



**HAL**  
open science

# Symmetry-aware Learning for Non-rigid Shape Matching

Abhishek Sharma, Maks Ovsjanikov

► **To cite this version:**

Abhishek Sharma, Maks Ovsjanikov. Symmetry-aware Learning for Non-rigid Shape Matching. 2021.  
hal-03251317v1

**HAL Id: hal-03251317**

**<https://hal.science/hal-03251317v1>**

Preprint submitted on 7 Jun 2021 (v1), last revised 6 Oct 2021 (v2)

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

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

---

# Symmetry-aware Learning for Non-rigid Shape Matching

---

**Abhishek Sharma**  
LIX, École Polytechnique  
kein.iitian@gmail.com

**Maks Ovsjanikov**  
LIX, École Polytechnique  
maks@lix.polytechnique.fr

## Abstract

This paper provides a novel framework that learns self symmetric functional maps with an end goal of exploiting symmetry-aware representation as an embedding for non-rigid shape matching. In contrast to prior work in this direction, our framework is trained end-to-end and thus avoids instabilities and constraints associated with the commonly-used Laplace-Beltrami basis. On multiple datasets, we demonstrate that learning self symmetry maps with a deep functional map projects 3D shapes into a low dimensional canonical embedding that facilitates non rigid shape correspondence between unseen shape categories via a simple nearest neighbor search. Our framework outperforms all recent learning based methods on two partial shape matching SHREC benchmarks while being computationally cheaper.

## 1 Introduction

Shape correspondence is a fundamental problem in computer vision, computer graphics and related fields, Thomas et al. [2021] since it facilitates many applications such as texture or deformation transfer and statistical shape analysis Bogoy et al. [2014] to name a few. Although shape correspondence has been studied from many viewpoints, we focus here on a functional map-based approaches Ovsjanikov et al. [2012] as this framework is quite general, scalable and thus, has been extended to various other applications such as pose estimation Neverova et al. [2020], matrix completion Sharma and Ovsjanikov [2021] and graph matching Wang et al. [2020].

While recent learning based deep functional map approaches have made impressive gains in non rigid isometric full shape matching, partial shape matching Rodolà et al. [2017], Litany et al. [2017b] has received little attention despite it being of great interest in robotics Chavdar et al. [2012] and Virtual reality applications Sharma et al. [2016]. The progress is mainly hindered by the difficulty of learning a suitable embedding or basis functions for partial 3D data. The majority of works in this domain use the Laplace-Beltrami basis, Ovsjanikov et al. [2017] which are biased towards near-isometries and can be unstable under significant partiality Kirgo et al. [2020]. Given a source and target shape as input, recent approaches either learn an alignment of their Laplacian eigen-functions Litany et al. [2017b], Roufousse et al. [2019], Halimi et al. [2019], Sharma and Ovsjanikov [2020] or learn a linearly invariant embedding directly from 3D data Marin et al. [2020]. While former approach is class specific and requires retraining a network for each class, the later employs a two stage optimization strategy without adequate regularization which, as we show later, is suboptimal.

Recently, it has been shown that learning an embedding to be used as a basis in the functional map framework, rather than relying on the Laplace-Beltrami eigenfunctions, can lead to improvement in robustness. Nevertheless, as we demonstrate below, learning a linearly invariant embedding for 3D shapes with a PointNet based feature extractor, as done in Marin et al. [2020], without exploiting natural priors on 3D shapes such as symmetry leads to overfitting as there is no regularization involved. Similarly, learning a linear transformation for a pair of shapes Marin et al. [2020], without any constraint on the linear transform itself can leave the network with a lot of degrees of freedom to

choose this transform during training which is harder to retrieve at test time. Our work builds on the idea that a functional map can be decomposed into symmetric and anti-symmetric parts of which symmetric part is easier to compute Lipman et al. [2010], Ovsjanikov et al. [2013]. While prior work explicitly factors out these subspaces from a functional map between the two shapes, in contrast, we obtain this representation implicitly by virtue of learning a self symmetry map during training. In this paper, we present a novel non-rigid shape matching method based on a nearest neighbour approach in canonical embedding by learning a self symmetric intrinsic functional map. Our key idea is to learn an embedding of each shape that would make the given self-symmetry map linear in some higher-dimensional space. We hypothesize that learning self symmetry map during training acts as a regularizer for learning a canonical embedding of 3D shapes and thus, improves generalization. Our method obtains superior results on two partial shape matching SHREC benchmarks when compared to recent learning based methods while being computationally cheaper.

