Learning Multi-resolution Functional Maps with Spectral Attention for Robust Shape Matching

Anonymous Author(s)
Affiliation
Address
email

Abstract

In this work, we present a novel non-rigid shape matching framework based on multi-resolution functional maps with spectral attention. Indeed, existing functional map learning methods all rely on a choice of the critical spectral resolution hyper-parameter, which can severely affect the overall accuracy or lead to overfitting, if not chosen carefully. In this paper, we show that spectral resolution tuning can be alleviated by introducing spectral attention. Our framework is applicable in both supervised and unsupervised settings, and we show that it is possible to train the network so that it can adapt the spectral resolution, depending on the given shape input. More specifically, we propose to compute multi-resolution functional maps that characterize correspondence across a wide range of spectral resolution, and introduce a spectral attention network that helps to combine this representation into a single coherent final correspondence. Our approach is not only accurate with near-isometric input, for which a high spectral resolution is typically preferred, but also robust and able to produce reasonable matching even in the presence of significant distortion, which poses great challenges to existing methods. We demonstrate the superior performance of our approach through experiments on a suite of challenging non-rigid shape matching benchmarks, including a new non-isometric correspondence dataset.

1 Introduction

Shape matching is a critical task in 3D shape analysis and has been paramount to a broad spectrum of downstream applications, including registration, deformation, and texture transfer [1, 2], to name a few. The algorithmic challenge of robust shape matching primarily lies in the fact that shapes may undergo significant variations, such as arbitrary non-rigid deformations. Earlier works to tackle non-rigid shape correspondence conventionally build upon hand-crafted features and pipelines [3], while now, in the time of deep learning with significantly increased computing power, the research focus has largely shifted to data-driven and learning-based approaches for improved matching robustness and accuracy [4].

To learn for non-rigid shape matching, a growing body of literature [7, 8, 9, 10, 11, 12] advocates spectral techniques, in particular, the functional map representation [13], which compactly encodes correspondences as small-sized matrices using a reduced spectral basis. A number of advances have been made to the functional map-based networks in terms of probe feature learning [5, 14], differentiable map regularization [5], supervised [7] and unsupervised learning [9, 8, 15], among many others.

Despite this progress, existing works nearly always learn functional maps in a single spectral resolution (or the number of basis functions used), which is often set empirically. However, the functional map resolution plays a crucial role in the non-rigid shape matching performance, as

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

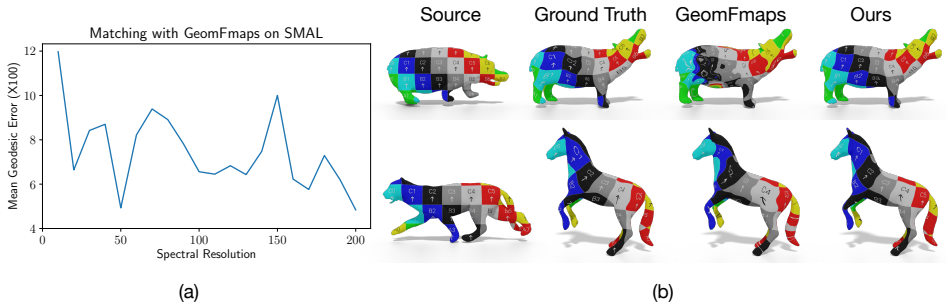

Figure 1: (a) The unstable matching performance of GeomFmaps [5] w.r.t. the spectral resolution parameter on an animal shape dataset SMAL [6]. (b) Correspondence visualization by texture transfer for near-isometric (top) and non-isometric (bottom) shapes. GeomFmaps is trained with supervision, while our approach is not.

observed in existing literature [9]. As a concrete example, Fig. 1-(a) shows that the matching performance of a state-of-the-art supervised learning method GeomFmaps [5] fluctuates significantly when trained with different map resolution. Therefore, to improve robustness, it is desirable to enable networks to adaptively make the resolution decision in a data-dependent manner: for near-isometric shapes, adopting a higher spectral resolution allows high-frequency details to be leveraged for more precise matching; while for non-isometric shapes, adopting a lower spectral resolution is advantageous to obtain approximate but potentially more robust maps.

Motivated by the above discussion, in this work, we propose a novel learning-based functional map framework that learns to adaptively combine multi-resolution maps with a spectral attention mechanism, aiming at accommodating both near-isometric and non-isometric shapes at the same time. Specifically, our framework consists in two novel components (Fig. 2): (1) multi-resolution functional maps and (2) spectral attention module.

Given as input a pair of non-rigidly deformed shapes, we first use a functional map network to estimate a series of maps with varying spectral resolution. Next, we feed the obtained functional maps to a spectral attention network to predict a weight for each map. The attention weights are used to combine all the intermediate maps into a final coherent map. To enable such an assembly, we design a differentiable spectral upsampling module that can transform the intermediate maps to the same spectral resolution within a learnable network. Finally, to train our network, we propose to impose penalties on the intermediate multi-resolution functional maps as well as the final map. This is different from existing approaches [7, 9, 5, 10], to name a few, that penalize a single hand-picked spectral resolution. Our method can be trained in both supervised and unsupervised settings and can directly benefit from other advances in deep functional map training, such as improved architectures or regularization. To evaluate our model, we perform a comprehensive set of experiments on several challenging non-rigid shape matching datasets, where our model achieves superior matching performance over existing methods.

In a nutshell, the main contributions of our work are as follows: (1) We introduce a powerful non-rigid shape matching framework equipped with multi-resolution functional maps with spectral attention for handling diverse shape inputs. (2) We propose a novel spectral attention network and a differentiable spectral upsampling module for robust functional map learning. (3) We demonstrate the superior performance of our model compared to existing approaches through extensive experiments on challenging non-rigid shape matching benchmarks. We will make our code and data publicly available upon publication.

2 Related Work

In shape analysis, the field of non-rigid shape matching is both extensive and well-studied. We review in the following paragraphs the works that constitute the backbone or those that are most closely related to our approach. A more complete overview can be found in recent surveys [16, 17], and more recently [4] (section 4).

