Data-Efficient Deep Functional Maps for 3D Shape and Graph Analysis

Abhishek Sharma

INSTITUT POLYTECHNIQUE DE PARIS

NNT : 2022IPPAX069

Thèse de doctorat

Data-Efficient Deep Functional Maps for 3D Shape and Graph Analysis

Thèse de doctorat de l'Institut Polytechnique de Paris
préparée à l'École polytechnique

École doctorale n°626 École doctorale de l'Institut Polytechnique de Paris (ED IP Paris)
Spécialité de doctorat : Informatique

Thèse présentée et soutenue à Palaiseau, le 25 Juillet 2022, par

ABHISHEK SHARMA

Composition du Jury :

Edmond Boyer Directeur de recherche, INRIA (MORPHEO) Grenoble	Rapporteur
Alex Bronstein Professor, Technion, Israel Institute of Technology	Rapporteur
Nikos Paragios Professor, CentraleSupelec, University Paris Saclay	Président
Michalis Vazirgiannis Professor, Ecole Polytechnique (LIX), IPP	Examinateur
Omri Azencot Senior Lecturer, Ben-Gurion Univeristy, Israel	Examinateur
Viorica Patraucean Researcher, Google DeepMind, UK	Examinateur
Maks Ovsjanikov Professeur, École Polytechnique(LIX), IPP	Directeur de thèse

Acknowledgment

I am grateful to Jean Ponce for showing me how to do research during my Masters stage (thesis) in barely four meetings I had with him besides many emails. I perhaps did not recognise at that time how much he cared for someone as busy as him and how rare it is in academia to invest time and effort in advancing a Masters student career just for the sake of science. I was asked (told) bluntly before, during an interview, by (apparently) a 100k citations researcher that I do not come from a good school. Such baseless biases disappeared because of him and it was unbelievable to see the change in people's behavior after they read Jean's letter. I thank Jean for starting my research career.

I am grateful to Nikos Paragios for supporting me at a time when everything went against me. Despite a single author NeurIPS paper (thanks to Jean Ponce), a very well cited first author ECCV paper and a MVA Masters, such were the circumstances that I needed a reference from him to get *an internship* that, if successful, would lead to a PhD offer. So this PhD would not have happened without him. I have learnt a lot from his remarks during his unique MVA course, via his personal blog and in so many interactions with him. I thank Dr. Catherine Pelachaud for bringing me to Paris and thus, introducing me to the totally free (no tuition fees) and yet, excellent French academia. I may not have gotten into the MVA masters without her recommendation. I thank Diego Borro and Francois Fleuret for the internship (reference) that perhaps convinced Catherine and Jean respectively.

I thank Jing for being my goto person for everything in geometry processing and Jean Michel for a fantastic start in the group. I thank Maks for his thoughtful comments on the manuscript and NeurIPS and ICASSP papers. In both papers, I had the right research direction as well as state of the art results. I needed a concrete justification and he justified them with symmetry disambiguation reasoning in NeurIPS and basis consistency in ICASSP. I thank the GeoVIC team for so many good lunches and gatherings before Covid and the rest of the group for bearing me. I am grateful to Benjamin Doerr for his selfless service towards the PhD administration. I think the doctoral school, based on adum, will collapse without his time and effort. I am grateful to my PhD thesis committee for their time, service and suggestions. In particular, I thank Dr. Edmond Boyer and Prof. Alex Bronstein for agreeing to review my thesis despite their super busy schedule and commitments.

I thank my friends at Cite Universitaire for making this Covid times pass very smoothly and also teaching me many simple things about people/life that I was stupidly unaware of. In particular, I am grateful to Irina, Katerina, Jaspreet, Gurkirt, Arthur, Heather, Evelyne, and the volleyball team for all the time spent together and for bearing me. This thesis is dedicated to my loving parents for their many sacrifice.

Contents

Abstract

Shape correspondence is a fundamental problem in computer vision, computer graphics, and related fields since it facilitates many applications such as texture or deformation transfer and statistical shape analysis to name a few. Although shape correspondence has been studied from many viewpoints, in this thesis, we focus on functional map-based approaches as this framework is quite general, scalable and thus, has been extended to various other applications such as pose estimation, matrix completion, and graph matching. In this thesis, we propose three contributions to deep functional maps: First, we propose a simple but effective method to estimate a high-dimensional functional map. Our method is based on first learning a low dimensional functional map and then refining it to a higher dimensional one based on iterative spectral upsampling. Second, we propose a new direction that advocates the use of approximate rigid alignment of shapes as a weak supervision signal. Our weakly supervised Deep Functional map obtains competitive performance compared to the fully supervised approach. Our main hypothesis is that the approximate rigid alignment provides the network with enough information to disambiguate symmetry issues. Although approximate alignment is easier to obtain than pointwise ground truth between a pair of shapes, it still suffers from scalability issues on large-scale 3D shape collections. Thus, we go beyond this prerequisite and consider the problem of learning simultaneously a self symmetry map and a pairwise map. Our third contribution is a novel commutative regularization that couples the self-symmetry map with a pairwise map and thus enable knowledge transfer between the two maps during training. Our last contribution is an application of the functional map framework to some graph-based machine learning problems such as geometric matrix completion and dimensionality reduction. We propose a simplified framework that is based on a key idea that using a reduced basis to represent functions on the product space is sufficient to recover a low-rank matrix approximation even from a sparse signal.

Chapter 1

Introduction

Recent years have witnessed a 3D revolution in various fields, largely thanks to advances in deep learning and 3D sensing technology. The three major fields with the largest scope of 3D deep learning can be broadly categorized into entertainment, robotics and life science. In the former, with the emergence of many 3D technologies in the consumer market such as depth field cameras like the Kinect and virtual reality devices like Oculus Rift, a lot of 3D media is growing in our digital world and will continue to increase in the near future. Similarly in robotics, emergence and affordability of new 3D sensor technologies like Lidar cameras allows autonomous agents to perceive the 3D environment in a much more complete way than previously possible with standard cameras. These agents require algorithms to process and analyse 3D signals in order to make decisions. Lastly, a recent breakthrough in bioinformatics [39] called alphafold has opened a new avenue for 3D shape analysis in biology. Jumper et al. [39] reliably predicts 3D structure of proteins given amino acid sequence. One of the major problems in biology is predicting protein-protein interaction [103] i.e. which protein is likely to bind with another protein. Correctly predicting protein-protein interaction has important applications in proteins engineering and drug development [103]. Interestingly, this phenomenon is largely driven by the similarity in the 3D shapes of individual proteins and thus necessitates 3D shape matching algorithms.

1.1 Non Rigid 3D Shape Matching

When asked about the most important problems in computer vision in late 90's, Takeo Kanade, one of the luminaries, replied: 'Correspondence, correspondence, correspondence' [112]. Shape correspondence is a fundamental problem in computer vision, computer graphics and related fields since it facilitates many applications such as texture or deformation transfer and statistical shape analysis [8] to name a few. It is fair to attribute similar significance even today to correspondence problems given its widespread applications in various fields described above. Although shape correspondence has been studied from many viewpoints [104, 107, 118, 117], in this dissertation we focus on a functional map-based approaches [68] as this framework is quite general, scalable and thus, has been extended to various other applications such as pose estimation [65], matrix completion [91] and graph matching [109].

Given a pair of objects containing m and n points respectively, finding correspondence entails finding a bijective (if possible) map between these points. e.g. in case of two images, such points can be keypoints, in case of two graphs, points translate to nodes. In

the simplest case, such problems are formulated as a labeling problem, where different points, E.g., in a template shape, correspond to labels to be predicted. This leads to an extremely large label space that scales linearly with the number of points and thus, requires lots of data to learn this space. In contrast, the authors in [68] introduced a functional view in visual correspondence problems by considering shapes or images as functional spaces. This functional map approach aligns descriptor functions from one visual object to another and thus, aims to infer an entire global map between a pair of shapes rather than aligning each point independently. Deep Functional Map [53] introduces learning into functional map approach by learning a non-linear transformation of the descriptor functions to be aligned later.

Two main inputs to a typical Functional Map framework are corresponding descriptor functions and basis functions defined on two objects. Prior to this dissertation, SHOT features [102] were most commonly used as descriptor functions and Laplacian eigenbasis [5] as the basis functions. Basis functions are required to perform dimensionality reduction by projecting the descriptors onto a spanning subspace of basis functions. Finally, one solves an optimization problem, seeking a matrix that best aligns the projected features. One of the key benefits of this framework is that it allows us to represent maps between shapes as small matrices, which encode relations between basis functions defined on the shapes. As a result, the objective function is independent of the number of points on the shape and can be optimized with simple linear solvers such as least squares.

Despite its simplicity and scalability, this framework is suboptimal in many ways. The first source of difficulty lies in estimating a high dimensional functional map accurately. It remains a challenge for both the axiomatic functional map pipeline as well as deep functional maps. Learning in high dimensions is known to be difficult in machine learning [7]. On the other hand, reduced dimensionality results in very approximate maps, losing medium and high-frequency details and leading to significant artifacts in applications. Prior to the work in this thesis, state of the art [54] relied on directly embedding the scalar functions into a high dimensional Laplace Beltrami eigenbasis. As we show in Chapter 2, this can lead to severe overfitting and thus, significant drop in the accuracy of resulting point to point map.

A major issue with the axiomatic functional map approach [68] is that descriptor functions computed on the shape are handcrafted or computed before and not learned from data. This makes the entire pipeline susceptible to changes in the representation of 3D geometry. e.g. the commonly used SHOT feature [102] heavily relies on the input mesh structure. Thus, a robust shape matching pipeline should learn features directly from point clouds rather than relying on input mesh connectivity. Deep Functional Maps [54] has partially addressed this problem by learning a non-linear transformation of SHOT features. However, that pipeline still requires an input mesh and can not learn directly from raw shape geometry.

Lastly, many natural as well man-made objects contain symmetries which poses a challenge to any matching algorithm. A simple example being left side of a human body is often mismatched to the right side. Symmetry disambiguation also poses a challenge for most techniques based on the functional map estimation pipeline. Inducing information exchange between a self-symmetry map and a pairwise map should improve the consistency between the two maps and thus, the resulting accuracy. Symmetry detection and shape matching are two problems that are inherently linked to each other. Yet, there is no unified learning framework that learns symmetry detection and shape matching simultaneously and thus couples the two maps during training.

In the following dissertation, we propose three contributions, each corresponding to these problems. We marry ideas from deep learning with the functional map framework thereby resulting in very efficient and robust algorithms for shape matching based on deep functional maps. Moreover, we extend the notion of functional maps to graphs and propose a functional view of the graph based matrix completion problem.

List of Publications

- J.M. Roufosse, A. Sharma, M. Ovsjanikov, Unsupervised Deep Learning for Structured Shape Matching [81] ICCV'2019.
- S. Melzi, J. Ren, E. Rodola, A. Sharma, P. Wonka, M. Ovsjanikov, ZoomOut: Spectral Upsampling for Efficient Shape Correspondence [60], Siggraph Asia'2019
- N. Donati, A. Sharma, M. Ovsjanikov, Deep geometric functional maps: Robust feature learning for shape correspondence [19], CVPR'2020
- A. Sharma, M. Ovsjanikov, Weakly Supervised Deep Functional Maps for Shape Matching [89], NeurIPS'2020
- A. Sharma, M. Ovsjanikov, Matrix Decomposition on Graphs: A Simplified Functional View [91], ICASSP'2022
- A. Sharma, M. Ovsjanikov, Joint Symmetry Detection and Shape Matching for Non-Rigid Point Cloud[90]

In Chapter 2, we explore the interaction between spectral quantities, and specifically the size of the functional map used in a deep learning pipeline, and overfitting between training and test data in different scenarios. We show that smaller functional maps tend to generalize better in difficult settings, and use this observation to motivate the use of a simple refinement strategy based on a spectral upsampling technique [60]. Our method is based on first learning a low dimensional functional map using a deep functional map [81] and then refining it to higher dimensional one based on iterative spectral upsampling [60]. With a combination of these two, we demonstrate that one can estimate a high dimensional functional map effectively. In contrast, directly learning a high dimensional map with deep functional maps directly leads to severe overfitting, a phenomenon we call

'Spectral Overfitting' in Chapter 2. This technique has been adopted in ZoomOut [60], GeomFmap [19], WSupFMNet [89] and constitutes a key contribution of this thesis.

In Chapter 3, we consider the problem of learning shape correspondence directly from point cloud data both on near isometric as well as partial shapes without explicit supervision. The first challenge to overcome is to design a learning framework that can learn scalar functions on shapes from scratch, without using pre-defined per vertex functions like SHOT or ground truth correspondences. We follow the approach of SURFMNet [81] that introduced the idea of learning correspondence with an unsupervised structural loss that acts as a proxy for ground truth geodesic error. However, learning from point cloud data with just unsupervised structural loss is insufficient in terms of supervision. The solution considered in [19] is to use the ground truth correspondence. In [89], we propose a new direction that advocates the use of approximate rigid alignment of shapes as a weak supervision signal. We demonstrate through extensive experiments that this weak supervision obtains competitive performance compared to the fully supervised approach [19]. Our main hypothesis is that the approximate rigid alignment provides the network enough information to disambiguate symmetry issues.

One limitation of the method we present in Chapter 3 is that it is only applicable to shapes that are approximately pre-aligned. Although approximate alignment is easier to obtain than pointwise ground truth between a pair of shapes, it still suffers from scalability issues on large scale 3D shape collections. In Chapter 4, we go beyond this prerequisite and consider the problem of learning simultaneously a self symmetry map and a pairwise map. In [90], we propose a novel commutative regularization that couples the self-symmetry map with a pairwise map and thus enables knowledge transfer between the two maps during training. To the best of our knowledge, we propose the first method that simultaneously learns symmetry detection and shape matching for non-rigid point clouds.

Our last contribution in Chapter 5 is an application of the functional map framework to some graph-based machine learning problems such as geometric matrix completion [62] and dimensionality reduction [84]. We propose a simplified framework that is based on a key idea that using a reduced basis to represent functions on the product space is sufficient to recover a low rank matrix approximation even from a sparse signal. We validate our framework on various synthetic as well as real world benchmark datasets.

In Chapter 6, we conclude with a brief overview of followup works that cite our work and build upon it. Lastly, we present cycle consistent deep functional maps as a future direction that is yet to be explored.

To summarize, Chapter 2 presents learning high dimensional functional map and is based upon the contributions made in SURFMNet [81], ZoomOut [60] and GeomFmap [19].

The method proposed in Chapter 3 contributed to Weakly Supervised Deep Functional Map for Point Clouds [89]. Chapter 4 is based on the following article

- Joint Symmetry Detection and Shape Matching for Non-Rigid Point Clouds [89].

. Chapter 5 contributed to the following paper:

- Matrix Decomposition on Graphs: A Simplified Functional View [91].

Chapter 2

Learning High Dimensional Functional Map

In this chapter, we introduce a new phenomenon named 'Spectral Overfitting'. We first show that directly learning high dimensional functional maps gives inaccurate correspondences. Instead, we propose a two stage algorithm that first learns lower resolution functional maps and then refines them using a iterative spectral upsampling method [60]. Our key idea is that low frequency information is more stable across different shapes/classes and thus learning a low dimensional functional map generalizes much better than a learned high dimensional functional map. This algorithm is used later throughout in this thesis to effectively estimate high dimension functional maps.

2.1 Background

In this chapter, we introduce Spectral Overfitting in deep functional maps. Since our work builds on the functional map framework, we briefly review the basic notions and pipeline for estimating functional maps, and its deep counterpart [53, 81]. FMNet [53] introduced learning into a functional map pipeline by learning a non-linear transformation of SHOT descriptor functions. In this thesis, we distinguish between FMNet and deep functional maps as the latter is a more general framework considering many followup works [32, 81, 89]. Specifically, we call deep functional maps any method that learns descriptors that are used to estimate a functional map inside a neural network, which then imposes some loss on this estimated functional map. In what follows, we describe FMNet as a specific deep functional map algorithm which learns on SHOT and deep functional maps as a general framework that learns features to be used with in-network functional map estimation.

2.1.1 Functional Map Pipeline

Given a pair of shapes, S_1, S_2 represented as triangle meshes, and containing, respectively, n_1 and n_2 vertices, the basic pipeline for computing a map between them using the functional map framework, consists of the following main steps :

1. Compute a small set of k_1, k_2 of basis functions on each shape, e.g. by taking the first few eigenfunctions of the respective Laplace-Beltrami operators.

2. Compute a set of descriptor *functions* on each shape that are expected to be approximately preserved by the unknown map. For example, a descriptor function can correspond to a particular dimension (e.g. choice of time parameter of the Heat Kernel Signature [99]) computed at every point. Store their coefficients in the respective bases as columns of matrices $\mathbf{A}_1, \mathbf{A}_2$.

3. Compute the optimal *functional map* $\mathbf{C}$ by solving the following optimization problem:

$$C_{\text{opt}} = \arg\min_{\mathbf{C}_{12}} E_{\text{desc}}(\mathbf{C}_{12}) + \alpha E_{\text{reg}}(\mathbf{C}_{12}), \tag{2.1}$$

where the first term aims at descriptor preservation: $E_{\text{desc}}(\mathbf{C}_{12}) = \|\mathbf{C}_{12}\mathbf{A}_1 - \mathbf{A}_2\|^2$, whereas the second term regularizes the map by promoting the correctness of its overall structural properties. The simplest approach penalizes the failure of the unknown functional map to commute with the Laplace-Beltrami operators:

$$E_{\text{reg}}(C_{12}) = \|\mathbf{C}_{12}\mathbf{\Lambda}_1 - \mathbf{\Lambda}_2\mathbf{C}_{12}\|^2 \tag{2.2}$$

where $\mathbf{\Lambda}_1$ and $\mathbf{\Lambda}_2$ are diagonal matrices of the Laplace-Beltrami eigenvalues on the two shapes.

4. Convert the functional map $\mathbf{C}$ to a point-to-point map, for example using nearest neighbor search in the spectral embedding, or using other more advanced techniques [79, 26].

One of the strengths of this pipeline is that typically Eq. (2.1) leads to a simple (e.g., least squares) problem with $k_1 k_2$ unknowns, independent of the number of points on the shapes. This formulation has been extended using e.g. manifold optimization [49], descriptor preservation constraints via commutativity [67] and with kernelization [110] among many others.

2.1.2 Deep Functional Maps

Despite its simplicity and efficiency, the functional map estimation pipeline described above is fundamentally dependent on the initial choice of descriptor functions. To alleviate this dependence, several approaches have been proposed to learn the optimal descriptors from data [53]. In this dissertation, we build upon a recent supervised deep learning-based framework, called FMNet, introduced by Litany et al. [53] that aims to transform a given set of descriptors so that the optimal map computed using them is as close as possible to some ground truth map [53]. The same framework is easily extended to an unsupervised setting by replacing the supervised loss with some structural loss that satisfies the desired properties of a functional map. SURFMNet [81] introduced this unsupervised approach.

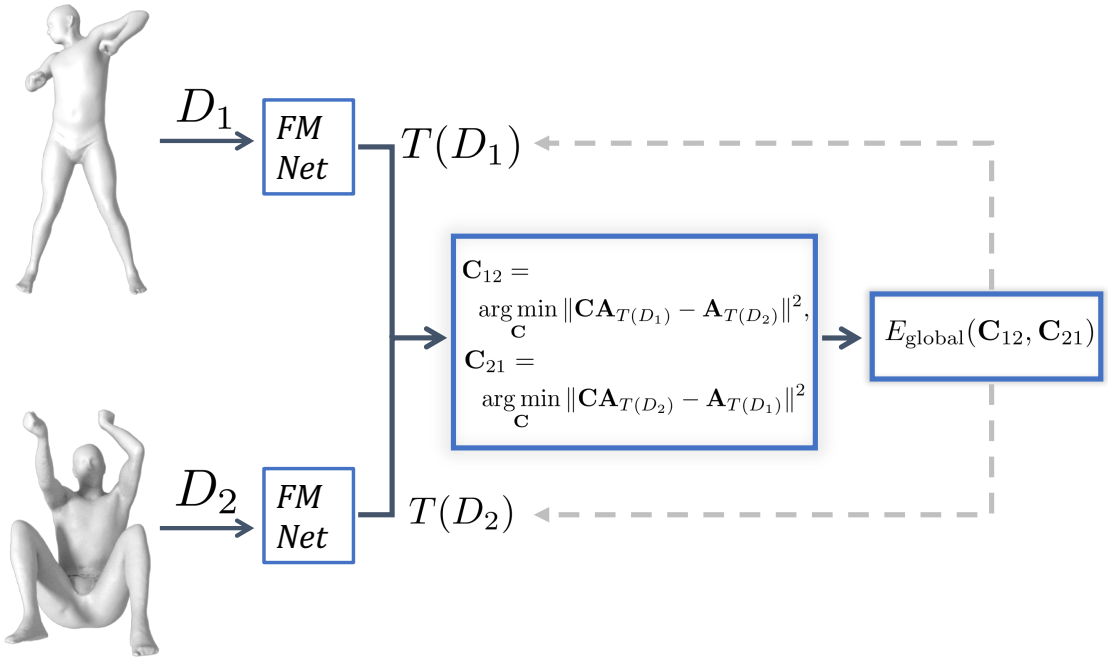

Figure 2.1: Overview of the Unsupervised Deep Functional Map approach SURFMNet [81]: given a pair of shapes and their descriptors D_1, D_2, it optimizes for a non-linear transformation T using a Siamese architecture so that the transformed descriptors lead to functional maps that best satisfy the structural constraints.

In general, given a set of shape pairs, the Deep Functional Map pipeline [53] aims to solve the following problem:

$$\min_{T} \sum_{(S_1,S_2)\in \text{Train}} l_F(\mathbf{C}_{\text{opt}}), \text{ where} \tag{2.3}$$

$$\mathbf{C}_{\text{opt}} = \arg\min_{\mathbf{C}} \|\mathbf{C}\mathbf{A}_{T(D_1)} - \mathbf{A}_{T(D_2)}\|. \tag{2.4}$$

Here T is a non-linear transformation, in the form of a neural network, to be applied to some input descriptor functions D, Train is the set of training pairs, l_F is a loss function which penalizes the deviation of the computed functional map $\mathbf{C}_{\text{opt}}$. In the supervised setting of FMNet [53], this deviation is measured from the ground truth correspondence. In the unsupervised setting of SURFMNet [81], such deviation is measured by a structural loss on estimated $\mathbf{C}_{\text{opt}}$. $\mathbf{A}_{T(D_1)}$ denotes the transformed descriptors D_1 written in the basis of shape S_1.

2.1.3 Deep Functional Map Regularization

The loss function, $l_F(\mathbf{C}_{\text{opt}})$, mentioned in Eq. 2.3 encourages desirable map properties by penalizing the deviation of estimated map from such properties. In the following, we

describe four such properties some of which are also used extensively later in Chapter 3 and Chapter 4.

