Learning Canonical Embedding for Non-rigid Shape Matching

Anonymous authors
Paper under double-blind review

Abstract

This paper provides a novel framework that learns canonical embeddings 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 or sequential optimization schemes. 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 via a simple nearest neighbor search. Our framework outperforms multiple recent learning based methods on FAUST and SHREC benchmarks while being computationally cheaper, data efficient, and robust.

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 (Bogo 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 & 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 (Litany et al., 2017b; Roufosse et al., 2019; Halimi et al., 2019; Sharma & Ovsjanikov, 2020), 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). As a consequence, there is no unified framework that excels at learning both partial as well as full non-rigid shape matching despite some efforts in this direction (Sharma & Ovsjanikov, 2020; Marin et al., 2020). While the former approach Sharma & Ovsjanikov (2020) is class-specific and requires retraining a network for each class, the latter (Marin et al., 2020) employs a two stage optimization strategy without adequate regularization which, as we show later, is suboptimal.

Instead of using predefined basis functions in the functional map framework, learning an embedding, to be used as a basis in the functional map, is a promising direction towards obtaining a unified shape matching framework for both partial as well as full shape matching. However, there exist few works that exploit prior information on such embeddings. Marin et al. (2020) make the first attempt by assuming such embeddings to be linearly invariant between a pair of shapes. However, as we show later, learning such an embedding without exploiting natural priors on 3D shapes, such as their symmetry structure, leads to overfitting as no regularization or constraint is enforced on the linear transformation between a pair of shape embeddings (Marin et al., 2020).

In this paper, we present a novel non-rigid shape matching method based on a nearest neighbour approach in canonical embedding. We work with point cloud representation of 3D shapes and assume to be given a self symmetry map for each shape during training. We make two assumptions on the canonical embedding: our first assumption is to learn an embedding of each shape that would make the given self-symmetry map *linear* in some higher-dimensional space. This is advantageous for two reasons: first, it significantly simplifies the embedding learning pipeline and makes it learnable end-to-end. This is because the linearly invariant assumption is made on a self-symmetry map and not on the pairwise map between shapes as done in (Marin et al., 2020). This alleviates the need to retrieve this linear transformation for shape matching at test time and thereby, reduces the two stage sequential optimization scheme of Marin et al. (2020) to joint optimization. Secondly, modelling a self symmetry map enables us to explicitly enforce the pointwise map between two shapes to take into account the intrinsic self-symmetry during training. Thus, our second contribution is a novel regularizer that, combined with the learned self symmetry map during training, significantly improves generalization and robustness to sampling resolution as well as the size of embedding. Our method obtains superior results on multiple shape matching benchmarks such as FAUST and SHREC when compared to recent learning-based methods while being computationally cheaper, more robust and data efficient.

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 & Ovsjanikov, 2017; Ren et al., 2018; Eisenberger et al., 2020; Ginzburg & 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 handcrafted features (Ovsjanikov et al., 2017), more recent methods directly aim to *learn* either the optimal transformations of hand crafted descriptors (Litany et al., 2017b; Roufosse et al., 2019) or even features directly from 3D geometry itself (Donati et al., 2020; Sharma & 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 (Roufosse 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 & 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 inject the symmetry information 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 Most of the functional map frameworks 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 an expensive optimization scheme and are instance specific. While Sharma & 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 (Raviv et al., 2010; Lipman et al., 2010; Mitra et al., 2012; Ovsjanikov et al., 2013). More recently, Shi et al. (2020) 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 method models or learn symmetry explicitly as a regularizer for shape matching.

Rest of the paper is structured as follows: In the next section, we briefly cover the necessary background on the functional map. Afterwards, we propose a novel learning strategy to learn canonical embedding and introduce our novel regularization based on intrinsic symmetry prior. Lastly, we validate our framework on three benchmark datasets by comparing it to various state-of-the-art methods and providing ablation studies.

3 Background

Before describing our method, we provide a brief overview of the basic pipeline to compute a functional map (Ovsjanikov et al., 2012).