Functional Maps Our method is based on the functional maps framework, first introduced in [13], and extended in various works such as [18, 19, 20, 21, 22, 23] among others. This paper proposed to encode maps between shapes using a reduced basis representation. Consequently, the problem of map optimization becomes both linear and more compact. Besides, this framework allows to represent natural constraints such as near-isometry or bijectivity as linear-algebraic regularization. It has also been extended to the partial setting [24, 25].

One of the bottlenecks of this well-known framework is the estimation of so-called "descriptor functions" that are key to the functional map computation. Early methods have relied on axiomatic features, mainly based on multi-scale diffusion-based descriptors, *e.g.*, HKS and WKS [26, 27].

Learning-based methods While learning maps between shapes can be treated as a dense segmentation problem, as in [28, 29, 30, 31, 32, 33], these methods *a)* usually require many labeled training shapes, which can be hard to obtain, and *b)* tend to overfit to the training connectivity, making the method unstable to triangulation change.

Closer to our approach are deep shape matching methods that also rely on the functional map framework, pioneered by FMNet [7]. In this work, SHOT descriptors [34] are given as input to the network, whose goal is to refine these descriptors in order to yield a functional map as close to ground-truth as possible. The key advantage of this approach is that it directly estimates and optimizes for the map itself, thus injecting more structure in the learning problem. FMNet introduced the idea of learning for shape *pairs*, using the same feature extractor (in their case, a SHOT MLP-based refiner) for the source and target shapes in a Siamese fashion to produce improved output descriptors for functional map estimation. However, later experiments conducted in [5] have highlighted that SHOT-based pipelines suffer greatly from connectivity overfitting. Thus, authors in [5, 15, 14] advocate for learning directly from the shapes' geometry, while introducing a strong regularizer for functional map estimation.

The major upside of using the functional map framework for deep shape matching is that it relies on the intrinsic information of the shapes, which results in overall good generalization from train to test, especially across pose changes, which involve minimal intrinsic deformation.

Unsupervised spectral learning The methods described above are supervised deep shape matching pipelines. While these methods usually give good correspondence prediction, they need ground-truth supervision at training time. Consequently, other methods have focused on training for shape matching using the functional map framework, *without ground-truth supervision*. This was originally performed directly on top of FMNet by enforcing either geodesic distance preservation [8, 35], or natural properties on the output functional map [9], as well as by promoting cycle consistency [36].

To disambiguate between symmetries present in many organic shapes, some works choose to rely on so-called "weak-supervision", by rigidly aligning all shapes (on the same three axes) as a pre-processing step [15, 37], and then use the extrinsic embedding information to resolve the symmetry ambiguity. This, however, limits their utility to correspondences between shapes with the same rigid alignment as the training set. Another solution is to use input signals that are independent to the shapes alignment, such as SHOT [34] descriptors as done in the original FMNet. One of these recent methods [11], makes use of optimal transport on top of this SHOT-refiner to align the shapes at different spectral scales. This method, like ours, thus computes the functional map at different scales via progressive upsample, but they only keep the last map as the output whereas we propose to let the network pick the best combination of different resolutions. Additionally, this method is dependent on SHOT input which makes it unstable towards change in triangulation. In-network refinement is also performed in DG2N [38], but not in spectral space.

Attention-based spectral learning The attention mechanism was originally introduced in deep learning for natural language processing, and consists in putting relative weights on the different words of an input sentence [39]. This mechanism can be applied in different contexts, including that of shape analysis. Indeed, attention can be utilised to focus on different parts of a 3D shape, for instance in partial shape matching, as done in [10]. As what we show in this paper, attention can also be used in the spectral domain by letting the network focus on different levels of details depending on the input shapes and their resulting functional maps at different spectral resolutions. Indeed, the utility of considering different resolutions of a functional map, *e.g.*, via upsampling of its size, has been highlighted in [40, 41]. Here we propose to let the network learn to adaptively combine all the intermediate functional maps into a final coherent correspondence.

3 Background

Our work proposes a learning-based framework for non-rigid shape matching by building upon the functional map representation [13], and especially its learning-based variant GeomFmaps, introduced in [5]. Before describing our approach in Sec. 4, we first briefly describe the basic learning pipeline with functional maps for shape correspondence below. We refer the interested reader to a recent course [42] for more technical details.

Deep Functional Map Pipeline We consider a pair of non-rigidly aligned shapes $\mathcal{S}_1$ and $\mathcal{S}_2$, represented as triangle meshes with n_1 and n_2 vertices, respectively. The goal is to compute a map as isometric as possible between the shapes in an efficient way. The basic learning pipeline estimates a functional map between $\mathcal{S}_1$ and $\mathcal{S}_2$ in the following four steps [42].

- Compute the first k eigenfunctions of the Laplace-Beltrami operator [43] on each shape, which will be used as a basis for decomposing smooth functions on these shapes. The eigenfunctions are stored as columns in matrices $\Phi_1 \in \mathbb{R}^{n_1 \times k}$ and $\Phi_2 \in \mathbb{R}^{n_2 \times k}$.
- Second, a set of descriptors (also known as probe, or feature functions) on each shape are extracted by a feature extractor network [7, 5] here denoted by $\mathcal{F}_\Theta$ with learnable parameters Θ. These feature functions are expected to be approximately preserved by the unknown map. We denote the learned feature functions as $\mathcal{F}_\Theta(\mathcal{S}_1) = G_1 \in \mathbb{R}^{n_1 \times d}$ and $\mathcal{F}_\Theta(\mathcal{S}_2) = G_2 \in \mathbb{R}^{n_2 \times d}$, where d is the number of descriptors, and project them onto the respective eigenbases. The resulting coefficients are stored as columns of matrices $\mathbf{A}_1, \mathbf{A}_2 \in \mathbb{R}^{k \times d}$, respectively.
- Next, we compute the optimal functional map $\mathbf{C} \in \mathbb{R}^{k \times k}$ by solving:

$$\mathbf{C} = \arg\min_{\mathbf{C}} \|\mathbf{C}\mathbf{A}_1 - \mathbf{A}_2\|^2 + \alpha \|\mathbf{C}\mathbf{\Delta}_1 - \mathbf{\Delta}_2\mathbf{C}\|^2, \tag{1}$$

  where the first term promotes preservation of the probe functions, and the second term regularizes the map by measuring its commutativity with the Laplace-Beltrami operators [13, 5], which in the reduced basis become diagonal matrices of the eigenvalues $\mathbf{\Delta}_1$ and $\mathbf{\Delta}_2$.
- As a last step, the estimated map $\mathbf{C}$ can be converted to a point-to-point map typically by nearest neighbor search between the spectral embeddings Φ_1, Φ_2 aligned by $\mathbf{C}$, with possible post-refinement applied [44, 21, 40, 45].