## 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 Sahillioğlu [2019] for an in-depth discussion. Our method is closely related to the functional map pipeline, introduced in Ovsjanikov et al. [2012] and then significantly extended in follow-up works (see, e.g., Ovsjanikov et al. [2017]). The key idea of this framework is to encode correspondences as small matrices, by using a reduced functional basis, thus greatly simplifying many resulting optimization problems. The functional map pipeline has been further improved in accuracy, efficiency and robustness by many recent works including Kovnatsky et al. [2013], Huang et al. [2014], Burghard et al. [2017], Rodolà et al. [2017], Nogneng and Ovsjanikov [2017], Ren et al. [2018], Eisenberger et al. [2020], Ginzburg and Raviv [2020]. There also exist other works Wei et al. [2016], Boscaini et al. [2016], Monti et al. [2017] that treat shape correspondence as a dense labeling problem but they typically require a lot of data as the label space is very large.

**Learning from raw 3D shape** Although early approaches in functional maps literature used hand-crafted features Ovsjanikov et al. [2017], more recent methods directly aim to *learn* either the optimal transformations of hand crafted descriptors Litany et al. [2017b], Roufousse et al. [2019] or even features directly from 3D geometry itself Donati et al. [2020], Sharma and Ovsjanikov [2020]. Initial efforts in this direction used classical optimisation techniques, Corman et al. [2014]. In contrast, Deep Functional Maps Litany et al. [2017a] proposed a deep learning architecture called FMNet to optimize a non-linear transformation of SHOT descriptors Tombari et al. [2010], that was further extended to unsupervised setting Roufousse et al. [2019], Halimi et al. [2019]. To alleviate the sensitivity to SHOT descriptor, recent works including Groueix et al. [2018], Donati et al. [2020], Sharma and Ovsjanikov [2020] 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.

**Self Supervised Learning** Self supervised learning has been exploited for learning representations and embedding in various domains where a proxy task is used to learn the representation. e.g. Sharma et al. [2016] uses an autoencoder to complete the partial shapes and uses the resulting representation of shape completion for shape classification task. Gidaris et al. [2018] learns to predict image rotations and uses the resulting representation for image classification. Our formulation is in the same spirit as we learn to predict the symmetry in a 3D shape and use the resulting representation for 3D shape matching. However, we choose symmetry learning as a proxy task for embedding learning for a principled reason which we describe in detail in the methodology section.

**Learning Basis from Data for Partial Shape Matching** Most of the functional map framework can not handle partiality in data as they rely on Laplacian eigenfunctions that are shown to be unstable under partial data. Rodolà et al. [2017], Litany et al. [2017b], Wu et al. [2020] deal with partiality but they are based on hand-crafted features and require expensive optimization scheme and is instance specific. While Sharma and Ovsjanikov [2020] proposes to learn a suitable alignment of pre-computed Laplacian Eigen basis functions, the approach still relies on the Laplacian basis and can therefore be unstable. Marin et al. [2020] proposed a two stage architecture to learn a linear transformation invariant shape embedding to bypass the difficulties associated with LBO. However,

as we demonstrate later in experiments, the two stage architecture is suboptimal due to the lack of adequate regularization.

**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 Mitra et al. [2012], Raviv et al. [2010]. More recently, Shi et al. [2020] propose 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 method models or learn symmetry explicitly as a regularizer for shape matching. Our work is mainly related to the prior work Lipman et al. [2010], Ovsjanikov et al. [2013] that proposes a principled framework for computation in symmetric subspaces. Ovsjanikov et al. [2013] shows a functional map decomposition into symmetric and anti-symmetric part and argues that it is easier to compute the symmetric part. However, this framework is based on handcrafted feature descriptors as well as precomputed Laplacian eigen Basis functions and thus, it is non-trivial to extend it directly to an end to end learning framework.

Rest of the paper is structured as follows: In the next section, we briefly cover the necessary background on functional map and its symmetrized counterpart. Afterwards, we propose a novel learning strategy to learn symmetrized embedding and generalize the functional maps framework to noisy and incomplete data. Lastly, we validate our framework on three benchmark datasets.

### 3 Background

Before describing our method, we provide a brief overview of the basic pipeline to compute a functional map and its symmetrized version.