Functional Map Computation The typical functional map pipeline (Ovsjanikov et al., 2012) assumes that we are given a source and a target shape, $\mathbf{X}, \mathbf{Y}$, containing, respectively, n_x and n_y vertices, a small set of k basis functions, e.g. of the respective Laplace-Beltrami operators (LBO). We are also given a set of descriptors on each shape, to be preserved by the unknown map, whose coefficients in the basis functions are stored as columns of matrices $\Phi_{\mathbf{X}}, \Phi_{\mathbf{Y}}$. The optimal *functional map* $\mathbf{C}_{\text{opt}}$ is computed by solving the following optimization problem:

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

where $E_{\text{desc}}(\mathbf{C}) = \left\|\mathbf{C}\Phi_{\mathbf{X}} - \Phi_{\mathbf{Y}}\right\|^2$ aims at the descriptor preservation whereas the second term acts as a regularizer on the map by enforcing its overall structural properties, such as bijectivity of the map. The optimization problem in equation 1 can be solved with any convex solver. Once the optimal functional map $\mathbf{C}_{\text{opt}}$ is computed, one can use nearest neighbor search in the spectral embedding to convert it to a point to point correspondence.

Note that when the basis functions are neural network-based, instead of optimizing over $\mathbf{C}$, we are optimizing the functional in equation 1 over $\mathbf{C}$, $\Phi_{\mathbf{X}}$ and $\Phi_{\mathbf{Y}}$. In this case, joint optimization over $\mathbf{C}$, $\Phi_{\mathbf{X}}$ and $\Phi_{\mathbf{Y}}$ is challenging as $\mathbf{C}$ is computed via an iterative solver itself.

4 Learning Canonical Embedding

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

Input Shape Representation In contrast to several recent works (Halimi et al., 2019; Sharma & Ovsjanikov, 2020) that assume to be given a mesh representation of 3D shapes in non-rigid shape matching, we do not impose any such constraint and directly work the with point cloud representation.

Type of Supervision In addition to the pointwise map between shapes, 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.

When compared to our main baseline (Marin et al., 2020), this is an additional supervision and thus, we also evaluate Marin et al. (2020) with the same supervision for fair comparison.

Our work is most closely related to a recent work (Marin et al., 2020) 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.

Linearly Invariant self-symmetry embedding We denote a map between a pair of shapes $\mathbf{X}$ and $\mathbf{Y}$ by $T_{\mathbf{XY}} : \mathbf{X} \rightarrow \mathbf{Y}$ such 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 the same notation T for self symmetry map $\mathbf{T}_{\mathbf{XX_f}}$ as well. We use $P_{\mathbf{X}}$ to denote the 3D coordinates of $\mathbf{X}$.

Our network takes a shape $\mathbf{X}$ as input as well as its point to point symmetry map denoted as $\mathbf{T}_{\mathbf{XX_f}}$. We then perform a reflection (flip) of 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 embeds these two shapes into some fixed k dimensional space. We illustrate in Figure 1 one such flip. The intuition behind this operation is to help the network learn representation that can disambiguate left from right in shape matching. Our first key idea is to learn an embedding of each shape that would make the given self-symmetry map linear in some higher-dimensional space.

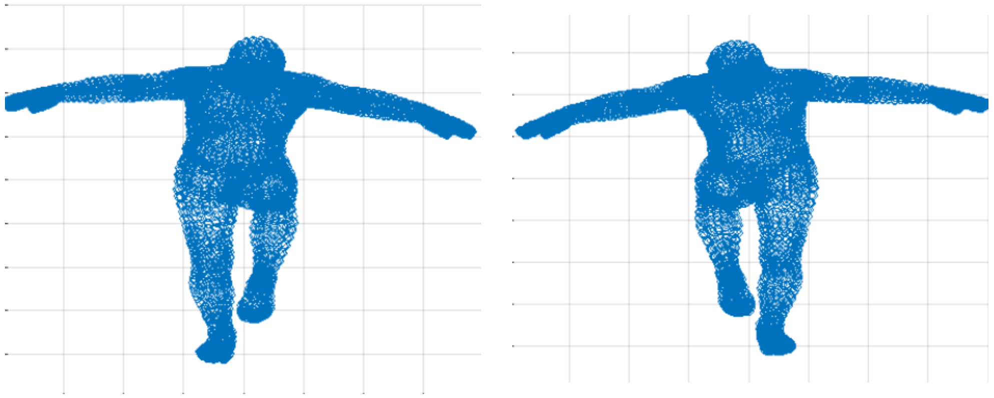


Figure 1: Original Shape and its Flipped version from Surreal dataset

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:

