Unsupervised Deep Multi-Shape Matching

Dongliang Cao[1,2] and Florian Bernard[1]

[1] University of Bonn, Germany
[2] Technical University of Munich, Germany

Abstract. 3D shape matching is a long-standing problem in computer vision and computer graphics. While deep neural networks were shown to lead to state-of-the-art results in shape matching, existing learning-based approaches are limited in the context of multi-shape matching: (i) either they focus on matching pairs of shapes only and thus suffer from cycle-inconsistent multi-matchings, or (ii) they require an explicit template shape to address the matching of a collection of shapes. In this paper, we present a novel approach for deep multi-shape matching that ensures cycle-consistent multi-matchings while not depending on an explicit template shape. To this end, we utilise a shape-to-universe multi-matching representation that we combine with powerful functional map regularisation, so that our multi-shape matching neural network can be trained in a fully unsupervised manner. While the functional map regularisation is only considered during training time, functional maps are not computed for predicting correspondences, thereby allowing for fast inference. We demonstrate that our method achieves state-of-the-art results on several challenging benchmark datasets, and, most remarkably, that our unsupervised method even outperforms recent supervised methods.

Keywords: 3D shape matching, multi-shape matching, functional maps, 3D deep learning

1 Introduction

The matching of 3D shapes is a long-standing problem in computer vision and computer graphics. Due to its wide range of applications, numerous approaches that address diverse variants of shape matching problems have been proposed over the past decades [46,44]. In recent years, with the success of deep learning, many learning-based methods were introduced for shape matching. One common way to address shape matching is to formulate it as classification problem [26,6,29,13,23,49]. The advantage of such methods is that after training the classifier, shape correspondences can efficiently and directly be predicted. A major downside is that for training such a classifier typically a large amount of data that is annotated with ground truth correspondences is required. However, specifically in the domain of 3D shapes, annotated data is scarce, since data annotation is particularly time-consuming and tedious. Thus, in practice, the above methods are often trained with small datasets, so that in turn they are prone to overfitting and lack the ability to generalise across datasets.
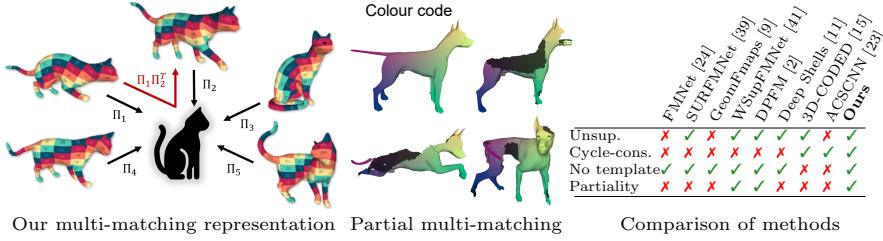
Fig. 1: **Left**: We present a novel unsupervised learning approach for cycle-consistent multi-shape matching based on matching each shape to a (virtual) universe shape. **Middle**: Our approach can successfully solve challenging *partial* multi-shape matching problems. **Right**: Our method is the first learning-based multi-shape matching approach that combines several favourable properties, i.e. it can be trained in an unsupervised manner, obtains cycle-consistent multi-matchings, does not require a template, and allows for partial matchings.

Another line of learning-based shape matching solutions build upon the functional map framework [24,16,39,9,41]. Functional maps can serve as a powerful regularisation, so that respective methods were even trained successfully in an unsupervised manner, i.e. without the availability of ground truth correspondences [39,41,2]. However, a downside of such approaches is that the conversion of the obtained functional map to a point-wise correspondence map is typically non-trivial. On the one hand, this may limit the matching accuracy, while on the other hand this may have a negative impact on inference speed (see Sec. 5.1).

In this work we aim to alleviate the shortcomings of both paradigms, while combining their advantages. To this end, we present a novel unsupervised learning approach for obtaining cycle-consistent multi-shape matchings, see Fig. 1. Our approach predicts shape-to-universe matchings based on a *universe classifier*, so that cycle-consistency of so-obtained pairwise matchings is guaranteed by construction. Unlike previous classification methods that rely on supervision based on ground truth correspondences, the training of our universe classifier purely relies on functional map regularisation, thereby allowing for a fully unsupervised training. Yet, at inference time, our method does not require to compute functional maps, and directly predicts shape-to-universe matchings via our classifier. We summarise our main contributions as follows:

- For the first time we enable the *unsupervised training* of a classification-based neural network for *cycle-consistent 3D multi-shape matching*.
- To this end, our method uses *functional maps as strong regularisation during training* but does *not require the computation of functional maps during inference*.
- Our method achieves *state-of-the-art results* on several challenging 3D shape matching benchmark datasets, even in comparison to most recent supervised methods.

2   Related work

Shape matching is a well-studied problem in computer vision and graphics [46,44]. Rather than providing an exhaustive literature survey, in the following we will focus on reviewing those methods that we consider most relevant to our work. First, we will provide an overview of works that rely on the functional map framework for 3D shape matching. Subsequently, we will focus on recent learning-based approaches that utilise functional maps. Afterwards, we will briefly discuss other learning-based methods. Eventually, we will discuss methods that are specifically tailored for the case of multi-shape matching, as opposed to the more commonly studied case of two-shape matching.

Functional maps. The functional map framework [31] enables to formulate the correspondence problem in the functional domain by computing functional maps instead of point-wise maps. The key advantage of functional maps is that they allow for a low-dimensional representation of shape correspondences, i.e. small matrices that encode how functions can be transferred between two domains (3D shapes). Unlike finding point-to-point correspondences, which can for example be phrased as the NP-hard quadratic assignment problem [22], functional maps can efficiently be obtained by solving a linear least-squares problem. The functional map framework has been extensively studied and was extended in numerous works, e.g. in terms of improving the accuracy or robustness [36], as well as extending its application to partial shape matching [38], non-isometric shape matching [30,37] and multi-shape matching [18,19]. Nevertheless, in these approaches functional maps are typically utilised in an axiomatic manner, meaning that they heavily rely on handcrafted feature descriptors, such as HKS [7], WKS [3] or SHOT [40], which potentially limits their performance.

Learning methods based on functional maps. In contrast to axiomatic approaches that use handcrafted features, a variety of methods have been proposed to learn the feature descriptors directly from data. Starting from [24], the (supervised) FMNet was proposed to learn a non-linear transformation of SHOT descriptors [40]. Later work [16] modified the loss to enable FMNet training in an unsupervised manner. However, both methods compute a loss that relies on geodesic distances, which is computationally expensive, particularly for high-resolution shapes. SURFMNet [39] proposed an unsupervised loss based on functional map regularisation, which, however, does not directly obtain point-to-point correspondences. More recently, several works [9,41] replaced FMNet with point-based networks [35,45] to achieve better performance. However, such point-based networks cannot utilise the connectivity information that exists in triangle meshes. To make use of it, DiffusionNet [42] introduced a diffusion layer, which was shown to achieve state-of-the-art performance for 3D shape matching. Most recently, DPFM [2] extended DiffusionNet with a cross-attention refinement mechanism [47] for partial shape matching. DPFM addresses two variants of partial matching problems: for partial-to-partial matching it relies on a supervised training strategy, and for partial-to-complete matching it can be trained in an unsupervised manner based on functional map regularisation [38,25]. While DPFM predicts functional maps and thereby requires a post-processing to ob-

tain point-wise maps, our proposed approach directly predicts point-wise maps without the need of computing functional maps during inference.

