Neural Correspondence Prior for Effective Unsupervised Shape Matching

Anonymous Author(s)
Affiliation
Address
email

Abstract

We present Neural Correspondence Prior (NCP), a new paradigm for computing correspondences between 3D shapes. Our approach is fully unsupervised and can lead to high quality correspondences even in challenging cases such as sparse point clouds or non-isometric meshes, where current methods fail. Our first key observation is that, in line with neural priors observed in other domains, recent network architectures on 3D data, *even without training*, tend to produce pointwise features that induce plausible maps between rigid or non-rigid shapes. Secondly, we show that given a noisy map as input, training a feature extraction network with the input map as supervision, tends to remove artifacts from the input and can act as a powerful correspondence denoising mechanism, both between individual pairs and within a collection. With these observations in hand, we propose a two-stage unsupervised paradigm for shape matching, by (i) performing unsupervised training by adapting an existing approach to obtain an initial set of noisy matches, (ii) using these matches to train a network in a supervised manner. We demonstrate that this approach significantly improves the accuracy of the maps, especially when trained within a collection. We show that NCP is data-efficient, fast, and achieves state-of-the-art results on many tasks. Our code will be released after publication.

1 Introduction

Establishing dense correspondences between 3D shapes is a fundamental problem in computer vision and computer graphics, as it enables many downstream applications such as statistical shape analysis [1, 2], texture [3] and deformation transfer [4, 5], and registration [6], to name a few.

A particularly challenging setting for this task is the computation of 3D correspondences between shapes that undergo non-rigid, non-isometric deformations, and that may exhibit some partiality, such as missing semantic parts, and can be represented as sparse point clouds.

The standard approach is to formulate shape correspondence as a supervised learning problem, by relying on ground truth maps within large datasets of shape pairs. Given such ground truth, it is possible to train neural networks to either produce deformation fields [7], segmentation maps [8, 9, 10] or functional maps [11, 12, 13] between the input shapes. However, all these methods rely on the presence of labeled point-to-point correspondences, which are expensive to obtain, and are only available in a handful of cases.

At the same time, *unsupervised* techniques try to solve the matching problem by imposing structural properties on the maps, either in the intrinsic [14, 15, 16] or extrinsic domains [17, 18]. However, the priors used by these methods tend to be purely geometric (e.g., near isometry, divergence-free deformation fields), and, as we demonstrate below, are not always applicable, especially in challenging non-isometric settings.

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

In this work, we demonstrate that the neural network itself can act as a powerful prior for shape correspondence problems. Specifically, we first observe that even without training, recent neural architectures for 3D shape analysis [19, 20, 21] produce pointwise features that capture local geometry and lead to well-structured maps between shapes. Secondly and perhaps more importantly, we demonstrate an effect that we call *neural correspondence prior*. Specifically, given noisy input maps, we formulate a supervised learning problem, where we aim to learn pointwise features, using the input maps for supervision. Remarkably, we show that the correspondences computed by the trained networks are typically of higher quality than the input. Our results are consistent with the notion of *noise impedance* observed in other works on neural priors, pioneered in [22], and *neural bias* [23], which states that neural networks tend to optimize lower frequency signals before higher-frequency ones. We show that in the context of shape matching, these effects imply that neural networks can learn powerful features by training to overfit to given noisy or incoherent correspondences.

Based on these observations, we develop a two stage algorithm for unsupervised shape matching. In the first stage, we adapt an existing unsupervised neural network, using variants of standard methods from the literature. We then train a new network from scratch in a supervised manner using the noisy maps as ground truth. We demonstrate that this second stage not only helps to denoise the maps, but also improves the matching result overall and achieves state-of-the-art results on multiple tasks. We show that *neural correspondence prior* is applicable within both collections and on individual shapes pairs, to provide test time refinement of dense input maps. Since this method makes no assumptions on the topology of the shapes, it can be used for difficult cases such as non-isometric matching.

Overall, our contributions can be summarized as follows:

- We demonstrate that untrained neural networks for 3D shapes can produce pointwise features that are on par with complex axiomatic local descriptors.
- We show that when trained using artifact-laden maps for supervision, the features learned by neural networks lead to correspondences that are more coherent and are of higher quality than the input. We call this effect *Neural Correspondnece Prior*.
- Based on these insights, we develop a two-stage algorithm for unsupervised 3D non-rigid shape matching that achieves SOTA results for difficult matching scenarios such as non-isometric and point cloud matching.
- Using our unsupervised point-to-point correspondences, we propose an approach for few shot keypoint detection

2 Related Work

Shape matching is a well-studied area in computer vision and graphics, and a full overview is beyond the scope of this paper. We refer the interested reader to recent surveys [24, 25, 26, 27] for a more in-depth treatment of this field. Below, we review methods that are most related to our work, with a focus on learning based techniques, both supervised and unsupervised.