$$C_{\mathbf{XX_f}} = \arg\min_{C} \|\Phi_{\mathbf{X}} C^T - \mathbf{T}_{\mathbf{XX_f}} \Phi_{\mathbf{X_f}}\|_2 \tag{2}$$

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 Section 3. Similarly, we can compute $C_{\mathbf{YY_f}}$ for shape $\mathbf{Y}$.

4.1 Loss functions

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 each defined for linearly invariant self symmetry embedding, nearest neighbour based loss for pairwise (shape pair) embedding and a commutativity loss for explicitly enforcing intrinsic symmetry during training.

Linearly Invariant Loss The first two loss functions are based on a soft-correspondence matrix, also used in (Litany et al., 2017a) and (Marin et al., 2020). 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:

$$(S_{\mathbf{X}\mathbf{X}_\mathbf{f}})_{ij} = \frac{e^{-\|\widehat{\Phi}^i_{\mathbf{X}} - \Phi^j_{\mathbf{X}_\mathbf{f}}\|_2}}{\sum_{k=1}^n e^{-\|\widehat{\Phi}^i_{\mathbf{X}} - \Phi^k_{\mathbf{X}_\mathbf{f}}\|_2}} \tag{3}$$

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.

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

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 Marin et al. (2020) 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.

Our next 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 $S_{\mathbf{X}\mathbf{Y}}$ 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 equation 3. We then evaluate the computed soft map, again, by evaluating how well it transfers the coordinate functions, compared to the given ground truth mapping.

$$L(\Phi_{\mathbf{X}}, \Phi_{\mathbf{Y}})_{euc.} = \sum \|S_{\mathbf{X}\mathbf{Y}} P_{\mathbf{Y}} - \mathbf{T}^{gt}_{\mathbf{X}\mathbf{Y}} P_{\mathbf{Y}}\|_2^2. \tag{5}$$

Note that unlike the linearly invariant loss that we impose on the symmetry maps, 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 Marin et al. (2020), based on computing an optimal linear transformation at test time.

Symmetry Commutativity Loss Our final 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 map on shapes $\mathbf{X}$ 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:

$$L(\Phi_{\mathbf{X}}, \Phi_{\mathbf{Y}})_{comm.} = \|S_{\mathbf{Y}_\mathbf{f}\mathbf{Y}} S_{\mathbf{X}\mathbf{Y}} - S_{\mathbf{X}\mathbf{Y}} S_{\mathbf{X}_\mathbf{f}\mathbf{X}}\|_2 \tag{6}$$

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 maps literature. 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 training Loss We combine the two embedding losses defined in equation 4 and equation 5 with that of commutativity loss defined in equation 6 and define the training loss as follows:

$$L_{tot.} = L_{euc.} + \lambda * L_{lin.} + \gamma * L_{comm.} \tag{7}$$

The scalars λ and γ allow us to weigh the symmetry information differently in partial and full shape matching. Naturally, we set them higher for full shape matching where enforcing symmetry structure makes more sense than partial setting where symmetry is partial at times. We set λ and γ to 5 for full shape matching and 1 and .1 for partial shape matching based on a small validation set.

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.

Implementation Details We implemented our method in Pytorch Paszke et al. (2019). All experiments are run on a Nvidia RTX 2080 graphics processing card and require 16 GB of GPU memory. For our experiments, similar to prior work (Sharma & Ovsjanikov, 2020; Marin et al., 2020), we train over randomly selected 5000 shapes from the SURREAL dataset (Varol et al., 2017), where each point cloud is resampled randomly at 3K vertices. We learn a $k = 50$ dimensional embedding (basis) for each point cloud.