Other learning-based methods. Despite the success of functional maps for diverse learning-based shape matching approaches, there is a wide variety of other learning-based methods. Many of the recent works formulate shape matching as a classification problem [26,6,29,13,23,49]. In these methods, special network architectures were proposed to extend and generalise convolutions from Euclidean grids to surfaces. However, these methods require ground truth correspondences as supervision for training the classifier. 3D-CODED [15] factorised a given input shape into a template shape and a learned global feature vector that encodes the deformation from the template shape to the input shape. In this way, it finds correspondences between input shapes and the template shape. In contrast, our method does not require to choose a template shape for matching. Deep shells [11] proposed a coarse-to-fine matching pipeline that utilises an iterative alignment of smooth shells [10]. While this iterative process may be time-consuming, our method directly predicts shape correspondences in one shot.

Multi-shape matching. There are several works that explicitly consider the matching of a collection of shapes, i.e. the so-called *multi-shape matching problem*. The key aspect in which multi-shape matching differs from the more commonly studied two-shape matching problem is that in the former one needs to ensure cycle-consistency between pairwise matchings across a collection of data. This is often achieved by first solving a quadratic number of pairwise matching problems (e.g. between all pairs of shapes), and subsequently establishing cycle consistency as a post-processing, e.g. based on permutation synchronisation [32,17]. The higher-order projected power iteration (HiPPI) method [4] circumvented this two-stage approach by generalising permutation synchronisation to explicitly consider geometric relations. In the context of functional maps, Consistent ZoomOut [19] extended ZoomOut [27] by adding functional map consistency constraints. IsoMuSh [14] simultaneously optimises for functional maps and point-wise maps that are both multi-shape consistent. In contrast to these axiomatic methods, which usually downsample the original shape for matching, our method can be directly applied on shapes up to 10,000 vertices. Several learning-based approaches, such as 3D-CODED [15], HSN [49] or ACSCNN [23], utilised an explicit template shape in order to ensure cycle-consistent multimatchings. However, the use of an explicit template shape poses severe limitations in practice, since a suitable template shape needs to be available, and the specific choice of template also induces a bias. In stark contrast, our method does not rely on an explicit template shape, thereby effectively alleviating such a bias while at the same time providing substantially more flexibility.

3   Background

Our approach is based on the functional map framework and aims for a cycle-consistent multi-shape matching. For completeness, we first recap the basic

pipeline for functional map computation and the desirable properties of functional maps. Then, we introduce the notion of cycle consistency for multi-shape matching.

3.1 Functional maps

Basic pipeline. Given is a pair of 3D shapes $\mathcal{X}$ and $\mathcal{Y}$ that are represented as triangle meshes with n_x and n_y vertices, respectively. The basic pipeline for computing a functional map between both shapes mainly consists of the following steps:

- Compute the first k eigenfunctions $\Phi_x \in \mathbb{R}^{n_x \times k}, \Phi_y \in \mathbb{R}^{n_y \times k}$ of the respective Laplacian matrix [34] as the basis functions.
- Compute feature descriptors $\mathcal{F}_x \in \mathbb{R}^{n_x \times c}, \mathcal{F}_y \in \mathbb{R}^{n_y \times c}$ on each shape, and (approximately) represent them in the (reduced) basis of the respective eigenfunctions, i.e. $A_x = \Phi_x^\dagger \mathcal{F}_x, A_y = \Phi_y^\dagger \mathcal{F}_y$.
- Compute the optimal functional map $C_{xy} \in \mathbb{R}^{k \times k}$ by solving the optimisation problem

$$C_{xy} = \arg\min_C \mathcal{L}_{\text{data}}(C) + \lambda \mathcal{L}_{\text{reg}}(C), \tag{1}$$

  where the data term $\mathcal{L}_{\text{data}}$ ensures that C maps between the feature descriptors represented in the reduced basis, and the regularisation term $\mathcal{L}_{\text{reg}}$ penalises the map by its structural properties (as explained below).
- Convert the functional map C_{xy} to a point map $\Pi_{yx} \in \{0,1\}^{n_y \times n_x}$, e.g. using nearest neighbour search or other post-processing techniques [27,33,48] based on the relationship

$$\Phi_y C_{xy} \approx \Pi_{yx} \Phi_x. \tag{2}$$

Structural properties. In the context of near-isometric shape pairs, functional maps have the following properties [39,41]:

- **Bijectivity.** Given functional maps in both directions C_{xy}, C_{yx}, bijectivity requires the map from $\mathcal{X}$ through $\mathcal{Y}$ to $\mathcal{X}$ to be the identity. The requirement can be formulated as the difference between their composition and the identity map [12]. Thus, the bijectivity regularisation for functional maps can be expressed in the form

$$\mathcal{L}_{\text{bij}} = \|C_{xy}C_{yx} - I\|_F^2 + \|C_{yx}C_{xy} - I\|_F^2. \tag{3}$$

- **Orthogonality.** A point map is locally area-preserving if and only if the associated functional map is an orthogonal matrix [31]. The orthogonality regularisation for functional maps can be expressed in the form

$$\mathcal{L}_{\text{orth}} = \|C_{xy}^\top C_{xy} - I\|_F^2 + \|C_{yx}^\top C_{yx} - I\|_F^2. \tag{4}$$

- **Laplacian commutativity.** A point map is an intrinsic isometry if and only if the associated functional map commutes with the Laplace-Beltrami operator [31]. The Laplacian commutativity regularisation for functional maps can expressed in the form

$$\mathcal{L}_{\text{lap}} = \|C_{xy}\Lambda_x - \Lambda_y C_{xy}\|_F^2 + \|C_{yx}\Lambda_y - \Lambda_x C_{yx}\|_F^2, \tag{5}$$

where Λ_x and Λ_y are diagonal matrices of the Laplace-Beltrami eigenvalues on the respective shapes.

3.2   Multi-shape matching

Given is a collection 3D shapes $\mathcal{S}$. For any pair $\mathcal{X}, \mathcal{Y} \in \mathcal{S}$ with n_x, n_y vertices, respectively, the point map Π_{xy} between them can be expressed in the form

$$\Pi_{xy} \in \left\{ \Pi \in \{0,1\}^{n_x \times n_y} : \Pi \mathbf{1}_{n_y} \leq \mathbf{1}_{n_x}, \mathbf{1}_{n_x}^{\top} \Pi \leq \mathbf{1}_{n_y}^{\top} \right\}, \tag{6}$$

where $\Pi_{xy}(i,j) = 1$ can be interpreted as the i-th vertex in shape $\mathcal{X}$ corresponding to the j-th vertex in shape $\mathcal{Y}$.