Bijectivity Given a pair of shapes and the functional maps in both directions, perhaps the simplest requirement is for them to be inverses of each other. It is enforced by penalizing the difference between their composition and the identity map:

$$E_1 = \|\mathbf{C}_{12}\mathbf{C}_{21} - \mathbf{I}\|^2 + \|\mathbf{C}_{21}\mathbf{C}_{12} - \mathbf{I}\|^2 \tag{2.5}$$

Orthogonality As observed in several works [68, 82] a point-to-point map is locally area preserving if and only if the corresponding functional map is *orthonormal*. Thus, another natural penalty in SURFMNet pipeline is:

$$E_2 = \|\mathbf{C}_{12}^\top\mathbf{C}_{12} - \mathbf{I}\|^2 + \|\mathbf{C}_{21}^\top\mathbf{C}_{21} - \mathbf{I}\|^2 \tag{2.6}$$

Laplacian commutativity Similarly, it is well-known that a pointwise map is an intrinsic isometry if and only if the associated functional map commutes with the Laplace-Beltrami operator [80, 68]. It is enforced by including the following penalty:

$$E_3 = \left\|\mathbf{C}_{12}\mathbf{\Lambda}_1 - \mathbf{\Lambda}_2\mathbf{C}_{12}\right\|^2 + \left\|\mathbf{C}_{21}\mathbf{\Lambda}_2 - \mathbf{\Lambda}_1\mathbf{C}_{21}\right\|^2 \tag{2.7}$$

Descriptor preservation via commutativity The last penalty promotes functional maps that arise from point-to-point maps, rather than more general soft correspondences. To achieve this, we follow the approach proposed in [67] based on preservation of pointwise products of functions. Namely, it is known that a non-trivial linear transformation $\mathcal{T}$ across function spaces corresponds to a point-to-point map if and only if $\mathcal{T}(f \odot h) = \mathcal{T}(f) \odot \mathcal{T}(h)$ for any pair of functions f, h. Here $\odot$ denotes the pointwise product between functions [95], i.e. $(f \odot h)(x) = f(x)h(x)$. When f is a descriptor function on the source and g is the corresponding descriptor on the target, the authors of [67] demonstrate that this condition can be rewritten in the reduced basis as follows: $\mathbf{C}\mathbf{M}_f = \mathbf{M}_g\mathbf{C}$, where $\mathbf{M}_f = \Phi^+\mathrm{Diag}(f)\Phi$, and $\mathbf{M}_g = \Psi^+\mathrm{Diag}(g)\Psi$. This leads to the following penalty:

$$\begin{aligned} E_4 = \sum_{(f_i,g_i)\in\text{Descriptors}} & ||\mathbf{C}_{12}\mathbf{M}_{f_i} - \mathbf{M}_{g_i}\mathbf{C}_{12}||^2 \\ & + ||\mathbf{C}_{21}\mathbf{M}_{g_i} - \mathbf{M}_{f_i}\mathbf{C}_{21}||^2, \\ \mathbf{M}_{f_i} = \Phi^+\mathrm{Diag}(f_i)\Phi, & \mathbf{M}_{g_i} = \Psi^+\mathrm{Diag}(g_i)\Psi. \end{aligned} \tag{2.8}$$

In this expression, f_i and g_i are the *optimized* descriptors on source and target shape, obtained by the neural network, and expressed in the full (hat basis), whereas Φ, Ψ are the fixed basis functions on the two shapes, and $+$ denotes the Moore-Penrose pseudoinverse.

Source descriptor before

Target descriptor before

Source descriptor after

Target descriptor after

Figure 2.2: Given a pair of shapes with noisy descriptors (top), SURFMNet [81] makes them more consistent (bottom) and automatically computes an accurate pointwise correspondence.

Source Ground-Truth SURFMNet

Figure 2.3: Illustration of SURFMNet computed point to point map via texture transfer between a source and a target shape.

Figure 2.1 illustrates the unsupervised Deep Functional Map pipeline of SURFMNet [81]. It aims to learn a transformation T of descriptors, so that the transformed descriptors $T(D_1)$, $T(D_2)$, *when used within the functional map pipeline* result in a desired map. Unlike methods based on formulating shape matching as a labeling problem this approach evaluates the quality of the *entire map*, obtained using the transformed descriptors and thus leads to significant improvement compared to several strong baselines.

The only unknowns in this optimization are the parameters of the neural network applied to the descriptor functions. The functional maps $\mathbf{C}_{12}$ and $\mathbf{C}_{21}$ are fully determined by the optimized descriptors via the solution of the optimization problems in Eq. (2.4). Note that although stated as an optimization problem, Eq. 2.3 reduces to a solution of a linear system of equations. This is easily differentiable using the well-known closed-form expression for derivatives of matrix inverses [71]. Namely for any matrix Y we have $\frac{\partial Y^{-1}}{\partial x} = -Y^{-1}\frac{\partial Y}{\partial x}Y^{-1}$. This expression allows to back-propagate the loss through the linear system used to solve for the functional maps in Eq. 2.3. Furthermore, the functionality of differentiating a linear system of equations is already implemented in TensorFlow and can be used directly.

Figure 2.2 illustrates how the network transforms the initial descriptor functions predefined on individual shapes so that they are better aligned afterwards. Figure 2.3 illustrates a point to point map computed by SURFMNet between a source and a target pair with texture transfer.

Motivation Although FMNet [53] and SURFMNet [81] propose an elegant solution to learning non-linear feature transformation, they rely on directly embedding the scalar functions into a high dimensional ($k = 120$) Laplace Beltrami eigen basis that leads to severe overfitting and thus, significant drop in the accuracy of resulting point to point map. In the following subsection, we illustrate it with experiments on standard benchmarks. In the next subsection, we briefly review the ZoomOut [60] algorithm that upsamples a lower resolution map to a higher resolution map. Afterwards, we introduce Spectral Overfitting.

2.1.4 Zoomout: Iterative Spectral Upsampling

Before we introduce spectral overfitting, we give a brief overview of ZoomOut [60]. We use it to iteratively upsample our learned functional map. ZoomOut is an iterative map refinement method that is particularly useful when the input map is noisy or is encoded as a 'small' functional map. It allows to recover a high-quality map, given noisy input. ZoomOut is a good fit for our purpose as it can upsample from a small-sized functional map, precisely the setting which is good for learning a functional map without overfitting.

As input we assume to be given either a small functional map $\mathbf{C_0}$ $k \times k$ or a point-to-point correspondence $T : S_1 \rightarrow S_2$. The goal of ZoomOut is to extend it to a new map $\mathbf{C_1}$ of size $(k+1) \times (k+1)$ without any additional information. It consists of the following two basic steps:

1. Convert a $k \times k$-size functional map to a pointwise map.
2. Convert the pointwise map to a $k+1 \times k+1$ functional map.

To compute a pointwise map from a given $\mathbf{C}$ in step (1), we solve the following problem:

$$T(p) = \arg\min_{q} \|\mathbf{C}(\Phi_{S_2}(q))^\top - (\Phi_{S_1}(p))^\top\|_2, \ \forall\ p \in S_1 \tag{2.9}$$

where $\Phi_{S_1}(p)$ denotes the p^{th} row of the matrix of eigenvectors Φ_{S_1}. This procedure gives a point-to-point map $T : S_1 \rightarrow S_2$, and can be implemented via a nearest-neighbor query in k-dimensional space.

Mathematically, ZoomOut can be written as:

1. Compute a point-to-point map T and encode it as a matrix Π.
2. Set $\mathbf{C_1} = (\Phi_{S_1}^{k+1})^\top \mathbf{A}_{S_1} \Pi\, \Phi_{S_2}^{k+1}$.

This procedure is then iterated to obtain progressively larger functional maps $\mathbf{C_0}, \mathbf{C_1}, \mathbf{C_2}, ., \mathbf{C_n}$ until some sufficiently large n. This remarkably simple procedure can be implemented in only a few lines of code and can result in very accurate functional and pointwise maps even when given a noisy input.

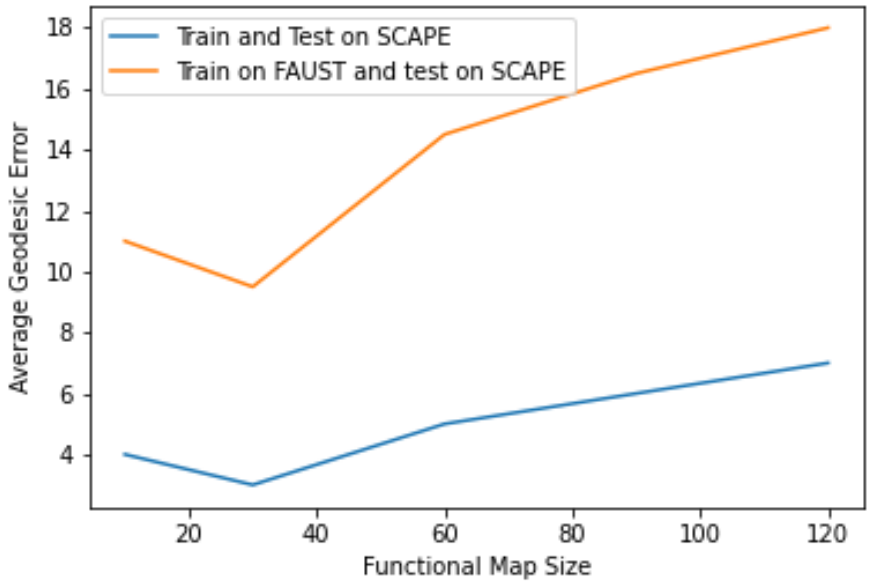

Figure 2.4: Accuracy of SURFMNet as the functional map size is varied from 10 to 120 in two different settings. In both cases, using 30 eigen basis gives optimal results.

2.2 Spectral Overfitting

In this chapter, we introduce a novel phenomenon that we call Spectral Overfitting. Figure 2.4 shows the sensitivity of SURFMNet on the SCAPE remeshed dataset as the number of eigen functions are varied from 10 to 150. We train the network in two settings to demonstrate the difference. In the first setting, we train and test on same dataset whereas in the second, we train on FAUST and test on SCAPE. As evident in Figure 2.4, we obtain a similar conclusion in both settings. We see severe over-fitting when using a large eigen-basis. However, using low number of eigen basis generalizes better to unseen test data. The generalization with large eigen basis gets even worse in cross dataset setting where the variation in input shapes is much larger.

We attribute the superior generalization ability with low eigen basis to the fact that low frequency information is more stable across different shapes/classes. Bias-Variance tradeoff [7] also plays an important role since we obtain lower accuracy when using a map resolution of 10. It implies that while low frequency information is stable across shapes, we also need a tradeoff between model capacity and stability. Thus, in both settings, we obtain the optimal result at a resolution of 30. We exploit these findings to estimate a high dimensional functional map using Zoomout. Zoomout is a good method in this scenario as it can upsample from a small-sized functional map, that is precisely the setting for learning without overfitting.

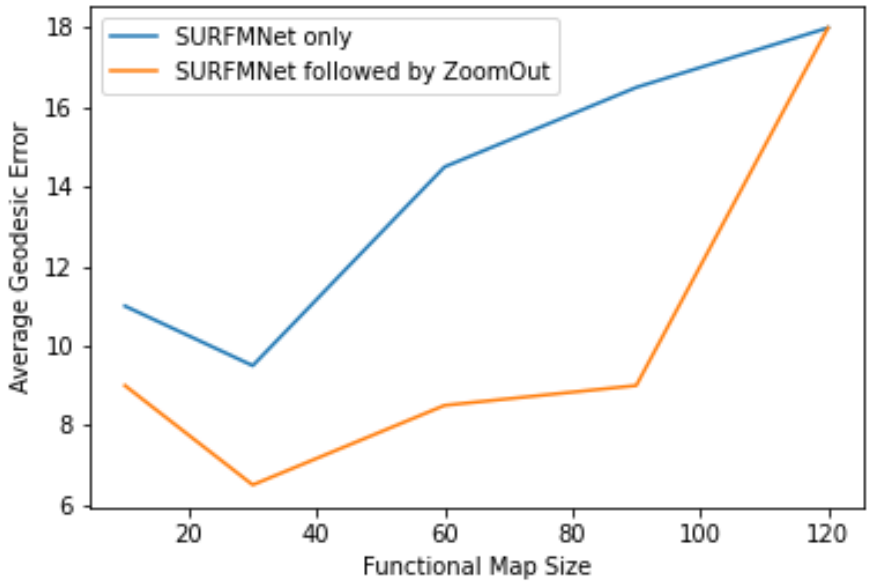

Figure 2.5: Average Geodesic error on SCAPE of learning a high resolution map directly with SURFMNet (in blue) and learning a small resolution map upsampled afterwards with ZoomOut (in orange). As evident, the later outperforms the former significantly on SCAPE testset (45 pairs). We used 80 shapes from FAUST for training.

Our method is based on first learning a low dimensional functional map based on a deep functional map [81, 19] and then refining it to higher dimensional based on an iterative spectral upsampling [60]. With a combination of these two, we demonstrate that one can estimate a high dimensional functional map effectively. To illustrate this experimentally, we upsample a $k \times k$ functional map obtained with a deep functional map [81]. We vary the size of the functional map k and always upsample it to a fixed maximum size of 120. As shown in Figure 2.5, the accuracy of the resulting 120×120 map is significantly higher, especially when upsampled from low resolutions of 30-90 than that of learning a large functional map directly.

Conclusion In this chapter, we marry two important concepts: learning without overfitting with efficient map refinement strategy that is parameter free, and can help refine even approximate maps. This leads to an effective general purpose pipeline that both avoids overfitting and leads to accurate maps. This technique is further used in GeomFmap [19] and WSupFMNet (Chapter 4 in this thesis.) where we directly learn features from point clouds.

Chapter 3

Weakly Supervised Deep Functional Maps for Point Clouds

A variety of deep functional maps have been proposed recently, from fully supervised to totally unsupervised, with a range of loss functions as well as different regularization terms. However, it is still not clear what are minimum ingredients of a deep functional map pipeline and whether such ingredients unify or generalize all recent work on deep functional maps. In this chapter, we show empirically the minimum components for obtaining state-of-the-art results with different loss functions, supervised as well as unsupervised. Furthermore, we propose a novel framework designed for both full-to-full as well as partial to full shape matching that achieves state of the art results on several benchmark datasets outperforming even the fully supervised methods. Our code is publicly available at https://github.com/Not-IITian/Weakly-supervised-Functional-map

3.1 Introduction

Classical correspondence methods are typically based on handcrafted features or deformation models [104]. In contrast, recent approaches have focused on *learning* an optimal model directly from 3D data. This includes approaches based on template fitting and reconstruction [30, 29], and methods that exploit different definitions of convolution and phrase correspondence as a dense labeling problem [113, 59, 9] among others.

As mentioned in the previous chapter, a prominent direction in learning-based shape matching was pioneered by the FMNet work, [53] by exploiting the *functional map representation* [68] and learning features that recover optimal functional maps rather than e.g. individual point labels. The use of the functional map representation allows to efficiently impose global correspondence constraints, and has been recently been extended in both unsupervised [32, 81] and supervised settings [19]. Despite significant progress in this area, there still exist three major issues.

First, the most recent approach, Deep Geometric Functional Maps [19] is limited to supervised setting that requires ground truth correspondences that are difficult to obtain considering the cost of annotating a dense point-to-point map on each shape pair. Second, despite a variety of deep functional maps-based methods, it is still not clear what are minimum ingredients of a deep functional map pipeline. More importantly, do such minimum ingredients unify or generalize all recent work on deep functional maps? While a battery of loss functions and regularization have been proposed for different

deep functional maps, as we demonstrate below, the devil is not in the loss functions. Instead, using a low number of Laplacian eigen-basis, very weak supervision in the form of rigid alignment and enforcing basic structural properties of the resulting functional map is sufficient to obtain high quality results. Moreover, our approach generalizes to many loss functions proposed recently and does not require geodesic matrices, as in FMNet [53] and UnSupFmnet [32], ground truth maps, as in GeomFmap [19] and FMNet, regularizers, such as descriptor preservation in SURFMNet [81] and regularized FMap layer in GeomFMap [19]. Third, recent learning-based approaches are neither designed nor tested for the *partial shape matching problem* [78, 54] which is of great interest in robotics [17] and Virtual reality applications [88]. To this end, we propose a weakly supervised framework that addresses all three major issues.

Broadly, there are three main components to any deep functional map pipeline, namely feature extractor, choice of basis functions and design of empirical loss or regularization on the functional map. In this chapter, we make contributions on all three fronts. First, we propose to learn feature descriptors directly from raw data with very weak supervision and establish that for non-rigid shape correspondence, rigid alignment supervision turns out to be sufficient to obtain accurate results. Remarkably, this approach also outperforms the fully supervised state-of-the-art methods, which rely on ground truth point-to-point correspondences, on challenging benchmarks. Secondly, we show that the combination of our feature extractor projected to low number of Laplacian eigen basis (30) and an unsupervised loss, consisting of simple regularization terms, suffice to obtain state-of-the-art results for any recently proposed loss functions. Thirdly, to address partial shape matching, we propose a novel data driven method to learn an optimal alignment between source and target Laplacian eigen basis functions which paves the way for future work on deep functional maps in partial shape matching.

3.2 Related Work

Functional Maps Computing point-to-point maps between two 3D discrete surfaces is a very well-studied problem. We refer to a recent survey [83] for an in-depth discussion. Our method builds upon the functional map pipeline. Functional maps encode correspondences as small matrices, expressed in a reduced basis, which greatly simplifies the associated optimization problems. A range of recent works, including [48, 34, 11, 78, 67, 77] among many others, have extended the generality and improved the robustness of the functional map estimation pipeline, by suggesting regularizers, robust penalties and powerful post-processing approaches. Nevertheless, existing non-learning based methods are strongly tied to the choice of descriptor (also known as 'probe') functions, which must be specified manually a priori. We also note that there also exist other techniques that learn correspondences without using the functional map representation, e.g., [113, 9, 62]. However, such techniques typically either require significantly more training data (essentially because they treat shape correspondence as a dense labeling problem with a very large number of labels), or do not learn from 3D geometry which is

the main goal of this chapter.

Supervised Learning from raw 3D shape In contrast to axiomatic approaches that use hand-crafted features, a variety of methods have also been proposed to *learn* the optimal features or descriptors from 3D data. In the functional maps domain, the seminal Deep Functional Maps work [53] proposed a deep learning architecture called FMNet to compute optimal features from data. This architecture was based on optimizing a non-linear transformation of SHOT descriptors, [102] to obtain maps that are as close as possible to given ground truth correspondences. Follow-up works have extended this approach to the unsupervised setting [81, 32] by modifying the training loss, but still used pre-defined descriptors for optimization. These methods generalize poorly across datasets as the input features such as SHOT descriptors are sensitive to the triangle mesh structure, which can vary drastically across different datasets.

Most recently, works including [30, 19] have shown that feature functions can be learned directly from the *raw 3D data* without relying on pre-defined descriptors, resulting in significantly more robust and accurate methods. However, to obtain good results these works had to rely on ground truth correspondences and do not generalize their empirical success beyond their own setup. Although PointNet [72] and its variants ([73]) achieve impressive results from raw point clouds for classification tasks, they are not yet competitive for the shape correspondence task.

In this chapter, weak supervision implies that datasets are only approximately rigidly aligned, which is necessary primarily due to the presence of symmetries. Since some poses (e.g. the neutral pose) are fully extrinsically symmetric, a PointNet like feature extractor cannot distinguish left/right unless the shapes are aligned. Interestingly, as we demonstrate below, such weak supervision is sufficient to obtain high quality results.

Partial Shape Matching While some formulations of functional maps allow to deal with the lack of isometry and presence of partiality, this framework is in principle not designed to deal with partial correspondence. Rodola et al. [78] have provided an empirical evidence and theoretical analysis of a surprising property of interaction between Laplacian eigenfunctions as a result of removing parts from surfaces. This work implies that there exists an unknown alignment between eigenfunctions of partial shapes and full shapes and knowing it results in a special slanted diagonal structure of the correspondence matrix. However, their solution relies on a complicated alternating optimization over the spectral domain and the spatial domain. Instead, [54] proposed an efficient and fully spectral method for finding this transformation matrix between the two eigen spaces. This approach, however, is still based on hand-crafted features, optimization on Stiefel manifold and is instance specific. Besides, replacing handcrafted features by learnable feature descriptors is not straightforward due to manifold optimization involved in the process. We address both of these issues by proposing a novel method that mitigates these issues by learning directly from raw data.

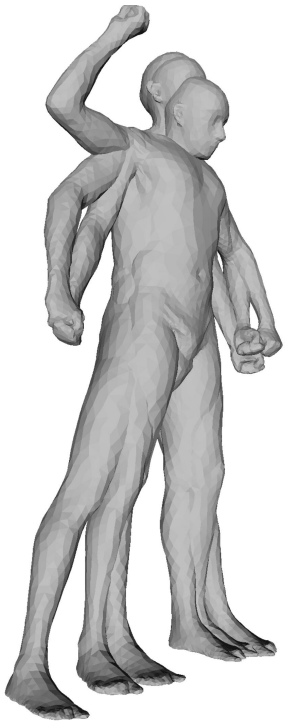

Figure 3.1: An example of approximate alignment from SCAPE that shows the weak supervision used in our work.

3.3 Method

In this section, we first introduce our approach to learning descriptors from raw 3D shapes for full to full shape matching. Afterwards, we detail our novel partial shape matching algorithm that learns an optimal alignment of Laplacian eigen basis functions, given the spectrum of partial and full shape. Note that the feature descriptor extraction is common to both approaches. However, our unsupervised loss function is totally different for partial and full shape matching.

Weak Supervision In both full and partial matching cases, our method is 'weakly supervised' in the sense that we expect the input non-rigid shapes to be approximately *rigidly* aligned. This means having a consistent 'up' direction (along, e.g., the y axis) and an approximate forward-facing direction (along, e.g., the z direction). Some existing datasets, such as partial SHREC, [18], already satisfy this assumption (See Figure 3.2 for example.)

When considering multiple datasets, we only need to make sure that these axes are consistent, which can be done with very little manual intervention. We stress that we *do not use* ground truth point-to-point or functional correspondences, and that obtaining reliable detailed ground truth maps requires significant effort especially when considering cross-dataset learning. Weak supervision is necessary primarily due to the presence of symmetries. Since some poses (e.g. the neutral pose) are fully extrinsically symmetric, *some way* to disambiguate left/right is necessary for accurate correspondence. In our case, we exploit weak supervision in the form of rigid alignment, and further use it to explain the performance drop of a fully supervised methods like GeomFmap, [19], when trained on aligned dataset and tested on SCAPE (non-aligned).

3.3.1 Overview of Architecture

Given a collection of shapes that are approximately rigidly pre-aligned, we train a network N, which takes a pair of point clouds P_1 and P_2, and produces feature functions D_1 and D_2. These feature functions are then projected onto LB eigenbasis. We then estimate a functional map $\mathbf{C}$ by aligning these feature functions. Lastly, we define a structural loss on estimated $\mathbf{C}$ to enable end to end learning. Note that this pipeline is similar to the pipeline shown in Figure 2.1 in Chapter 2. However, in this chapter, we learn directly from point cloud coordinates and with a novel supervision.