3D Non Rigid Shape Correspondence There is a vast literature on learning non-rigid shape correspondences. In terms of supervised methods, an important direction is to consider the matching problem as a dense semantic segmentation problem where the labels are vertices on some template shape [8, 9, 10], or by mapping via template deformation [7]. However, these techniques are known to require a considerable amount of data, and may fail when discretization changes [19, 11]. Another approach is to use the functional map framework that was introduced in [28] and extended in many follow-up works [29, 30, 31, 32, 33, 34, 35, 36] to name a few (see [37] for an overview). The functional map (fmap) framework is based on encoding maps and optimizing maps as small matrices in a reduced basis. The fmap formulation has been successfully used in the supervised setting to establish correspondences between complete and partial non-rigid shapes [11, 12, 13]. However, it is typically restricted to near-isometric matching between relatively clean discretizations.

The functional maps framework has also been adapted to the unsupervised setting, either by imposing structural properties on the functional maps such as bijectivity, orthogonality and commutativity with the Laplacian [15, 38], or by enforcing cycle coherence [39], geodesic distortion preservation [14], or by combining intrinsic and extrinsic properties [17, 18]. However, these methods all assume that the shapes are nearly isometric, not partial, and thus fail in these contexts. Another line of

work proposes to do unsupervised shape matching by reconstructing the input shapes in a canonical order. These approaches are based on the encoder-decoder architecture, and find matches either by nearest neighbor in the latent space [40], or by nearest neighbor on the reconstruction of the two input shapes [41, 42]. However, these methods require a lot of shapes and learning time, and have only successfully been applied to man-made shapes that do not undergo significant non-rigid deformations.

Neural Prior The neural prior was first introduced in the seminal work of Ulyanov *et al.*[22], where the authors introduced the Deep Image Prior (DIP), and showed that a randomly initialized convolutional neural network can serve as a good prior for many 2D inverse problems, such as inpainting and denoising. Since then, several modifications and improvements have been made to DIP [43, 44, 45, 46, 47], in order to improve its performance or adapt it to new tasks. Another line of work [48, 49] has theoretically studied DIP either through the lens of regularization theory or through its connection to Gaussian processes. Despite its success in 2D computer vision, neural prior has been little exploited in the 3D domain, the only application being shape reconstruction. Indeed, [50, 51] attempt to reconstruct a water-tight mesh from a point cloud, either by overfitting several local patches to the point cloud, or by shrinking an input mesh by a neural network. The objective of our work is to fill this gap, and to propose a neural prior technique for 3D matching.

Unsupervised Keypoint Detection Unlike the 2D case [52, 53, 54, 55], unsupervised 3D keypoint detection is relatively under-explored in the literature. The recently introduced KeyPointNet [56] benchmark provides a good testing bed. Prominent unsupervised keypoint detection methods are based on self-supervised paradigm using an encoder-decoder network, with a reconstruction error. The encoder of Skeleton Merger [57] predicts a set of salient keypoints, that the decoder uses to produce a skeleton of the shape. UKPGAN [58] harnesses the power of GANs to detect keypoints, by forcing the decoder to reconstruct the input shape using solely the set of keypoints produced by the encoder. Finally, KeypointDeformer [59] predicts a set of keypoints that allow for efficient shape manipulation. In particular, given a source and a target shape, the keypoints predicted on the source shape are used to deform it to the target shape using a cage skinning deformer [60].

3 Motivation & Method Overview

Our main goal is to develop an unsupervised learning-based shape correspondence method that would take as input a pair of shapes $\mathcal{M}$ and $\mathcal{N}$ and produce a dense correspondence, also called a map, $T : \mathcal{M} \to \mathcal{N}$. For this, we explore the direction of neural priors, and show how they can be used for 3D shape matching. The motivation behind this direction is twofold. First, as in the 2D case [61, 45], we hope that the structure of an untrained 3D neural network captures the low-level statistics of a single 3D shape, and thus can enable fine-tunning using a single pair. Second, as demonstrated in DIP [22], deep neural networks have high noise impedance, which allows them to learn good features from noisy inputs. This property can be used to guide the network to good local minima while trying to use, in our case, noisy point-to-point (p2p) matches for supervision.

Feature embeddings. In this work, we formulate the shape matching problem for both man-made and organic shapes, by learning pointwise features. Such features can be used to compute correspondences either through nearest neighbor search or with minimal post-processing. We focus on features that are learned by a neural network. Specifically, the network N takes as input a 3D shape $\mathcal{M}$ and produces, as output, a feature vector $N(\mathcal{M})_x \in \mathbb{R}^d$, for every point of the shape $x \in \mathcal{M}$. When considering the entire object as a whole, we call the set $N(\mathcal{M})_x$ for all $x \in \mathcal{M}$ a *feature embedding* of shape $\mathcal{M}$.