Cycle-consistency is a desirable property between pairwise matchings in a collection, as it must hold for the true matchings. Cycle consistency means that for any given triplet $\mathcal{X}, \mathcal{Y}, \mathcal{Z} \in \mathcal{S}$, the matching composition from $\mathcal{X}$ through $\mathcal{Y}$ to $\mathcal{Z}$ should be identical to the direct matching from $\mathcal{X}$ to $\mathcal{Z}$, i.e.

$$\Pi_{xz} = \Pi_{xy} \Pi_{yz}. \tag{7}$$

We note that if cycle consistency holds for all triplets in $\mathcal{S}$, it also holds for any higher-order tuples of matching compositions, since the latter can be constructed by composing triplet matching compositions.

Alternatively, one can use a shape-to-universe matching representation [32,17] to avoid explicitly modelling the (non-convex) cycle-consistency constraint in Eq. (7). This idea builds upon a *virtual* universe shape, which can be thought of a shape that is never explicitly instantiated, i.e. as opposed to a template shape we do not require a 3D mesh of the universe shape. Instead, we merely assume that there is such a shape, so that for all points of the shapes in $\mathcal{S}$ there exists a corresponding (virtual) universe point. We denote the number of universe points as d. For $\Pi_x \in \{0,1\}^{n_x \times d}$ being the matching from shape $\mathcal{X}$ to the universe shape, and $\Pi_y^{\top} \in \{0,1\}^{d \times n_y}$ being the matching from the universe shape to shape $\mathcal{Y}$, this shape-to-universe representation allows to compute pairwise matchings as

$$\Pi_{xy} = \Pi_x \Pi_y^{\top}. \tag{8}$$

4   Our unsupervised multi-shape matching method

Our novel unsupervised learning approach for cycle-consistent multi-shape matching is illustrated in Fig. 2. Conceptually, our pipeline comprises of two main components that are trained in an end-to-end manner.

Analogous to other learning-based approaches [39,41], the first main component (blue in Fig. 2) performs feature extraction. Given the source and target shapes $\mathcal{X}$ and $\mathcal{Y}$, a Siamese feature extraction network with (shared) trainable weights Θ extracts features $\mathcal{F}_x$ and $\mathcal{F}_y$ from the input shapes, respectively. Then, a (non-trainable but differentiable) FM solver (yellow in Fig. 2) is applied
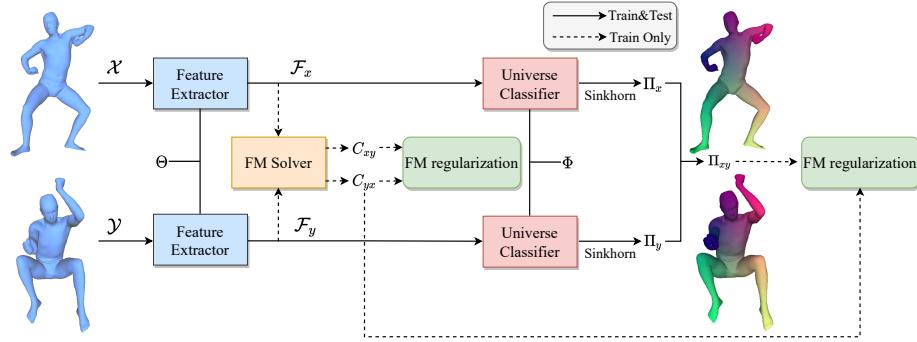
Fig. 2: Overview of our unsupervised learning approach for cycle-consistent multi-shape matching. First, features $\mathcal{F}_x$ and $\mathcal{F}_y$ are extracted from the input shapes $\mathcal{X}$ and $\mathcal{Y}$. The feature descriptors are then used to compute the bidirectional functional maps C_{xy} and C_{yx} based on the FM solver. Subsequently, our novel universe classifier takes the feature descriptors as input and predicts a shape-to-universe matching (Π_x and Π_y) for each shape. The pairwise matching Π_{xy} is obtained by the composition of shape-to-universe point maps $\Pi_x\Pi_y^{\top}$. During training (solid and dashed lines) we utilise functional maps (FM) as regularisation, whereas at inference time no FM have to be computed (solid lines).

to compute the bidirectional functional maps C_{xy} and C_{yx}. The second main component (red in Fig. 2) is a Siamese universe classifier with shared weights Φ. It takes features from the first part as input to predict the shape-to-universe matchings Π_x and Π_y for each shape. The pairwise matching Π_{xy} is based on their composition, see Eq. (8). To allow for an unsupervised end-to-end training of our architecture, we build upon functional map regularisation (green in Fig. 2) described in Sec. 3.1. In the following we explain the individual parts in detail.

4.1   Feature extractor

The goal of the feature extraction module is to compute feature descriptors of 3D shapes that are suitable both for functional map computation and shape-to-universe matching prediction. Our feature extraction is applied in a Siamese manner, i.e. the identical network is used for both $\mathcal{X}$ and $\mathcal{Y}$. The outputs of this module are point-wise feature descriptors for each shape, which we denote as $\mathcal{F}_x$ and $\mathcal{F}_y$ respectively.

4.2   Functional map solver

The FM solver aims to compute the bidirectional functional maps C_{xy} and C_{yx} based on the extracted feature descriptors $\mathcal{F}_x$ and $\mathcal{F}_y$ (see Sec. 3.1). We use a regularised formulation to improve the robustness when computing the optimal

functional map [2], i.e. we consider

$$C_{xy} = \arg\min_{C} \|CA_x - A_y\|_F^2 + \lambda \sum_{ij} C_{ij}^2 M_{ij}, \qquad (9)$$

where

$$M_{ij} = \left(\frac{\Lambda_y(i)^\gamma}{\Lambda_y(i)^{2\gamma} + 1} - \frac{\Lambda_x(j)^\gamma}{\Lambda_x(j)^{2\gamma} + 1} \right)^2 + \left(\frac{1}{\Lambda_y(i)^{2\gamma} + 1} - \frac{1}{\Lambda_x(j)^{2\gamma} + 1} \right)^2. \quad (10)$$

The regulariser can be viewed as an extension of Laplacian commutativity defined in Eq. (5) with well-justified theoretical foundation, see [36] for details.

4.3   Universe classifier

The goal of the universe classifier module is to utilise the extracted feature descriptors $\mathcal{F}_x$ and $\mathcal{F}_y$ in order to predict the point-to-universe maps Π_x and Π_y, respectively. Similarly to our feature extractor, the universe classifier is applied in a Siamese way. The output dimension of our universe classifier is equal to the number of points d in the universe shape (see supp. mat. for details how it is chosen).