During training, we require self-symmetry ground truth as well as pairwise ground truth map. Following (Sharma & Ovsjanikov, 2020; Marin et al., 2020), our feature extractor is also based on the semantic segmentation architecture of PointNet. We use a batch size of 20 and learning rate of $1e - 4$ 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 of 500 shapes from the SURREAL dataset. Unlike Marin et al. (2020) that uses fixed 1000 points during train and test time, we use full resolution at test time. In the SURREAL dataset, all point cloud contain 6890 points. During training, we randomly sample 3000 points from the point cloud and also augment the training set with random rotations applied to input data. We obtain an embedding of 50 dimensions during training by validating from the set $40, 50, 60$. Our results are not sensitive to small changes in these two parameters. We probe the effect of changing size of embedding as well as the amount of training data on the resulting performance and report it in the last section.

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 full shape matching. Section 5.2 demonstrates the effectiveness on partial 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 Full Shape Matching

We present our results on a full shape matching benchmark dataset FAUST Bogo 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 a test set and report the performance on this test set. We compare our results with Marin et al. (2020); Sharma & Ovsjanikov (2020) in Table 2 as they are applicable, in principle, to both partial and complete shape matching. Note that Marin et al. (2020) presents results on the FAUST data that is subsampled to 1000 points both during train and test. Our method obtains significantly better results than Marin et al. (2020). Methods based on LBO eigen functions already form a good basis for shapes and thus, prior work based on LBO eigen basis obtains impressive performance. However, performance of this line of work degrades significantly under partiality, as shown in the next section and also in Marin et al. (2020).

Baselines We compare with the following two state-of-the-art methods that are shown to outperform existing competitors and our ablated baselines:

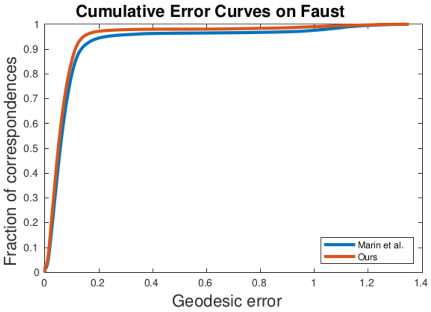
Sharma & Ovsjanikov (2020) This baseline, although weakly supervised, assumes to be given as input a mesh representation of a shape and LBO basis. We include it to demonstrate we can achieve competitive performance with the methods that excel in full shape matching where LBO basis are stable.

Marin et al. (2020) This is considered state of the art for learning embedding directly from data. Since we are testing with point clouds of much higher resolution compared to the experiments in Marin et al. (2020), we retrain their models with $3k$ subsampled points for each point cloud and show their results with the best performing resolution of $3k$. The rest of the parameters such as embedding size are used as specified in their paper as they were found to be optimal. We use their open source code and retrain it on our subset of Surreal dataset. Note that this baseline is somewhat different from others since it requires and thus, learns both basis functions and probe functions (feature descriptors).

Euc. Emb. This baseline 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. We denote our results with **Euc. Emb. + comm. + Lin. Inv.** in the following Table. We also show the corresponding curves below that are consistent with average geodesic error shown in the Table below.

Method \ Dataset	Faust
Marin et al. (2020)-3k	08
Marin et al. (2020) + sym.	09
Euc. Emb.	12
Euc. Emb. + comm.	10
Euc. Emb. + comm. + Lin. Inv. (Ours)	**05**
Sharma & Ovsjanikov (2020)	05



Discussion Our ablation study shows that all three loss functions improve the overall performance. We note that the performance gains brought in by linearly invariant loss on self-symmetry embeddings are significant. We also remark that 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 embedding is learned end to end in one phase. Second, none of the state-of-the-art methods takes symmetry structure 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 the size of embedding 20. In contrast, our method is quite robust and can train with twice their embedding size. We compete well with Sharma & Ovsjanikov (2020) even though this approach relies on LBO and exploiting mesh connectivity.