To train the feature extractor network $\mathcal{F}_\Theta$, one defines a set of training shape *pairs*, and a set of testing shape *pairs*. As shown in [5], the solution to Eq. 1 can be obtained in closed form within a neural network in a differentiable manner, and constitutes what is called "FMReg" in Fig. 2. Using this insight, during training time the network aims at reducing a loss $L(\mathbf{C})$, defined on the output functional map $\mathbf{C}(\mathcal{F}_\Theta(\mathcal{S}_1), \mathcal{F}_\Theta(\mathcal{S}_2))$ estimated from the learned descriptors using the closed form solution of Eq. 1. The network parameters Θ are then back-propagated and updated to make the network produce better features for the next pair of shapes.

We stress that in this pipeline, the size of the functional map k is a critical non-learned hyper-parameter, which can strongly affect matching results (as highlighted in Fig. 1 and in existing literature [9]).

4 Method

The main goal of our work is to robustly and adaptively estimate functional maps for shape pairs with diverse geometric properties, including both near-isometric and non-isometric transformations. In this section, we describe the technical details of our proposed non-rigid shape matching framework. We illustrate the whole pipeline in Fig. 2. Our framework has two main stages: multi-resolution functional map learning (Sec. 4.1) and spectral attention learning (Sec. 4.2).

4.1 Multi-resolution Functional Maps

In the first stage of our framework, given the input shapes $\mathcal{S}_1$ and $\mathcal{S}_2$, we follow the basic learning pipeline, as described in Sec. 3, to infer multi-resolution functional maps for extensively characterizing shape matching in the spectral domain. As mentioned above, this differs from existing learning-based

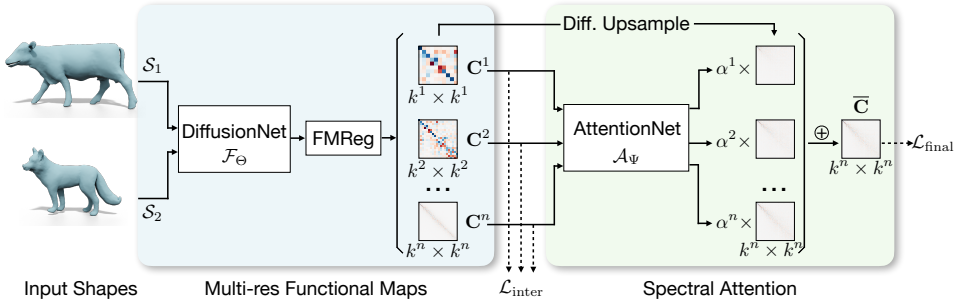

Figure 2: Illustration of our non-rigid shape matching pipeline based on multi-resolution functional maps with spectral attention. Our network takes as input a pair of shapes $\mathcal{S}_1$ and $\mathcal{S}_2$. First, we estimate a set of intermediate functional maps $\{\mathbf{C}^i\}_{i=1}^n$ with varying resolution, using DiffusionNet $\mathcal{F}_\Theta$ as the feature backbone and the differentiable FMReg module for the map computation. Next, we design a spectral attention network $\mathcal{A}_\Psi$ to predict a set of attention weights $\{\alpha^i\}_{i=1}^n$ for combining the functional maps at different resolutions, transformed by differentiable spectral upsampling, into the final map $\overline{\mathbf{C}}$. We impose losses $\mathcal{L}_{\text{inter}}$ and $\mathcal{L}_{\text{final}}$ during training.

functional map works typically computing a single map at a specific hand-picked resolution, which may not be universally applicable due to the rich diversity of shapes, thus putting limit on the generalization of networks across datasets.

To learn multi-resolution maps, we adopt a recent surface feature learning network DiffusionNet [14] as the backbone to learn the d probe functions. DiffusionNet is shown to be robust to discretization and widely applicable to non-rigid shape analysis tasks [14, 12]. Following the notation in Sec. 3, let $\mathcal{F}_\Theta$ denote this DiffusionNet backbone. We feed the input shapes to the network a Siamese manner and obtain $G_1 = \mathcal{F}_\Theta(\mathcal{S}_1)$ and $G_2 = \mathcal{F}_\Theta(\mathcal{S}_2)$. These learned probe functions are shared in the subsequent computation of multi-resolution maps.

To solve for a functional map with Eq. (1) in a differentiable manner, we employ the FMReg module proposed in [5]. With slight abuse of notation, let $\mathcal{C} = \{\mathbf{C}^i\}_{i=1}^n$ denote a series of n estimated functional maps with varying resolution (Fig. 2-left), where the ith map $\mathbf{C}^i$ obtained via Eq. (1) is of size $k^i \times k^i$. Adjacent maps differ in their resolution by τ rows and τ columns, *i.e.*, $k^{i+1} = k^i + \tau$ for $\mathbf{C}^{i+1}$, meaning that τ more eigenfunctions are included in Φ_1 and Φ_2 for the map optimization.

Acceleration In the above approach, FMReg is repeatedly used to optimize Eq. (1) for each $\mathbf{C}^i \in \mathcal{C}$, incurring noticeable computation cost. To address this issue, we point out an acceleration scheme based on principal submatrices to significantly simplify the computation of $\mathcal{C}$. Specifically, we observe that for $i < n$, the principal $k^i \times k^i$ submatrix of $\mathbf{C}^n$ can be treated as an approximation of $\mathbf{C}^i$ computed in the standard way with Eq. (1). Accordingly, we can first employ FMReg once to compute the largest map $\mathbf{C}^n$ and then construct the multi-resolution functional maps $\mathcal{C}$ as a set of principal submatrices of $\mathbf{C}^n$, thus avoiding the optimization for $\{\mathbf{C}^i\}_{i=1}^{n-1}$. In practice, we find this acceleration scheme to work reasonably well, showing comparable matching performance with the above standard procedure (Sec. 5).