PointNet++ Feature extractor As described in Chapter 2, we aim to learn descriptor functions that will be used in the deep functional map framework. Thus, our main goal is to learn functional characterizations of point clouds that will later be used to compute spectral descriptors and then functional maps. Thus, this network must be applied with the same weights to the source and target shapes in a Siamese way using shared learnable parameters. Our feature extractor is based on PointNet ++[73], that extracts local features capturing fine geometric structures from small neighborhoods. Such local

features are further grouped into larger units and processed to produce higher level features. Such hierarchical feature learning with increasing scales of contexts is inspired from convolutional neural networks. To deal with non-uniform densities, it proposes special sampling layers that are able to intelligently aggregate information from different scales.

Our feature extraction network is based on the standard architecture consisting of 4 sampling layers, with first layer sampling 1024 points and 4 feature propagation layers such that final layer outputs 128 dimension feature descriptor for each input shape.

Unsupervised loss for full shape matching Given the extracted feature functions, we first project them onto the Laplacian basis and then compute the optimal functional map by minimizing $\min_{\mathbf{C}} \left\|\mathbf{CA} - \mathbf{B}\right\|^2$. As noted in [53], this leads to a simple linear system of equation, whose solution can be differentiated during training. We therefore train our feature extraction network by imposing an unsupervised loss on the optimized functional map. Our loss follows the approach of [81] and is based on three key structural properties of a functional map between two approximately isometric shapes.

Bijectivity Transporting functions on a shape and transporting them back should yield the same functions. Following [25, 81], we therefore enforce that composition between $\mathbf{C_{12}}$ and $\mathbf{C_{21}}$ to be as closely as possible to $\mathbf{I}$, the identity matrix, which leads to: $E_1 = \|\mathbf{C}_{12}\mathbf{C}_{21} - \mathbf{I}\|^2 + \|\mathbf{C}_{21}\mathbf{C}_{12} - \mathbf{I}\|^2$

Orthogonality As observed in the functional map literature, [68, 82, 81] a point-to-point map is locally area preserving if and only if the corresponding functional map is *orthonormal*. Thus, for shape pairs, approximately satisfying this assumption, a natural penalty in our unsupervised pipeline is: $E_2 = \|\mathbf{C}_{12}^\top\mathbf{C}_{12} - \mathbf{I}\|^2 + \|\mathbf{C}_{21}^\top\mathbf{C}_{21} - \mathbf{I}\|^2$

Laplacian commutativity Having functional maps that commute with the Laplace-Beltrami operators is known to be a common regularizer in the functional map pipeline [80, 68]. We recall that this constraint helps find better mappings since it promotes near-isometric point-to-point maps: $E_3 = \left\|\mathbf{C}_{12}\mathbf{\Lambda}_1 - \mathbf{\Lambda}_2\mathbf{C}_{12}\right\|^2 + \left\|\mathbf{C}_{21}\mathbf{\Lambda}_2 - \mathbf{\Lambda}_1\mathbf{C}_{21}\right\|^2$ where $\mathbf{\Lambda}_1$ and $\mathbf{\Lambda}_2$ are diagonal matrices of the Laplace-Beltrami eigenvalues on the two shapes.

Note that orthogonality and commutativity does not imply bijectivity. One counter example is $\mathbf{I}$ and $-\mathbf{I}$. Both are orthogonal and would commute with any Laplacian. However, they are not inverse of each other.

Our unsupervised loss function is a combination of all three structural properties and weighted as follows: $L = E_1 + E_2 + 0.001E_3$ where the weighing scalars are found empirically.

3.3.2 Basis Alignment for Partial Shape Matching

The basic pipeline described above for shape matching breaks down in the case of partial shape matching. This is primarily because structural properties of the maps such as bijectivity, area preservation (orthogonality) are not applicable anymore. Rodola et al. [78] show that for each partial eigenfunction (i.e., each eigenfunction of a partial shape), there exists a corresponding full eigenfunction (i.e., some eigenfunction of the full shape). The problem then reduces to finding alignment in k dimensional eigen space that is achieved by optimizing for a new basis on one shape only and keeping the other fixed to the standard Laplacian eigenfunctions. Due to this coupling, the new basis functions will behave consistently resulting in almost perfectly diagonal C even in the absence of a perfect isometry. Keeping the same notation as before, where $\mathbf{A}$ and $\mathbf{B}$ represents the PointNet++ descriptors projected onto the laplacian eigen basis, it is written as follows:

$$\min_{\mathbf{X}} \left\|\mathbf{A_r} - \mathbf{X}^\top \mathbf{B}\right\|^2 + \text{off}(\mathbf{X}^\top \mathbf{\Lambda} \mathbf{X}), \tag{3.1}$$

where $\mathbf{A_r}$ contains the $r \times k$ submatrix of $\mathbf{A}$ (the first r rows of matrix $\mathbf{A}$) and $\mathbf{X}$ of size $k \times r$ is a transformation matrix between the two eigen spaces that stores the coefficients of desired linear combination. Note that $\mathbf{A}$ is the partial shape and $\mathbf{X}$ is a complete shape. The value of r is estimated from the spectrum of partial and full shape as follows: $r = \max_{i=1}^{k_p}\{i \mid \lambda_i^p < \max_{j=1}^{k_f} \lambda_j^f\}$ after setting $k_p = k_f = 60$ where f denotes the full shape and p denotes the partial one. We upper bound the rank obtained by 40. $\mathbf{\Lambda}$ is a diagonal matrix of the first k eigenvalues of partial shape. The second term in Eq. (3.1) is a regularizer on $\mathbf{X}$ that ensures that resulting eigen basis functions on partial shape minimize the Dirchelet energy on its Laplace Beltrami operator Δ. The method of [54] obtains the descriptor function matrix $\mathbf{A}$ and $\mathbf{B}$ using precomputed SHOT descriptors. Besides, it constrains $\mathbf{X}$ to be an orthogonal matrix and thus optimize it using manifold optimization solver on Stiefel Manifold. However, we do not impose any orthogonality constraint on $\mathbf{X}$ and optimize Eq. (3.1) differently since our descriptor functions are PointNet ++ based and need to be learned simultaneously. So, instead of optimzing over $\mathbf{X}$, we are optimizing the functional over $\mathbf{X}$, $\mathbf{A}$ and $\mathbf{B}$.

$$\min_{\mathbf{X},\mathbf{A},\mathbf{B}} \left\|\mathbf{A_r} - \mathbf{X}^\top \mathbf{B}\right\|^2 + \text{off}(\mathbf{X}^\top \mathbf{\Lambda} \mathbf{X}), \tag{3.2}$$

We split the functional in Eq. 3.2 in two parts and first optimize for $\mathbf{X}$ by solving $\left\|\mathbf{A_r} - \mathbf{X}^\top \mathbf{B}\right\|^2$ with a simple linear system for which the derivatives can be computed in closed form. Given this optimal $\mathbf{X}$, we then impose the loss on $\mathbf{X}$ by computing the second part of Eq (3.2) and use this unsupervised loss to backpropagate gradients to learn the appropriate descriptor functions. Note that for partial matching this loss term is *the only* one we use, whereas in the full shape matching setting we use a more powerful loss described in Section 3.3.1.

Implementation We implemented our method in TensorFlow [1]. We train our network with a batch size of 24 shape pairs for 10000 steps. We use a learning rate of $1e-4$ with Adam optimizer. During training, we randomly sample 4000 points from each shape

Figure 3.2: An example of approximate alignment from SHREC'16

while training with SURREAL dataset whose shapes contain 7000 points each. For other datasets such as SCAPE and FAUST remesh, that contain roughly 5000 points each, we randomly sample 3000 points during for training. We describe partial shape matching experimental setup later in Section 3.4.3. For a fair comparison with baseline methods, we always use a recent and efficient refining algorithm, called ZoomOut [60] described in Chapter 2 based on navigating between spatial and spectral domains while progressively increasing the number of spectral basis functions.

3.4 Results

This section is divided into three subsections where each provides a separate evaluation of our contributions. Section 3.4.1 shows the experimental comparison of our weakly supervised approach with fully supervised state-of-the art methods for near-isometric shape matching. Section 3.4.2 demonstrates that weak rigid alignment of datasets, low

Table 3.1: Results on remeshed FAUST and SCAPE. F on S means trained on FAUST and tested on SCAPE whereas S on F means trained on SCAPE and tested on FAUST. Note that our weak supervision obtains competitive or even better results than fully supervised methods such as GeomFmap.

Method \ Dataset	F	S	F on S	S on F
SURFMNet	15.	12.	32.	32.
SURFMNet+icp	7.4	6.1	19.	23.
Unsup FMNet	10.	16.	29.	22.
Unsup FMNet+pmf	5.7	10.	12.	9.3
FMNet	11.	17.	30.	33.
FMNet+pmf	5.9	6.3	11.	14.
3D-CODED	2.5	31.	31.	33.
GeomFmap	3.1	4.4	11.	6.0
GeomFmap +zo	**1.9**	**3.0**	9.2	**4.3**
Ours(WSupFMNet)	3.3	7.3	11.7	6.2
Ours(WSupFMNet) + zo	**1.9**	4.9	**8.0**	**4.3**

number of Laplacian eigenbasis and enforcing structural properties of a map suffice to obtain excellent results across a variety of loss functions. Finally, Section 3.4.3 demonstrates the effectiveness of our novel partial shape matching framework. We evaluate all results by reporting the per-point-average geodesic distance between the ground truth map and the computed map. All results are multiplied by 100 for the sake of readability.

3.4.1 Near-isometric Shape Matching

In this section, we evaluate our method for complete (full to full) near isometric shape matching. We compare our method with state-of-the-art approaches while focusing especially on the the very recent functional map-based technique [19], which was shown to outperform existing competitors.

Datasets For a fair comparison with [19], we follow the same experimental setup and test our method on a wide spectrum of datasets: first, the re-meshed versions of FAUST dataset [8] and the SCAPE [3], made publicly available by [77]. Lastly, we also use the training dataset of 3D-CODED, consisting in 230K synthetic shapes generated using SURREAL [105] with the parametric model SMPL introduced in [56]. We use a subset of it for training purposes to compare the generalization ability of different methods to changes in connectivity and triangulation. This is achieved by training on this synthetic data and testing on re-meshed datasets such as FAUST and SCAPE.

Table 3.2: Results when trained on SURREAL and tested on remeshed FAUST and SCAPE. Our weak supervision obtains significantly better results on SCAPE when compared to fully supervised methods.

Method \ Dataset	F	S
GeomFmap +Zo	**2.5**	9.2
3D-CODED	4.9	6.0
Ours (WSupFMNet)	5.0	8.3
Ours (WSupFMNet)+zo	**2.8**	**5.5**

Baselines We compare our method to several state of the art methods: the first category includes a variety of unsupervised deep functional maps proposed recently with SHOT descriptors. The second category includes supervised methods that directly learn from 3D data. This includes the supervised template based approach of 3D-CODED [30] as well as the recent work GeomFmap [19]. All baseline results are taken from [19]. In the case of SHOT based deep functional maps [53, 32, 81], all results are invariant by any rigid transformation of the input shapes and therefore, no alignment is required. For a fair comparison with other methods, we show our results with and without ZoomOut [60] refinement, referred to as ZO. For conciseness, we refer to our method as Ours in the following text. We compare our approach to these different methods in Table 3.1.

Table 3.3: Ablation study of individual losses with and without alignment when trained with Surreal.

Losses	All	E1	E2	E3	(E1+E3)	All-not-aligned
SCAPE	**8.3**	13	16	10.5	9.2	22
FAUST	**5.0**	11	14	9.0	6.3	8.0

Generalization Experiments Following the standard protocol, we split FAUST re-meshed and SCAPE re-meshed into training and test sets containing 80 and 20 shapes for FAUST, and 51 and 20 shapes for SCAPE. **F** and **S** in Table 3.1 shows the results for training and testing on same dataset, FAUST and SCAPE, respectively whereas **F on S** means trained on FAUST and tested on SCAPE. In Table 3.2, results are shown with the SURREAL dataset from which we sample 500 shapes for training and test the trained models on test sets of FAUST re-meshed, SCAPE re-meshed. We compare with 3D-CODED and GeomFMap since they outperform every other method and learn from raw 3D geometry. We report baseline numbers from [19] which report performance of different methods by varying the size of training set from few hundred to thousands. We pick the best results obtained with any number of shapes. All results are multiplied by 100 for the sake of readability. We also report results without ZoomOut refinement.

Results and Discussion As evident in Table 3.1, our weak supervision performs on par with the fully supervised approaches such as 3D-CODED [30] and GeomFMap [19]. We observe comparable or superior performance to the supervised approach in Table 3.2. Note that shapes are pre-aligned for our weak supervision whereas for rest of the approaches, they are not pre-aligned. We obtain a particularly remarkable performance on the SCAPE dataset at test-time when trained with any other dataset. On FAUST, our weak supervision is comparable with GeomFMap even though it is trained with ground truth correspondences.

We would like to stress that baselines such as 3D-CODED and GeomFMap require hundreds of SURREAL shapes, 2000 for 3D-CODED, in order to obtain reasonable results on SCAPE whereas we can obtain high quality results with significant improvement over GeomFMap with as few as 100 and 50 shapes. We stress that no other method is able to achieve such a generalization with this low number of shapes. We attribute our superior results over GeomFmap to a range of factors. First, in contrast to our unsupervised loss, GeomFmap uses a supervised loss without adequate regularization that leads to severe overfitting on challenging datasets with different poses such as SCAPE. This underscores the importance of enforcing structural properties of functional map in any loss function. Second, GeomFmap achieves a robustness to changes in shape orientation using *data augmentation* and ground truth functional map supervision, whereas we align the shapes manually. Third, we assume to be given pre-aligned shapes on the test set (weak supervision) but use the unaligned test set for GeomFmaps that relies on point to point ground truth and data augmentation during training.

These experimental results confirm our findings that the devil in non-rigid shape matching lies in approximate *rigid* alignment and such weak supervision is equivalent to having supervised ground truth correspondence. Compared to GeomFmap, we obtain better results with ZoomOut as it refines initial maps better if they do not contain large errors, e.g. due to symmetries, which we observe with GeomFmap. Also, when the initial maps are good, the refined maps are often similar regardless of initial maps.

Ablation Study We show in Table 3.3 the ablation of our method trained on SURREAL and tested on FAUST and SCAPE. E3 (Laplacian commutativity) is the most important while E2 (Orthonormality) is the least among the three losses. Drastic decrease in performance of our method (All) without weak supervision underlines its importance. The drop is less severe in case of FAUST where one axis is already aligned in contrast to SCAPE that is not aligned at all.

3.4.2 Deep Functional Maps with any Loss Function

The goal of this section is to unpack the minimum ingredients of a deep functional map pipeline such that it leads to unification of all the recent work under these minimum conditions. To this end, we test one representative deep functional map approach each from supervised setting and unsupervised setting with different loss functions. For both

Table 3.4: Comparative results of different loss functions when trained with our framework on Surreal and tested on remeshed Faust and Scape. Note that we train and test various methods on pre-aligned shapes here except for the first row.

Method \ Dataset	F	S
GeomFap+zo	2.5	9.2
Unsup FMNet loss + Ours	6.3	7.7
Unsup FMNet loss + Ours +zo	4.4	5.2
GeomFap loss + Ours	5.0	7.7
GeomFap loss + Ours +zo	2.7	4.6
Ours(WSupFMNet)	5.0	8.3
Ours(WSupFMNet) +zo	2.8	5.5

Table 3.5: Comparative results on the partial SHREC benchmark

Method \ Dataset	Holes	Cuts
Litany et. al	16	**12**
Ours(WSupFMNet)	**12**	15

approaches, we assume to be given the pre-aligned shapes as in our weak supervision. We optimize their loss functions with our PointNet++ feature extractor with low eigenbasis (30) and our regularizers in both functional map pipeline and discard any other regularizer or feature extractor as proposed in these works. We train on the SURREAL dataset from which we sample 500 shapes for training and test the trained models on test sets of FAUST re-meshed, SCAPE re-meshed.

Unsup FMNet loss + Ours UnsupFMNet [32] is an unsupervised approach that uses a soft correspondence based loss with geodesic matrix. Note that in their paper, Unsup FMNet relies on the SHOT descriptor that we replace with a PointNet++ feature extractor. We use their unsupervised loss in addition to our regularization terms.

GeomFmap loss + Ours We also evaluate GeomFmap [19], a supervised approach where the ground truth functional map is computed in the spectral domain. We use this supervised loss function but discard their regularization proposed to alleviate overfitting. We also discard their feature extractor and do not perform any data augmentation. We sample 6000 vertices randomly for each shape as more vertices should lead to a better ground truth functional map estimation. *GeomFmap* simply reports the performance of [19] without any modifications.

Results and Discussion We summarize the findings in Table 3.4. Remarkably, we obtain state of the art results with both loss functions. In particular, GeomFmap supervised spectral loss when optimized with our framework leads to significant increase in accuracy on the challenging SCAPE dataset. This shows the generalization capability of our framework. Similar performance boost is observed with Unsup FMNet on both datasets. It must be noted that memory footprint/training time of [32] is 50 times more as it requires either geodesic matrices to fit to RAM or load them on the fly for each pair. Our approach does not require geodesic matrices, as in FMnet and UnSupFmnet, ground truth maps, as in GeomFmap and FMnet, regularizers, such as descriptor preservation in SURFMNet and regularized FMap layer in GeomFMap. Furthermore, when we remove these components from the respective works and include our minimum components, we get comparable or better results, thus proving the redundancy empirically.

In the figure below, we show the correspondence error curves, using the protocol introduced in [41], that are consistent with average geodesic error shown before.

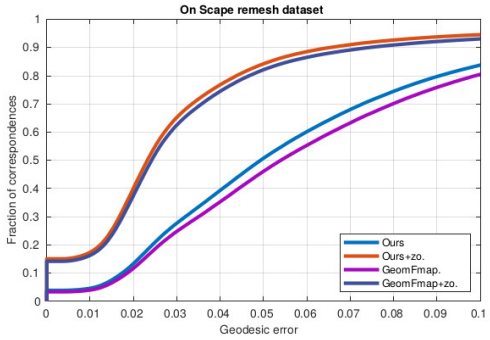

Figure 3.3: Train on SURREAL,Test on SCAPE

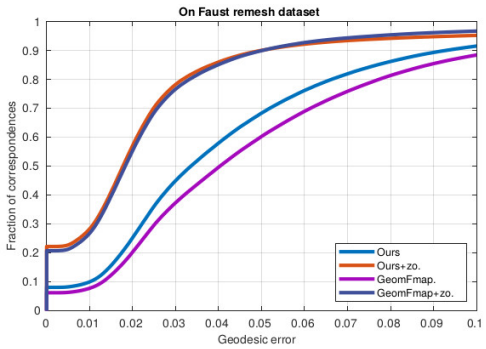

Figure 3.4: Train on SURREAL, Test on FAUST

3.4.3 Partial Shape Matching

Finally, we quantitatively evaluate our method in the partial matching scenario on the challenging SHREC'16 Partial Correspondence benchmark [18]. The dataset is composed of 200 partial shapes (from a few hundred to 9K vertices each) belonging to 8 different classes (humans and animals), undergoing nearly-isometric deformations in addition to having missing parts of various forms and sizes. Each class comes with a null shape in a standard pose which is used as the full template to which partial shapes are to be matched. The dataset is split into two subsets, namely cuts (removal of a few large parts) and holes (removal of many small parts).

Experimental Setup The dataset contains several shapes whose number of points range from few hundreds to 2500. We use some of these shapes as a validation set and separate them from training or test set. Holes dataset is shown to be more challenging

than cuts in [54]. Our loss function for partial shape matching does not contain any hyperparameters. Thus, we use validation set to only validate the training iterations. We consider [54] as our main baseline as it is considered state of the art for partial shape matching and run their code to obtain results. Remark that no existing functional maps learning-based approach has yet been proposed for *partial* non-rigid shape matching. For each class, we obtain $10 - 12$ training shapes and $4 - 5$ test shapes.

Results and Discussion We present our findings on partial shape matching in Table 3.5. We obtain superior results on holes dataset. However, on cuts dataset, Litany et al. obtains better results. We attribute it mainly to the fact that convergence is found to be different for different shapes with our learned model. Thus, a shape in the test set obtains optimal matching at time that is different for other shapes in test set. This could be due to a large fluctuations in the number of points per shape. Results shown here are obtained when our model was trained with a fixed number of iterations for the whole test set. Note that the method of [54] is not learning based but relies on expensive manifold optimization for every pair of shapes at test time. In contrast, our method obtains a correspondence directly with pre-trained features and without the need for any test time optimization.

3.5 Conclusion

We presented a novel weakly supervised method based on the functional map representation for both full and partial shape matching. Our main observation is that weak supervision in the form of approximately rigidly aligned input data is sufficient for learning powerful features to solve the non-rigid correspondence problem from raw data. Moreover, we establish that the key to cross dataset generalization lies in working with low number of eigen basis and enforcing very basic structural properties of a functional map. Our method for partial shape matching is also the first approach towards learning partial functional map and is of independent interest. We believe that this method will set the future direction of research, especially towards simpler techniques and weak supervision, in both near isometric as well as partial shape matching.

Chapter 4

Joint Symmetry Detection and Shape Matching

In Chapters 2 and 3, we introduced two different deep functional map pipelines for shape matching. We have shown that approximate rigid alignment provides a simple way to disambiguate left/right that is necessary for accurate correspondence. However, such approach suffers from scalability issues on large scale 3D shape collections. In Chapter 4, we go beyond this prerequisite and consider the problem of learning simultaneously a self symmetry map and a pairwise map. Despite the success of deep functional maps in non-rigid shape matching, there exists no learning framework that models both self-symmetry and shape matching simultaneously. This is despite the fact that errors due to symmetry mismatch are a major challenge in non-rigid shape matching. In this chapter, we propose a novel framework that simultaneously learns both self symmetry as well as a pairwise map between a pair of shapes. Our key idea is to couple a self symmetry map and a pairwise map through a regularization term that provides a joint constraint on both of them, thereby leading to more accurate correspondences. We validate our method on several benchmarks where it outperforms many competitive baselines on both tasks.

4.1 Related Work

Learning from raw 3D shape Although early approaches in functional maps literature used hand-crafted features, more recent methods aim to *learn* either the optimal transformations of hand crafted descriptors [54, 81] or even features directly from 3D geometry itself [89]. Deep Functional Maps [53] proposed a deep learning architecture called FMNet to optimize a non-linear transformation of SHOT descriptors [102], that was further extended to unsupervised setting [81, 32, 23]. To alleviate the sensitivity of the SHOT descriptor to changes in mesh structure, recent works including [30, 89] learn shape matching directly from the *raw 3D data* without relying on pre-defined descriptors, thus leading to improvements in both robustness and accuracy. However, all these works are aimed at *full* (complete) shape correspondence and do not handle partial shape matching effectively. Our work is also related to a recent work [58] that proposes to replace the Laplace-Beltrami basis with learned embeddings. However, unlike [58], we do not impose linearly invariant constraint between the learned embeddings.