5.2 Partial Shape Matching

Datasets For a fair comparison with (Sharma & Ovsjanikov, 2020; Litany et al., 2017b), we follow the same experimental setup and test our method on the challenging SHREC'16 Partial Correspondence dataset (Cosmo et al., 2016). 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 (Sharma & Ovsjanikov, 2020). Overall, this test set contains 20 shapes each for cuts and holes datasets chosen randomly from the two sets respectively. In addition to Marin et al. (2020),

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

Method \ Dataset	Holes	Cuts
Litany et al. (2017b)	16	13
Sharma & Ovsjanikov (2020)	14	16
Marin et al. (2020)-3k	12	15
Euc. Emb.	18	20
Ours	**10**	**13**

we compare with the following two baselines:

Sharma & Ovsjanikov (2020). This baseline 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 these results by running the code provided by the authors.

Litany et al. (2017b). 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 shapes at test time. Thus, in terms of computation and ground truth map requirement, it is most expensive.

Results and Discussion We present our findings on partial shape matching in Table 1 where we obtain superior performance on both benchmark datasets for partial shape matching. In addition, our result outperforms the euclidean 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 & Ovsjanikov (2020) are class specific and need to be trained each time whereas our method is class agnostic and can obtain good results with a fraction of computational time. Similarly, Marin et al. (2020) trains a similar network as our 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 at least twice faster than this baseline in computational complexity.

Robustness to Embedding size Our method is not sensitive to small variations in the embedding size. We demonstrate this by changing the embedding size and plotting the average error on the FAUST test set. Note that in contrast, Marin et al. (2020) is extremely sensitive to changes in the size of feature descriptors (probe functions). We show the sensitivity in Figure 2.

Training Data Efficient We experiment with 3 training set of different sizes $1000, 3000, 5000$ sampled randomly and plot the results in Figure 3. We observe a drop in the performance of our method but the drop is slightly better than our baseline (Marin et al., 2020).

Figure 2:

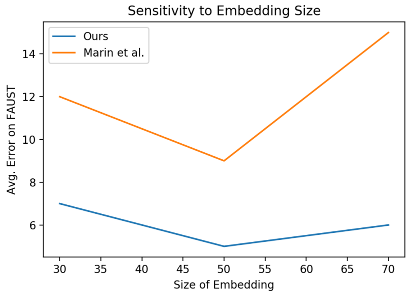
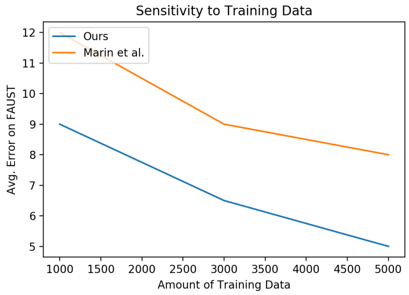


Figure 3:



6 Conclusion and Future Work

In shape correspondence literature, partial shape matching and complete shape matching are generally tackled by two different sets of methods which obtain impressive results in one of the two respective domains. 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 competitive performance on multiple shape matching benchmarks in comparison to all recent learning based methods.

There are several promising future directions to our work. First, our architecture is based on a very simple PointNet feature extractor and thus, there is a scope to integrate more advanced feature extractor that can exploit the neighbourhood structure better while learning the embedding. Second, following the advances in unsupervised deep functional maps, it would be interesting to explore learning of canonical embedding with minimal supervision.

7 Reproducibility

We have not included the code in the supplement due to the time constraints before the deadline to clean and anonymize it but we promise to upload it in the next chance to update this submission. We use publicly available datasets which can be downloaded by anyone.

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*, pp. 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*, pp. 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)*, pp. 230–246, 2018.

Oshri Halimi, Or Litany, Emanuele Rodol'a, 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, pp. 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*, pp. 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, pp. 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, pp. 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*, pp. 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*, pp. 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*, pp. 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*, pp. 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*, pp. 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 Poulenard, 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, pp. 222–236. Wiley Online Library, 2017.

Jean-Michel Roufosse, Abhishek Sharma, and Maks Ovsjanikov. Unsupervised deep learning for structured shape matching. In *ICCV*, pp. 1617–1627, 2019.

Yusuf Sahillioğlu. Recent advances in shape correspondence. *The Visual Computer*, pp. 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)*, pp. 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*, pp. 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.