For our shape-to-universe matching representation, each point of a shape needs to be classified into exactly one universe class, and a universe class cannot be chosen multiple times (per shape). Mathematically, this means that the point-to-universe map Π_x must be a partial permutation matrix as defined in Eq. (6), where in addition the rows of Π_x sum up to one, i.e.

$$\Pi_x \in \left\{ \Pi \in \{0,1\}^{n_x \times d} : \Pi \mathbf{1}_d = \mathbf{1}_{n_x}, \mathbf{1}_{n_x}^\top \Pi \leq \mathbf{1}_d^\top \right\}. \qquad (11)$$

We approximate these combinatorial constraints in terms of Sinkhorn normalisation [43,28] to make the prediction differentiable. Sinkhorn normalisation iteratively normalises rows and columns of a matrix based on the softmax operator and a temperature parameter τ (see [28]).

4.4   Unsupervised loss

Our unsupervised loss is composed of two main parts:

FM regularisation for feature extractor. Following [39,41], we use functional map regularisation to compute unsupervised losses for optimised bidirectional functional maps C_{xy} and C_{yx}. Specifically, we use the bijectivity loss $\mathcal{L}_{\text{bij}}$ in Eq. (3), the orthogonality loss $\mathcal{L}_{\text{orth}}$ in Eq. (4), and the Laplacian commutativity loss $\mathcal{L}_{\text{lap}}$ in Eq. (5) to regularise the functional maps. As such, the total loss for training the feature extractor can be expressed in the form

$$\mathcal{L}_{\text{ft}} = w_{\text{bij}}\mathcal{L}_{\text{bij}} + w_{\text{orth}}\mathcal{L}_{\text{orth}} + w_{\text{lap}}\mathcal{L}_{\text{lap}}. \qquad (12)$$

In case of partial shape matching, the functional map from the complete shape to the partial shape becomes a slanted diagonal matrix [38]. We follow DPFM [2] to

regularise the predicted functional maps based on this observation. For $\mathcal{X}$ being the complete shape and $\mathcal{Y}$ being the partial shape, in the partial matching case the loss terms can be expressed as

$$\mathcal{L}_{\mathrm{bij}} = \|C_{xy}C_{yx} - \mathbf{I}_r\|_F^2, \text{ and } \mathcal{L}_{\mathrm{orth}} = \|C_{xy}C_{xy}^\top - \mathbf{I}_r\|_F^2, \tag{13}$$

where $\mathbf{I}_r$ is a diagonal matrix in which the first r elements on the diagonal are equal to 1, and r is the slope of the functional map estimated by the area ratio between two shapes, i.e. $r = \max\left\{i \mid \Lambda_y^i < \max\left(\Lambda_x\right)\right\}$.

FM regularisation for universe classifier. To train our universe classifier in an unsupervised manner, we build upon the relationship $\Phi_x C_{yx} \approx \Pi_{xy}\Phi_y$ between the functional map C_{yx} and the point-wise map Π_{xy}, as explained in Eq. (2). In our case, the pairwise point map Π_{xy} is obtained from the composition of the shape-to-universe point maps $\Pi_{xy} = \Pi_x \Pi_y^\top$. With that, the unsupervised loss for training the universe classifier can be expressed in the form

$$\mathcal{L}_{\mathrm{cls}} = \|\Phi_x C_{yx} - \Pi_{xy}\Phi_y\|_F^2. \tag{14}$$

This loss is differentiable with respect to both the functional map C_{yx} and the point map Π_{xy}, so that we can train our network in an end-to-end manner. By doing so, the predicted functional map and the predicted point map are improved during training. Overall, we demonstrate that we are able to achieve better matching results even in comparison a network trained with ground truth correspondences (see Sec. 5.3).

The total loss combines the loss terms of the feature extractor and the universe classifier and has the form

$$\mathcal{L}_{\mathrm{total}} = \mathcal{L}_{\mathrm{ft}} + \lambda_{\mathrm{cls}}\mathcal{L}_{\mathrm{cls}}. \tag{15}$$

4.5   Implementation details

We implemented our method in PyTorch. Our feature extractor takes SHOT descriptors [40] as inputs. We use DiffusionNet [42] as the network architecture for both our feature extractor and universe classifier. In terms of training, we use the Adam optimiser [21] with learning rate $1e^{-3}$ in all experiments. More details are provided in the supp. mat.

5   Experimental results

For our experimental evaluation, we consider complete shape matching, partial shape matching, as well as an ablation study that analyses the importance of individual components of our method. We note that in all experiments we train a single network for all shapes in a dataset (as opposed to a per-shape category network).

Geodesic error (×100)	FAUST	SCAPE	F on S	S on F	△
Axiomatic Methods					
BCICP [37]	6.4	11	-	-	✗
ZOOMOUT [27]	6.1	7.5	-	-	✗
Smooth Shells [10]	2.5	4.7	-	-	✗
Supervised Methods					
FMNet [24]	11	17	30	33	✗
+ *pmf*	5.9	6.3	11	14	✗
3D-CODED [15]	2.5	31	31	33	✓
GeomFmaps-KPC [9]	3.1	4.4	11	6.0	✗
+ *zo*	1.9	3.0	9.2	4.3	✗
GeomFmaps-DFN [42]	2.6	3.0	3.3	3.0	✗
+ *zo*	1.9	2.4	2.4	1.9	✗
HSN [49]	3.3	3.5	25.4	16.7	✓
ACSCNN [23]	2.7	3.2	8.4	6.0	✓
Unsupervised Methods					
SURFMNet [39]	15	12	32	32	✗
+ *icp*	7.4	6.1	19	23	✗
Unsup FMNet [16]	10	16	29	22	✗
+ *pmf*	5.7	10	12	9.3	✗
WSupFMNet [41]	3.3	7.3	11.7	6.2	✗
+ *zo*	1.9	4.9	8.0	4.3	✗
Deep Shells [11]	1.7	2.5	5.4	**2.7**	✗
Ours (*fine-tune*)	**1.5**	**2.0**	7.3 (**3.2**)	8.6 (3.2)	✓

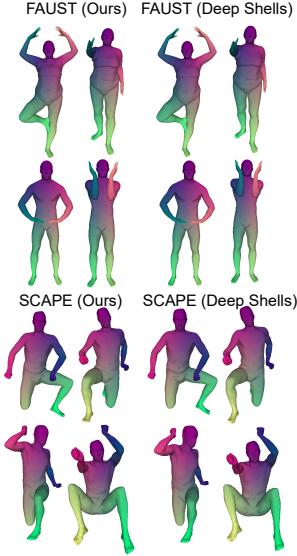FAUST (Ours) FAUST (Deep Shells)

SCAPE (Ours) SCAPE (Deep Shells)

Fig. 3: **Left:** Quantitative results on the FAUST and SCAPE datasets in terms of mean geodesic errors (×100). 'F on S' stands for training on FAUST and testing on SCAPE datasets ('S on F' analogously). The rows in grey show refined results using the indicated post-processing procedure. Ours is the only unsupervised method that obtains cycle-consistent (△) multi-matchings. **Right:** Qualitative multi-matching results using our method and Deep Shells. Although qualitatively both methods perform similarly, ours directly predicts shape correspondences without iterative refinement, thereby leading to a faster inference, cf. Fig. 5.