Symmetry for Non Rigid Shape Matching Matching shapes with intrinsic symmetries involves dealing with symmetric ambiguity problem which has been very well studied and explored in axiomatic methods [75, 52, 61, 69, 64, 76]. More recently, [93, 27]

proposes an end to end method to learn extrinsic 3D symmetries from a RGB-D image. However, none of the existing learning based non-rigid shape matching methods models or learn symmetry explicitly as a regularizer for shape matching.

Joint Learning of similar tasks Computer vision literature is full of problems that are inherently linked [46, 87, 22] and thus, should be learned simultaneously. In 3D shape analysis, NeuroMorph [22] simultaneously learns shape correspondence and interpolation. Our work also follows a similar direction as we aim to learn shape matching and symmetry detection simultaneously. Our work is most similar in spirit to [87] that couples image segmentation and detection via linear constraints and thus, induces information transfer/sharing between the segmentation map and detection map via these constraints. In our formulation, we enable this information transfer during training via a commutative loss that couples the self-symmetry and pairwise map.

The rest of this chapter is structured as follows: in the next section, we first propose our method to learn a canonical embedding for joint shape matching and symmetry detection and introduce our novel regularization term that constrains self-symmetry and pairwise map. We then consider the unsupervised setting in which symmetry supervision is not provided. Lastly, we validate our framework on three benchmark datasets by comparing it to various state-of-the-art methods and providing ablation studies.

4.2 Joint Shape Matching and Symmetry Detection

Due to the instability of Laplace-Beltrami operator, LBO, on partial 3D shapes [42] and noise [58], our main goal is to avoid using its eigenfunctions and instead we aim to *learn* an embedding that can replace the spectral embedding given by the LBO. This section details how to learn such an embedding while working in the symmetric space.

Input Shape Representation In contrast to several recent works [32, 89] that assume to be given a mesh representation of 3D shapes in terms of LBO operator, we do not impose such a constraint and work directly with the point cloud representation. We denote a map between a pair of shapes $\mathbf{X}$ and $\mathbf{Y}$ by $T_{\mathbf{XY}} : \mathbf{X} \to \mathbf{Y}$ so that $T_{\mathbf{XY}}(x_i) = y_j$, $\forall i \in \{1, \dots, n_{\mathbf{X}}\}$ and some $j \in \{1, \dots, n_{\mathbf{Y}}\}$. This map can be represented by a matrix $\Pi_{\mathbf{XY}} \in \mathbb{R}^{n_{\mathbf{X}} \times n_{\mathbf{Y}}}$ such that $\Pi_{\mathbf{XY}}(i, j) = 1$ if $T_{\mathbf{XY}}(x_i) = y_j$ and 0 otherwise. We use $P_{\mathbf{X}}$ to denote the 3D coordinates of $\mathbf{X}$.

4.2.1 Supervised Loss functions

In the supervised setting, we assume to be given a set of pairs of shapes $\mathbf{X}, \mathbf{Y}$ for which ground truth correspondences $\mathbf{T}^{gt}_{\mathbf{XY}}$ as well as the ground truth self-symmetry map $\mathbf{T}^{sym}$ are known. Our main goal in the supervised setting is to construct descriptors that lead to self-symmetries, which match $\mathbf{T}^{sym}$. We then show that this regularization in the descriptor space helps to obtain significantly higher-quality correspondences across shape pairs. Our network takes input as $P_{\mathbf{X}}$, 3D coordinates of point clouds, computes

an embedding $\Phi_{\mathbf{X}} \in \mathbb{R}^{n_{\mathbf{X}} \times k}$ for each shape based on a PointNet [72] feature extractor that embeds the shapes into some fixed k dimensional space. The parameters of this feature extractor are learned by optimizing the sum of two loss functions during training as described below.

Cosine Similarity Our loss functions are based on a soft-correspondence matrix, also used in [53] and [58]. The *soft* correspondence matrix $S_{\mathbf{XY}}$ is a soft version of the binary correspondence matrix $\Pi_{\mathbf{XY}}$. We compare the rows of $\Phi_{\mathbf{X}}$ to those of $\Phi_{\mathbf{Y}}$ to obtain the *soft* correspondence matrix $S_{\mathbf{XY}}$ that approximates the pairwise map in a differentiable way as follows:

$$(S_{\mathbf{XY}})_{ij} = \frac{e^{\Phi_{\mathbf{X}}^{i^T} \Phi_{\mathbf{Y}}^{j} / \tau}}{\sum_j e^{\Phi_{\mathbf{X}}^{j^T} \Phi_{\mathbf{Y}}^{j} / \tau}} \tag{4.1}$$

where $\Phi_{\mathbf{X}}^{i^T} \Phi_{\mathbf{Y}}^{j}$ measures the similarity between any two pointwise embeddings and is defined as their inner product and the scalar τ is set to .3.

Nearest Neighbour Loss Our Nearest Neighbour loss links the embeddings of the two shapes and is designed to preserve the given ground truth mapping. Specifically, we first compute the soft correspondence matrix $S_{\mathbf{XY}}$ between a pair of shapes, by comparing the rows of $\Phi_{\mathbf{X}}$ to those of $\Phi_{\mathbf{Y}}$ in a differentiable way as done in Eq. (4.9). We then evaluate the computed soft map, again, by evaluating how well it transfers the coordinate functions, compared to the given ground truth mapping.

$$L(\Phi_{\mathbf{X}}, \Phi_{\mathbf{Y}})_{NN.} = \sum \| S_{\mathbf{XY}} P_{\mathbf{Y}} - \mathbf{T}_{\mathbf{XY}}^{gt} P_{\mathbf{Y}} \|_2^2. \tag{4.2}$$

Note that unlike the linearly invariant loss imposed in [58], this loss is based on comparing $\Phi_{\mathbf{X}}$ and $\Phi_{\mathbf{Y}}$ directly, without computing any linear transformations. This significantly simplifies the learning process and in particular, reduces the computation of the correspondence at test time to a simple nearest-neighbor search. Despite this, as we show below, due to our strong regularization, our approach achieves superior results compared to the method of [58], based on computing an optimal linear transformation at test time.

Symmetry Commutativity Loss Our next loss aims to link the symmetry map computed for each shape and the correspondence across the two shapes. We achieve this by using the algebraic properties of the functional representation, and especially using the fact that map composition can simply be expressed as matrix multiplication.

Specifically, given a self-symmetry pointwise groundtruth maps on shape $\mathbf{X}$ and shape $\mathbf{Y}$, we aim to promote the *consistency* between the computed correspondence and the symmetries on each shape. We do this by imposing the following commutativity loss during training:

$$L(\Phi_{\mathbf{X}}, \Phi_{\mathbf{Y}})_{comm.}^{sup} = \| \mathbf{T}_{\mathbf{X}}^{sym} S_{\mathbf{XY}} - S_{\mathbf{XY}} \mathbf{T}_{\mathbf{Y}}^{sym} \|_2 \tag{4.3}$$

Intuitively, this loss considers the difference between mapping from $\mathbf{X}$ to $\mathbf{Y}$ and applying the symmetry map on $\mathbf{Y}$, as opposed to applying the symmetry on $\mathbf{X}$ and then mapping from $\mathbf{X}$ to $\mathbf{Y}$. Note that this is similar to the commonly used *Laplacian* commutativity in the functional map literature [68]. However, rather than promoting isometries, our loss enforces that the computed map respects the self-symmetry structure of each shape, which holds regardless of the deformation class, and is not limited to isometries.

Overall Loss We combine the two embedding losses defined in (4.11) and (4.3) and write the overall loss as follows:

$$L_{sup.} = L_{NN.} + \gamma * L^{sup}_{comm.} \tag{4.4}$$

4.2.2 Unsupervised Setting

In the previous section, we assume to be given the ground truth self-symmetry map $\mathbf{T}^{sym}$. In this subsection, we propose an unsupervised method that does not require $\mathbf{T}^{sym}$ to be given. To this end, our network takes a shape $\mathbf{X}$ as input and then performs a reflection (flip) of each shape along X-axis resulting in a shape denoted as $\mathbf{X_f}$. The intuition behind such extrinsic flip is to let the network learn two different embeddings for the same shape from which a symmetry map can be computed when no symmetry ground truth is given. We also experimented with other axis but chose a flip along X axis as most of the datasets by default have a symmetry bias along this axis and thus, best performance. The original and flipped shapes are then forwarded to a Siamese architecture, based on a PointNet [72] feature extractor, that embeds these two shapes into some fixed k dimensional space. Let $\Phi_{\mathbf{X}}$ and $\Phi_{\mathbf{X_f}}$ denote the matrices, whose rows can be interpreted as embeddings of the points of $\mathbf{X}$ and $\mathbf{X_f}$.

Self-Symmetry Map We compare the rows of $\Phi_{\mathbf{X}}$ to those of $\Phi_{\mathbf{X_f}}$ to obtain the *soft* correspondence matrix $S_{\mathbf{XX_f}}$ that approximates the self-symmetry map in a differentiable way as follows:

$$(S_{\mathbf{XX_f}})_{ij} = \frac{e^{\Phi^{i^T}_{\mathbf{X}} \Phi^{j}_{\mathbf{X_f}}/\tau}}{\sum_j e^{\Phi^{j^T}_{\mathbf{X}} \Phi^{j}_{\mathbf{X_f}}/\tau}} \tag{4.5}$$

Unsupervised Symmetry Commutativity We enforce a *consistency* between the computed correspondence and the estimated symmetry map in an unsupervised way using commutativity loss as follows:

$$L(\Phi_{\mathbf{X}}, \Phi_{\mathbf{Y}})^{uns}_{comm.} = \|S_{\mathbf{X_f X}} S_{\mathbf{XY}} - S_{\mathbf{XY}} S_{\mathbf{YY_f}}\|_2 \tag{4.6}$$

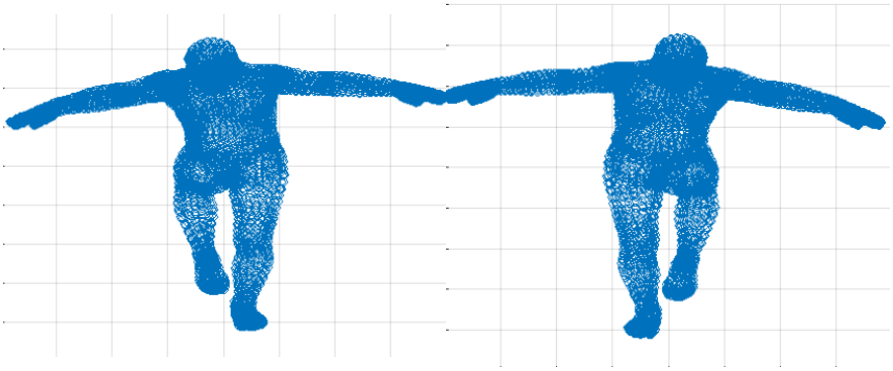

Figure 4.1: On the left, we show a source shape and on the right, we show the flipped version. We compute features from these two shapes in the unsupervised version of our pipeline and use them to compute an unsupervised self-symmetry map.

Overall Loss We combine the two embedding losses defined in (4.11) with that of commutativity loss defined in (4.6) and define the training loss in the unsupervised setting as follows:

$$L_{unsup.} = L_{NN.} + \gamma * L^{uns}_{comm.} \tag{4.7}$$

The scalar γ allows us to weigh the symmetry information differently in a supervised setting where we assume to be given a self-symmetry map and in an unsupervised setting where we work without a symmetry map. Naturally, we set it higher for the supervised case where enforcing symmetry structure makes more sense than unsupervised case where symmetry is induced by a pairwise matching NN loss and transferred by commutativity loss. We set γ to 1 for supervised setting and .2 for unsupervised setting.

Test Phase At test time, once the network is trained, we simply compute the embedding $\Phi_{\mathbf{X}}$ and $\Phi_{\mathbf{Y}}$ and do a nearest neighbour search between them to find correspondence between the two shapes. Similarly, to estimate a self-symmetry map, we compute the embedding $\Phi_{\mathbf{X}}$ and $\Phi_{\mathbf{X_f}}$ and do a nearest neighbour search between them.

Implementation Details We implemented our method in Pytorch [70]. All experiments are run on a Nvidia RTX 2080 graphics processing card with 16 GB of GPU memory. We learn a $k = 20$ dimensional embedding (basis) for each point cloud. Following [89, 58], our feature extractor is also based on the architecture of PointNet. We use a batch size of 8 and learning rate of $1e-4$ and optimize our objective with Adam optimizer in Pytorch [70]. During training, we randomly sample 3000 points from the point cloud and obtain an embedding of 20 dimensions. Our results are not sensitive to small changes in these two parameters. We experimented with an embedding size of $20, 40, 60$ and obtained an average geodesic error in the range $33-36$ on FAUST-R. Similarly, in addition to the 3000 point cloud resolution during training, we also tried a point cloud resolution in the range $2k-4k$ and found our network to be robust to small changes. This can be explained by PointNet resilience to change in point cloud density.

4.3 Results

This section is divided into three subsections. First subsection 4.3.1 shows the experimental comparison of our approach with state-of-the art methods for shape matching and tests our method on a wide spectrum of datasets: the re-meshed versions [77] of the FAUST dataset [8] and SHREC'16 Partial Correspondence dataset [18]. These experiments validate the promising direction of our embedding-based method as it obtains competitive performance on these two benchmarks and especially outperforms LBO based methods on benchmarks with noise. The next subsection 4.3.2 ablates the overall performance and experimentally validates our claim that shape matching with canonical embedding with appropriate regularization outperforms the linearly invariant embeddings proposed in Marin et al. [58]. We demonstrate this with both symmetry supervision as well as without symmetry supervision. Lastly, Section 4.3.3 shows the effectiveness of our method on the symmetry detection task in the presence of noise. We evaluate all results by reporting the per-point-average geodesic distance between the ground truth map and the computed map. All results are multiplied by 100 for the sake of readability. We conclude with an illustration showing a failure case of our method.

4.3.1 Shape Matching

We present our results on a full shape matching benchmark dataset FAUST remesh [8, 77], denoted in future subsections as FAUST-R. We also use its two other versions used previously: the FAUST aligned dataset used in [89], denoted as FAUST-A and noisy FAUST version [58] denoted as FAUST-N. All these datasets contain 100 shapes of 10 different subjects in different poses where each point cloud contains roughly 5000 points. Following prior work, we use the last 20 shapes as a test set and report the performance on this test set. We compare our results with various LBO based methods [19, 23, 89] in Table 4.1 as well as embedding based methods [30, 58] as they are applicable, in principle, to both partial and complete shape matching.

Baselines We compare with the following two broad approaches that are shown to outperform existing competitors:

LBO based Methods. Such baselines [89, 23] assume to be given as input a mesh representation of a shape as they rely on LBO. While [19, 89] directly learn features from raw 3D data similar to our method, they project them into LBO basis. DeepShells [23] refines pre-computed SHOT descriptors [102] to learn shape matching. We provide results after refining the point to point map with ZoomOut [60] where applicable for all the methods. Note that in presence of outliers and noise, such a refinement makes the resulting point to point map worse and thus, for FAUST-N, we do not apply it. DeepShells [23] already has a refinement built in their architecture.

Embedding based Methods. 3D-Coded [30] and Marin et al. [58] are considered state-of-the-art in learning correspondences directly from the point cloud representation

Table 4.1: Avg. Geodesic Error for Shape Matching on FAUST

Method \ Dataset	FAUST-R	FAUST-N
GeomFM[19]+Zo	**1.9**	32
DeepShell[23]	**1.7**	24
Sharma-Ovsjanikov[89]+Zo	5.0	28
3D-Coded[30]	2.5	**6.8**
Marin et al.[58]	7.0	9.0
Marin et al.[58] +Zo	5.0	-
Ours-sym-Sup.	3.3	**5.8**
Ours-sym-Sup.+Zo	**1.8**	-
Ours-sym-Unsup.	5.0	**6.9**
Ours-sym-Unsup.+Zo	**1.8**	-

without relying on LBO. Note that the baseline [58] is somewhat different from others since it requires and thus, learns both basis functions and probe functions (feature descriptors).

Ours. For all results in this chapter, we denote our method with symmetry supervision as Ours-sym-Sup and without symmetry supervision as Ours-sym-Unsup. Here symmetry supervision means the access to the ground truth self-symmetry map that is publicly available for FAUST-R point clouds. While our method already achieves good performance without symmetry ground truth during training, we include Ours-sym-Sup to show the additional gain brought in by additional symmetry supervision during training.

Results and Discussion. As evident in Table 4.1, we obtain competitive performance on FAUST-R. LBO eigen functions already form a good basis for shapes and thus, prior work based on it obtains impressive performance. However, the performance of this line of work degrades significantly under noise, as shown in the Table 4.1 and also in [58]. Thus, our method is significantly more resilient to noise than LBO-based methods. Compared to embedding-based approaches, we obtain slightly better accuracy. In particular, our symmetry-unsupervised version, Ours-sym-Unsup, obtains slightly better performance than our main baseline [58]. We also provide a qualitative example to show comparison with [58] in Figure 4.3. Note that the right foot is mismatched with the method of Marin et al. whereas we transfer it comparatively well without left-right ambiguity. We also provide more qualitative comparison in Figure 4.2. We note that 3D-Coded is also resilient to noise in point clouds and achieves competitive performance in both scenarios.

We also note that the method described in Chapter 3 [89] proposes to align the shapes rigidly and shows that manual rigid alignment resolves some symmetry problems arising in shape matching.

Figure 4.2: On the left, we show a source shape that is to be matched with a target shape. In the middle, we show the color coded map on target shape using the method presented in Chapter 3, WSupFMNet, whereas on the right, we show our color coded map. Notice that WSupFMNet mismatches left and right, our method does not.

In this work, we go further and provide an alternative based on a principled approach that combines symmetry with shape correspondence.

Partial Shape Matching. For a fair comparison with [89, 54], we follow the same experimental setup and test our method on the challenging SHREC'16 Partial Correspondence dataset [18]. The dataset is composed of 200 partial shapes, each containing about few hundreds to 9000 vertices, belonging to 8 different classes (humans and animals), undergoing nearly-isometric deformations in addition to having missing parts of various forms and sizes. Each class comes with a null shape in a standard pose which is used as the full template to which partial shapes are to be matched. The dataset is split into two sets, namely cuts (removal of a few large parts) and holes (removal of many small parts). We use the same test set following [89]. Overall, this test set contains 20 shapes each for cuts and holes datasets chosen randomly from the two sets respectively. In addition to [58], we compare with the following two baselines:

WSupFMNet (Chapter 3) . This baseline relies on learning LBO alignment and thus, is dependent on class and needs to be retrained for each of the 8 classes. We include their results even though our results are class agnostic and thus, significantly more robust and efficient.

Litany et al. [54] . This baseline is not learning based and relies on hand crafted features and an expensive optimization scheme on the Stiefel manifold for every pair of

Table 4.2: Avg. Geodesic Error on partial SHREC benchmarks

Method \ Dataset	Holes	Cuts
Litany et al.[54]	16	13
Sharma-Ovsjanikov [89]	14	16
Marin et al.[58]	12	15
Ours-sym-UnSup.	**10**	**12**

shapes at test time. Thus, in terms of computation, it is most expensive.

Results and Discussion We present our findings on partial shape matching in Table 4.2 where we obtain superior performance on both benchmark datasets for partial shape matching. We would like to stress that baseline methods such as WSupFMNet and Litany et al. are class specific and need to be trained each time for a class whereas our method is class agnostic and can obtain good results with a fraction of computational time. Similarly, [58] trains a similar network as ours two times. First, it learns an embedding with a network similar to ours, followed by a similar network training to compute the optimal linear transformation between the two embeddings. Moreover, the test phase also requires running the network twice. Therefore, our method is 2 times faster than this baseline in computational complexity.

Figure 4.3: On the left, we show the source shape. In the middle, we transfer a color function on a target shape using Marin et al. [58] whereas on the right, we show the transfer using our results.

4.3.2 Ablation Study

In Table 4.3, we ablate the overall performance and validate our claim on two different correspondence map coupling via the commutativity loss.

NN : This baseline ablates the overall performance of our method and quantifies the gain brought in by the pairwise point to point ground truth map alone during training.

Table 4.3: Ablation Study for Shape Matching: Note that just using the nearest neighbour loss on a self-symmetry map and a pairwise map denoted as $NN + NN_{sym}$ overfits badly as there is no coupling between the two maps.

Method \ Dataset	FAUST-R
NN	61
$NN + NN_{sym}$	108
NN+ comm.(sup)	33
NN+ comm.(unsup)	50

It shows the performance if we learn an embedding by just projecting the shapes into a canonical space using point to point pairwise map.

NN+NN_{sym} : This baseline shows the results obtained for shape matching with strong supervision i.e. instead of using commutativity loss defined in Eq 4.3, we replace it with a nearest neighbour loss that preserves the ground truth symmetry map for each shape. This baseline is most important to quantify the coupling effect brought in by our commutative loss.

NN + comm.(sup) : This baseline represents Ours-sym-Sup and quantifies the gain brought in by commutativity loss (Eq 4.3) in supervised setting.

NN + comm.(unsup) : shows the gain brought in by coupling with a commutativity loss in an unsupervised way (Eq 4.6) and represents Ours-sym-Unsup.

Figure 4.4: Left shows a source shape and the right shows our self-symmetry map.

Discussion Our ablation study shows the individual importance of the two loss functions. We note that the performance gains brought in by commutative loss on self-symmetry embeddings are significant. More specifically, as evident in Table 4.3, using just the nearest neighbour loss on a self-symmetry map and a pairwise map, denoted as NN + NN_{sym} in Table 4.3, overfits badly as there is no explicit information transfer or constraint between the two maps.

Table 4.4: Avg. Geodesic Error for self-symmetry maps

Method \ Dataset	Faust-A	Scape-A	Faust-N	Scape-N
Nagar-Raman[64]	34	60	-	-
Ren et al.+Zo[76]	**19**	**54**	166	193
Our-sym-Sup.+Zo	29	63	58	88
Our-sym-Unsup+Zo	50	75	66	95

4.3.3 Symmetry Detection

This subsection evaluates our method on the task of symmetry detection in non-rigid shapes. We evaluate it on FAUST aligned dataset (FAUST-A), SCAPE-A as well as its noisy version. We use the usual train-test split where we test on the last 20 shapes for FAUST-A and last 12 shapes for SCAPE-A. We show the comparative results in Table 4.4 where we compare with multiple baselines.

In particular, Ren et al.[76] is considered state-of-the-art and heavily relies on LBO to estimate self-symmetry maps. We show our results with both symmetry supervision, denoted as Ours-sym-Sup as well as without symmetry supervision denoted as Ours-sym-Unsup in Table 4.4. Similar to Ren et al.[76], we also refine our point to point map by applying ZoomOut to initial maps. For the noisy setting, we simply show the results as such and do not apply ZoomOut refinement as it is based on LBO that is unreliable in a noisy setup. We provide a qualitative example from SCAPE-A in Figure 4.4 and from FAUST-A in Figure 4.5 to illustrate our results.

Figure 4.5: Left shows a source shape and the right shows our self-symmetry map.