**Functional Map Computation** We assume that we are given a source and a target shape,  $S_1, S_2$ , containing, respectively,  $n_1$  and  $n_2$  vertices, a small set of  $k_1, k_2$  of basis functions, e.g. of the respective Laplace-Beltrami operators (LBO). For the sake of simplicity, we also assume that we are given a set of descriptors on each shape, to be preserved by the unknown map, whose coefficients in the LB basis are stored as columns of matrices  $\mathbf{A}, \mathbf{B}$ . The optimal *functional map* is computed  $\mathbf{C}$  by solving the following optimization problem:

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

where  $E_{\text{desc}}(\mathbf{C}) = \|\mathbf{C}\mathbf{A} - \mathbf{B}\|^2$  aims at the descriptor preservation whereas the second term acts as a regularizer on the map by enforcing its overall structural properties, such as commutativity with the Laplacian, which corresponds to promoting near-isometries (see Chapter 2.4.4 and Algorithm 1 in Ovsjanikov et al. [2017] for details). The optimization problem in Eq (1) can be solved with any convex solver. Once functional map  $\mathbf{C}$  is computed, one can use nearest neighbor search in the spectral embedding to convert it to point to point correspondence.

Note that when descriptor functions are neural network based, instead of optimizing over  $\mathbf{C}$ , we are optimizing the functional in Eq.1 over  $\mathbf{C}, \mathbf{A}$  and  $\mathbf{B}$ . In this case, joint optimization over  $\mathbf{C}, \mathbf{A}$  and  $\mathbf{B}$  is challenging as  $\mathbf{C}$  is computed via an iterative solver itself. Deep functional map Litany et al. [2017a] propose to optimize for descriptors, which define the spectral coefficients  $\mathbf{A}$  and  $\mathbf{B}$  simply through projection onto the Laplacian basis, whereas the functional map matrix  $\mathbf{C}$  is defined implicitly via  $\mathbf{C} = \arg \min_X \|X\mathbf{A} - \mathbf{B}\|$ . This results in a simple linear system for which the derivatives can be computed in closed form.

**Symmetrized Functional Map Pipeline** The functional map pipeline described above is known to have issues disambiguating intrinsic symmetries Ovsjanikov et al. [2013]. To avoid this, Ovsjanikov et al. [2013] propose a simple pipeline that splits the shape matching problem into two steps – first a linear transformation is computed between a set of *symmetric* basis functions, which is then extended to a complete point-to-point map by continuity. This approach is summarized as follows: 1. given a self-symmetry map on each shapes, a set of *symmetrized* basis functions is computed by projecting the Laplacian eigen-basis onto the space of functions that are invariant under the given self-maps. 2. The functional map between symmetrized bases is computed by using standard descriptors and regularization terms, 3. This map is then extended to the complete space by enforcing symmetry-preservation during the conversion of functional to point-to-point maps. The main observation

of Ovsjanikov et al. [2013] is that computing functional maps between symmetric basis functions only is easier, since this space does not suffer from ambiguities and is thus more stable. At the same time, this still relies on hand-crafted descriptors and the Laplace-Beltrami basis, making it parameter-dependent and limited to near-isometries.

## 4 Learning Symmetrized Embedding

In the previous section, we outlined a basic mechanism to take any functional map and compute its symmetric part. However, this requires as input a space of symmetric functions on each shape and a set of basis functions. Due to the instability of LBO on partial 3D shapes, our main goal is to avoid using LBO and instead *learn* an embedding that can replace the spectral embedding given by the Laplace-Beltrami eigenfunctions. This section details how to learn such an embedding whilst working in the symmetric space.

We assume to be given a collection of shapes with a self symmetry ground truth map for each shape and our goal is to find an embedding that respects the given symmetry of each shape and that ultimately can reduce shape correspondence between a pair of shapes to a nearest neighbor search between their embeddings. We draw our inspiration from a recent work Marin et al. [2020] that has proposed to replace the Laplace-Beltrami basis by learning embeddings that are related by linear transforms 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 space, in which transformations across shapes are linear. However, using a supervised loss to learn this transform without enforcing any structural properties on this linear transform provides no guarantee that the learned transform will generalize from train to test setting.