5.1   Complete shape matching

Datasets. To be consistent and comparable with prior works, we evaluate our method on two standard benchmarks, FAUST [5] and SCAPE [1] (for both, we use the more challenging remeshed versions from [37]). The FAUST dataset contains 100 shapes consisting of 10 people, each in 10 poses. The SCAPE dataset comprises 71 different poses of the same person. Consistent with previous works, we split both datasets into training sets with 80 and 51 shapes, respectively, and test sets with 20 shapes.

Quantitative results. For the evaluation we use the mean geodesic error defined in the Princeton benchmark protocol [20]. We compare our method with state-of-the-art axiomatic, supervised and unsupervised methods, as shown in Fig. 3. Our method outperforms the previous state-of-the-art in most settings, even in comparison to the supervised methods. The last two columns in the table shown in Fig. 3 (left) show generalisation results. Our method generalises better compared to previous unsupervised methods based on functional map regularisation [16,39,41]. In comparison to Deep Shells [11], our method shows comparative results after fine-tuning our pipeline with the loss in Eq. (15) for each shape pair
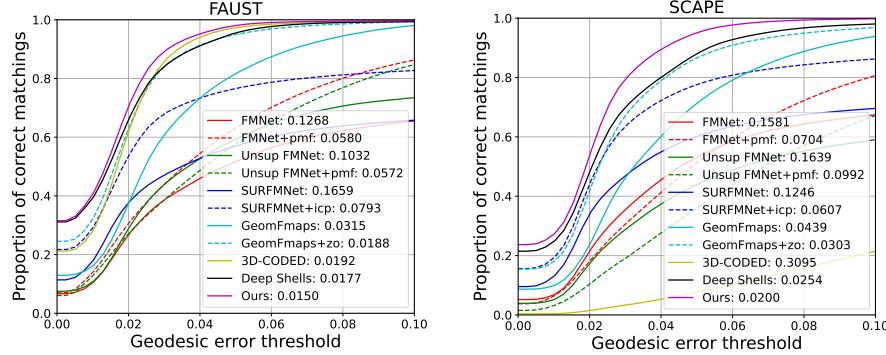
Fig. 4: PCK curves for the FAUST and SCAPE dataset. Dashed lines indicate methods with refinement. Our method achieves the best scores on both datasets.

individually (see details on fine-tuning in supp. mat.). We plot the percentage of correct keypoints (PCK) curves for both datasets in Fig. 4, where it can be seen that our method achieves the best results in comparison to a wide range of methods. Moreover, our method does not require post-processing techniques, such as ZOOMOUT [27] or PMF [48], which are often time-consuming, see Fig. 5 for runtime comparisons.

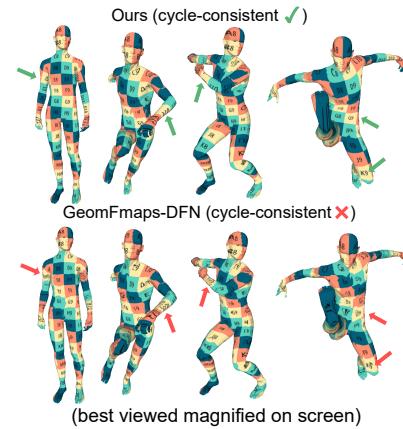| Runtime (s)	Inference	Refine	Total
Axiomatic Methods			
BCICP [37]	881.	-	881.
ZoomOut [27]	43.	-	43.
Smooth Shells [10]	125.	-	125.
Supervised Methods			
FMNet [24]	5.1	223.	228.
3D-CODED [15]	725.	-	725.
GeomFmaps-KPC [9]	1.9	35.	37.
GeomFmaps-DFN [42]	1.5	35.	37.
Unsupervised Methods			
SURFMNet [39]	5.7	43.	49.
Unsup FMNet [16]	5.1	216.	221.
WSupFMNet [41]	2.1	35.	37.
Deep Shells [11]	14.	-	14.
Ours (*fine-tune*)	**0.6** (5.0)	-	**0.6** (5.0)

Fig. 5: **Left:** Runtimes for the matchings in the experiments of Fig. 3. **Right:** Texture transfer using our method and the supervised GeomFmaps-DFN (erroneous matchings indicated by red arrows). Cycle consistency ensures that our method consistently transfers textures.
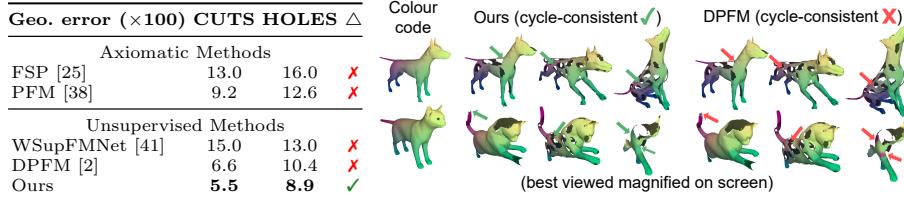
Geo. error (×100) CUTS HOLES △			
Axiomatic Methods			
FSP [25]	13.0	16.0	✗
PFM [38]	9.2	12.6	✗
Unsupervised Methods			
WSupFMNet [41]	15.0	13.0	✗
DPFM [2]	6.6	10.4	✗
Ours	**5.5**	**8.9**	✓

Fig. 6: **Left:** Quantitative results on the CUTS and HOLES subsets of the SHREC'16 Partial Correspondence Benchmark. Reported are mean geodesic errors (×100). Ours is the only method that obtains cycle-consistent (△) multi-matchings. **Right:** Qualitative multi-matching results using our method and DPFM on the HOLES subset (erroneous matchings indicated by red arrows).

5.2   Partial shape matching

Datasets. We evaluate our method in the setting of partial shape matching on the challenging SHREC'16 Partial Correspondence Benchmark [8]. This dataset consists of 200 shapes with 8 categories (humans and animals). The dataset is divided into two subsets, namely CUTS (removal of a large part) with 120 pairs and HOLES (removal of many small parts) with 80 pairs. We follow the training procedure as in WSupFMNet [41].

Quantitative results. We compare our method with two axiomatic methods FPS [25], PFM [38] and two unsupervised methods WSupFMNet [41], unsupervised DPFM [2]. In Fig. 6 we show quantitative results (in terms of the mean geodesic error) and qualitative results. Our method outperforms previous axiomatic and unsupervised methods. The major difference in comparison to the unsupervised DPFM that predicts partial-to-complete functional maps, is that our method directly predicts shape-to-universe correspondences based on our universe classifier, so that ours leads to cycle-consistent multi-matchings.

5.3   Ablation study

The goal of this section is to evaluate the importance of the individual components of our approach, including the universe classifier and our unsupervised loss. For all ablation experiments, we consider the same experimental protocol as in Sec. 5.1. We summarise the results for all ablative experiments and our full method in Fig. 7. We study a total of three different ablative settings that consider the removal of the universe classifier, or the modification of the loss. In the following we explain these in detail.