Discussion Table 4.4 shows that axiomatic approach of Ren et al.[76] obtains slightly better performance than us on both FAUST-A and SCAPE-A. However, in the presence of noise, its performance suffers significantly. We also remark that we are not aware of any other work that investigates the performance of axiomatic approach for symmetry detection in

the presence of outliers. Our method also undergoes a decrease in accuracy. However, our approach is still resilient to noise and performs significantly better than that of Ren et al.

Failure Case We show a failure case from SCAPE-A in Figure 4.6 where our method finds it challenging to disambiguate symmetry. It maps the right foot of source shape to the left foot on target shape. The method of Marin et al. still performs worse than us as it fails to disambiguate the lower leg of source shape from the lower right leg of target shape. Human poses are quite diverse and this example shows a failure for symmetry detection when training with small data.

Figure 4.6: Failure Case: On the left, we show the source shape. In the middle, we transfer a color function on a target shape using the approach of Marin et al. [58] whereas on the right, we show our result.

4.4 Conclusion

In shape correspondence literature, partial shape matching and full shape matching are generally tackled by two different sets of methods which obtain impressive results in one of the two respective domains. Similarly, symmetry detection and shape matching are also learned or modelled separately. We presented a simple, general but promising approach that provides a unifying framework and reduces pairwise as well as self-symmetry map estimation to a nearest neighbour search in a canonical embedding. Our approach is significantly more resilient to noise than methods based on predefined basis/embedding functions. We believe our key idea of coupling a self-symmetry and a pairwise map via commutativity will encourage future work to explore similar constraints in unsupervised or weakly supervised learning of canonical embeddings.

4.5 Alternative Formulation

In the preceding section, we described a novel way to tackle symmetry issues in a shape matching pipeline and also simultaneously learn a self-symmetry map for each shape. In this section, we propose a slightly different formulation whose aim is to structure the embedding (feature) space so that symmetry is a linear map in it. Thus, different

from preceding section, we enforce a linearly invariant embedding regularization on self-symmetry embedding in the loss function. It comes with a few caveats and trade-off that we explain next.

Our approach is closely related to a recent work [58] that proposes to replace the Laplace-Beltrami basis by learning embeddings that are related by a linear transformation across pairs of shapes. Intuitively, this formulation aims to embed a shape from the 3D space, in which complex non-rigid deformations could occur, to another higher-dimensional space, in which transformations across shapes are linear. However, using a supervised loss to learn this embedding without enforcing any structural properties on the underlying linear transform provides little guarantee that the learned transform will generalize from the train to test setting. Instead, in our formulation, we learn an embedding of each shape that would make the given self-symmetry map linear in some higher-dimensional space. Thus, this formulation imposes a linearly invariant constraint on a dual space and in this way, facilitates shape matching in canonical space. This is advantageous since at test time, we can simply discard the dual space and do shape matching in canonical space without estimating the linear transform. However, we cannot easily estimate a self-symmetry map in this setting as it involves estimating the linear transform enforced in the dual symmetry space and the problems associated with it.

4.5.1 Learning Canonical Embedding

Let $\Phi_{\mathbf{X}}$ and $\Phi_{\mathbf{X_f}}$ denote the matrices, whose rows can be interpreted as embeddings of the points of $\mathbf{X}$ and $\mathbf{X_f}$. In the functional map framework, there exists a functional map $C_{\mathbf{XX_f}}$ that aligns the corresponding embeddings. Given a self symmetry ground truth pointwise map $\mathbf{T}_{\mathbf{XX_f}}$, we can estimate $C_{\mathbf{XX_f}}$ by solving the following optimization problem:

$$C_{\mathbf{XX_f}} = \arg\min_{C} \|\Phi_{\mathbf{X}} C^T - \mathbf{T}_{\mathbf{XX_f}} \Phi_{\mathbf{X_f}}\|_2 \tag{4.8}$$

The optimal symmetry map $C_{\mathbf{XX_f}}$ is given by: $C_{\mathbf{XX_f}} = (\Phi^+_{\mathbf{X}} \mathbf{T}_{\mathbf{XX_f}} \Phi_{\mathbf{X_f}})^T$, that is differentiable using the closed-form expression of derivatives of matrix inverses, as also mentioned in Chapter 2. Similarly, we can compute $C_{\mathbf{YY_f}}$ for shape $\mathbf{Y}$.

Loss functions Similar to previous section, given a set of pairs of shapes $\mathbf{X}, \mathbf{Y}$ for which ground truth correspondences $\mathbf{T}^{gt}_{\mathbf{XY}}$ are known along with a pointwise symmetry map, our network computes an embedding $\Phi_{\mathbf{X}}, \Phi_{\mathbf{Y}}$ for each shape as well as a self symmetry functional map $C_{\mathbf{XX_f}}$ and $C_{\mathbf{YY_f}}$ respectively as described above. We then optimize the sum of three loss functions, one defined for linearly invariant self symmetry embedding, nearest neighbour based loss for pairwise (shape pair) embedding and a commutativity loss. We remark that the last two loss functions are similar to preceding section. So, we only explain linearly invariant loss on self symmetry embedding.

Linearly Invariant Loss The first two loss functions are based on a soft-correspondence matrix, also used in [53] and [58]. To define it for self symmetry map, we *transform*

each shape embedding $\widehat{\Phi}_{\mathbf{X}} = \Phi_{\mathbf{X}} C^T_{\mathbf{X}\mathbf{X}_\mathbf{f}}$ by applying the optimal symmetry map. We then compare the rows of $\widehat{\Phi}_{\mathbf{X}}$ to those of $\Phi_{\mathbf{X}_\mathbf{f}}$ to obtain the *soft* correspondence matrix $S_{\mathbf{X}\mathbf{X}_\mathbf{f}}$ that approximates the self-symmetry map in a differentiable way as follows:

$$(S_{\mathbf{X}\mathbf{X}_\mathbf{f}})_{ij} = \frac{e^{-\|\widehat{\Phi}^i_{\mathbf{X}} - \Phi^j_{\mathbf{X}_\mathbf{f}}\|_2}}{\sum_{k=1}^{n} e^{-\|\widehat{\Phi}^i_{\mathbf{X}} - \Phi^k_{\mathbf{X}_\mathbf{f}}\|_2}} \tag{4.9}$$

We then define our loss that uses this soft-map to transfer the Euclidean coordinates and compares the result to transferring the coordinates using the ground truth map.

$$L(\Phi_{\mathbf{X}}, \Phi_{\mathbf{X}_\mathbf{f}}, \Phi_{\mathbf{Y}}, \Phi_{\mathbf{Y}_\mathbf{f}})_{lin.} = \sum \|S_{\mathbf{X}\mathbf{X}_\mathbf{f}} P_{\mathbf{X}_\mathbf{f}} - \mathbf{T}^{gt}_{\mathbf{X}\mathbf{X}_\mathbf{f}} P_{\mathbf{X}_\mathbf{f}}\|_2^2 + \sum \|S_{\mathbf{Y}\mathbf{Y}_\mathbf{f}} P_{\mathbf{Y}_\mathbf{f}} - \mathbf{T}^{gt}_{\mathbf{Y}\mathbf{Y}_\mathbf{f}} P_{\mathbf{Y}_\mathbf{f}}\|_2^2 \tag{4.10}$$

Note that this does not assume that the Euclidean coordinates to correspond. Instead, this loss measures how well the predicted map transfers a particular set of functions, compared to the ground truth map. This loss was introduced in [58] but we enforce it on the self-symmetry map.

Euclidean Loss The loss described in the previous paragraph only considers the embedding of each shape independently and aims to promote the structural property of this embedding: i.e., that the symmetry map should be linear in the embedding space.

This loss is same as described before in previous section. We first compute the soft correspondence matrix $S_{\mathbf{X}\mathbf{Y}}$ between a pair of shapes as done previously in (4.9). We then evaluate the computed soft map $S_{\mathbf{X}\mathbf{Y}}$, again, by evaluating how well it transfers the coordinate functions, compared to the given ground truth mapping.

$$L(\Phi_{\mathbf{X}}, \Phi_{\mathbf{Y}})_{euc.} = \sum \|S_{\mathbf{X}\mathbf{Y}} P_{\mathbf{Y}} - \mathbf{T}^{gt}_{\mathbf{X}\mathbf{Y}} P_{\mathbf{Y}}\|_2^2. \tag{4.11}$$

Due to an additional regularization, our approach achieves superior results compared to the method of [58], based on computing an optimal linear transformation at test time. We also achieve slightly better results than the approach presented in preceding section.

Overall training Loss We combine the two embedding losses defined in Eq. (4.10) and Eq. (4.11) with that of commutativity loss defined in Eq. (4.3) and define the training loss as follows:

$$L_{tot.} = L_{euc.} + \lambda * L_{lin.} + \gamma * L_{comm.} \tag{4.12}$$

We set λ and γ to 5 for full shape matching.

Test Phase At test time, once the network is trained, we simply compute the embedding $\Phi_{\mathbf{X}}$ and $\Phi_{\mathbf{Y}}$ and do a nearest neighbour search between them to find correspondence between the two shapes. This significantly simplifies the learning process and in particular, reduces the computation of the correspondence at test time to a simple nearest-neighbor search.

Table 4.5: Avg. Geodesic Error on FAUST. We obtain slightly better results than our approach described in preceding section denoted here as Euc. Emb. + comm. We attribute this to linearly invariant constraint on symmetry embedding.

Method \ Dataset	Faust
Marin et al. [58]-3k	08
Marin et al. [58] + sym.	09
Euc. Emb.	12
Euc. Emb. + comm.	07
Euc. Emb. + comm. + Lin. Inv. (Ours)	**06**

Results We briefly discuss our results on FAUST original benchmark. We first describe our baseline methods. **Euc. Emb.** ablates the overall performance of our method and quantifies the gain brought in by the euclidean loss alone during training. It shows the performance if we learn an embedding by just projecting the shapes into a 50 dimensional space with a nearest neighbour euclidean loss. **Euc. Emb. + comm.** This baseline combines the above baseline with the commutativity loss and quantifies what can be achieved without the linearly invariant assumption on self-symmetry map. This is same as the approach described in preceding section. We denote our results with **Euc. Emb. + comm. + Lin. Inv.** in Table 4.5.

Discussion Note that we obtain slightly better results than our approach described in preceding section, denoted here as Euc. Emb. + comm. We attribute this to the regularization brought in by linearly invariant constraint on self-symmetry embedding. However, due to the linearly invariant assumption on self-symmetry embedding, inference becomes much harder to compute a self-symmetry map. Moreover, unlike preceding section, it is not clear in this formulation how to leverage symmetry information in an unsupervised way.

Chapter 5

Graph Alignment for Matrix Completion

In the previous chapters, we described learning based formulations where the optimal neural network parameters were learned from training data. However, in this chapter, we mainly focus on an optimization problem and thus, there is no neural network learning involved. In this chapter, we propose a simplified functional view of matrix decomposition problems on graphs such as geometric matrix completion. Our unifying framework is based on the key idea that using a reduced basis to represent functions on the product space is sufficient to recover a low rank matrix approximation even from a sparse signal. We validate our framework on several real and synthetic benchmarks where it either outperforms very competitive baselines or achieves competitive results at a fraction of the computational effort of prior work. We first define the geometric matrix completion problem below and then propose our functional view of it.

The assumption that high-dimensional data samples lie on or close to a smooth low-dimensional manifold is exploited as a regularizer or prior in many machine learning algorithms. Often, the low-dimensional manifold information is exploited via a graph structure between the data samples. As a result, graphs are often used as a regularizer in various machine learning problems such as dimensionality reduction [37], hashing [55] or matrix completion [40] to name a few. In this chapter, we focus on geometric matrix completion.

Matrix completion deals with the recovery of missing values of a matrix of which we have only measured a subset of the entries. In general, without any constraints, this problem is ill-posed. However if the rank of the underlying matrix is small, it is common to find the lowest rank matrix that agrees with known measurements [14]. Under this low rank assumption, the problem is very similar to dimensionality reduction and can be rewritten as,

$$\min_{\mathbf{X}} \ \operatorname{rank}(\mathbf{X}) + \frac{\mu}{2} \left\| (\mathbf{X} - \mathbf{M}) \odot \mathbf{S} \right\|_F^2 . \tag{5.1}$$

Here $\mathbf{X}$ stands for the unknown matrix, $\mathbf{M} \in \mathbb{R}^{m \times n}$ for the ground truth matrix, $\mathbf{S}$ is a binary mask representing the input support, and $\odot$ denotes the Hadamard product. Various problems in collaborative filtering can be posed as a *matrix completion* problem [40, 74], where for example the columns and rows represent users and items, respectively, and matrix values represent a score determining the preference of user for that item. Often, additional structural information is available in the form of column

and row graphs representing similarity of users and items, respectively. Most of the prior work that incorporates geometric structure into matrix completion problems is either based on highly engineered frameworks, e.g., [63] or a non convex formulation with several hyperparameters [10] thereby making the overall optimization harder to optimize. Instead, our simple formulation based on the functional map representation [68], with a single regularizer, mitigates the problems associated with [10].

Contributions. We propose a convex and smooth matrix decomposition formulation that explicitly imposes and optimizes for a low rank approximation and, as we demonstrate below, is empirically more accurate in recovering a low rank matrix approximation than competitive baselines. Our simplified functional framework also proposes a novel regularization that is shown to be competitive with a combination of several regularizers on various real world datasets. Moreover, we also outline a condition under which a functional map based framework can recover the low rank matrix.

5.1 Related work

Matrix completion has been studied from many viewpoints. In this section, we first briefly cover related work on geometric matrix completion and then describe prior work that is directly related to our work.

Geometric matrix completion. A prominent relaxation of the rank operator in Equation (5.1) is to constrain the space of solutions to be smooth w.r.t. some geometric structure of the matrix rows and columns. There exist several prior works on geometric matrix completion that exploit geometric information [6, 40, 74] such as graphs encoding relations between rows and columns. More recent works leverage deep learning on geometric domains [6, 63] to extract relevant information from graphs. While these techniques achieve state-of-the-art results, their design is highly engineered and thus, non-intuitive and often lacks a proper theoretical foundation.

Graph Regularized Dimensionality Reduction. Jiang et. al. proposed Graph Laplacian PCA (GLPCA) [37] which imposes the graph regularization of principal components using the Dirichlet term for clustering in the low dimensional space. Similarly, the models proposed in [37, 119, 38, 100] leverage the graph structure to learn enhanced class structures. All of these methods still suffer from non-convexity [37, 38, 100]. RPCAG [84] is convex but uses the nuclear norm relaxation that involves an expensive SVD step inhibiting its scalabilty to large datasets. The idea of using two graph regularization terms has also been applied in co-clustering [31], Non negative matrix factorization [86] and more recently in the context of low-rank representation [115]. The co-clustering & NMF based models which use such a scheme [31], [86] suffer from non-convexity and the works of [115] use a nuclear-norm formulation making it difficult to scale. Note that there also exist methods that learn a union of low dimensional subspaces where each

class belongs to a different subspace [24, 106] but they are not directly related to our approach.

Low Rank Estimators. In classical matrix completion or estimation literature, there is large body of work that assumes the underlying signal matrix $\mathbf{M}$ to be low rank and then tries to estimate it using truncated SVD methods [47, 44, 20, 16, 45] as it is the best approximation of a given rank r to the data in the least squares sense. Most of these works estimate this unknown rank and provide bounds on optimality of hard thresholded SVD in an asymptotic framework. Our method is not directly related to these work and we explain it in more detail in the methodology section 5.3.

Functional Maps. Our work is mainly inspired by the functional map framework [68]. This framework has recently been adapted for geometric matrix completion in [10], where the authors propose to build a functional map between graphs of rows and columns. However, they 1) impose several non convex regularization terms each with a scaling hyperparameter and some even with different initialization 2) explore a huge range of hyperparameter space. Instead, we propose a simple and convex formulation with one hyperaparameter.

5.2 Preliminaries

In this section, we cover some preliminaries about product graphs and functional maps.

Product graphs Let $\mathcal{G} = (V, E, W)$ be a (weighted) graph with its vertex set V, edge set E and adjacency matrix denoted by W. Graph Laplacian $\mathbf{L}$ is given by $\mathbf{L} = \mathbf{D} - \mathbf{W}$, where $\mathbf{D} = \text{diag}(\mathbf{W})$ is the *degree matrix*. $\mathbf{L}$ is symmetric and positive semi-definite and therefore admits a spectral decomposition $\mathbf{L} = \mathbf{\Phi}\mathbf{\Lambda}\mathbf{\Phi}^\top$. It is well-known that spectrum of the Laplacian contains the structural information about the graph [98]. Let $\mathcal{G}_1 = (V_1, E_1, W_1)$, $\mathcal{G}_2 = (V_2, E_2, W_2)$ be two graphs, with $\mathbf{L}_1 = \mathbf{\Phi}\mathbf{\Lambda}_1\mathbf{\Phi}^\top$, $\mathbf{L}_2 = \mathbf{\Psi}\mathbf{\Lambda}_2\mathbf{\Psi}^\top$ being their corresponding graph Laplacians. We define the Cartesian product of $\mathcal{G}_1$ and $\mathcal{G}_2$, denoted by $\mathcal{G}_1\mathcal{G}_2$, as the graph with vertex set $V_1 \times V_2$.

Functional maps. Let $\mathbf{X}$ be a function defined on $\mathcal{G}_1\mathcal{G}_2$. It can be encoded as a matrix of size $|V_1| \times |V_2|$. Then it can be represented using the bases $\mathbf{\Phi}, \mathbf{\Psi}$ of the individual graph Laplacians, $\mathbf{C} = \mathbf{\Phi}^\top\mathbf{X}\mathbf{\Psi}$. In the shape processing community, such $\mathbf{C}$ is called a *functional map* [68] as it is used to map between the functional spaces of $\mathcal{G}_1$ and $\mathcal{G}_2$. One of the advantages of working with the functional map representation $\mathbf{C}$ rather than the matrix $\mathbf{X}$ is that its size is typically much smaller, and is only controlled by the size of the basis, independent of the number of nodes in G_1 and G_2, resulting in simpler optimization problems. Moreover, the projection onto a basis also provides a strong regularization, which can itself be beneficial for both shape matching, and, as we show below, matrix completion.

5.3 Low Rank Matrix Decomposition

We assume that we are given a set of samples in some matrix $\mathbf{M} \in \mathbb{R}^{m \times n}$. In addition, we construct two graphs $\mathcal{G}_{\mathrm{r}}, \mathcal{G}_{\mathrm{c}}$, encoding relations between the rows and the columns, respectively. We represent the Laplacians of these graphs and their spectral decompositions by $\mathbf{L}_{\mathrm{r}} = \mathbf{\Phi}\mathbf{\Lambda}_{\mathrm{r}}\mathbf{\Phi}^{\top}$, $\mathbf{L}_{\mathrm{c}} = \mathbf{\Psi}\mathbf{\Lambda}_{\mathrm{c}}\mathbf{\Psi}^{\top}$. For the matrix completion problem, the matrix $\mathbf{M}$ is not completely known so we are also given a binary indicator mask $\mathbf{S}$ that indicates 1 for measured samples and 0 for missing ones. We minimize the objective function of the following form:

$$\min_{\mathbf{X}} \; E_{\mathrm{data}}(\mathbf{X}) + \mu E_{\mathrm{reg}}(\mathbf{X}) \tag{5.2}$$

with E_{data} denoting a data term of the form

$$E_{\mathrm{data}}(\mathbf{X}) = \|(\mathbf{X} - \mathbf{M}) \odot \mathbf{S}\|_F^2 \,, \tag{5.3}$$

As observed in [10], we can decompose $\mathbf{X} = \mathbf{\Phi}\mathbf{C}\mathbf{\Psi}^{\top}$ where $\mathbf{C}$ is some unknown matrix to be optimized. Remarkably, the data term itself, as we show in our experiments, when expressed through the functional map i.e.$\mathbf{X} = \mathbf{\Phi}\mathbf{C}\mathbf{\Psi}^{\top}$ already recovers low-rank matrices and outperforms the recent approach of [10] on synthetic geometric experiments for matrix completion and obtains competitive results on dimensionality reduction tasks. Before we explain the choice and motivation of our regularizer E_{reg}, we explain next why the data term itself already works remarkably well on rank constrained geometric problems.

5.3.1 Motivation and Analysis

Suppose that we constrain $\mathbf{X}$ to be a matrix such that $\mathbf{X} = \mathbf{\Phi}\mathbf{C}\mathbf{\Psi}^{\top}$ for some matrix $\mathbf{C}$. Note that if $\mathbf{\Phi}$ and $\mathbf{\Psi}$ have k columns each then $\mathbf{C}$ must be a $k \times k$ matrix. We would like to argue that solving the optimization problem in Equation (5.3) under the constraint that $\mathbf{X} = \mathbf{\Phi}\mathbf{C}\mathbf{\Psi}^{\top}$ will recover the underlying ground truth signal if it is low rank and satisfies an additional condition that we call basis consistency.

For this suppose that the ground truth hidden signal $\mathbf{M}$ has rank r. Consider its singular value decomposition $\mathbf{M} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\top}$. As $\mathbf{M}$ has rank r, $\mathbf{\Sigma}$ is a diagonal matrix with r non-zero entries. We will call $\mathbf{M}$ *basis-consistent* with respect to $\mathbf{\Phi}, \mathbf{\Psi}$ if the first r left singular vectors U_r (i.e., those corresponding to non-zero singular values) lie in the span of $\mathbf{\Phi}$, and the first r right singular vectors V_r lie in the span of $\mathbf{\Psi}$. In other words, there exist some matrices $\mathbf{R}, \mathbf{Q}$ s.t. $U_r = \mathbf{\Phi}\mathbf{R}$ (note that this implies $k \geq r$) and $V_r = \mathbf{\Psi}\mathbf{Q}$. Given this definition, it is easy to see that all basis-consistent matrices with rank $r \leq k$ can be represented by some functional map $\mathbf{C}$. In other words, given $\mathbf{Y}$ that is basis-consistent, there is some functional map $\mathbf{C}$ s.t. $Y = \mathbf{\Phi}\mathbf{C}\mathbf{\Psi}^{T}$. Conversely any $\mathbf{X} = \mathbf{\Phi}\mathbf{C}\mathbf{\Psi}^{T}$ has rank at most k and must be basis-consistent by construction.

Second, suppose we are optimizing Equation (5.3) under the constraint $\mathbf{X} = \mathbf{\Phi C \Psi}^\top$ and that the optimum, i.e., the ground truth matrix $\mathbf{M}$, is basis-consistent. Then since the energy $E_{\text{data}}(C)$ is convex, given k^2 known samples to fully constrain the corresponding linear system, we are guaranteed to recover the optimum low-rank basis-consistent matrix.