Our key idea is to learn an embedding of each shape that would make the given self-symmetry map linear in some higher-dimensional space. Learning self symmetry map simultaneously also acts as a regularizer. Our network takes a shape as input  $\mathbf{X}$  and its point to point symmetry map denoted as  $\mathbf{T}_{\mathbf{X}\mathbf{X}_f}$ . We use  $P_{\mathbf{X}}$  to denote the 3D coordinates of  $\mathbf{X}$ . It then flips each shape along one axis resulting in a shape denoted as  $\mathbf{X}_f$ . The original and flipped shapes are then forwarded to a Siamese architecture, based on a PointNet Qi et al. [2017] feature extractor, that projects 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  $\mathbf{X}_f$ . In the functional map framework, there exists a functional map  $C_{\mathbf{X}\mathbf{X}_f}$  that aligns the corresponding embeddings. Given a self symmetry ground truth pointwise map  $\mathbf{T}_{\mathbf{X}\mathbf{X}_f}$ , we can estimate  $C_{\mathbf{X}\mathbf{X}_f}$  by solving the following optimization problem:

$$C_{\mathbf{X}\mathbf{X}_f} = \arg \min_C \|\Phi_{\mathbf{X}} C^T - \mathbf{T}_{\mathbf{X}\mathbf{X}_f} \Phi_{\mathbf{X}_f}\|_2 \quad (2)$$

The optimal symmetry map  $C_{\mathbf{X}\mathbf{X}_f}$  is given by:  $C_{\mathbf{X}\mathbf{X}_f} = (\Phi_{\mathbf{X}}^+ \mathbf{T}_{\mathbf{X}\mathbf{X}_f} \Phi_{\mathbf{X}_f})^T$ , that is differentiable using the closed-form expression of derivatives of matrix inverses, as also mentioned in Section 3. Similarly, we can compute  $C_{\mathbf{Y}\mathbf{Y}_f}$  for shape  $Y$ .

**Training Loss** Given a set of pairs of shapes  $\mathbf{X}, \mathbf{Y}$  for which ground truth correspondences  $\mathbf{T}_{\mathbf{X}\mathbf{Y}}^{gt}$  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  $C_{\mathbf{X}\mathbf{X}_f}$  and  $C_{\mathbf{Y}\mathbf{Y}_f}$  respectively as described above. We then optimize the sum of two loss functions, one each defined for self symmetry map and pairwise map computation. Both loss functions are based on a soft-correspondence matrix, also used in Litany et al. [2017a] with geodesic weights and without in Marin et al. [2020], that penalizes incorrect correspondences based on the Euclidean distances of associated points. To define it for self symmetry map, we *transform* each embedding  $\hat{\Phi}_{\mathbf{X}} = \Phi_{\mathbf{X}} C_{\mathbf{X}\mathbf{X}_f}^T$  by applying the optimal symmetry map. We then compare the rows of  $\hat{\Phi}_{\mathbf{X}}$  to those of  $\Phi_{\mathbf{X}_f}$  to obtain the *soft* correspondence matrix  $S_{\mathbf{X}\mathbf{X}_f}$  that approximates the self-symmetry map in a differentiable way as follows:

$$(S_{\mathbf{X}_f\mathbf{X}})_{ij} = \frac{e^{-\|\hat{\Phi}_{\mathbf{X}}^i - \Phi_{\mathbf{X}_f}^j\|_2}}{\sum_{k=1}^n e^{-\|\hat{\Phi}_{\mathbf{X}}^i - \Phi_{\mathbf{X}_f}^k\|_2}} \quad (3)$$

To compute the soft correspondence matrix  $S_{\mathbf{X}\mathbf{Y}}$  between a pair of shapes, we compare the rows of  $\Phi_{\mathbf{X}}$  to those of  $\Phi_{\mathbf{Y}}$  in a differentiable way as done in Eq. 3. Note that there is no embedding

transformation involved in this case. Finally, we combine the two loss functions and write the overall loss as follows:

$$L(\Phi_{\mathbf{X}}, \Phi_{\mathbf{X}_f}, \Phi_{\mathbf{Y}}, \Phi_{\mathbf{Y}_f}) = \sum \|S_{\mathbf{X}_f \mathbf{X}} P_{\mathbf{X}} - \mathbf{T}_{\mathbf{X} \mathbf{X}_f}^{gt} P_{\mathbf{X}_f}\|_2^2 + \sum \|S_{\mathbf{Y}_f \mathbf{Y}} P_{\mathbf{Y}} - \mathbf{T}_{\mathbf{Y} \mathbf{Y}_f}^{gt} P_{\mathbf{Y}_f}\|_2^2 + \sum \|S_{\mathbf{X} \mathbf{Y}} P_{\mathbf{Y}} - \mathbf{T}_{\mathbf{X} \mathbf{Y}}^{gt} P_{\mathbf{Y}}\|_2^2. \quad (4)$$

**Testing** : 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. .

**Implementation Details** We implement our method in Pytorch Paszke et al. [2019]. All experiments are run on a Nvidia RTX 2080 graphics processing card and require 5 GB of GPU memory. For our experiments, similar to prior work Sharma and Ovsjanikov [2020], Marin et al. [2020], we train over randomly selected 1K shapes from the SURREAL dataset Varol et al. [2017], where each point cloud is resampled randomly at 3K vertices. We learn a  $k = 40$  dimensional embedding (basis) for each point cloud.