Classifier-free. We remove the universe classifier in our pipeline and only train the feature extractor with the unsupervised loss $\mathcal{L}_{ft}$ defined in Eq. (12). At test time, we convert the optimised functional maps to point maps using nearest neighbour search. In comparison to our complete method, the matching performance drops substantially, especially in the last two columns of Fig. 7, indicating that this results in poor generalisation ability across datasets. Overall,

this implies that our universe classifier is able to predict more accurate point maps compared to point maps converted from functional maps.

Feature similarity. In the previous ablative experiment, we modify our network architecture by removing the classifier and change our unsupervised loss at the same time. Here, we focus on the universe classifier only, i.e. we remove it from our pipeline, while keeping our unsupervised loss. To this end, we construct a soft pairwise point map based on the feature similarity between two shapes with the help of Sinkhorn normalisation, similar to Deep Shells [11]. By doing so, the pairwise point map can be expressed in the form $\Pi_{xy}(i,j) \propto \exp\left(-\frac{1}{\lambda}\|\mathcal{F}_x(i) - \mathcal{F}_y(j)\|_2^2\right)$. In comparison to classifier-free experiment, we observe that point maps based on such a feature similarity have similar performance on the intra-dataset experiments, while it significantly improves the generalisation ability across datasets. However, there is still a performance gap compared with our complete method.

Supervised training. The goal of this ablative experiment is to show the superiority of our unsupervised loss based on functional map regularisation compared to a supervised classification loss. In this experiment, we use the same network architecture as for our complete method. However, we replace our unsupervised loss defined in Eq. (15) by a cross entropy loss between the predicted correspondences and the ground truth correspondences. In comparison to our complete method, we observe that the supervised alternative achieves better performance on FAUST dataset, but leads to a worse performance on other datasets, especially for the generalisation cases. We believe that the main reason is that the supervised approach is overfitting to the training data, but lacks the ability to generalise across datasets.

Geo. error (×100)	FAUST	SCAPE	F on S	S on F
Classifier-free	2.1	3.8	17.4	22.9
Feat. similarity	2.1	3.7	10.6	13.9
Supervised	**1.4**	2.8	9.8	18.5
Ours	1.5	**2.0**	**7.3**	**8.6**

Fig. 7: **Left:** Quantitative results of our ablation study on the FAUST and SCAPE datasets. **Right:** Qualitative results for the considered ablative experiments on the SCAPE dataset (erroneous matchings indicated by red arrows).

6 Limitations and future work

Our work is the first unsupervised learning-based approach for finding cycle-consistent matchings between multiple 3D shapes, and additionally pushes the current state of the art in multi-shape matching. Yet, there are also some limitations that give rise to interesting future research questions.

For our approach it is not necessary that the universe shape is explicitly instantiated (e.g. in the form of 3D mesh). However, we require to fix the maximum number of points d of this (virtual) universe shape for training, since d corresponds to the number of output classes that are predicted by our classifier. With that, a universe classifier trained with a fixed number of d classes is not able to predict (unique) correspondences of shapes with more than d vertices. The exploration of alternative formalisms that do not require to fix d as a-priori is an interesting future direction.

Our universe classifier can be trained in an unsupervised manner both with datasets comprising of complete shapes only, as well as with mixed datasets (comprising of partial shapes and at least a single complete shape of the same category). As such, our current neural network does not allow for the unsupervised training with datasets that contain only partially observed shapes. This is due to limitations for partial-to-partial matchings that our approach inherits from the functional map framework. We note that DPFM [2] also shares this limitation – in their case, the authors utilise a *supervised* training strategy when considering the partial-to-partial matching case. Since DPFM uses different network architectures for partial-to-complete and partial-to-partial settings, it cannot be applied to different settings during training and inference. For example, it cannot be trained using partial-to-complete shape pairs and then be applied to predict partial-to-partial correspondences. In contrast, our method is more flexible, as we use a single neural network architecture based on our universe classifier, where the shape-to-universe matching formalism naturally allows to represent complete, partial-to-complete as well as partial-to-partial matchings.

7   Conclusion

We introduce the first approach for the unsupervised training of a deep neural network for predicting cycle-consistent matchings across a collection of 3D shapes. Our approach builds upon the powerful functional map framework in order to allow for an unsupervised training. Yet, during inference we directly predict point-wise matchings and do not require to compute functional maps, which has a positive impact on the runtime. The major strength of our approach is that it combines a unique set of favourable properties: our approach can be trained in an unsupervised manner, obtains cycle-consistent multi-matchings, does not rely on a template shape, and can handle partial matchings. Overall, due to the conceptual novelties and the demonstrated state-of-the-art performance on diverse shape matching benchmarks, we believe that our work advances the field of 3D shape matching.

References

1. Anguelov, D., Srinivasan, P., Koller, D., Thrun, S., Rodgers, J., Davis, J.: Scape: shape completion and animation of people. In: ACM SIGGRAPH (2005)
2. Attaiki, S., Pai, G., Ovsjanikov, M.: Dpfm: Deep partial functional maps. In: International Conference on 3D Vision (3DV) (2021)
3. Aubry, M., Schlickewei, U., Cremers, D.: The wave kernel signature: A quantum mechanical approach to shape analysis. In: ICCV (2011)
4. Bernard, F., Thunberg, J., Swoboda, P., Theobalt, C.: Hippi: Higher-order projected power iterations for scalable multi-matching. In: ICCV (2019)
5. Bogo, F., Romero, J., Loper, M., Black, M.J.: Faust: Dataset and evaluation for 3d mesh registration. In: CVPR (2014)
6. Boscaini, D., Masci, J., Rodolà, E., Bronstein, M.: Learning shape correspondence with anisotropic convolutional neural networks. NIPS (2016)
7. Bronstein, M.M., Kokkinos, I.: Scale-invariant heat kernel signatures for non-rigid shape recognition. In: CVPR (2010)
8. Cosmo, L., Rodola, E., Bronstein, M.M., Torsello, A., Cremers, D., Sahillioglu, Y.: Shrec'16: Partial matching of deformable shapes. Proc. 3DOR **2**(9), 12 (2016)
9. Donati, N., Sharma, A., Ovsjanikov, M.: Deep geometric functional maps: Robust feature learning for shape correspondence. In: CVPR (2020)
10. Eisenberger, M., Lahner, Z., Cremers, D.: Smooth shells: Multi-scale shape registration with functional maps. In: CVPR (2020)
11. Eisenberger, M., Toker, A., Leal-Taixé, L., Cremers, D.: Deep shells: Unsupervised shape correspondence with optimal transport. NIPS (2020)
12. Eynard, D., Rodola, E., Glashoff, K., Bronstein, M.M.: Coupled functional maps. In: International Conference on 3D Vision (3DV) (2016)
13. Fey, M., Lenssen, J.E., Weichert, F., Müller, H.: Splinecnn: Fast geometric deep learning with continuous b-spline kernels. In: CVPR (2018)
14. Gao, M., Lahner, Z., Thunberg, J., Cremers, D., Bernard, F.: Isometric multi-shape matching. In: CVPR (2021)
15. Groueix, T., Fisher, M., Kim, V.G., Russell, B.C., Aubry, M.: 3d-coded: 3d correspondences by deep deformation. In: ECCV (2018)
16. Halimi, O., Litany, O., Rodola, E., Bronstein, A.M., Kimmel, R.: Unsupervised learning of dense shape correspondence. In: CVPR (2019)
17. Huang, Q.X., Guibas, L.: Consistent shape maps via semidefinite programming. In: Computer Graphics Forum. vol. 32, pp. 177–186 (2013)
18. Huang, Q., Wang, F., Guibas, L.: Functional map networks for analyzing and exploring large shape collections. ACM Transactions on Graphics (ToG) **33**(4), 1–11 (2014)
19. Huang, R., Ren, J., Wonka, P., Ovsjanikov, M.: Consistent zoomout: Efficient spectral map synchronization. In: Computer Graphics Forum. vol. 39, pp. 265–278. Wiley Online Library (2020)
20. Kim, V.G., Lipman, Y., Funkhouser, T.: Blended intrinsic maps. ACM transactions on graphics (TOG) **30**(4), 1–12 (2011)
21. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: ICLR (2015)
22. Lawler, E.L.: The quadratic assignment problem. Management science **9**(4), 586–599 (1963)
23. Li, Q., Liu, S., Hu, L., Liu, X.: Shape correspondence using anisotropic chebyshev spectral cnns. In: CVPR (2020)