This simple observation suggests that by restricting $\mathbf{X} = \mathbf{\Phi C \Psi}^\top$ and optimizing over the matrices $\mathbf{C}$ instead of $\mathbf{X}$ already provides a strong regularization that can help recover appropriate low-rank signals even without any other regularization. Further, it avoids solving complex optimization problems involving iterative SVD, since $\mathbf{C}$ becomes the only free variable, which can be optimized directly. For problems such as geometric matrix completion, we observe that a weak additional regularization is often sufficient to obtain state-of-the-art results. More formally, we state our result as follows

Proposition 1 *We recover an optimal low rank matrix with high probability as long as the underlying latent matrix* $\mathbf{X}$ *is basis consistent.*

Proof: The proof is based on the main result (Theorem 1 in [14]) in low rank exact matrix recovery method. [14] prove that there is a unique rank k matrix that agrees with the sampled values with high probability and thus, recovers this underlying hidden signal matrix. Our method also recovers a rank k matrix by construction. Since our problem is convex, our method will recover the best rank k matrix that is within the span of the eigenfunctions. If the underlying matrix is basis consistent, then our method will recover the same exact matrix as a low rank exact recovery method (by definition of basis consistency).

Note that we verified on the Synthetic Netflix dataset that our basis consistency condition is indeed satisfied for small values of k such as 30 and 50.

5.3.2 Laplacian Commutativity as a Regularizer

Our E_{reg} contains a single regularization term on the functional map induced between row space and column spaces described next. We propose to use the simplest possible regularizer, which furthermore leads to a convex optimization problem and can achieve state-of-the-art results. For this, we borrow a condition that is prominent in the functional map literature [68]. Namely, in the context of surfaces, the functional map is often expected to *commute with the Laplace-Beltrami operator*: $E_{\text{reg}} = \left\| \mathbf{C\Lambda}_{\text{r}} - \mathbf{\Lambda}_{\text{c}}\mathbf{C} \right\|^2$ where $\mathbf{\Lambda}_{\text{r}}$ and $\mathbf{\Lambda}_{\text{c}}$ are diagonal matrices of Laplacian eigenvalues of the source graph (row graph) and target graph (column graph). More broadly, commutativity with the Laplacian imposes a diagonal structure of the functional map, which intuitively promotes preservation of low frequency eigenfunctions used in the basis. In the context of matrix completion, this can be interpreted simply as approximate preservation of global low frequency signals defined on the two graphs.

Given these above definitions, our problem defined in equation (5.2) reduces to

$$\min_{\mathbf{C}} \|(\mathbf{X}-\mathbf{M})\odot\mathbf{S}\|_F^2 + \mu * \left\|\mathbf{C}\mathbf{\Lambda}_\mathrm{r} - \mathbf{\Lambda}_\mathrm{c}\mathbf{C}\right\|^2 \quad \text{where } \mathbf{X} = \mathbf{\Phi}\mathbf{C}\mathbf{\Psi}^\top \tag{5.4}$$

In practice, we observe faster convergence if we replace $\mathbf{C}$ with $\mathbf{P}\mathbf{C}\mathbf{Q}^\top$, and let all three $\mathbf{P}, \mathbf{C}$ and $\mathbf{Q}$ be free variables.

Differences from SGMC [10] Even though both methods, ours and SGMC build on the functional map framework, there is a fundamental difference between the two. SGMC focuses on high complexity functional map based model (large values of $\mathbf{C}$, multiple resolutions of $\mathbf{C}, \mathbf{P}, \mathbf{Q}$) and thus, requires a variety of (non-convex) regularizers. In contrast, our core idea is to represent the low rank matrix recovery based on the functional map based decomposition alone $\mathbf{X} = \mathbf{\Phi}\mathbf{C}\mathbf{\Psi}^\top$ (See 'Ours-FM' baseline in experiments Section 5.2).

To outline the differences more precisely, in addition to the Dirichlet energy on the two graphs, [10] also introduces two non-convex regularizations on the transformation matrix $\mathbf{P}$ and $\mathbf{Q}$. Non-convexity comes from orthogonality constraint on $\mathbf{P}$ and $\mathbf{Q}$. Additionaly, [10] also uses a multi-resolution spectral loss named SGMC-Zoomout (SGMC-Z) [60] with its own hyperparameters (step size between different resolutions) besides several scalars to weigh different regularizations.

Hyperparameters The optimization is carried out using gradient descent in Tensorflow [1]. For all experiments, we set μ and the learning rate to be .00001. We report the size of $\mathbf{C}$ explicitly in each experiment below. For geometric matrix completion, we divide the number of available entries in the matrix randomly into training and validation set in a 95 to 5 ratio respectively.

5.4 Experiments

In the first half of this section, we extensively compare between our approach and Spectral Geometric Matrix Completion **(SGMC)**[10] on a synthetic example of a community structured graphs. In the latter half, we compare with all approaches on various real world recommendation benchmarks. For a fair comparison with [10], we use graphs taken from the synthetic Netflix dataset. Synthetic Netflix is a small synthetic dataset constructed by [40] and [63], in which the user and item graphs have strong community structure. Similar to [10], we use a randomly generated low rank matrix on the product graph $\mathcal{G}_\mathrm{c}\mathcal{G}_\mathrm{r}$ to test the matrix completion accuracy. Synthetic Netflix is useful in conducting controlled experiments to understand the behavior of geometry-exploiting algorithms.

Table 5.1: Testing the dependence on the density of the sampling set for a random rank 10 matrix of size 150×200. In the data-poor regime, our regularization is strong enough to recover the matrix unlike other methods.

Density in %	Ours	Ours-FM	SGMC
1	**2e-2**	2e-2	1e-1
5	**8e-7**	1e-3	5e-4
10	**2e-7**	5e-5	2e-4
20	**1e-7**	2e-5	1e-4

Table 5.2: Testing the robustness to noisy graphs. Our method is robust to substantial amounts of noise in graphs.

Noise	Ours	Ours-FM	SGMC
5	**1e-3**	2e-3	5e-3
10	4e-3	**3e-3**	1e-2
20	**6e-3**	6e-3	1e-2

Graph Construction We follow the setup of [40] and use the graphs constructed by them as described next. The row graph $\mathcal{G}_{\mathrm{r}}$ of the matrix $\mathbf{M}$ is constructed as follows. The rows of $\mathbf{M}$ are grouped into 10 communities of different sizes. We connect nodes within a community using a 3-nearest neighbors graph and then add different amounts of erroneous edges, that is, edges between vertices belonging to different communities. The erroneous edges form a standard Erdos-Renyi graph with variable probability. We follow the same construction process for the column graph $\mathcal{G}_{\mathrm{c}}$ that contains 12 communities. For both graphs, binary edge weights are used. The intuition behind this choice of graphs is that users form communities of people with similar taste. Likewise, movies can be grouped according to their type, so that movies of the same group obtain similar ratings.

We consider the following two baselines:

Ours-FM : This baseline only optimizes for $\mathbf{C}$ without any regularization. All results are obtained with $\mathbf{C}$ of size 30×30. This value was chosen after a cross validation from a set of $20, 30, 40$. **SGMC**: All results are obtained with their open source code with their optimal parameters.

Test Error. To evaluate the performance of the algorithms in this section, we report the *root mean squared error*,

$$\mathrm{RMSE}(\mathbf{X}, \mathbf{S}) = \sqrt{\frac{\|(\mathbf{X} - \mathbf{M}) \odot \mathbf{S}\|_F^2}{\sum_{i,j} \mathbf{S}_{i,j}}} \tag{5.5}$$

Table 5.3: Perturbation in the rank of the underlying matrix. As the rank increases, it becomes harder for other methods to recover the matrix

Rank	Ours	Ours-FM	SGMC
5	**1e-7**	2e-5	1e-4
10	**2e-7**	2e-5	2e-4
12	**5e-7**	4e-5	9e-4
15	**6e-3**	1e-3	1e-2

computed on the complement of the training set. Here $\mathbf{X}$ is the recovered matrix and $\mathbf{S}$ is the binary mask representing the support of the set on which the RMSE is computed. We compare the two approaches on different constraints ranging from rank of the underlying matrix to the sampling density. Note that optimality bounds for classical matrix completion algorithms also depend on constraints such as sampling density, noise variance etc.

Rank of the underlying matrix. We observe that as the rank increases up to 15, it becomes harder for both methods to recover the matrix. We remark that Ours-FM alone recovers the low rank very effectively. However, on real data, we find the additional regularizer in Ours to be more effective than Ours-FM. We also remark that Ours-FM consistently outperforms SGMC for all rank values. For the training set we used 10% of the points chosen at random (same training set for all experiments summarized in Table 5.3).

Sampling density. We demonstrate that in the data-poor regime, our regularization is strong enough to recover the matrix, compared to performance achieved by incorporating geometric regularization through SGMC. These experiments are summarized in Table 5.1. Note that gap between us and SGMC remains high even when the sample density increases to 20%. Even when using 1% of the samples, we perform better than SGMC.

Noisy graphs. We follow the same experimental setup as [10] and perturb the edges of the graphs by adding random Gaussian noise with zero mean and tunable standard deviation to the adjacency matrix. We discard the edges that became negative as a result of the noise, and symmetrize the adjacency matrix. Table 5.2 demonstrates that our method is robust to substantial amounts of noise in graphs. Surprisingly, Ours-FM demonstrates even stronger resilience to noise.

Runtime Comparison. Our method runs 20 times faster than SGMC when compared on synthetic experiments described above. This is not surprising as SGMC involves optimizing various regularizers and with high values of $\mathbf{P}, \mathbf{C}, \mathbf{Q}$.

Table 5.4: Test error on Flixster and Movielens-100K

Model	Flixster	ML-100K
MC [14]	1.533	0.973
GMC [40]	–	0.996
GRALS [74]	1.245	0.945
RGCNN [63]	0.926	0.929
GC-MC [6]	0.917	0.910
Ours-FM	1.02	1.12
DMF[4]	1.06	0.922
SGMC	0.900	**0.912**
SGMC-Z	**0.888**	0.913
Ours	**0.888**	0.915

In addition to synthetic Netflix, we also validate our method on two more recommender systems datasets for which row and column graphs are available. Movielens-100K [33] contains ratings of 1682 items by 943 users whereas Flixter [36] contains ratings of 3000 items by 3000 users. All baseline numbers, except Ours-FM, are taken from [63] and [10]. In addition to **SGMC** and **SGMC(Z)**, we also compare with **DMF**[4]. This is a matrix factorization approach that was adapted for matrix completion tasks by [10]. Note that this approach does not incorporate any geometric information. We explain some observations from Table 5.4: first, our baseline, Ours-FM, obtains surprisingly good performance across all datasets. This underscores the regularization brought in by the Laplacian eigen-basis of row and column graphs. Second, the non geometric model DMF shows competitive performance with all the other methods on ML-100K. This suggests that the geometric information is not very useful for this dataset. Third, our proposed algorithm is competitive with the other methods while being simple and interpretable. Lastly, these experimental results validate the effectiveness of our single regularization when compared to the combination of several non-convex regularizations introduced in [10].

5.5 Graph Regularized Dimensionality reduction

In the previous section, we describe the application of functional maps to geometric matrix completion problem. In this section, we extend the same formulation to Graph regularized Dimensionality reduction. Given a data matrix $\mathbf{M} \in \mathbb{R}^{m \times n}$ with n m-dimensional data vectors, we seek to find a lower dimensional representation of this data which will lead to better clustering and classification performance. In contrast to the previous setting, we are given the entire matrix $\mathbf{M}$ as input here.

Most prior work related to PCA [2] can be broadly categorised in two themes: 1) matrix factorization approach of the classical PCA and its variants 2) matrix subtraction approach of robust PCA [13] and its variants. The former learns the projection of M in some lower d-dimensional linear space. Several followup works including Graph Laplacian

PCA [37, 119, 38, 100] have shown that the clustering quality of PCA can be significantly improved by incorporating the data manifold information in the form of some underlying graph structure. All of these methods still suffer from non-convexity [37, 38, 100].

Instead of relying on matrix factorization, the second line of work directly estimates clean low rank approximation $\mathbf{X}$ of data matrix $\mathbf{M}$ by separating noise with a matrix additive model. Along these lines, Fast Robust PCA on graphs (FRPCAG [85]) introduces a joint notion of reduced rank for the rows and columns of a data matrix and proposes to jointly minimize the Dirichlet energy on the row and column graphs:

$$\min_{\mathbf{X}} \|\mathbf{M}-\mathbf{X}\|_1 + \gamma_1 \operatorname{tr}(\mathbf{X}\mathbf{L}_1\mathbf{X}^\top) + \gamma_2 \operatorname{tr}(\mathbf{X}^\top\mathbf{L}_2\mathbf{X}). \tag{5.6}$$

Here $\mathbf{L}_1, \mathbf{L}_2$ are Laplacian matrices of graphs built, respectively, from the rows and columns of the data matrix $\mathbf{M}$. We describe the graph construction in next section. Conceptually, minimizing the Dirichlet energy, $\operatorname{tr}(\mathbf{X}\mathbf{L}_1\mathbf{X}^\top)$, promotes smoothness of $\mathbf{X}$ by penalizing high frequency components of the signals on corresponding graphs. The authors of FRPCAG [85] demonstrate theoretically that under certain assumptions this minimization is connected with the spectrum of the underlying low rank matrix $\mathbf{X}$. Building on this idea, we instead directly constrain the low rank approximation by decomposing it using the first few eigenvectors of row and column graph Laplacians $\mathbf{X} = \mathbf{\Phi}\mathbf{C}\mathbf{\Psi}^\top$ and optimizing over the coupling matrix $\mathbf{C}$ only.

Our Approach for Dimensionality Reduction. For dimensionality reduction, we optimize the data term alone i.e. $E_{\text{data}}(\mathbf{X}) = \|(\mathbf{X}-\mathbf{M})\|_F^2$ under the constraint $\mathbf{X} = \mathbf{\Phi}\mathbf{C}\mathbf{\Psi}^\top$. The resulting low rank matrix is then considered a new representation of original data matrix $\mathbf{M}$. One can use this new representation directly for clustering using k-means algorithm.

Differences from FRPCAG [85] We do not target the Robust PCA problem [13] as done in FRPCAG. FRPCAG obtains a low rank approximation by minimizing the Dirichlet energy on the two graphs and thus, only implicitly obtains a low rank approximation. In contrast, we explicitly factorize the data matrix. As shown in our experiments below, this explicit control over the resulting low rank of matrix, by optimizing over $\mathbf{C}$, yields superior clustering results over FRPCAG.

Graphs Construction Following [85], we use two types of graphs G_1 and G_2 in our proposed model. The graph G_1 is constructed between the data samples or the columns of the data matrix and the graph G_2 is constructed between the features or the rows of the data matrix. The graphs are undirected and built using a standard K-nearest neighbor strategy. We connect each x_i to its K nearest neighbors x_j where K is 10. This is followed by the graph weight matrix A computation as

$$A_{ij} = \begin{cases} \exp\left(-\frac{\|(x_i - x_j)\|_2^2}{\sigma^2}\right) & \text{if } x_j \text{ is connected to } x_i \\ 0 & \text{otherwise.} \end{cases}$$

Table 5.5: Clustering purity on Benchmark Datasets.

Dataset	Samples	PCA	LE	GLPCA	GRPCA	FGRPCA	Ours
ORL	400	57	56	68	74	77	**79**
COIL20	1404	67	56	66	65	68	**71**
MFEAT	400	82	90	71	80	85	**90**
BCI	400	52	52	52	53	52	**53**

5.5.1 Graph Regularized Dimensionality Reduction

Datasets We use 4 well-known benchmarks and perform our clustering experiments on following databases: ORL, BCI, COIL20, and MFEAT. ORL[1] is a face database with small pose variations. COIL20 [2] is a dataset of objects with significant pose changes. MFeat[3] consists of features extracted from handwritten numerals whereas BCI database consists of features extracted from a Brain Computer Interface setup [4].

Baselines

We compare the clustering performance of our model with 5 other dimensionality reduction models. Apart from classical PCA, the rest of the models exploit graph information.

Models using graph structure : We compare 1) Graph Laplacian PCA (GLPCA) [37] 2) Laplacian Eigenmaps (LE) 3) Robust PCA on graphs RPCAG [84] 4) Fast Robust PCA on graphs FRPCAG [85] 5) Our proposed model. Note that RPCAG and FRPCAG are closest to our approach and known to outperform other graph regularized models such as Manifold Regularized Matrix Factorization (MMF) [119], Non-negative Matrix Factorization (NMF)[50], Graph Regularized Non-negative Matrix Factorization (GNMF) [12]. We obtain FRPCAG and RPCAG results by running the open source implementation provided by the authors on the four datasets. Note that we run the clustering on all the samples of COIL20 and all the features of MFEAT whereas FRPCAG only use a subset of them in their paper. FRPCAG contains two hyperparameters, namely weighing scalars for Dirichlet energy. For these scalars, we pick the best value from the set (1,10,50,100) based on empirical performance. For PCA, we chose the first 40 principal components from a set (30, 40, 50). For our method, the only hyper-parameter is the dimensionality of matrix $\mathbf{C}$. We pick the best value out of (50, 100).We pre-process the datasets to zero mean and unit standard deviation along the features.

Clustering Metric We follow the standard evaluation protocol and use clustering purity to evaluate our method. To compute purity, each cluster is assigned to the class which is most frequent in the cluster, and then the accuracy of this assignment is measured

[1] cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html
[2] cs.columbia.edu/CAVE/software/softlib/coil-20.php
[3] archive.ics.uci.edu/ml/datasets/Multiple+Features
[4] olivier.chapelle.cc/ssl-book/benchmarks.html

by counting the number of correctly assigned and dividing by the total no. of samples. We report the maximum clustering error from 10 runs of k-means and summarize our findings in Table 5.5.

Table 5.6: Classification accuracy on Benchmark Datasets.

Dataset	**PCA**	**LE**	**FGRPCA**	**Ours**
ORL	63	56	66	**68**
COIL20	88	78	88	**89**
MFEAT	**97**	94	**97**	**97**
BCI	52	48	53	**55**

As shown in Table 5.5, our model obtains superior or competitive performance over all other baselines.

Classification We further evaluate our framework on the classification task on the same 4 datasets. We perform classification with PCA, LE and our data representations using a KNN classifier. We randomly select 30% of labeled data, and use the rest to evaluate. We repeat this 5 times and summarize the average classification accuracy in Table 5.6. Our method obtains competitive accuracy compared to other baselines. PCA representation with first 40 components already provides very competitive classification results on several datasets.

Conclusion We establish empirically and theoretically that using a reduced basis to represent a function on the product space of two graphs already provides a strong regularization, that is sufficient to recover a low rank matrix approximation. Moreover, by extensive experiments, we show that our functional map based framework is very competitive when compared to some complex baselines proposed before for geometric matrix completion as well graph regularized dimensionality reduction.

Chapter 6

Conclusion, Extensions and Future Work

In this thesis, we have made several contributions to Deep Functional Maps which we summarize in this chapter. In the next subsection, we provide a quick overview of the impact of our contributions since their publication and of other closely related works. Afterwards, we propose possible future extensions of our work.

In Chapter 2, we coin the term spectral overfitting and demonstrate that naively learning a high dimensional map directly with Deep Functional Maps [53] leads to severe overfitting. Besides, we propose to learn a higher resolution map by learning a low resolution map with a Deep Functional Map approach and then refining it with iterative spectral upsampling.

In Chapter 3, we propose a novel form of weak supervision that allows learning shape matching over point clouds with Deep Functional Maps. We show that approximate alignment provides the network enough information to disambiguate symmetry mismatch and thus, acts as a proxy for ground truth correspondences. Moreover, we also propose a simple but effective loss to learn partial shape matching.

In Chapter 4, we go beyond this prerequisite of approximate rigid-alignment and consider the problem of learning simultaneously a self symmetry map and a pairwise map. We propose a novel commutative regularization that couples the self-symmetry with a pairwise map computation and thus, enables knowledge transfer between the two maps during training. To the best of our knowledge, we propose the first method that simultaneously learns symmetry detection and shape matching for non-rigid point clouds.

In Chapter 5, we extend the notion of functional spaces to graphs and propose a novel application of functional map framework to graph based machine learning problems such as geometric matrix completion.

6.0.1 Follow up works

Since its publication, our work has been cited in multiple papers. In this section we list some recent papers that either reuse or cite the work presented in this thesis or simply propose new approaches to the problems we considered. For each one we explain how it relates to our work.

Learning High Dimensional Functional Map

In Chapter 2, we introduced a simple but remarkably effective technique to learn high resolution functional map. We advocate learning with lower resolution embedding and refining it later with an axiomatic spectral upsampling approach. Many follow up works have used our technique or our insights to obtain accurate correspondence for non-rigid 3D shapes. In the following, we list some of the works inspired by us or that build upon our approach.

- Following the spectral overfitting phenomenon, Marin et al. [58] propose to learn a low dimensional linearly invariant embedding. In essence, it replaces the LB eigen basis typically used in the functional map pipeline with a learned embedding of 3D point clouds. However, similar to our approach, it also advocates learning a low dimensional embedding. We would like to emphasise that this is in sharp contrast to classical functional map framework where working with high dimensional embedding was considered inevitable before this dissertation. Moreover, similar to our approach, in their follow up work [57], they also refine these learned embeddings with ZoomOut.

- One potential shortcoming of our approach is that it is not differentiable end to end. On a high level, a recent work [23] addresses this problem by including our two stage algorithm into an end to end differentiable learning pipeline. Instead of refining it by axiomatic spectral upsampling, Eisenberger et al. [23] propose to upsample the map to a higher resolution with a differentiable and end to end loss.

Weakly Supervised Deep Functional Map

In Chapter 3, we proposed a novel form of weak supervision that can be used to learn shape matching from point clouds. We show that approximate alignment provides an extrinsic feature extractor network enough information to disambiguate symmetry issues. Our extrinsic feature extractor is based on PointNet++. We also propose a simple algorithm for partial shape matching that is differentiable end to end. In the following, we mention couple of works that cite our method.

- DiffusionNet [92] proposes a novel feature learning architecture for surfaces such as 3D triangle meshes and point clouds. DiffusionNet features demonstrates impressive performance across all forms of supervision including our weak supervision.

- NeuroMorph [22] is another unsupervised shape correspondence method that outperforms our method on various datasets. Its high level approach is based on 3D-CODED that simultaneously models deformation and correspondence between two shapes. However, it exploits our weak supervision based on approximate rigid alignment.

- [15] is another work that heavily builds on our weakly supervised structural loss. It proposes a cycle consistency loss in addition to the structural properties of the functional map proposed in our work [81, 89].

Graph Alignment for Matrix Completion

In Chapter 5, we proposed a functional view of graph based machine learning problems such as Geometric Matrix completion. In the following, we mention one work, MAF-GNN [97] that compares and improves upon our result on matrix completion and another [96] that cites our method and proposes a novel solution to the geometric matrix completion problem.

- MAF-GNN [97] develops a novel deep GNN model with multi-graph attention fusion (MAF). It constructs two feature graph attention modules and a multi-scale latent features module, to generate better user and item latent features from input information. Through extensive experiments, the authors show that it improves upon our results on both Synthetic Netflix as well as real world datasets.

- A straightforward extension of graph based matrix completion problem is to apply them to tensor completion where one can exploit higher order correlation among entities. Sofuoglu et al. [96] propose a graph regularized tensor-completion method where the graph regularization is applied across each mode of the tensor to incorporate the local geometry.

Applications in Biology

In this subsection, we mention couple of works that extensively apply Deep Functional Maps for applications in bioinformatics or shape matching on biological data and extensively cite our work.