Our method is based on two main observations: *neural bias* for untrained networks, and an effect that we call *neural correspondence prior*. Below we describe each of these effects, and then describe our unsupervised shape matching method in Sec. 4.1.

3.1 Neural bias for pointwise features

Our first observation is that feature embeddings produced by neural networks have a particular structure, which can be exploited in the context of computing features for shape correspondence. To highlight this effect we performed the following experiment. We take the test set of the FAUST-remeshed (**FR**) and SCAPE-remeshed (**SR**) datasets introduced in [33], and used in many recent works [11, 17, 38], and compute the correspondences using the pointwise features extracted *by a randomly initialized* DiffusionNet network [19]. We use the default variant of DiffusionNet, which

Table 1: Matching accuracy of DiffusionNet *with random weights*, compared to multiple shape matching methods, on the test set of **FR** and **SR**. Values are mean geodesic error $\times 100$ on unit-area shapes. We report the mean and standard deviation of three random runs.

Method / Dataset	Method Type	**FR**	**SR**
HKS [68] + NN	*Axiom*	26.8	28.6
WKS [69] + NN	*Axiom*	24.8	26.0
SHOT [70] + NN	*Axiom*	37.7	38.4
DiffusionNet [19] + NN	*Rand.init*	17.8 ± 0.1	20.3 ± 0.3
BCICP [33]	*Axiom*	15.	16.
FMNet [13]	*Sup*	11.	17.
SURFMNet [15]	*Unsup*	15.	12.
Unsup FMNet [14]	*Unsup*	10.	16.
HKS + FMAP	*Axiom*	20.2	26.9
WKS + FMAP	*Axiom*	20.1	27.1
SHOT + FMAP	*Axiom*	19.8	20.0
DiffusionNet + FMAP	*Rand.init*	$\mathbf{8.9 \pm 0.02}$	$\mathbf{13.9 \pm 0.3}$

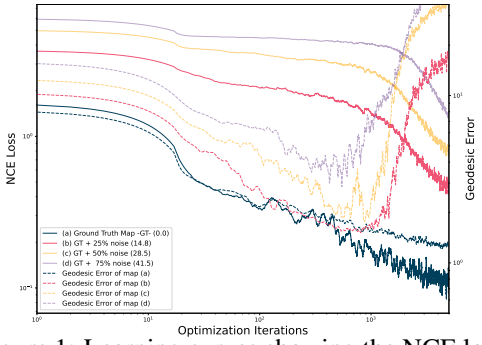

Figure 1: Learning curves showing the NCE loss for: the ground truth map (GT), and some noisy maps with different level of noise. Numbers in parentheses represent the geodesic error of the input maps. Observe the noise impedance and geodesic error decrease of intermediate maps.

takes as input the XYZ coordinates of the shapes and produces a 128-dimensional feature vector for every point. We emphasize that the weights of the network are set randomly and without any training. We then used the extracted features to produce p2p maps either via simple nearest neighbor (NN) search in the feature space, or with the functional map framework [62] (FMAP). We compare the results to maps produced using the same procedure with classical axiomatic features such as SHOT [63], or using other axiomatic and training-based non-rigid correspondence methods.

As shown in Tab. 1, remarkably, the features computed with an *untrained* DiffusionNet network perform on par or better than axiomatic features and even outperform the *supervised* learning method, FMNet [64], based on training MLPs to refine pre-defined axiomatic features, as well as recent unsupervised extensions of that approach. We attribute the fact that a randomly-initialized DiffusionNet performs better than *trained* point-wise MLPs to the the spatially-aware nature of the architecture of the network, which uses diffusion to simulate intrinsic convolution, thereby providing a strong neural prior [65, 66, 51]. Notice that using untrained features for matching was recently used in the context of 2D keypoint matching in [67].

3.2 NCP: Neural Correspondence Prior

While the previous effect relates to properties of untrained networks, in this work, we also observe another, complementary phenomenon that we call *Neural Correspondence Prior*, which we exploit in our approach below to train powerful feature extractors. This effect can be formulated as follows. Suppose we are given a fixed map $T_{\mathcal{MN}}$, between some shapes $\mathcal{M}$ and $\mathcal{N}$. We can formulate an optimization problem, where we aim to learn the feature embedding of $\mathcal{M}$ and $\mathcal{N}$ so that the induced map, e.g., computed via nearest neighbors in the feature space: i.e. $x \rightarrow \arg\min_y \|N(\mathcal{M})_x - N(\mathcal{N})_y\|$, is as close as possible to $T_{\mathcal{MN}}$. Note that unlike the neural bias mentioned in the previous section, here we formulate an optimization problem, where the parameters of the network are optimized to fit to the given input map. Our key observation is that when the target map T is noisy, networks tend to produce well-structured feature embeddings until very late stages of the optimization. I.e., throughout the early stages of optimization, the maps computed by the optimized features tend to be of *higher accuracy than the input* noisy map used for supervision.