4.2 Spectral Attention

Given the estimated multi-resolution functional maps $\mathcal{C}$, there are two main questions: (1) how to prioritize the maps, and (2) how to address the resolution difference for combining the maps together. Thus in the second stage of our framework, instead of performing hard assignment, we propose to learn for each map $\mathbf{C}^i \in \mathcal{C}$ a soft attention weight $\alpha^i \in [0, 1]$, representing its contribution to a final map assembly. To enable the assembly, we also introduce a differentiable spectral upsampling module to align the map resolution within the network.

Network To predict $\{\alpha^i\}_{i=1}^n$, the attention network needs to jointly assess the multi-resolution functional maps with their associated spectral information (Fig. 2-right). Directly taking the eigenfunctions as input signals to the network can be problematic due to the known issues of sign flipping and order changes [13]. In this work, we opt for spectral alignment residual as pointwise features and build the network upon a PointNet-based architecture [46].

Specifically, let Φ_1^i and Φ_2^i denote the spectral embeddings associated with $\mathbf{C}^i$, *i.e.*, matrices of the first k^i eigenfunctions on $\mathcal{S}_1$ and $\mathcal{S}_2$, respectively, as defined in Sec. 3. For a point $q \in \mathcal{S}_2$, we define its spectral alignment residual r_q^i as

$$r_q^i = \min_{p \in \mathcal{S}_1} \delta_{qp}^i, \qquad \delta_{qp}^i = \|(\Phi_2^i[q])^\top - \mathbf{C}^i(\Phi_1^i[p])^\top\|_2, \tag{2}$$

where $\Phi_1^i[p]$ denotes the pth row of the matrix Φ_1^i, similarly for $\Phi_2^i[q]$. We gather the residuals across $\mathcal{C}$ into a feature vector $\mathbf{r}_q = [..., \ r_q^i/\sqrt{k^i}, \ r_q^{i+1}/\sqrt{k^{i+1}}, \ ...]^\top$ for the point q, where the scaling factor $\sqrt{k^i}$ is to counteract the dimensionality difference across the spectral embeddings. The feature $\mathbf{r}_q$ can be interpreted as a form of confidence of each map $\mathbf{C}^i$ at the point q.

The vectors of spectral alignment residuals $\{\mathbf{r}_q\}_{q \in \mathcal{S}_2}$ collectively form an unordered set. We then feed this n-dimensional point set to our spectral attention network, denoted as $\mathcal{A}_\Psi$ with learnable parameters Ψ. We build $\mathcal{A}_\Psi$ with the classification architecture of PointNet [46] with feature transformations and global feature pooling. We forward the pooled global features through multi-layer perceptrons and the softmax function to obtain the attention weights *s.t.* $\sum_{i=1}^n \alpha^i = 1$.

Differentiable Spectral Upsampling Due to the dimensionality difference, the functional maps in $\mathcal{C}$ cannot be straightforwardly combined together with the learned spectral attention. Inspired by [40], we propose a differentiable spectral upsampling module to transform all the maps to the same spectral resolution. The main idea consists of two differentiable steps: (1) Convert the $k^i \times k^i$-size functional map $\mathbf{C}^i$ to a soft pointwise map; (2) Convert the pointwise map to a $k^n \times k^n$-size functional map. Specifically, in the first step, we compute a soft pointwise map, denoted as $\Pi^i \in \mathbb{R}^{n_2 \times n_1}$, from $\mathbf{C}^i$ as

$$\Pi_{qp}^i = \frac{\exp(-\delta_{qp}^i/t)}{\sum_{p'} \exp(-\delta_{qp'}^i/t)}, \tag{3}$$

where $p, p' \in \mathcal{S}_1$ and $q \in \mathcal{S}_2$, and t is a learnable temperature parameter. Next, to compute an upsampled map $\widehat{\mathbf{C}}^i$ of size $k^n \times k^n$, we project Π^i onto the corresponding spectral basis:

$$\widehat{\mathbf{C}}^i = (\Phi_2^n)^\dagger \Pi^i \Phi_1^n, \tag{4}$$

where $\dagger$ denotes the Moore-Penrose inverse. After this differentiable upsampling, we obtain a set of functional maps $\widehat{\mathcal{C}} = \{\widehat{\mathbf{C}}^i\}_{i=1}^n$ of size $k^n \times k^n$. Moreover, since they all represent maps between the same shape pair $\mathcal{S}_1$ and $\mathcal{S}_2$, they can be directly compared to each other and linearly combined.

Map Assembly With the estimated spectral attention $\{\alpha^i\}_{i=1}^n$ and the upsampled functional maps $\widehat{\mathcal{C}}$, the last step of our framework is to assemble the intermediate maps together as a final coherent map output for shapes $\mathcal{S}_1$ and $\mathcal{S}_2$ (Fig. 2-rightmost). We denote the final map as $\overline{\mathbf{C}}$, which is computed by a simple linear combination as follows:

$$\overline{\mathbf{C}} = \sum_{i=1}^n \alpha^i \widehat{\mathbf{C}}^i. \tag{5}$$

4.3 Training

Loss To train our network, we impose losses on both the intermediate multi-resolution functional maps $\mathcal{C}$ and the final map output $\overline{\mathbf{C}}$, as shown in Fig. 2. Following prior works [5, 9], our network can be trained in a supervised or unsupervised way, and we present comprehensive results of both training paradigms in Sec. 5. In both the supervised and unsupervised settings, the two loss terms for maps $\mathcal{C}$ and $\overline{\mathbf{C}}$ are formulated as

$$\mathcal{L}_{\text{inter}} = \frac{1}{n} \sum_{i=1}^n \left(\frac{k^n}{k^i}\right)^2 L(\mathbf{C}^i), \qquad \mathcal{L}_{\text{final}} = L(\overline{\mathbf{C}}), \tag{6}$$

where $L(\mathbf{C})$ is the penalty for a given map $\mathbf{C}$ depending on the training paradigm and described below. The map penalty in $\mathcal{L}_{\text{inter}}$ is multiplied by a scaling factor to counterbalance the spectral resolution difference. The overall training loss combines the two losses as $\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{inter}} + \mathcal{L}_{\text{final}}$.