24. Litany, O., Remez, T., Rodola, E., Bronstein, A., Bronstein, M.: Deep functional maps: Structured prediction for dense shape correspondence. In: ICCV (2017)
25. Litany, O., Rodolà, E., Bronstein, A.M., Bronstein, M.M.: Fully spectral partial shape matching. In: Computer Graphics Forum. vol. 36, pp. 247–258. Wiley Online Library (2017)
26. Masci, J., Boscaini, D., Bronstein, M., Vandergheynst, P.: Geodesic convolutional neural networks on riemannian manifolds. In: ICCV (2015)
27. Melzi, S., Ren, J., Rodola, E., Sharma, A., Wonka, P., Ovsjanikov, M.: Zoomout: Spectral upsampling for efficient shape correspondence. arXiv preprint arXiv:1904.07865 (2019)
28. Mena, G., Belanger, D., Linderman, S., Snoek, J.: Learning latent permutations with gumbel-sinkhorn networks. In: ICLR (2018)
29. Monti, F., Boscaini, D., Masci, J., Rodola, E., Svoboda, J., Bronstein, M.M.: Geometric deep learning on graphs and manifolds using mixture model cnns. In: CVPR (2017)
30. Nogneng, D., Ovsjanikov, M.: Informative descriptor preservation via commutativity for shape matching. In: Computer Graphics Forum. vol. 36, pp. 259–267. Wiley Online Library (2017)
31. Ovsjanikov, M., Ben-Chen, M., Solomon, J., Butscher, A., Guibas, L.: Functional maps: a flexible representation of maps between shapes. ACM Transactions on Graphics (ToG) **31**(4), 1–11 (2012)
32. Pachauri, D., Kondor, R., Singh, V.: Solving the multi-way matching problem by permutation synchronization. NIPS (2013)
33. Pai, G., Ren, J., Melzi, S., Wonka, P., Ovsjanikov, M.: Fast sinkhorn filters: Using matrix scaling for non-rigid shape correspondence with functional maps. In: CVPR (2021)
34. Pinkall, U., Polthier, K.: Computing discrete minimal surfaces and their conjugates. Experimental mathematics **2**(1), 15–36 (1993)
35. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. NIPS (2017)
36. Ren, J., Panine, M., Wonka, P., Ovsjanikov, M.: Structured regularization of functional map computations. In: Computer Graphics Forum. vol. 38, pp. 39–53. Wiley Online Library (2019)
37. Ren, J., Poulenard, A., Wonka, P., Ovsjanikov, M.: Continuous and orientation-preserving correspondences via functional maps. ACM Transactions on Graphics (ToG) **37**, 1–16 (2018)
38. Rodolà, E., Cosmo, L., Bronstein, M.M., Torsello, A., Cremers, D.: Partial functional correspondence. In: Computer Graphics Forum. vol. 36, pp. 222–236. Wiley Online Library (2017)
39. Roufosse, J.M., Sharma, A., Ovsjanikov, M.: Unsupervised deep learning for structured shape matching. In: ICCV (2019)
40. Salti, S., Tombari, F., Di Stefano, L.: Shot: Unique signatures of histograms for surface and texture description. Computer Vision and Image Understanding **125**, 251–264 (2014)
41. Sharma, A., Ovsjanikov, M.: Weakly supervised deep functional maps for shape matching. NIPS (2020)
42. Sharp, N., Attaiki, S., Crane, K., Ovsjanikov, M.: Diffusionnet: Discretization agnostic learning on surfaces. arXiv preprint arXiv:2012.00888 (2020)
43. Sinkhorn, R., Knopp, P.: Concerning nonnegative matrices and doubly stochastic matrices. Pacific Journal of Mathematics **21**(2), 343–348 (1967)

44. Tam, G.K., Cheng, Z.Q., Lai, Y.K., Langbein, F.C., Liu, Y., Marshall, D., Martin, R.R., Sun, X.F., Rosin, P.L.: Registration of 3d point clouds and meshes: A survey from rigid to nonrigid. IEEE transactions on visualization and computer graphics **19**(7), 1199–1217 (2012)
45. Thomas, H., Qi, C.R., Deschaud, J.E., Marcotegui, B., Goulette, F., Guibas, L.J.: Kpconv: Flexible and deformable convolution for point clouds. In: ICCV (2019)
46. Van Kaick, O., Zhang, H., Hamarneh, G., Cohen-Or, D.: A survey on shape correspondence. In: Computer Graphics Forum. vol. 30, pp. 1681–1707 (2011)
47. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. NIPS (2017)
48. Vestner, M., Litman, R., Rodola, E., Bronstein, A., Cremers, D.: Product manifold filter: Non-rigid shape correspondence via kernel density estimation in the product space. In: CVPR (2017)
49. Wiersma, R., Eisemann, E., Hildebrandt, K.: Cnns on surfaces using rotation-equivariant features. ACM Transactions on Graphics (TOG) **39**(4), 92–1 (2020)

8   Supplementary Material

In this supplementary document we first introduce the implementation details of our method. Subsequently, we provide details on the unsupervised loss for partial shape matching. Afterwards, we discuss our network fine-tuning. Eventually, we also show additional qualitative results of our method.

8.1   Implementation details

We implemented our method in PyTorch. Our feature extractor takes 352-dimensional pre-computed SHOT descriptors [40] as inputs. We use Diffusion-Net [42] composed of 4 diffusion blocks with width 128 as the network architecture for both our feature extractor and universe classifier. In the context of the FM solver, we set $\lambda = 100$ in Eq. (9) and $\gamma = 0.5$ in Eq. (10) for our partial shape matching (for complete shape matching we use $\lambda = 0$). For the basis functions for functional maps computation, we choose the number to be 80 for the FAUST and SCAPE datasets for full shape matching. For partial shape matching, we choose the number to be 50 and 30 for the CUTS and HOLES subsets of the SHREC'16, respectively, to be consistent with DPFM [2]. We apply Sinkhorn normalisation with the number of iterations equal to 10 and the temperature parameter τ equal to 0.2.