During training, we require self-symmetry ground truth as well as pairwise ground truth map. We train on Surreal dataset instead of ShrecCosmo et al. [2016] as Shrec only has ground truth symmetry data for one whereas we require it for both null and partial shapes. Following Sharma and Ovsjanikov [2020], Marin et al. [2020], our feature extractor is also based on the semantic segmentation architecture of PointNet. For the network details, we attach here the source code. We use a batch size of 12 and learning rate of  $1e - 5$  and optimize our objective with Adam optimizer in Pytorch Paszke et al. [2019]. Training iterations as well as other hyperparameters are validated on a small validation set.

Unlike Marin et al. [2020] that uses fixed 1000 points during train and test time, we use full resolution at test time. During training, all point cloud contains 6890 points. We experimented with 3 sampling density from the set  $2k, 3k, 4k$  and found  $2k$  as well as  $3k$  to provide similar results. However, increasing the point cloud density further during training leads to a gradual performance drop on testing at Cosmo et al. [2016] dataset later. This can be explained by the fact that this dataset has several shapes containing less than 2000 points. We obtain an embedding of 40 dimensions during training by validating from set 30, 40, 50. Our results are not sensitive to small variations in the size of embedding.

## 5 Results

This section is divided into two subsections. First subsection 5.1 shows the experimental comparison of our approach with two state-of-the art methods for near-isometric partial shape matching. Section 5.2 demonstrates the effectiveness on full shape matching. 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.

### 5.1 Partial Shape Matching

We compare our method with state-of-the-art approaches focusing especially on the the very recent functional map-based techniques Sharma and Ovsjanikov [2020], Marin et al. [2020], which was shown to outperform existing competitors. Note that Marin et al. [2020] is the only baseline that learns embedding for shape matching between point clouds and thus, our main competitor.

**Datasets** For a fair comparison with Sharma and Ovsjanikov [2020], Litany et al. [2017b], we follow the same experimental setup and test our method on on the challenging SHREC’16 Partial Correspondence dataset Cosmo et al. [2016]. 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 sets, namely cuts (removal of a few large parts) and holes (removal of many small parts). We use the same test set following Sharma and Ovsjanikov [2020]. However, we also include the wolf class missing in their setup. Overall, this test set contains 24 shapes each for cuts and holes datasets chosen randomly from the two sets respectively.

Table 1: Avg. Geodesic Error on two partial SHREC benchmarks

Method \ Dataset	Holes	Cuts
Litany et al. [2017b]	16	13
Sharma and Ovsjanikov [2020]	14	16
Marin et al. [2020]-1k	19	22
Marin et al. [2020]-3k	12	15
Canonical Emb.	18	20
Canonical Emb. + symmetry(Ours)	<b>9</b>	<b>12</b>

**Baselines** We compare with the following four very competitive baselines:

**Litany et al. [2017b]** This baseline is not learning based and relies on hand crafted features and expensive optimization scheme on Stiefel manifold for every pair of shapes at test time. Thus, in terms of computation and ground truth map requirement, it is most expensive.

**Sharma and Ovsjanikov [2020]** This baseline, although weakly supervised, also relies on learning LBO alignment and thus, is dependent on class that 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. We obtain their results by running the code provided by the authors.

**Marin et al. [2020]** This is considered state of the art for learning embedding directly from data. They obtain all their results by training on  $1k$  subsampled points from each point cloud and testing on  $1k$ . Since we are testing with point clouds of much higher resolution, we retrain their models with  $3k$  subsampled points for each point cloud and show their results with both resolutions  $1k$  as well as  $3k$ . The rest of the parameters such as embedding size are used as such as they were found to be optimal. We use their open source code to retrain on our subset of Surreal dataset. Note that this baseline is somewhat different from others since it requires both basis training and descriptors (feature functions).