- μ-Match [43] packages a deep functional map approach [32] and its refinement with ZoomOut into a user-friendly, open-source end-to-end Python pipeline. The authors demonstrate μ-Match's ability on a biologically-relevant benchmark dataset for shape correspondence, and to recover previously reported morphological differences in embryonic limb development.

- The authors of [101] models functional correspondence between bones and propose an approach that characterizes the shape variation within a dataset. This approach classifies biological shapes to the Genus level and demonstrate which aspects of bone shapes differ most between groups.

6.1 Future Work

In this section, we describe a future direction in Deep Functional Maps that has largely been left unexplored until now.

Despite the significant progresses in Deep Functional Maps, little is known about joint optimization of their composite maps. In axiomatic functional map estimation, consistency of maps along cycles serves as a strong regularizer for correcting and improving noisy initial maps computed between pairs of shapes in isolation [34]. Moreover, cycle consistency also allows one to convert the difficult task of computing maps between two

dissimilar objects into an easy task by composing maps along a path of similar object pairs. While some recent work enforce the consistency between a forward and a reverse map between a pair of shapes [81, 28], pairwise consistency by default is not enough to guarantee cycle consistency over a collection of shapes.

Cycle consistency has been used in various forms ranging from regularization to a proxy for self supervision [21] in various computer vision and graphics problem formulations [120]. For instance, early methods attempt to detect and eliminate inconsistent cycles to enforce consistency constraints [116, 66, 114, 28]. Especially, in [66], the authors show that, when the map network is complete (i.e., maps between any two shapes exist), 3-cycle consistency implies cycle consistency along a cycle of arbitrary length. More recent approaches have associated cycle consistency with low-rank properties of matrices that encode the map network [35, 111, 51], which leads to relatively simple formulations. More importantly, this line of work also comes up with a theoretical guarantee on the optimality [35]. On the other hand, the matrix nature of the functional map representation enables a convenient access to map composition, which naturally bridges the functional map framework and consistent map refinement techniques [108, 34, 94].

2-Cyclic Deep Functional Map We remark that two previous works already enforce 2-cycle consistency in Deep Functional Maps. SURFMNet enforces that composition between $\mathbf{C_{12}}$ and $\mathbf{C_{21}}$ to be as closely as possible to $\mathbf{I}$. In contrast, Cyclic Functional Map [28] enforces the same constraint on pairwise maps and not spectral maps. However, both these works do not explore 3-cycle consistency in their formulation which we detail next.

6.1.1 Cycle Consistency in Deep Functional Maps

In this section, we state a simple observation about 3-cycle consistency in deep functional maps. Based on the following observation, we design some baselines and experiments to analyze cycle consistency and its utility in Deep Functional Maps.

Proposition 2 *Deep Functional maps are cycle consistent as long as* $E_{desc}(\mathbf{C})$ *is satisfied exactly.*

Proof: For a shape pair i and j, functional map $\mathbf{C}_{ij}$ is obtained by $\mathbf{C}_{ij} = \arg\min_{\mathbf{C}} \left\|\mathbf{C}\mathbf{A}_i - \mathbf{A}_j\right\|^2$ where $\mathbf{A}_i$ and $\mathbf{A}_j$ are the coefficients of learned descriptors.

We can also write this as $\mathbf{C}_{ij} = \mathbf{A}_j\mathbf{A}_i^+$ where $\mathbf{A}_i^+$ is the pseudo inverse of $\mathbf{A}_i$. Extending this to a triplet of shapes, we obtain:

$$\mathbf{C}_{jk}\mathbf{C}_{ij} = \mathbf{A}_k\mathbf{A}_j^+\mathbf{A}_j\mathbf{A}_i^+ = \mathbf{A}_k\mathbf{A}_i^+ = \mathbf{C}_{ik} \tag{6.1}$$

Eq 6.1 is the definition of cycle consistency for a shape triplet (i, j, k). This suggests that learning-based functional maps are *only* capable of producing consistent maps networks, as long as the descriptor losses are indeed satisfied. It implies that either the networks already produce cycle consistent functional maps or if not, we could simply

add the residual of the linear system $\left\|\mathbf{C}_{ij}\mathbf{A}_i - \mathbf{B}_j\right\|^2$ into the loss and make it satisfy cycle consistency. We test this hypothesis by defining some baselines in next section. We define the normalized residuals by computing them as follows:

$$Res. = \left\|\mathbf{CA} - \mathbf{B}\right\|^2 / \left\|\mathbf{B}\right\|^2 \tag{6.2}$$

Triplet Loss Another way to enforce cycle consistency in Deep Functional Map is by training with a triplet loss. Given a triplet of shapes i,j and k along with their corresponding pairwise functional map, triplet loss is defined as follows:

$$\begin{aligned} E3 = (l_t(\mathbf{C}_{ji}, \mathbf{C}_{kj}, \mathbf{C}_{ki}) + l_t(\mathbf{C}_{jk}, \mathbf{C}_{ij}, \mathbf{C}_{ik}) + l_t(\mathbf{C}_{ki}, \mathbf{C}_{jk}, \mathbf{C}_{ji}) + \\ l_t(\mathbf{C}_{kj}, \mathbf{C}_{ik}, \mathbf{C}_{ij}) + l_t(\mathbf{C}_{ij}, \mathbf{C}_{ki}, \mathbf{C}_{kj}) + l_t(\mathbf{C}_{ik}, \mathbf{C}_{ji}, \mathbf{C}_{jk}))/6 \end{aligned} \tag{6.3}$$

where $l_t(U, V, W) = \left\|UV - W\right\|^2$.

6.1.2 Experiments and Results

In this section, we analyze various baselines performance on a challenging dataset SMAL [121]. The goal of this section is to test the residual hypothesis and also find out the utility of triplet loss in enforcing cycle consistency. SMAL dataset comprises 49 remeshed shapes of different animal groups (big cats, canines, horses, bovine, and hippos), exhibiting significant non-isometries, making regularization potentially more important.

Baselines In the following, we describe the three baseline approaches based on SURFMNet. Note that SURFMNet is unsupervised so it makes sense to add this regularization.

SURFMNet denotes a modified version of SURFMNet architecture [81]. It follows the architecture proposed in [89] that uses only 30 eigenbasis functions as opposed to 120 and 4 layers of non-linear transformation as opposed to 7.

SURFMNet+Tri. This baseline takes SURFMnet and replaces the bijectivity loss with triplet loss as defined above. It quantifies if minimizing triplet loss during training could enforce cycle consistency more strictly during training when compared to the residual loss.

SURFMNet+Tri.+Res. This baseline adds the residual of linear solver into the SURFMNet+Tri.

We next plot the residuals of SURFMNet on SMAL in Figure 6.1. We observe an interesting phenomenon with regards to our Proposition 2 in Figure 6.1. Enforcing cycle consistency with triplet loss increases the residual values whereas explicitly minimizing the residual decreases its value.

To measure the accuracy of the different baselines, we divide SMAL dataset as follows. First, we cluster the dataset into similar looking classes manually. e.g. fox and dogs are merged into one class. As a consequence, we create a cluster of 5 classes namely dogs, cows, lions, hippos and horses each containing 6-19 shapes. The purpose of this division is to facilitate an analysis in an intermediate challenging setting as described next.

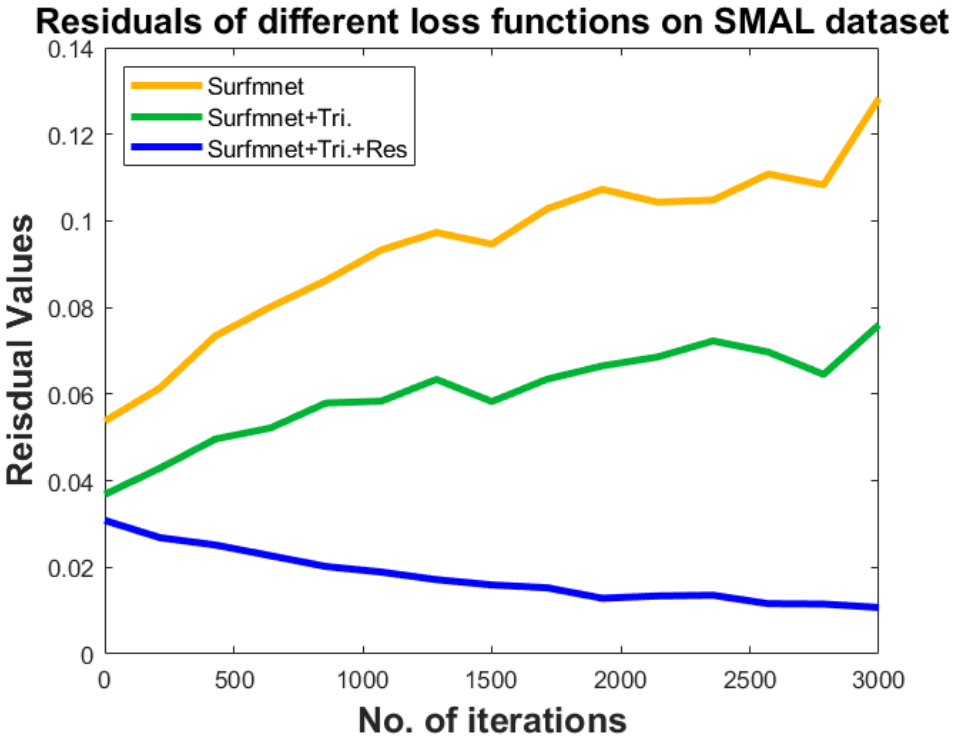

Figure 6.1: The normalized residuals of SURFMNet and triplet loss on SMAL suggest that residuals go higher even after cycle consistency is enforced with Triplet loss in SURFMNet. Nevertheless, interestingly the accuracy of the maps is better with SurfMnet + Tri. as shown in Table 6.1

2-class triplets One can train with all possible triplets but we focus on a special case where a triplet only contains 3 shapes of 2 classes only. We remark that such triplets are slightly less challenging than the triplets where each shape comes from a different class. We find that enforcing cycle consistency in Deep Functional Maps during training gets easier if we work with 2 class triplets instead of 3 classes. We now show the performance of these 3 baselines on 3 different data splits of SMAL. In the first split denoted as S1, we train with 40 train shapes and test on the remaining 9. In the second split S2, we train with 30 shapes whereas we test on the remaining 19. Third split is extreme as we divide the whole dataset almost equally into train and test by 25 and 24 shapes respectively. We show the performance of various baselines on SMAL dataset in Table 6.1.

We notice in Table 6.1 that SURFMNet+Tri. outperforms SURFMNet in low data regime. Moreover, adding residual into the loss function degrades the overall performance. Although in theory zero residual implies zero triplet loss, we observe that in practice the triplet loss is better correlated with the the geodesic error with respect to the ground truth. A more in-depth study is necessary to analyze the exact behavior of these different losses and their correlation to ground truth maps.

Table 6.1: Results on SMAL dataset. Triplet loss enforces cycle consistency and improves SURFMNet results in low data splits S1 and S2.

Method \ Class	S1	S2	S3
SURFMNet	.15	.31	.39
SURFMNet+Tri.	.17	.25	.25
SURFMNet+Tri.+Res.	.20	.27	.29

Bibliography

[1] M. Abadi, A. Agarwal, and P. B. et al., *TensorFlow: Large-scale machine learning on heterogeneous systems*, 2015. Software available from tensorflow.org. (Cited on pages 27 and 56.)

[2] H. Abdi and L. J. Williams, *Principal component analysis*, *Wiley Interdisciplinary Reviews: Computational Statistics, (2010). (Cited on page 59.)*

[3] D. Anguelov, P. Srinivasan, D. Koller, S. Thrun, J. Rodgers, and J. Davis, SCAPE: Shape Completion and Animation of People, *ACM Transactions on Graphics (TOG), 24 (2005), pp. 408–416. (Cited on page 29.)*

[4] S. Arora, N. Cohen, W. Hu, and Y. Luo, Implicit regularization in deep matrix factorization, *in NeurIPS, 2019. (Cited on page 59.)*

[5] M. Belkin and P. Niyogi, Laplacian eigenmaps for dimensionality reduction and data representation, *Neural computation, 15 (2003), pp. 1373–1396. (Cited on page 6.)*

[6] R. v. d. Berg, T. N. Kipf, and M. Welling, Graph convolutional matrix completion, *arXiv preprint arXiv:1706.02263, (2017). (Cited on pages 52 and 59.)*

[7] C. M. Bishop, Pattern Recognition and Machine Learning (Information Science and Statistics), *Springer-Verlag, Berlin, Heidelberg, 2006. (Cited on pages 6 and 18.)*

[8] F. Bogo, J. Romero, M. Loper, and M. 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. (Cited on pages 5, 29 and 40.)*

[9] D. Boscaini, J. Masci, E. Rodola, and M. M. Bronstein, Learning shape correspondence with anisotropic convolutional neural networks, *in Proc. NIPS, 2016, pp. 3189–3197. (Cited on pages 21 and 22.)*

[10] A. Boyarski, S. Vedula, and A. Bronstein, Deep matrix factorization with spectral geometric regularization, *arXiv preprint arXiv:1911.07255, (2020). (Cited on pages 52, 53, 54, 56, 58 and 59.)*

[11] O. Burghard, A. Dieckmann, and R. Klein, Embedding shapes with Green's functions for global shape matching, *Computers & Graphics, 68 (2017), pp. 1–10. (Cited on page 22.)*

[12] D. Cai, X. He, J. Han, and T. Huang, Graph regularized nonnegative matrix factorization for data representation, *IEEE transactions on pattern analysis and machine intelligence, 33 8 (2011), pp. 1548–60. (Cited on page 61.)*

[13] E. Candès, X. Li, Y. Ma, and J. Wright, Robust principal component analysis?, *J. ACM, 58 (2011), pp. 11:1–11:37. (Cited on pages 59 and 60.)*

[14] E. J. Candès and B. Recht, Exact matrix completion via convex optimization, *Foundations of Computational mathematics, 9 (2009), p. 717. (Cited on pages 51, 55 and 59.)*

[15] D. Cao and F. Bernard, Unsupervised deep multi-shape matching, *in ECCV, 2022. (Cited on page 64.)*

[16] S. Chatterjee, Matrix estimation by universal singular value thresholding, *The Annals of Statistics, 43 (2012). (Cited on page 53.)*

[17] P. Chavdar, H. Sami, P. Sven, K. Kai, and B. Darius, Rigid 3d geometry matching for grasping of known objects in cluttered scenes, *IJRR, 31 (2012). (Cited on page 22.)*

[18] L. Cosmo, E. Rodola, J. Masci, A. Torsello, and M. M. Bronstein, Matching deformable objects in clutter, *in 3D Vision (3DV), 2016 Fourth International Conference on, IEEE, 2016, pp. 1–10. (Cited on pages 25, 33, 40 and 42.)*

[19] N. Donati, A. Sharma, and M. Ovsjanikov, Deep geometric functional maps: Robust feature learning for shape correspondence, *in CVPR, 2020. (Cited on pages 7, 8, 19, 21, 22, 23, 25, 29, 30, 31, 32, 40, 41 and 82.)*

[20] D. Donoho and M. Gavish, The optimal hard threshold for singular values is, *IEEE Transactions on Information Theory, 60 (2013). (Cited on page 53.)*

[21] D. Dwibedi, Y. Aytar, J. Tompson, P. Sermanet, and A. Zisserman, Temporal cycle-consistency learning, *in The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2019. (Cited on page 66.)*

[22] M. Eisenberger, D. Novotny, G. Kerchenbaum, P. Labatut, N. Neverova, D. Cremers, and A. Vedaldi, Neuromorph: Unsupervised shape interpolation and correspondence in one go, *in CVPR, 2021, pp. 7473–7483. (Cited on pages 36 and 64.)*

[23] M. Eisenberger, A. Toker, L. Leal-Taixé, and D. Cremers, Deep shells: Unsupervised shape correspondence with optimal transport, *in NeurIPS, vol. 34, 2020. (Cited on pages 35, 40, 41 and 64.)*

[24] E. Elhamifar and R. Vidal, Sparse subspace clustering: Algorithm, theory, and applications, *IEEE Transactions on Pattern Analysis and Machine Intelligence, 35 (2013), pp. 2765–2781. (Cited on page 53.)*

[25] D. Eynard, E. Rodola, K. Glashoff, and M. M. Bronstein, Coupled functional maps, *in 3D Vision (3DV), IEEE, 2016, pp. 399–407. (Cited on page 26.)*

[26] D. Ezuz and M. Ben-Chen, Deblurring and denoising of maps between shapes, *Computer Graphics Forum, 36 (2017), pp. 165–174. (Cited on page 12.)*

[27] C. Fernandez-Labrador, A. Chhatkuli, D. P. Paudel, J. J. Guerrero, C. Demonceaux, and L. Van Gool, Unsupervised learning of category-specific symmetric 3d keypoints from point sets, *in ECCV, 2020. (Cited on page 35.)*

[28] D. Ginzburg and D. Raviv, Cyclic functional mapping: Self-supervised correspondence between non-isometric deformable shapes, *in ECCV, 2020. (Cited on page 66.)*

[29] T. Groueix, M. Fisher, V. Kim, B. Russell, and M. Aubry, Unsupervised cycle-consistent deformation for shape matching, *in Computer Graphics Forum, 2019. (Cited on page 21.)*

[30] T. Groueix, M. Fisher, V. G. Kim, B. C. Russell, and M. Aubry, 3d-coded: 3d correspondences by deep deformation, *in Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 230–246. (Cited on pages 21, 23, 30, 31, 35, 40 and 41.)*

[31] Q. Gu and J. Zhou, Co-clustering on manifolds, *in KDD, 2009. (Cited on page 52.)*

[32] O. Halimi, O. Litany, E. Rodol'a, A. Bronstein, and R. Kimmel, Unsupervised learning of dense shape correspondence, *in CVPR, 2019. (Cited on pages 11, 21, 22, 23, 30, 32, 33, 35, 36 and 65.)*

[33] F. M. Harper and J. A. Konstan, The movielens datasets: History and context, *Acm transactions on interactive intelligent systems (tiis), 5 (2016), p. 19. (Cited on page 59.)*

[34] Q. Huang, F. Wang, and L. Guibas, Functional map networks for analyzing and exploring large shape collections, *TOG, 33 (2014). (Cited on pages 22, 65 and 66.)*

[35] Q.-X. Huang and L. Guibas, Consistent shape maps via semidefinite programming, *in Computer Graphics Forum, 2013. (Cited on page 66.)*

[36] M. Jamali and M. Ester, A matrix factorization technique with trust propagation for recommendation in social networks, *in Proceedings of the fourth ACM conference on Recommender systems, ACM, 2010, pp. 135–142. (Cited on page 59.)*

[37] B. Jiang, C. Ding, B. Luo, and J. Tang, Graph-laplacian pca: Closed-form solution and robustness, *in CVPR, 2013, pp. 3492–3498. (Cited on pages 51, 52, 60 and 61.)*

[38] T. Jin, J. Yu, J. You, K. Zeng, C. Li, and Z. Yu, Low-rank matrix factorization with multiple hypergraph regularizer, *Pattern Recognition., 48 (2015), pp. 1011–1022. (Cited on pages 52 and 60.)*

[39] J. M. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Zídek, A. Potapenko, A. Bridgland, C. Meyer, S. A. A. Kohl, A. Ballard, A. Cowie, B. Romera-Paredes, S. Nikolov, R. Jain, J. Adler, T. Back, S. Petersen, D. A. Reiman, E. Clancy, M. Zielinski, M. Steinegger, M. Pacholska, T. Berghammer, S. Bodenstein, D. Silver, O. Vinyals, A. W. Senior, K. Kavukcuoglu, P. Kohli, and D. Hassabis, Highly accurate protein structure prediction with alphafold, *Nature, 596 (2021), pp. 583 – 589. (Cited on page 5.)*

[40] V. Kalofolias, X. Bresson, M. Bronstein, and P. Vandergheynst, Matrix completion on graphs, *arXiv preprint arXiv:1408.1717, (2014). (Cited on pages 51, 52, 56, 57 and 59.)*

[41] V. Kim, Y. Lipman, and T. Funkhouser, Blended intrinsic maps, *ACM Transactions on Graphics (Proc. SIGGRAPH), 30 (2011). (Cited on page 33.)*

[42] M. Kirgo, S. Melzi, G. Patanè, E. Rodolà, and M. Ovsjanikov, Wavelet-based heat kernel derivatives: Towards informative localized shape analysis, *in Computer Graphics Forum, Wiley Online Library, 2020. (Cited on page 36.)*

[43] J. Klatzow, G. Dalmasso, N. Martínez-Abadías, J. Sharpe, and V. Uhlmann, μmatch: 3d shape correspondence for biological image data, *Frontiers in Computer Science, 4 (2022). (Cited on page 65.)*

[44] O. Klopp, Rank penalized estimators for high-dimensional matrices, *Electronic Journal of Statistics, 5 (2011), pp. 1161–1183. (Cited on page 53.)*

[45] O. Klopp, Matrix completion by singular value thresholding: sharp bounds, *Electronic Journal of Statistics, 9 (2015), pp. 2348–2369. (Cited on page 53.)*

[46] I. Kokkinos, Ubernet: Training universal cnn for low mid and high level vision with diverse datasets and limited memory, *in CVPR, 2017. (Cited on page 36.)*

[47] V. Koltchinskii, K. Lounici, and A. B. Tsybakov, Nuclear-norm penalization and optimal rates for noisy low-rank matrix completion, *Annals of Statistics, 39 (2011), pp. 2302–2329. (Cited on page 53.)*

[48] A. Kovnatsky, M. M. Bronstein, A. Bronstein, and R. Kimmel, Coupled quasi-harmonic bases, *Computer Graphics Forum, 32 (2013), pp. 439–448. (Cited on page 22.)*

[49] A. Kovnatsky, K. Glashoff, and M. Bronstein, Madmm: a generic algorithm for non-smooth optimization on manifolds, *in ECCV, Springer, 2016, pp. 680–696. (Cited on page 12.)*

[50] D. Lee and H. S. Seung, Learning the parts of objects by non-negative matrix factorization, *Nature, 401 (1999), pp. 788–791. (Cited on page 61.)*

[51] S. Leonardos, X. Zhou, and K. Daniilidis, Distributed consistent data association via permutation synchronization, *in 2017 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2017, pp. 2645–2652. (Cited on page 66.)*

[52] Y. Lipman, X. Chen, I. Daubechies, and T. Funkhouser, Symmetry factored embedding and distance, *in SIGGRAPH, 2010. (Cited on page 35.)*

[53] O. Litany, T. Remez, E. Rodolà, A. M. Bronstein, and M. M. Bronstein, Deep functional maps: Structured prediction for dense shape correspondence, *2017 IEEE International Conference on Computer Vision (ICCV), (2017), pp. 5660–5668. (Cited on pages 6, 11, 12, 13, 17, 21, 22, 23, 26, 30, 35, 37, 47 and 63.)*

[54] O. Litany, E. Rodolà, A. M. Bronstein, and M. M. Bronstein, Fully spectral partial shape matching, *36, (2017), pp. 247–258. (Cited on pages 6, 22, 23, 27, 34, 35, 42 and 43.)*

