animals used in test; (g) test animals from **TOSCA_r**.

9. Label Preparation in DT4D-H

Note that the inter-class correspondence annotations from **DT4D-H** are only available between category *crypto* and the other 7 categories. In order to train and test on this benchmark in a *category-agnostic* manner, we compute an inter map between two shapes, S_1, S_2 , from two categories other than *crypto*, with the following composition:

$$T_{12} = T_{c2} \circ T_{1c},$$

where T_{c2}, T_{1c} are the annotated inter-class maps regarding the center category, *crypto*. Note again, we exclude categories *mousey* and *ortiz* in the experimental setting reported in the main submission, simply due to their lack of inter-

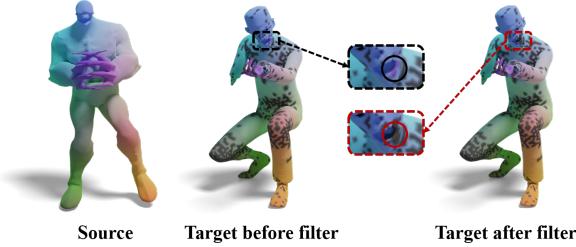


Figure 6. We filter out the erroneous correspondences via consistency prior. See the text for details.

class correspondence annotation with respect to the center category.

However, empirically we observe that certain noise in the original annotation is amplified through the above composition, leading to a small portion of erroneous labels. To alleviate such discrepancy for better evaluation, we filter the composed correspondences as follows: Given composed maps T_{12}, T_{21} , we further compose them to obtain self maps on S_1 and S_2 , respectively. That is, $T_{11} = T_{21} \circ T_{12}, T_{22} = T_{12} \circ T_{21}$. Then, we evaluate per-vertex Euclidean errors of the self-maps with respect to the ground truth identity maps. Finally, we filter out all the annotated correspondences involving vertices such that $\|T_{ii}(p) - p\| > 0.1$ (all shapes are normalized to unit total area).

In Fig. 6, we visualize the correspondences before and after our post-processing. We remark that the ground-truth annotations are not dense. That is, there exists a portion of vertices on one shape corresponding to no vertex on the other, which is indicated by the black color in the transferred texture. As illustrated within the circles of the zoom-in regions, our post-processing manages to remove the wrongly mapped points (see the discontinuous purple regions at the top). As a result, the removed region is now in no correspondence (see the black region at the bottom). On average, about 1% of the points in the original annotation are filtered out.

10. More Experimental Results and Details

In this section, we provide not only implementation details, but also more experimental results, both quantitatively and qualitatively, to further clarify and support our claims made in the main submission.

10.1. Single Pair Fine-tune

In this section, we perform a challenging fine-tuning task to test our method and two state-of-the-art unsupervised methods – AttentiveFMaps [29] and UDMSM [9].

We select two pairs of non-isometric animals from **TOSCA_r**. Then we use the weights trained on the **human** dataset **FAUST_r** from our experiments as initialization and perform fine-tuning on the selected **animal** pairs. All meth-

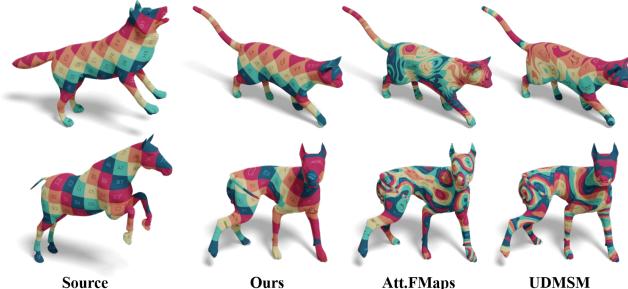


Figure 7. We perform unsupervised fine-tune on two pairs of non-isometric animals with weights initialized from models trained on **FAUST_r**. Our results clearly outperform the competing methods.

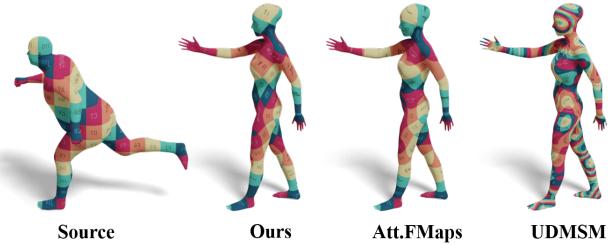


Figure 8. We train models on **FAUST_r** and test on **SHREC19_r**.

ods are optimized for 100 epochs over the given pair. The qualitative comparisons are shown in Fig. 7. Note that our method is the only one that leads to good maps, by which the grid texture (e.g., on the torsos) is well-preserved.

10.2. Additional Baselines on Near-isometric Datasets

Due to the space limit, we only present the more recent and stronger baselines in Tab. 1 in the main submission. In Tab. 6, we provide more complete results on near-isometric shape matching. Note that the newly introduced baselines (highlighted in light gray) are in general weaker than the baselines we report in the main submission, therefore their absence does not affect our experimental analysis.

In Fig. 8, we provide qualitative results to demonstrate the generalization power of our method. Specifically, we train models on **FAUST_r** and infer a pair of shapes from **SHREC19_r**. The qualitative results are consistent with the quantitative results in Tab. 6.