**Canonical Emb.** This baseline ablates the overall performance of our method and quantifies the gain brought in by learning self symmetry map during training. It shows the performance if we simply remove the symmetry loss and learn an embedding by just projecting the shapes into a 50 dimensional space. We denote our results with **Canonical Emb.+ symmetry** in the Table 1.

**Results and Discussion** We present our findings on partial shape matching in Table 1. As shown in that table, we obtain superior performance on both benchmark datasets for partial shape matching. In addition, our result outperforms the canonical embedding competitive baseline by a significant margin and thus, validating the importance of working in the symmetric space while learning canonical embedding. We would like to stress that baseline such as Sharma and Ovsjanikov [2020] are class specific and needs to be trained each time whereas our method is class agnostic and can obtain high quality results with a fraction of time. Similarly, Marin et al. [2020] trains similar network as our two times. First, it computes the embedding and then it again trains a similar network to compute the optimal linear transformation between the two embedding. Moreover, testing it also requires running a network twice. Therefore, our method is at least twice faster than nearest baseline in computational complexity.

We attribute our superior quantitative results over other learning based methods to a range of factors. First, in contrast to Marin et al. [2020] that is based on two stage sequential architecture, our symmetry factored embedding is trained end to end in one phase. As a result, embedding obtained by Marin et al. [2020] optimization is suboptimal with the canonical embedding as we also show in the supplement. Second, none of the baseline methods takes symmetry into account even though symmetry ambiguities for shape matching is a well known problem and studied extensively in axiomatic methods. Third, performance of Marin et al. [2020] is sensitive to a specific point cloud resolution and low size of embedding 20. In contrast, due to the regularization provided by learning symmetry map, we can train with twice their embedding size.

**Cumulative Curves** In Figure 1 and 2, we show the corresponding curves below that are consistent with average geodesic error shown in Table 1. Y axis shows fraction of correspondence whereas X axis shows the geodesic error. We include comparison with rest of the methods in supplement.

Figure 1: Cumulative curves on SHREC Cuts

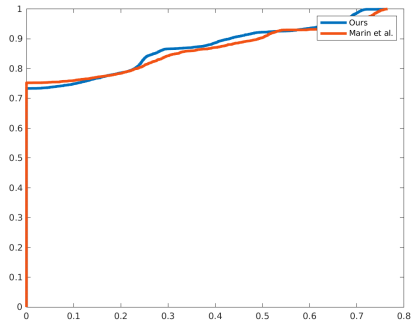


Figure 2: Cumulative curves on SHREC Holes

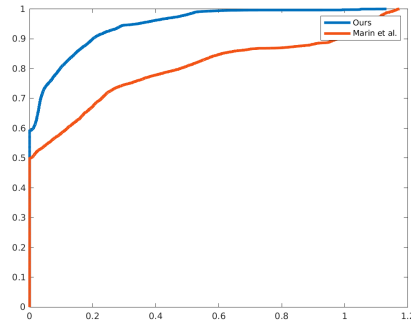


Table 2: Avg. Geodesic Error on FAUST benchmark

Method \ Dataset	Faust
Marin et al. [2020]-3k	20
Canonical Emb.	19
Canonical Emb. + symmetry(Ours)	<b>12</b>

## 5.2 Full Shape Matching

We also present our results on full shape matching benchmark dataset FAUSTBogo et al. [2014]. This dataset contains 100 shapes of 10 different subjects in different poses where each point cloud contains 6890 points. Following prior work, we use the last 20 shapes as test set and report the performance on this test set. We compare our results with Marin et al. [2020] in Table 2 as this approach is applicable, in principle, to both partial and complete shape matching. Note that in their work, Marin et al. [2020] presents results on FAUST data that is subsampled to 1000 points both during train and test. Our method obtains significantly better results than Marin et al. [2020] although our results are not state of the art for complete shape matching. Note that in the case of complete shape matching, LBO eigen functions already form a good basis for shapes and thus, prior work based on finding LBO eigen basis alignment obtains impressive performance. However, performance of this line of work collapses badly under partiality, as also shown in Marin et al. [2020].

## 6 Conclusion and Limitation

In shape correspondence literature, partial shape matching and complete shape matching are generally tackled by two different set of methods which obtain impressive results in one of the two respective domain. We presented a simple, general but effective method that reduces shape matching to a nearest neighbour search problem in a canonical embedding and apply it to both partial and complete shape matching. Our key idea is to learn an embedding of each shape that would make the given self-symmetry map linear in some higher-dimensional space. Our idea of injecting symmetry into the learning pipeline also serves as a regularizer and provides superior performance on two partial shape matching benchmarks in comparison to all recent learning based methods.