For supervised training, we define the map penalty $L(\mathbf{C}) = \|\mathbf{C} - \mathbf{C}^{\text{gt}}\|^2$, where $\mathbf{C}^{\text{gt}}$ is the corresponding ground truth functional map of the same spectral resolution, and we use the Frobenius norm. For unsupervised training, we follow [9] to define the map penalty as $L(\mathbf{C}) = \|\mathbf{C}^\top \mathbf{C} - \mathbf{I}\|^2$, which promotes orthogonality of the given map. Note that a Laplacian commutativity penalty [9] is not necessary because of the regularization imposed in Eq. (1).

Implementation We implement our network with PyTorch [47]. The feature backbone consists of four DiffusionNet blocks [14] and produces $d = 256$ dimensional features as output. We use $\alpha = 10^{-3}$ in Eq. (1). We compute $n = 20$ multi-resolution functional maps with sizes ranging from 10×10 to 200×200 with a step size $\tau = 10$. We set the batch size to 1 and use ADAM [48] for optimization. We use the WKS [27] descriptor as input to the network for all experiments, except for the SMAL dataset [6] in Sec. 5.2, where the 3D coordinates are used as input. This XYZ signal input is augmented with rotations around the up axis (here Y), since the shapes in SMAL are randomly rotated around this axis. This input signal can better address overfitting in the hard case of different animal species than WKS. More implementation details are provided in the supplementary material.

5 Experiments

In this section, we present extensive experimental results on a wide range of challenging non-rigid shape correspondence benchmarks to demonstrate the superior performance of our approach. We perform evaluations on widely adopted human shape matching datasets such as FAUST [2] and SCAPE [49] as well as a non-isometric dataset SHREC'19 [50]. We also evaluate on an animal shape matching dataset SMAL [6] to stress-test generalization. Moreover, we construct another challenging humanoid shape dataset from DeformingThings4D [51] for both near-isometric and highly non-isometric correspondence. We use remeshed variants for these datasets, introduced in [21] and used in recent learning-based methods [5, 15, 11]. This ensures that the shapes do not share identical mesh connectivity.

5.1 Matching on FAUST, SCAPE, and SHREC'19

Datasets Following [5, 11], the FAUST dataset, which has 100 human shapes, is split into 80 and 20 shapes for training and testing, respectively. For the SCAPE dataset consisting of 71 human shapes, the training and testing split is 51/20. The SHREC'19 dataset has 44 human shapes and is used only as a test set. Shape 40 in SHREC'19 is a partial shape and removed from the dataset, since it is outside the scope of this work.

Baselines We perform comparison with existing non-rigid shape matching works categorized as follows: (1) Axiomatic approaches including BCICP [21], ZoomOut [40], and Smooth Shells [52]; (2) Supervised learning approaches including FMNet [7], 3D-CODED [53], HSN [33], ACSCNN [54], and GeomFmaps [5]; (3) Unsupervised learning approaches including SURFMNet [9], UnsupFMNet [8], NeuroMorph [37], DeepShells [11], and GeomFmaps [5]. In particular, GeomFmaps is a strong baseline closely related to our work. For a fair comparison, we also use DiffusionNet as the feature backbone in GeomFmaps, with the same input signal as ours, for improved performance [14]. The supervised and unsupervised training losses for GeomFmaps are the same as the map penalties L defined in Sec. 4.3. GeomFmaps uses 30 eigenfunctions, as recommended in the original work [5].

Table 1: Mean geodesic error ($\times 100$) on **F**AUST, **S**CAPE, and **S**HREC'**19**

Train		**F**			**S**			**F + S**		
Test		**F**	**S**	**S19**	**F**	**S**	**S19**	**F**	**S**	**S19**
FMNet	sup	11.0	30.0	-	33.0	30.0	-	-	-	-
3D-CODED	sup	2.5	31.0	-	33.0	31.0	-	-	-	-
HSN	sup	3.3	25.4	-	16.7	3.5	-	-	-	-
ACSCNN	sup	2.7	8.4	-	6.0	3.2	-	-	-	-
GeomFmaps	sup	2.6	3.6	9.9	2.9	2.9	12.2	2.6	2.9	7.9
Ours-Fast	sup	**1.3**	2.9	**7.1**	1.8	**1.8**	**11.7**	**1.3**	**1.8**	7.1
Ours	sup	1.4	**2.2**	9.4	**1.7**	**1.8**	12.2	**1.3**	**1.8**	**6.2**
BCICP		6.1	-	-	-	11.	-	-	-	-
ZoomOut		6.1	-	-	-	7.5	-	-	-	-
SmoothShells		2.5	-	-	-	4.7	-	-	-	-
SURFMNet	unsup	15.0	32.0	-	32.0	12.0	-	33.0	29.0	-
UnsupFMNet	unsup	10.0	29.0	-	22.0	16.0	-	11.0	13.0	-
NeuroMorph	unsup	8.5	28.5	26.3	18.2	29.9	27.6	9.1	27.3	25.3
DeepShells	unsup	**1.7**	5.4	27.4	2.7	2.5	23.4	**1.6**	2.4	21.1
GeomFmaps	unsup	3.5	4.8	8.5	4.0	4.3	11.2	3.5	4.4	7.1
Ours-Fast	unsup	1.9	**2.6**	**5.8**	**1.9**	**2.1**	**8.1**	1.9	**2.3**	6.3
Ours	unsup	1.9	**2.6**	6.4	2.2	2.2	9.9	1.9	**2.3**	**5.8**

The **best** results are highlighted separately for supervised and unsupervised methods.