10.3. Implementation Details on SMAL_r

In this part, we clarify our experiments setting of **SMAL_r** (see Tab. 2 in the main submission). We follow the setting of [29], where the training and testing data contain 5 and 3 species, respectively. Also following [29], we use the XYZ signal input augmented with random rotations around the up (or Y) axis as the input signal to the network. The same settings are applied to the baseline GeomFMaps [12]. For UDMSM [9] and DeepShell [15], we have imple-

Table 6. Mean geodesic errors ($\times 100$) on **FAUST_r**, **SCAPE_r**, and **SHREC19_r**. The **best** and the **second best** are highlighted correspondingly.

Method	Train	FAUST_r			SCAPE_r		
	Test	FAUST_r	SCAPE_r	SHREC19_r	SCAPE_r	FAUST_r	SHREC19_r
ZM[36]		6.1	\	\	7.5	\	\
BCICP[43]		6.4	\	\	11.0	\	\
IsoMuSh[17]		4.4	\	\	5.6	\	\
Smooth Shell[13]		2.5	\	\	4.7	\	\
CZO[25]		2.2	\	\	2.5	\	\
FMNet[32]		11.0	30.0	\	17.0	33.0	\
3D-CODED[19]		2.5	31.0	\	31.0	33.0	\
HSN[54]		3.3	25.4	\	3.5	16.7	\
ACSCNN[30]	sup	2.7	8.4	\	3.2	6.0	\
TransMatch[51]		2.7	33.6	21.0	18.3	18.6	38.8
GeomFMaps[12]		2.6	3.3	9.9	3.0	3.0	12.2
AttentiveFMaps[29]		1.4	2.2	9.4	1.7	1.8	12.2
SURFMNet[45]		6.0	16.5	\	6.8	18.5	\
UnsupFMNet[20]		10.0	29.0	\	16.0	22.0	\
WSupFMNet[46]		3.3	11.7	\	7.3	6.2	\
NeuroMorph[14]		8.5	28.5	26.3	29.9	18.2	27.6
SyNoRiM[16]		7.9	21.7	25.5	9.5	24.6	26.8
Deep Shell[15]		1.7	5.4	27.4	2.5	2.7	23.4
AttentiveFMaps[29]	unsup	1.9	2.6	6.4	2.2	2.2	9.9
UDMSM[9]		1.5	7.3	21.5	2.0	8.6	30.7
DUO-FM[11]		2.5	4.2	6.4	2.7	2.8	8.4
Ours		2.3	2.6	3.8	2.4	2.5	4.5
Ours (80 dim)		1.7	2.6	5.5	2.2	2.0	5.8



Figure 9. We train models on **SMAL_r** and test on **TOSCA_r**.

mented the official codes by the regarding authors with *both* SHOT [50] (the common default descriptors) and XYZ as input. And in the end, we select the better output from the two. In fact, both methods work better with SHOT as input.

In Fig. 9, we train models on **SMAL_r** and test on **TOSCA_r**. The qualitative results suggest our better generalization performance, which agrees with the quantitative results reported in Tab. 2 in the main submission.

10.4. Implementation details on DT4D-H

We find in the official code of [29] that the authors *train* and test inter maps with a fixed source category (*crypto*). On the other hand, in the main submission, we advocate a category-agnostic *training* scheme, which is more practical as well as challenging (see Tab. 3 in the main submission for comparison). For the sake of completeness and fairness, we follow the exact experimental settings of AttentiveFMaps to train our model and report the results in the same manner as [29] in Tab. 7. We outperform [29] by a significant mar-

Table 7. Mean geodesic errors ($\times 100$) on **DT4D-H** followed AttentiveFMaps. The **best** and the **second best** are highlighted correspondingly.

Method	DT4D		
	intra-class	inter-class	
GeomFMaps[12]	2.1	4.1	
AttentiveFMaps[29]	sup	1.8	4.6
DeepShell[15]		3.4	31.1
GeomFMaps[12]	unsup	3.3	22.6
AttentiveFMaps[29]		1.7	11.6
Ours		1.2	6.1

gin (6.1 vs. 11.6 for inter-class maps) in their setting. Remarkably, as an unsupervised method, our inter-class score is even comparable with the baselines with supervision (see the top two rows).

10.5. Implementation Details on Plugin with SURFMNet

We implement our two-branch variant of SURFMNet with PyTorch [41]. The dimension of the Laplace-Beltrami eigenbasis is set to 40. SHOT [50] descriptors are used as the input signal of the network. The dimensions of the input and the output descriptors are both set to 352. During training, the value of the learning rate is set to 1e-3 with ADAM optimizer. In all experiments, we set the batch size to 1. We initialize α to 1 and increase it by 1 per epoch. Note that this learning scheme is different from the one we reported

in our main submission, where the backbone is Diffusion-Net [47]. Here α is augmented slower as the backbone network of SURFMNet [45] is weaker. We keep all the losses used in SURFMNet [45], and just simply add our proposed new branch, as shown in Fig. 2 in the main submission.