We have demonstrated our results on near isometric benchmarks and extension to non-isometric shapes is left as a future work.

## 7 Acknowledgement

Parts of this work were supported by the ERC Starting Grant StG-2017-758800 (EXPROTEA), and ANR AI chair AIGRETTE.



## References

- Federica Bogo, Javier Romero, Matthew Loper, and Michael J. Black. FAUST: Dataset and evaluation for 3D mesh registration. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Piscataway, NJ, USA, June 2014. IEEE.
- Davide Boscaini, Jonathan Masci, Emanuele Rodola, and Michael M. Bronstein. Learning shape correspondence with anisotropic convolutional neural networks. In *Proc. NIPS*, pages 3189–3197, 2016.
- Oliver Burghard, Alexander Dieckmann, and Reinhard Klein. Embedding shapes with Green’s functions for global shape matching. *Computers & Graphics*, 68:1–10, 2017.
- P Chavdar, H Sami, P Sven, Krieger Kai, and B. Darius. Rigid 3d geometry matching for grasping of known objects in cluttered scenes. *IJRR*, 31(4), 2012.
- Etienne Corman, Maks Ovsjanikov, and Antonin Chambolle. Supervised descriptor learning for non-rigid shape matching. In *Proc. ECCV Workshops (NORDIA)*, 2014.
- Luca Cosmo, Emanuele Rodola, Jonathan Masci, Andrea Torsello, and Michael M Bronstein. Matching deformable objects in clutter. In *3D Vision (3DV), 2016 Fourth International Conference on*, pages 1–10. IEEE, 2016.
- Nicolas Donati, Abhishek Sharma, and Maks Ovsjanikov. Deep geometric functional maps: Robust feature learning for shape correspondence. In *CVPR*, 2020.
- Marvin Eisenberger, Zorah Lahner, and Daniel Cremers. Smooth shells: Multi-scale shape registration with functional maps. In *CVPR*, 2020.
- Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *ICLR*, 2018.
- Dvir Ginzburg and D. Raviv. Cyclic functional mapping: Self-supervised correspondence between non-isometric deformable shapes. In *ECCV*, 2020.
- Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. 3d-coded: 3d correspondences by deep deformation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 230–246, 2018.
- Oshri Halimi, Or Litany, Emanuele Rodolà, Alex Bronstein, and Ron Kimmel. Unsupervised learning of dense shape correspondence. In *CVPR*, 2019.
- Qixing Huang, Fan Wang, and Leonidas Guibas. Functional map networks for analyzing and exploring large shape collections. *ACM Transactions on Graphics (TOG)*, 33(4):36, 2014.
- Maxime Kirgo, Simone Melzi, Giuseppe Patanè, Emanuele Rodolà, and Maks Ovsjanikov. Wavelet-based heat kernel derivatives: Towards informative localized shape analysis. In *Computer Graphics Forum*. Wiley Online Library, 2020.
- Artiom Kovnatsky, Michael M Bronstein, Alexander M Bronstein, Klaus Glashoff, and Ron Kimmel. Coupled quasi-harmonic bases. In *Computer Graphics Forum*, volume 32, pages 439–448, 2013.
- Y. Lipman, Xiaobai Chen, I. Daubechies, and T. Funkhouser. Symmetry factored embedding and distance. In *SIGGRAPH*, 2010.
- Or Litany, Tal Remez, Emanuele Rodolà, Alexander M. Bronstein, and Michael M. Bronstein. Deep functional maps: Structured prediction for dense shape correspondence. In *ICCV*, pages 5660–5668, 2017a.
- Or Litany, Emanuele Rodolà, Alex M Bronstein, and Michael M Bronstein. Fully spectral partial shape matching. In *Computer Graphics Forum*, volume 36, pages 247–258. Wiley Online Library, 2017b.
- Riccardo Marin, Marie-Julie Rakotosaona, Simone Melzi, and Maks Ovsjanikov. Correspondence learning via linearly-invariant embedding. In *NeurIPS*, volume 33, pages 1608–1620, 2020.