[55] W. Liu, J. Wang, S. Kumar, and S. Chang, Hashing with graphs, *in ICML, 2011. (Cited on page 51.)*

[56] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black, Smpl: A skinned multi-person linear model, *ACM Trans. Graph., 34 (2015), pp. 248:1–248:16. (Cited on page 29.)*

[57] R. Marin, S. Attaiki, S. Melzi, E. Rodolà, and M. Ovsjanikov, Why you should learn functional basis, *ArXiv, abs/2112.07289 (2021). (Cited on page 64.)*

[58] R. Marin, M.-J. Rakotosaona, S. Melzi, and M. Ovsjanikov, Correspondence learning via linearly-invariant embedding, *in NeurIPS, vol. 33, 2020, pp. 1608–1620. (Cited on pages 35, 36, 37, 39, 40, 41, 42, 43, 46, 47, 48, 49 and 64.)*

[59] J. Masci, D. Boscaini, M. Bronstein, and P. Vandergheynst, Geodesic convolutional neural networks on riemannian manifolds, *in Proceedings of the IEEE international conference on computer vision workshops, 2015, pp. 37–45. (Cited on page 21.)*

[60] S. Melzi, J. Ren, E. Rodola, M. Ovsjanikov, and P. Wonka, Zoomout: Spectral upsampling for efficient shape correspondence, *ACM Transactions on Graphics (Proc. SIGGRAPH Asia), (2019). (Cited on pages 7, 8, 11, 17, 19, 28, 30, 40, 56 and 82.)*

[61] N. J. Mitra, M. Pauly, M. Wand, and D. Ceylan, Symmetry in 3d geometry: Extraction and applications, *in EUROGRAPHICS State-of-the-art Report, 2012. (Cited on page 35.)*

[62] F. Monti, D. Boscaini, J. Masci, E. Rodolà, J. Svoboda, and M. M. Bronstein, Geometric deep learning on graphs and manifolds using mixture

model cnns, *in CVPR, IEEE Computer Society, 2017, pp. 5425–5434. (Cited on pages 8 and 22.)*

[63] F. Monti, M. Bronstein, and X. Bresson, Geometric matrix completion with recurrent multi-graph neural networks, *in Advances in Neural Information Processing Systems, 2017, pp. 3697–3707. (Cited on pages 52, 56 and 59.)*

[64] R. Nagar and S. Raman, Fast and accurate intrinsic symmetry detection, *in ECCV, 2018. (Cited on pages 35 and 45.)*

[65] N. Neverova, D. Novotny, V. Khalidov, M. Szafraniec, P. Labatut, and A. Vedaldi, Continuous surface embeddings, *in NeurIPS, 2020. (Cited on page 5.)*

[66] A. Nguyen, M. Ben-Chen, K. Welnicka, Y. Ye, and L. Guibas, An optimization approach to improving collections of shape maps, *in Computer Graphics Forum, 2011. (Cited on page 66.)*

[67] D. Nogneng and M. Ovsjanikov, Informative descriptor preservation via commutativity for shape matching, *Computer Graphics Forum, 36 (2017), pp. 259–267. (Cited on pages 12, 14 and 22.)*

[68] M. Ovsjanikov, M. Ben-Chen, J. Solomon, A. Butscher, and L. Guibas, Functional Maps: A Flexible Representation of Maps Between Shapes, *ACM Transactions on Graphics (TOG), 31 (2012), p. 30. (Cited on pages 5, 6, 14, 21, 26, 38, 52, 53 and 55.)*

[69] M. Ovsjanikov, Q. Mérigot, V. Pătrăucean, and L. Guibas, Shape matching via quotient spaces, *in SGP, 2013, p. 1–11. (Cited on page 35.)*

[70] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, 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, *in NeurIPS, 2019, pp. 8024–8035. (Cited on page 39.)*

[71] K. B. Petersen and M. S. Pedersen, The Matrix Cookbook, *Technical University of Denmark, 2012. (Cited on page 16.)*

[72] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, Pointnet: Deep learning on point sets for 3d classification and segmentation, *in Proc. CVPR, 2017, pp. 652–660. (Cited on pages 23, 37 and 38.)*

[73] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, Pointnet++: Deep hierarchical feature learning on point sets in a metric space, *in NeurIPS, 2017, pp. 5105–5114. (Cited on pages 23 and 25.)*

[74] N. Rao, H.-F. Yu, P. K. Ravikumar, and I. S. Dhillon, Collaborative filtering with graph information: Consistency and scalable methods, *in Advances in neural information processing systems, 2015, pp. 2107–2115. (Cited on pages 51, 52 and 59.)*

[75] D. Raviv, A. Bronstein, M. Bronstein, and R. Kimmel, Full and partial symmetries of non-rigid shapes, *International Journal of Computer Vision, 89 (2010), pp. 18–39. (Cited on page 35.)*

[76] J. Ren, S. Melzi, M. Ovsjanikov, and P. Wonka, Maptree: Recovering multiple solutions in the space of maps, *ACM Trans. Graph., 39 (2020). (Cited on pages 35 and 45.)*

[77] J. Ren, A. Poulenard, P. Wonka, and M. Ovsjanikov, Continuous and orientation-preserving correspondences via functional maps, *ACM Transactions on Graphics (TOG), 37 (2018). (Cited on pages 22, 29 and 40.)*

[78] E. Rodolà, L. Cosmo, M. M. Bronstein, A. Torsello, and D. Cremers, Partial functional correspondence, *Computer Graphics Forum, 36 (2017), pp. 222–236. (Cited on pages 22, 23 and 27.)*

[79] E. Rodolà, M. Moeller, and D. Cremers, Point-wise map recovery and refinement from functional correspondence, *in Proc. Vision, Modeling and Visualization (VMV), 2015. (Cited on page 12.)*

[80] S. Rosenberg, The Laplacian on a Riemannian manifold: an introduction to analysis on manifolds, *vol. 31, Cambridge University Press, 1997. (Cited on pages 14 and 26.)*

[81] J.-M. Roufosse, A. Sharma, and M. Ovsjanikov, Unsupervised deep learning for structured shape matching, *in ICCV, 2019, pp. 1617–1627. (Cited on pages 7, 8, 11, 12, 13, 15, 16, 17, 19, 21, 22, 23, 26, 30, 35, 64, 66, 67 and 82.)*

[82] R. Rustamov, M. Ovsjanikov, O. Azencot, M. Ben-Chen, F. Chazal, and L. Guibas, Map-based exploration of intrinsic shape differences and variability, *ACM Trans. Graphics, 32 (2013), pp. 72:1–72:12. (Cited on pages 14 and 26.)*

[83] Y. Sahillioğlu, Recent advances in shape correspondence, *The Visual Computer, (2019), pp. 1–17. (Cited on page 22.)*

[84] N. Shahid, V. Kalofolias, X. Bresson, M. Bronstein, and P. Vandergheynst, Robust principal component analysis on graphs, *in ICCV, 2015, pp. 2812–2820. (Cited on pages 8, 52 and 61.)*

[85] N. Shahid, N. Perraudin, V. Kalofolias, G. Puy, and P. Vandergheynst, Fast robust pca on graphs, *IEEE Journal of Selected Topics in Signal Processing, 10 (2016), pp. 740–756. (Cited on pages 60 and 61.)*

[86] F. Shang, L. Jiao, and F. Wang, Graph dual regularization non-negative matrix factorization for co-clustering, *Pattern Recognition., 45 (2012), pp. 2237–2250. (Cited on page 52.)*

[87] A. Sharma, Foreground clustering for joint segmentation and localization in videos and images, *in NeurIPS, vol. 31, 2018. (Cited on page 36.)*

[88] A. Sharma, O. Grau, and M. Fritz, Vconv-dae: Deep volumetric shape learning without object labels, *in ECCV, 2016. (Cited on page 22.)*

[89] A. Sharma and M. Ovsjanikov, Weakly supervised deep functional maps for shape matching, *in NeurIPS, vol. 33, 2020. (Cited on pages 7, 8, 9, 11, 35, 36, 39, 40, 41, 42, 43, 64, 67 and 82.)*

[90] A. Sharma and M. Ovsjanikov, Joint symmetry detection and shape matching for non-rigid point cloud, under review, *in 3DV, 2022. (Cited on pages 7, 8 and 82.)*

[91] A. Sharma and M. Ovsjanikov, Matrix decomposition on graphs: A simplified functional view, *in ICASSP, 2022. (Cited on pages 5, 7, 9 and 82.)*

[92] N. Sharp, S. Attaiki, K. Crane, and M. Ovsjanikov, Diffusionnet: Discretization agnostic learning on surfaces, *ACM Transactions on Graphics, (2022). (Cited on page 64.)*

[93] Y. Shi, J. Huang, H. Zhang, X. Xu, S. Rusinkiewicz, and K. Xu, Symmetrynet: Learning to predict reflectional and rotational symmetries of 3D shapes from single-view RGB-D images, *ACM Transactions on Graphics (Proc. SIGGRAPH Asia), 39 (2020). (Cited on page 35.)*

[94] M. Shoham, A. Vaxman, and M. Ben-Chen, Hierarchical functional maps between subdivision surfaces, *in Computer Graphics Forum, 2019. (Cited on page 66.)*

[95] R. K. Singh and J. S. Manhas, Composition Operators on Function Spaces, *ELSEVIER, 1993. (Cited on page 14.)*

[96] S. E. Sofuoglu and S. Aviyente, Graph regularized low-rank tensor-train for robust principal component analysis, *IEEE Signal Processing Letters, (2022), pp. 1–1. (Cited on page 65.)*

[97] Y. Song, H. Ye, M. Li, and F. Cao, Deep multi-graph neural networks with attention fusion for recommendation, *Expert Systems with Applications, 191 (2022), pp. 116–240. (Cited on page 65.)*

[98] D. Spielman, Spectral graph theory, *Lecture Notes, Yale University, (2009), pp. 740–0776. (Cited on page 53.)*

[99] J. Sun, M. Ovsjanikov, and L. Guibas, A Concise and Provably Informative Multi-Scale Signature Based on Heat Diffusion, *Computer Graphics Forum, 28 (2009), pp. 1383–1392. (Cited on page 12.)*

[100] L. Tao, H. Ip, Y. Wang, and X. Shu, Low rank approximation with sparse integration of multiple manifolds for data representation, *Applied Intelligence, 42 (2014), pp. 430–446. (Cited on pages 52 and 60.)*

[101] O. O. Thomas, H. Shen, R. L. Raaum, W. E. Harcourt-Smith, J. D. Polk, and M. Hasegawa-Johnson, Automated morphological phenotyping using learned shape descriptors and functional maps: A novel approach to geometric morphometrics, *bioRxiv, (2021). (Cited on page 65.)*

[102] F. Tombari, S. Salti, and L. Di Stefano, Unique signatures of histograms for local surface description, *in International Conference on Computer Vision (ICCV), 2010, pp. 356–369. (Cited on pages 6, 23, 35 and 40.)*

[103] R. J. L. Townshend, R. Bedi, P. Suriana, and R. O. Dror, End-to-end learning on 3d protein structure for interface prediction, *in NeurIPS, 2019. (Cited on page 5.)*

[104] O. Van Kaick, H. Zhang, G. Hamarneh, and D. Cohen-Or, A survey on shape correspondence, *Computer Graphics Forum, 30 (2011), pp. 1681–1707. (Cited on pages 5 and 21.)*

[105] G. Varol, J. Romero, X. Martin, N. Mahmood, M. J. Black, I. Laptev, and C. Schmid, Learning from synthetic humans, *in CVPR, 2017. (Cited on page 29.)*

[106] R. Vidal and P. Favaro, Low rank subspace clustering (lrsc), *Pattern Recognit. Lett., 43 (2014), pp. 47–61. (Cited on page 53.)*

[107] C. Wang, M. M. Bronstein, A. M. Bronstein, and N. Paragios, Discrete minimum distortion correspondence problems for non-rigid shape matching, *in International Conference on Scale Space and Variational Methods in Computer Vision (SSVM), 2011. (Cited on page 5.)*

[108] F. Wang, Q. Huang, and L. J. Guibas, Image co-segmentation via consistent functional maps, *in CVPR, 2013, pp. 849–856. (Cited on page 66.)*

[109] F. Wang, G.-S. Xia, N. Xue, Y. Zhang, and M. Pelillo, A functional representation for graph matching, *IEEE Transactions on Pattern Analysis and Machine Intelligence, 42 (2020), pp. 2737–2754. (Cited on page 5.)*

[110] L. Wang, A. Gehre, M. M. Bronstein, and J. Solomon, Kernel functional maps, *Computer Graphics Forum, 37 (2018), pp. 27–36. (Cited on page 12.)*

[111] L. Wang and A. Singer, Exact and stable recovery of rotations for robust synchronization, *Information and Inference: A Journal of the IMA, 2 (2013), pp. 145–193. (Cited on page 66.)*

[112] X. Wang, A. Jabri, and A. A. Efros, Learning correspondence from the cycle-consistency of time, *in CVPR, 2019. (Cited on page 5.)*

[113] L. Wei, Q. Huang, D. Ceylan, E. Vouga, and H. Li, Dense human body correspondences using convolutional networks, *in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 1544–1553. (Cited on pages 21 and 22.)*

[114] L. Yang, W. Liu, Z. Cui, N. Chen, and W. Wang, Mapping in a cycle: Sinkhorn regularized unsupervised learning for point cloud shapes, *in ECCV, 2020. (Cited on page 66.)*

[115] M. Yin, J. Gao, Z. Lin, Q. Shi, and Y. Guo, Dual graph regularized latent low-rank representation for subspace clustering, *IEEE Transactions on Image Processing, 24 (2015), pp. 4918–4933. (Cited on page 52.)*

[116] C. Zach, M. Klopschitz, and M. Pollefeys, Disambiguating visual relations using loop constraints, *in 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, IEEE, 2010, pp. 1426–1433. (Cited on page 66.)*

[117] Y. Zeng, C. Wang, X. Gu, D. Samaras, and N. Paragios, Higher-order graph principles towards non-rigid surface registration, *IEEE Transactions on Pattern Analysis and Machine Intelligence, 38 (2016), pp. 2416–2429. (Cited on page 5.)*

[118] Y. Zeng, C. Wang, Y. Wang, X. Gu, D. Samaras, and N. Paragios, Dense non-rigid surface registration using high-order graph matching, *in CVPR, 2010, pp. 382–389. (Cited on page 5.)*

[119] Z. Zhang and K. Zhao, Low-rank matrix approximation with manifold regularization, *PAMI, 35 (2013), pp. 1717–1729. (Cited on pages 52, 60 and 61.)*

[120] T. Zhou, P. Krähenbühl, M. Aubry, Q. Huang, and A. A. Efros, Learning dense correspondence via 3d-guided cycle consistency, *in CVPR, 2016. (Cited on page 66.)*

[121] S. Zuffi, A. Kanazawa, D. Jacobs, and M. J. Black, 3D menagerie: Modeling the 3D shape and pose of animals, *in CVPR, 2017. (Cited on page 67.)*

Résumé

Étant donné une paire d'objets contenant respectivement m et n points, trouver la correspondance consiste à trouver une carte bijective (si possible) entre ces points. Dans le cas le plus simple, ces problèmes sont formulés comme un problème d'étiquetage, où différents des points correspondent à des étiquettes à prédire. Cela mène à un espace d'étiquette extrêmement grand qui évolue linéairement avec le nombre de points et donc,nécessite beaucoup de données pour apprendre cet espace. En revanche, les auteurs dans [69] ont introduit une vue fonctionnelle dans les problèmes de correspondance visuelle en considérant les formes ou les images comme espaces fonctionnels. Cette approche de carte fonctionnelle aligne les fonctions de descripteur d'un objet visuel à un autre et vise ainsi à déduire une carte globale entière entre une paire de formes plutôt que d'aligner chaque point indépendamment.

Carte fonctionnelle approfondie [54] introduit l'apprentissage dans l'approche de la carte fonctionnelle en apprenant une transformation non linéaire des fonctions descripteurs à aligner ultérieurement. Deux entrées principales d'un cadre de carte fonctionnelle typique sont le descripteur correspondant fonctions et fonctions de base définies sur deux objets. Avant cette thèse, SHOT[104] étaient les plus couramment utilisées comme fonctions de description et base propre laplacienne[5] comme fonctions de base. Les fonctions de base sont nécessaires pour effectuer la réduction de dimensionnalité en projetant les descripteurs sur un sous-espace couvrant de fonctions de base. Enfin, un résout un problème d'optimisation, en recherchant une matrice qui aligne au mieux les caractéristiques projetées. L'un des principaux avantages de ce cadre est qu'il nous permet de représenter des cartes entre formes sous forme de petites matrices, qui encodent les relations entre les fonctions de base définies sur le formes. Par conséquent, la fonction objectif est indépendante du nombre de points sur la forme et peut être optimisé avec des solveurs linéaires simples tels que les moindres carrés.

Malgré sa simplicité et son évolutivité, ce cadre est sous-optimal à bien des égards. La première source de difficulté réside dans l'estimation précise d'une carte fonctionnelle de grande dimension. Cela reste un défi à la fois pour le pipeline de cartes fonctionnelles axiomatiques et pour cartes fonctionnelles. L'apprentissage en haute dimension est connu pour être difficile en apprentissage automatique [7]. D'autre part, une dimensionnalité réduite se traduit par des cartes très approximatives, perdant détails à moyenne et haute fréquence et conduisant à des artefacts importants dans les applications. Avant les travaux de cette thèse, l'état de l'art [55] reposait sur l'intégration directe des fonctions scalaires dans une base propre de Laplace Beltrami de grande dimension.

Dans la thèse qui suit, nous proposons trois contributions, chacune correspondant à ces problèmes. Nous marions les idées de l'apprentissage en profondeur avec le cadre de la carte fonctionnelle résultant ainsi en des algorithmes très efficaces et robustes pour

l'appariement de formes basés sur cartes fonctionnelles profondes. De plus, nous étendons la notion d'applications fonctionnelles aux graphes et proposer une vue fonctionnelle du problème de complétion matricielle basé sur les graphes.

Liste des publications

- J.M. Roufosse, A. Sharma, M. Ovsjanikov, Apprentissage en profondeur non supervisé pour la correspondance de forme structurée [81] ICCV'2019.
- S. Melzi, J. Ren, E. Rodola, A. Sharma, P. Wonka, M. Ovsjanikov, ZoomOut : suréchantillonnage spectral pour une correspondance de forme efficace [60], Siggraph Asia'2019
- N. Donati, A. Sharma, M. Ovsjanikov, Cartes fonctionnelles géométriques profondes : apprentissage robuste des caractéristiques pour la correspondance des formes [19], CVPR'2020
- A. Sharma, M. Ovsjanikov, Cartes fonctionnelles profondes faiblement supervisées pour la correspondance de forme [89], NeurIPS'2020
- A. Sharma, M. Ovsjanikov, Décomposition matricielle sur les graphes : une vue fonctionnelle simplifiée [91], ICASSP'2022
- A. Sharma, M. Ovsjanikov, Détection de symétrie conjointe et correspondance de forme pour un nuage de points non rigide [90]

ECOLE
DOCTORALE

Titre : Cartes fonctionnelles profondes efficaces en données pour l'analyse de formes et de graphiques 3D

Mots clés : cartes fonctionnelles ; correspondance de forme ; surveillance faible ; complétion matricielle ; alignement graphique

Résumé : La correspondance de forme est un problème fondamental en vision par ordinateur, en infographie et dans les domaines connexes, car elle facilite de nombreuses applications telles que le transfert de texture ou de déformation et l'analyse statistique de forme, pour n'en nommer que quelques-unes. Bien que la correspondance de forme ait été étudiée sous de nombreux points de vue, dans cette thèse, nous nous concentrons sur les approches fonctionnelles basées sur des cartes, car ce cadre est assez général, évolutif et a donc été étendu à diverses autres applications telles que l'estimation de pose, la complétion de matrices et le graphe. correspondant à. Dans cette thèse, nous proposons trois contributions aux cartes fonctionnelles profondes : Premièrement, nous proposons une méthode simple mais efficace pour estimer une carte fonctionnelle de grande dimension. Notre méthode est basée sur l'apprentissage d'une carte fonctionnelle de faible dimension, puis sur son raffinement en une carte de dimension supérieure basée sur un suréchantillonnage spectral itératif. Deuxièmement, nous proposons une nouvelle direction qui préconise l'utilisation d'un alignement rigide approximatif des formes comme signal de supervision faible. Notre hypothèse principale est que l'alignement rigide approximatif fournit au réseau suffisamment d'informations pour lever l'ambiguïté des problèmes de symétrie. Bien que l'alignement approximatif soit plus facile à obtenir que la vérité terrain ponctuelle entre une paire de formes, il souffre toujours de problèmes d'évolutivité sur les collections de formes 3D à grande échelle. Ainsi, nous allons au-delà de ce prérequis et considérons le problème de l'apprentissage simultané d'une auto-symétrie et d'une application par paires. Notre troisième contribution est une nouvelle régularisation commutative qui couple la carte d'autosymétrie avec une carte par paires et permet ainsi le transfert de connaissances entre les deux cartes pendant l'apprentissage. Notre dernière contribution est une application du cadre de la carte fonctionnelle à certains problèmes d'apprentissage automatique basés sur des graphes. Nous proposons un cadre simplifié basé sur une idée clé selon laquelle l'utilisation d'une base réduite pour représenter des fonctions sur l'espace produit est suffisante pour récupérer une approximation matricielle de rang bas même à partir d'un signal clairsemé.

Title : Data-Efficient Deep Functional Maps for 3D Shape and Graph Analysis

Keywords : Deep functional maps ; Shape Matching ; Weak supervision ; Matrix completion ; Graph alignment

Abstract : Shape correspondence is a fundamental problem in computer vision, computer graphics, and related fields since it facilitates many applications such as texture or deformation transfer and statistical shape analysis to name a few. Although shape correspondence has been studied from many viewpoints, in this thesis, we focus on functional map-based approaches as this framework is quite general, scalable and thus, has been extended to various other applications such as pose estimation, matrix completion, and graph matching. In this thesis, we propose three contributions to deep functional maps : First, we propose a simple but effective method to estimate a high-dimensional functional map. Our method is based on first learning a low dimensional functional map and then refining it to a higher dimensional one based on iterative spectral upsampling. Second, we propose a new direction that advocates the use of approximate rigid alignment of shapes as a weak supervision signal. Our main hypothesis is that the approximate rigid alignment provides the network with enough information to disambiguate symmetry issues. Although approximate alignment is easier to obtain than pointwise ground truth between a pair of shapes, it still suffers from scalability issues on large-scale 3D shape collections. Thus, we go beyond this prerequisite and consider the problem of learning simultaneously a self symmetry map and a pairwise map. Our third contribution is a novel commutative regularization that couples the self-symmetry map with a pairwise map and thus enable knowledge transfer between the two maps during training. Our last contribution is an application of the functional map framework to some graph-based machine learning problems. We propose a simplified framework that is based on a key idea that using a reduced basis to represent functions on the product space is sufficient to recover a low-rank matrix approximation even from a sparse signal.

Institut Polytechnique de Paris
91120 Palaiseau, France