Results Tab. 1 reports the matching performance in terms of mean geodesic error on unit-area shapes [55]. Ours-Fast refers to our method with the acceleration scheme (Sec. 4.1). We observe that our method outperforms all baselines in both the supervised and unsupervised settings. The slightly better correspondences obtained on FAUST by DeepShells [11] can be explained by the fact that they use refinement as a post-processing step, which our method *does not require*. Our method directly produces high-quality maps by choosing the best weighted combination of functional maps obtained from matching at different spectral resolutions. In particular, notice that how we improve upon the closest competitor, GeomFmaps [5], by enabling the network to optimize the spectral resolution.

Besides, the fast version of our method gives almost the same results as the complete approach, and even better results in some cases. This underlines the fact that the approximation made in this version of our model works well in practice.

5.2 Matching on SMAL

Dataset The SMAL dataset has 49 four-legged animal shapes of eight species. To stress-test the generalization of learning-based approaches, we use five species for training and three species for testing, resulting in a 29/20 split of the shapes. Thus no shapes similar to the testing data are seen during training, and the correspondence across species is highly non-isometric, presenting great challenges to learning-based approaches.

Results Tab. 2-left reports the matching performance. Again, our method significantly improves upon its predecessor GeomFmaps in both supervised and unsupervised settings. Besides, DeepShells fails to predict accurate maps in this hard setting, due to its tendency to overfit to the training set, since they use SHOT input which is very triangulation-sensitive. Our method thus has strong generalization power, being able to predict correspondence between two animals whose species are never encountered during training.

Table 2: Results ($\times 100$) on SMAL and DT4D-H

		SMAL	DT4D-H	
			intra-class	inter-class
GeomFmaps	sup	8.4	2.1	**4.1**
Ours-Fast	sup	5.8	2.0	4.7
Ours	sup	**5.3**	**1.8**	4.6
DeepShells	unsup	29.3	3.4	31.1
GeomFmaps	unsup	7.6	3.3	22.6
Ours-Fast	unsup	5.8	**1.2**	14.6
Ours	unsup	**5.4**	1.7	**11.6**

5.3 Matching on DeformingThings4D

Dataset We also test on a new non-rigid shape matching benchmark consisting of shapes from a recently introduced large-scale animation dataset DeformingThings4D [51]. We select nine classes of humanoid shapes for our benchmark. The inter-class ground truth was found by non-rigid ICP using manually selected landmarks. In total, there are 198/95 non-rigid shapes for training/testing, significantly more than the previous datasets. We denote this humanoid shape benchmark as DT4D-H.

Results Tab. 2-right presents the performance for both intra and inter class matching. In the supervised setting, we obtain performance comparable to GeomFmaps. This may be due to artefacts in the inter-class ground truth computed by non-rigid ICP across strongly non-isometric shape classes, and that matching accuracy below 5 becomes saturated. Nevertheless, the improvement becomes remarkably noticeable in the unsupervised setting. DeepShells suffers from overfitting, as it is shown to succeed in intra-class matching but *not in inter-class matching*. Meanwhile, our approach is the only one to give reasonable overall results in this challenging unsupervised case.

Visualization In addition to the SMAL results in Fig. 1-(b), we show in Fig. 3 more qualitative correspondence results on the DT4D-H dataset, where the compared methods are trained in the unsupervised setting. We observe that our method is able to produce high-quality correspondences for both intra and inter class shapes, compared to other methods. In Fig. 3-right, we also plot the spectral attention (*i.e.*, $\{\alpha^i\}_{i=1}^n$ defined in Sec. 4.2) estimated by our network with the unsupervised training. The plots demonstrate that our network can adapt the spectral resolution by distributing attention weights according to the shape input for more robust matching.

5.4 Ablation Study

We perform ablation studies w.r.t. our network components and show the results in Tab. 3, where unsupervised training on SMAL is used. First, we train our network without the loss on intermediate

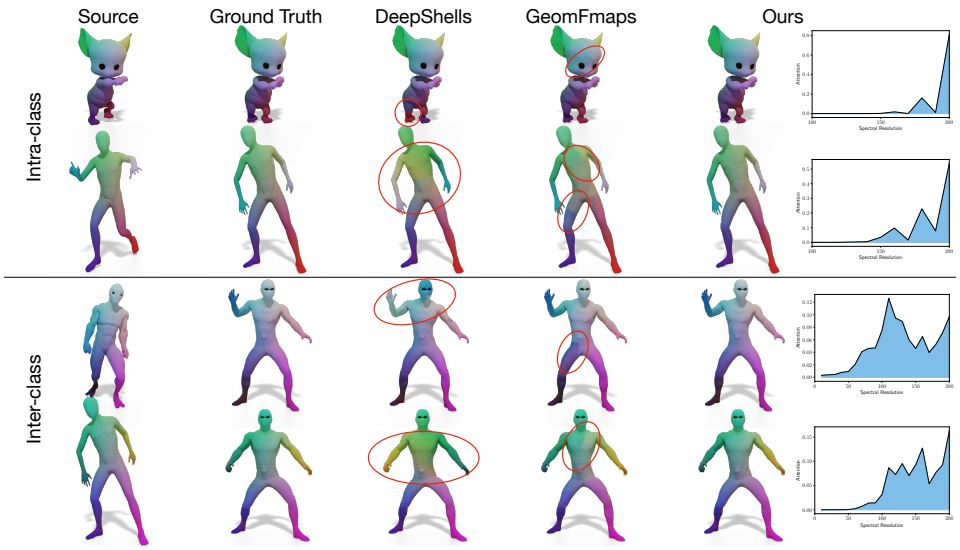

Figure 3: Correspondence visualization by color transfer for shapes from the DT4D-H dataset, where the methods are trained in the unsupervised setting. The rightmost plots are the learned spectral attention by our method.

multi-resolution functional maps and denote it as w/o $\mathcal{L}_{\text{inter}}$ in Tab. 3. We observe that using $\mathcal{L}_{\text{inter}}$ can greatly improve the matching performance owing to the better intermediate representations produced.