- Niloy J. Mitra, Mark Pauly, Michael Wand, and Duygu Ceylan. Symmetry in 3d geometry: Extraction and applications. In *EUROGRAPHICS State-of-the-art Report*, 2012.
- Federico Monti, Davide Boscaini, Jonathan Masci, Emanuele Rodolà, Jan Svoboda, and Michael M. Bronstein. Geometric deep learning on graphs and manifolds using mixture model cnns. In *CVPR*, pages 5425–5434. IEEE Computer Society, 2017.
- Natalia Neverova, David Novotny, Vasil Khalidov, Marc Szafraniec, Patrick Labatut, and Andrea Vedaldi. Continuous surface embeddings. In *NeurIPS*, 2020.
- Dorian Nogneng and Maks Ovsjanikov. Informative descriptor preservation via commutativity for shape matching. *Computer Graphics Forum*, 36(2):259–267, 2017.
- Maks Ovsjanikov, Mirela Ben-Chen, Justin Solomon, Adrian Butscher, and Leonidas Guibas. Functional Maps: A Flexible Representation of Maps Between Shapes. *ACM Transactions on Graphics (TOG)*, 31(4):30, 2012.
- Maks Ovsjanikov, Quentin Mérigot, Viorica Pătrăucean, and Leonidas Guibas. Shape matching via quotient spaces. In *SGP*, page 1–11, 2013.
- Maks Ovsjanikov, Etienne Corman, Michael Bronstein, Emanuele Rodolà, Mirela Ben-Chen, Leonidas Guibas, Frederic Chazal, and Alex Bronstein. Computing and processing correspondences with functional maps. In *ACM SIGGRAPH 2017 Courses*, pages 5:1–5:62, 2017.
- Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, pages 8024–8035, 2019.
- Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proc. CVPR*, pages 652–660, 2017.
- D. Raviv, A. Bronstein, M. Bronstein, and R. Kimmel. Full and partial symmetries of non-rigid shapes. *International Journal of Computer Vision*, 89:18–39, 2010.
- Jing Ren, Adrien Poulénard, Peter Wonka, and Maks Ovsjanikov. Continuous and orientation-preserving correspondences via functional maps. *ACM Transactions on Graphics (TOG)*, 37(6), 2018.
- Emanuele Rodolà, Luca Cosmo, Michael M Bronstein, Andrea Torsello, and Daniel Cremers. Partial functional correspondence. In *Computer Graphics Forum*, volume 36, pages 222–236. Wiley Online Library, 2017.
- Jean-Michel Roufousse, Abhishek Sharma, and Maks Ovsjanikov. Unsupervised deep learning for structured shape matching. In *ICCV*, pages 1617–1627, 2019.
- Yusuf Sahillioğlu. Recent advances in shape correspondence. *The Visual Computer*, pages 1–17, 2019.
- Abhishek Sharma and Maks Ovsjanikov. Weakly supervised deep functional maps for shape matching. In *NeurIPS*, volume 33, 2020.
- Abhishek Sharma and Maks Ovsjanikov. Matrix decomposition on graphs: A functional view. *arXiv*, 2021.
- Abhishek Sharma, Oliver Grau, and Mario Fritz. Vconv-dae: Deep volumetric shape learning without object labels. In *ECCV*, 2016.
- Yifei Shi, Junwen Huang, Hongjia Zhang, Xin Xu, Szymon Rusinkiewicz, and Kai 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.

- Oshane O. Thomas, Hongyu Shen, Ryan L. Raaum, William E.H. Harcourt-Smith, John D. Polk, and Mark Hasegawa-Johnson. Automated morphological phenotyping using learned shape descriptors and functional maps: A novel approach to geometric morphometrics. *bioRxiv*, 2021. URL <https://www.biorxiv.org/content/early/2021/05/18/2021.05.18.444628>.
- Federico Tombari, Samuele Salti, and Luigi Di Stefano. Unique signatures of histograms for local surface description. In *International Conference on Computer Vision (ICCV)*, pages 356–369, 2010.
- Gül Varol, Javier Romero, Xavier Martin, Naureen Mahmood, Michael J. Black, Ivan Laptev, and Cordelia Schmid. Learning from synthetic humans. In *CVPR*, 2017.
- Fudong Wang, Gui-Song Xia, Nan Xue, Yipeng Zhang, and M. Pelillo. A functional representation for graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42:2737–2754, 2020.
- Lingyu Wei, Qixing Huang, Duygu Ceylan, Etienne Vouga, and Hao Li. Dense human body correspondences using convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1544–1553, 2016.
- Yan Wu, Jun Yang, and Jinlong Zhao. Partial 3d shape functional correspondence via fully spectral eigenvalue alignment and upsampling refinement. *Comput. Graph.*, 92:99–113, 2020.