If a dataset provides ground truth correspondences based on a reference shape, we set the number of universe vertices to the number of vertices of the reference shape. Otherwise, we set the number of universe vertices to the largest number of vertices among the given shapes. For network training, we use $w_{\mathrm{bij}} = 1.0, w_{\mathrm{orth}} = 1.0, w_{\mathrm{lap}} = 10^{-3}$ for $\mathcal{L}_{\mathrm{ft}}$ in Eq. (12) (for partial shape matching we use $w_{\mathrm{lap}} = 0$, since in this case we already enforce Laplacian commutativity regularisation in our regularised FM solver). The final loss is a linear combination of $\mathcal{L}_{\mathrm{ft}}$ and $\mathcal{L}_{\mathrm{cls}}$, where we set $\lambda_{\mathrm{cls}} = 0.01$ for complete shape matching. The loss for the universe classifier $\mathcal{L}_{\mathrm{cls}}$ is slightly different from Eq. (14) for partial shape matching, for which we provide the details in Sec. 8.2. We train our network with a batch size of 1 for all datasets. We use the ADAM optimiser with a learning rate of 10^{-3} for all experiments. The total number of training iterations for each dataset is 20000. During the first 4000 training iterations, when computing $\mathcal{L}_{\mathrm{cls}}$ defined in Eq. (14), we detach the gradient for C_{yx} and only regularise it based on its structural properties defined in $\mathcal{L}_{\mathrm{ft}}$. Afterwards, we will use the gradients for both C_{yx} and Π_{xy} to optimise our network. In this way, it can lead to faster convergence and better network performance.

8.2   Unsupervised loss for partial shape matching

In the context of partial-to-complete shape matching, we can assume that the complete shape plays the role of the universe shape, since it is guaranteed that each point in the partial shapes is in correspondence with exactly one point in the complete shape. We modify the unsupervised loss for universe classifier based

on it. For $\mathcal{X}$ being the complete shape and $\mathcal{Y}$ being the partial shape, the loss term can be expressed in the form

$$\mathcal{L}_{\text{cls}} = \mathcal{L}_{\text{ce}}^{\text{smooth}}(\Pi_x, \mathbf{I}_d) + \mathcal{L}_{\text{ce}}^{\text{smooth}}(\Pi_y, \hat{\Pi}_y), \tag{16}$$

where $\mathbf{I}_d$ is the identity matrix of size d, $\hat{\Pi}_y$ is the partial-to-complete correspondences obtained by nearest neighbour search between $\Phi_y C_{xy}$ and Φ_x, and $\mathcal{L}_{ce}^{\text{smooth}}$ is the cross entropy loss with label smoothing, where we set the smoothing factor equal to 0.1. The first term of the equation encourages the correspondences between the complete shape and the (virtual) universe shape to be identical, while the second term regularises the predicted partial-to-universe correspondence based on functional map regularisation. Similar to complete shape matching, the total unsupervised loss is a linear combination of $\mathcal{L}_{\text{ft}}$ and $\mathcal{L}_{\text{cls}}$, where we set $\lambda_{\text{cls}} = 1.0$.

8.3   Network fine-tuning

We observe that the generalisation ability of our method across different datasets can be improved by network fine-tuning, as shown in Fig. 3 (main paper). In order to achieve this, we first train our network on the training dataset in the ordinary way, and afterwards we use an unsupervised fine-tuning of the pre-trained network for the test dataset. Specifically, during fine-tuning, we update the network weights for each shape pair independently. To this end, we use the same loss defined in Eq. (15) to optimise the network with a fixed number of five forward/backward passes (for each shape pair individually). The advantage of network fine-tuning compared to post-processing techniques is that it directly optimises the network itself, thus leading to better performance.

8.4   Additional qualitative results

We show additional qualitative results on the FAUST dataset in Fig 8, on the SCAPE dataset in Fig 9, as well as on the SHREC'16 datset in Fig 10. Our method predicts shape-to-universe correspondences for each shape to obtain cycle-consistent multi-shape matchings among a collection of shapes.

Fig. 8: Qualitative multi-matching results using our method on the FAUST dataset.

Fig. 9: Qualitative multi-matching results using our method on the SCAPE dataset.

Colour code

Fig. 10: Qualitative partial-to-partial multi-matching results using our method on the SHREC'16 dataset. The full shape is shown merely for visualisation purposes (colour code).

8.5   Inter-class shape matching

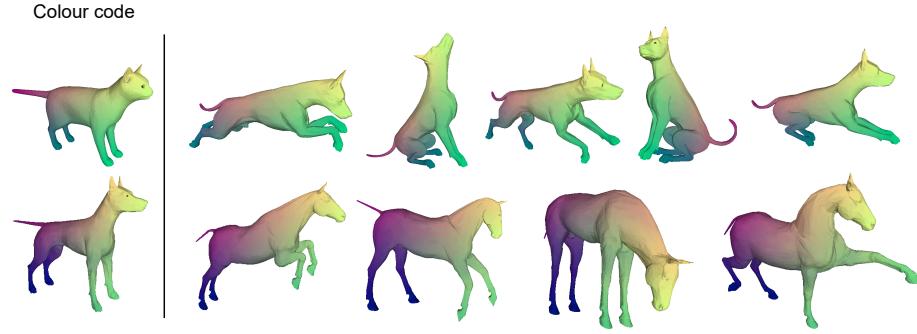We evaluate our method for the challenging inter-class multi-shape matching on the TOSCA dataset.

Fig. 11: Qualitative inter-class multi-matching results using our method on the TOSCA dataset.

8.6   Shape matching on SHREC'19 dataset

We evaluate our method on the more challenging SHREC'19 dataset. Furthermore, we randomly remesh each shape to different resolution to evaluate the robustness of our method with respect to different meshings.
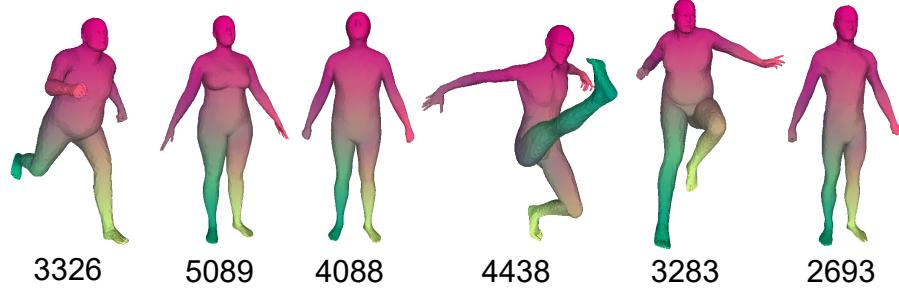

Fig. 12: Qualitative shape matching results using our method on the SHREC'19 dataset with different resolution (numbers refer to #vertices).