Next, to validate the spectral attention network $\mathcal{A}_\Psi$, we remove it and compute spectral attention with two non-learning schemes. One is to use a uniform weight (*i.e.*, $\alpha^i = 1/n$), denoted as w/o $\mathcal{A}_\Psi$ w/ uni. in Tab. 3. The other is to use mean spectral alignment residual (*i.e.*, soft-max on $-\sum_{q\in\mathcal{S}_2} r_q^i/n_2$ for α^i), denoted as w/o $\mathcal{A}_\Psi$ w/ avg. res. The results show that the matching performance degrades without the spectral attention network, in particular for Ours-Fast, which fails to converge with the uniform weight during training, strongly indicating the robustness brought by learned spectral attention. Moreover, we also test different values for the spectral step size τ (Sec. 4.1) and set it to 10 for the balanced performance and computation cost.

Table 3: Ablation results ($\times 100$)

	Ours-Fast	Ours
Full model	5.8	5.4
w/o $\mathcal{L}_{\text{inter}}$	13.1	13.4
w/o $\mathcal{A}_\Psi$ w/ uni.	56.2	5.9
w/o $\mathcal{A}_\Psi$ w/ avg. res.	11.1	5.6
$\tau = 5$	5.1	7.2
$\tau = 20$	6.4	6.9

6 Conclusion, Limitations, and Societal Impacts

To conclude, we introduced a robust non-rigid shape correspondence framework applicable in both supervised and unsupervised settings. We leverage a multi-resolution functional map representation and propose to learn spectral attention for a final coherent correspondence estimation, resulting in a powerful deep model that can adaptively accommodate near-isometric and non-isometric shape input. We demonstrate the improved matching performance of our model through extensive experiments on challenging non-rigid shape matching benchmarks.

One limitation of our approach is, similar to existing functional map works, that we need to pre-compute spectral decomposition for each input shape. While the pre-computation is efficient for moderately sized shapes, down-sampling or remeshing may be needed for large-size input. Besides, we focus on full shape matching and assume input shapes represented as triangle meshes. It would be interesting to investigate an extension of our approach to partial shapes or point clouds as input.

Lastly, we do not see any immediate ethical issue with the proposed method, but note that considering the superior performance on human shape matching, unintended uses, such as surveillance, may be a potential negative societal impact of our work.

References

[1] Huong Quynh Dinh, Anthony Yezzi, and Greg Turk. Texture transfer during shape transformation. *ACM TOG*, 24(2):289–310, 2005.

[2] Federica Bogo, Javier Romero, Matthew Loper, and Michael J Black. FAUST: Dataset and evaluation for 3d mesh registration. In *CVPR*, 2014.

[3] Oliver Van Kaick, Hao Zhang, Ghassan Hamarneh, and Daniel Cohen-Or. A survey on shape correspondence. In *CGF*, volume 30, pages 1681–1707. Wiley Online Library, 2011.

[4] Bailin Deng, Yuxin Yao, Roberto M. Dyke, and Juyong Zhang. A survey of non-rigid 3d registration, 2022.

[5] Nicolas Donati, Abhishek Sharma, and Maks Ovsjanikov. Deep geometric functional maps: Robust feature learning for shape correspondence. In *CVPR*, 2020.

[6] Silvia Zuffi, Angjoo Kanazawa, David W. Jacobs, and Michael J. Black. 3D Menagerie: Modeling the 3d shape and pose of animals. In *CVPR*, 2017.

[7] Or Litany, Tal Remez, Emanuele Rodola, Alex Bronstein, and Michael Bronstein. Deep functional maps: Structured prediction for dense shape correspondence. In *ICCV*, 2017.

[8] Oshri Halimi, Or Litany, Emanuele Rodola, Alex M. Bronstein, and Ron Kimmel. Unsupervised learning of dense shape correspondence. In *CVPR*, 2019.

[9] Jean-Michel Roufosse, Abhishek Sharma, and Maks Ovsjanikov. Unsupervised deep learning for structured shape matching. In *ICCV*, 2019.

[10] Souhaib Attaiki, Gautam Pai, and Maks Ovsjanikov. DPFM: Deep partial functional maps. In *3DV*, 2021.

[11] Marvin Eisenberger, Aysim Toker, Laura Leal-Taixé, and Daniel Cremers. Deep Shells: Unsupervised shape correspondence with optimal transport. *NeurIPS*, 2020.

[12] Nicolas Donati, Etienne Corman, and Maks Ovsjanikov. Deep orientation-aware functional maps: Tackling symmetry issues in shape matching. In *CVPR*, 2022.

[13] Maks Ovsjanikov, Mirela Ben-Chen, Justin Solomon, Adrian Butscher, and Leonidas Guibas. Functional maps: a flexible representation of maps between shapes. *ACM TOG*, 31(4):1–11, 2012.

[14] Nicholas Sharp, Souhaib Attaiki, Keenan Crane, and Maks Ovsjanikov. DiffusionNet: Discretization agnostic learning on surfaces. *ACM TOG*, 41(3):1–16, 2022.

[15] Abhishek Sharma and Maks Ovsjanikov. Weakly supervised deep functional maps for shape matching. *NeurIPS*, 33, 2020.

[16] Silvia Biasotti, Andrea Cerri, Alex Bronstein, and Michael Bronstein. Recent trends, applications, and perspectives in 3d shape similarity assessment. In *CGF*, volume 35, pages 87–119, 2016.

[17] Yusuf Sahillioğlu. Recent advances in shape correspondence. *The Visual Computer*, 36(8):1705–1721, 2020.

[18] Davide Eynard, Emanuele Rodola, Klaus Glashoff, and Michael M Bronstein. Coupled functional maps. In *3DV*, pages 399–407. IEEE, 2016.

[19] Dorian Nogneng and Maks Ovsjanikov. Informative descriptor preservation via commutativity for shape matching. *CGF*, 36(2):259–267, 2017.

[20] Oliver Burghard, Alexander Dieckmann, and Reinhard Klein. Embedding shapes with green's functions for global shape matching. *Computers & Graphics*, 68:1–10, 2017.

[21] Jing Ren, Adrien Poulenard, Peter Wonka, and Maks Ovsjanikov. Continuous and orientation-preserving correspondences via functional maps. *ACM TOG*, 37(6):1–16, 2018.

[22] Anne Gehre, Michael Bronstein, Leif Kobbelt, and Justin Solomon. Interactive curve constrained functional maps. In *CGF*, volume 37, pages 1–12, 2018.

[23] Meged Shoham, Amir Vaxman, and Mirela Ben-Chen. Hierarchical Functional Maps between Subdivision Surfaces. *CGF*, 2019.

[24] Emanuele Rodolà, Luca Cosmo, Michael M Bronstein, Andrea Torsello, and Daniel Cremers. Partial functional correspondence. In *CGF*, volume 36, pages 222–236, 2017.

[25] Or Litany, Emanuele Rodolà, Alex M Bronstein, and Michael M Bronstein. Fully spectral partial shape matching. In *CGF*, volume 36, pages 247–258. Wiley Online Library, 2017.

[26] Jian Sun, Maks Ovsjanikov, and Leonidas Guibas. A concise and provably informative multi-scale signature based on heat diffusion. In *CGF*, volume 28, pages 1383–1392, 2009.

[27] Mathieu Aubry, Ulrich Schlickewei, and Daniel Cremers. The wave kernel signature: A quantum mechanical approach to shape analysis. In *ICCV workshops*, pages 1626–1633, 2011.

[28] Jonathan Masci, Davide Boscaini, Michael Bronstein, and Pierre Vandergheynst. Geodesic convolutional neural networks on riemannian manifolds. In *ICCV Workshops*, pages 37–45, 2015.

[29] Davide Boscaini, Jonathan Masci, Emanuele Rodolà, and Michael Bronstein. Learning shape correspondence with anisotropic convolutional neural networks. *NeurIPS*, 29, 2016.

[30] Federico Monti, Davide Boscaini, Jonathan Masci, Emanuele Rodola, Jan Svoboda, and Michael M Bronstein. Geometric deep learning on graphs and manifolds using mixture model cnns. In *CVPR*, pages 5115–5124, 2017.

[31] Matthias Fey, Jan Eric Lenssen, Frank Weichert, and Heinrich Müller. Splinecnn: Fast geometric deep learning with continuous b-spline kernels. In *CVPR*, pages 869–877, 2018.

[32] Adrien Poulenard and Maks Ovsjanikov. Multi-directional geodesic neural networks via equivariant convolution. *ACM TOG*, 37(6):1–14, 2018.

[33] Ruben Wiersma, Elmar Eisemann, and Klaus Hildebrandt. CNNs on surfaces using rotation-equivariant features. *ACM TOG*, 39(4):92–1, 2020.

[34] Samuele Salti, Federico Tombari, and Luigi di Stefano. SHOT: Unique signatures of histograms for surface and texture description. *CVIU*, 125, 2014.

[35] Mehmet Aygün, Zorah Lähner, and Daniel Cremers. Unsupervised dense shape correspondence using heat kernels. In *3DV*, pages 573–582. IEEE, 2020.

[36] Dvir Ginzburg and Dan Raviv. Cyclic functional mapping: Self-supervised correspondence between non-isometric deformable shapes. In *ECCV*, pages 36–52. Springer, 2020.

[37] Marvin Eisenberger, David Novotny, Gael Kerchenbaum, Patrick Labatut, Natalia Neverova, Daniel Cremers, and Andrea Vedaldi. NeuroMorph: Unsupervised shape interpolation and correspondence in one go. In *CVPR*, 2021.

[38] Ginzburg Dvir and Raviv Dan. Dual geometric graph network (DG2N) - zero-shot refinement for dense shape correspondence. *CoRR*, abs/2011.14723, 2020.

[39] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[40] Simone Melzi, Jing Ren, Emanuele Rodolà, Abhishek Sharma, Peter Wonka, and Maks Ovsjanikov. ZoomOut: Spectral upsampling for efficient shape correspondence. *ACM TOG*, 38(6), November 2019.

[41] Jing Ren, Simone Melzi, Peter Wonka, and Maks Ovsjanikov. Discrete optimization for shape matching. In *SGP*, 2021.

[42] 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 *SIGGRAPH ASIA Courses*, 2016.

[43] Bruno Vallet and Bruno Lévy. Spectral geometry processing with manifold harmonics. In *CGF*, volume 27, pages 251–260, 2008.

[44] Danielle Ezuz and Mirela Ben-Chen. Deblurring and denoising of maps between shapes. In *CGF*, volume 36, pages 165–174, 2017.

[45] Gautam Pai, Jing Ren, Simone Melzi, Peter Wonka, and Maks Ovsjanikov. Fast Sinkhorn Filters: Using matrix scaling for non-rigid shape correspondence with functional maps. In *CVPR*, pages 384–393, 2021.

[46] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. PointNet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*, 2017.

[47] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. PyTorch: An imperative style, high-performance deep learning library. *NeurIPS*, 2019.

[48] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[49] Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis. SCAPE: shape completion and animation of people. In *SIGGRAPH*, pages 408–416. 2005.

[50] Simone Melzi, Riccardo Marin, Emanuele Rodolà, Umberto Castellani, Jing Ren, Adrien Poulenard, Peter Wonka, and Maks Ovsjanikov. Matching humans with different connectivity. In *3DOR*, pages 121–128, 2019.

[51] Yang Li, Hikari Takehara, Takafumi Taketomi, Bo Zheng, and Matthias Nießner. 4DComplete: Non-rigid motion estimation beyond the observable surface. In *ICCV*, 2021.

[52] Marvin Eisenberger, Zorah Lahner, and Daniel Cremers. Smooth shells: Multi-scale shape registration with functional maps. In *CVPR*, 2020.

[53] Thibault Groueix, Matthew Fisher, Vladimir G. Kim, Bryan C. Russell, and Mathieu Aubry. 3D-CODED: 3d correspondences by deep deformation. In *ECCV*, 2018.

[54] Qinsong Li, Shengjun Liu, Ling Hu, and Xinru Liu. Shape correspondence using anisotropic Chebyshev spectral cnns. In *CVPR*, June 2020.

[55] Vladimir G Kim, Yaron Lipman, and Thomas Funkhouser. Blended intrinsic maps. *ACM TOG*, 30(4):1–12, 2011.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [Yes]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]

3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No] The code and the data will be made publicly available upon publication.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] More details can be found in the supplementary material.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]