To demonstrate the effect of NCP quantitatively, we performed an experiment in which we corrupted the ground truth map between a pair of shapes from the **FR** dataset, and then trained the DiffusionNet network, with a gradient descent optimizer using standard NCE loss [71] to overfit to those corrupted maps. Specifically, to supervise this training, we use: (*a*) the ground truth (GT) map between the shapes, (*b*) the GT map where 25% of its entries have been assigned to random vertices, (*c*) the GT map with 50% noise, (*d*) the GT map with 75% noise. The results of this experiment are shown in Fig. 1. We plot both the NCE loss being optimized as well as the *geodesic error* of the intermediate maps with respect to the ground truth.

As shown in Fig. 1, while the network can easily adapt to the correct map, it has difficulty minimizing the loss in the case of noisy maps. Moreover, and perhaps more remarkably, observe that the geodesic

error starts to decrease *towards lower values than the input map*, in the first iterations of optimization and diverges only in the later stages, when the network starts to overfit to the noise in the map. This highlights that the intermediate maps tend to be of higher quality than the supervisory signal, and implies that such optimization with a neural network and early stopping can be used as an effective map denoising mechanism (Sec. 4.2). It should be mentioned that this effect does not depend on the shape pair, and other examples of different pairs are provided in Sec. 5.1.2 and the supplementary.

We observe this effect to hold very broadly in shape correspondence problems. For a single shape pair with a noisy input map, eventually any map can be learned by the network if the parametrization and the number of training iterations is sufficient. However, the network architecture strongly regularizes the map search space, providing low impedance to 'signal' and high impedance to 'noise', resulting in an optimization trajectory that either converges to a good local minimum or passes near one [22]. Furthermore, in the presence of *shape collections*, this effect is even stronger and, as we demonstrate below, significantly helps to regularize computed features so that no early stopping or post-processing of the results becomes necessary (see our Algorithm 1 that we explain in detail in Sec. 4.1).

In addition to the network architecture itself, we attribute the *Neural Correspondence Prior* to the spectral bias principle [23, 72], which states that neural networks tend to learn low frequencies in the early stages of training, and that low frequencies are more robust to random perturbations of network parameters. Since we formulate shape matching as the problem of computing optimal features, spectral bias, in our context, implies that even given a noisy map as input, the *feature embeddings* produced by the optimized neural network tend *to be smooth*, especially in the early stages of the optimization. In other words, it is significantly harder for the network to produce a noisy, high-frequency feature embedding than a smooth one. Furthermore, a smooth feature embedding will tend to produce smooth correspondences between shapes, as suggested by the following theorem, which we state here informally and prove in the supplementary.

Theorem 1 *Given two smooth surfaces $\mathcal{M}, \mathcal{N}$, if the feature embeddings given by a network N are smooth injective maps into $\mathbb{R}^d$, i.e., the mapping $\mathcal{M} \to \mathbb{R}^d$ given by $x \to N(\mathcal{M})_x$ for all $x \in \mathcal{M}$ is smooth (and similarly for $\mathcal{N}$), then the map $T_{nn} : \mathcal{M} \to \mathcal{N}$ given by $T_{nn}(x) = \arg\min_y \|N(\mathcal{M})_x - N(\mathcal{N})_y\|$ will generically be at least C^1 continuous up to sets of measure 0.*

This result highlights the fact that smooth feature embeddings, generically, translate into continuous maps, which is a desirable property in most typical correspondence scenarios.

4 Method Description

Based on the observations above, in this section, we introduce contexts in which the NCP can be exploited, resulting in two algorithms: one for unsupervised shape matching, and the second for p2p map denoising.

4.1 NCP within a shape collection

Given a collection of shapes $\{\mathcal{N}_i\}$, we propose to exploit the NCP in two ways. The first configuration assumes that we are also given a collection of noisy maps $\{T_j\}$ between some shape pairs in $\{\mathcal{N}_i\}$ as input. In that case, we propose to train a randomly-initialized neural network to produce feature embedding for each shape, s.t., the induced maps are as close as possible to the target maps $\{T_j\}$. We provide the exact choice of the loss function depending on the nature of the underlying shapes in Sec. 5 below.

In the second configuration, no maps are given as input. In this case, we propose a two stage pipeline where in the first stage, we adapt an existing unsupervised correspondence method to obtain possibly very noisy initial correspondences. We then formulate a supervised learning problem, where we learn feature embeddings that would induce these computed correspondences as a target. Once the network is trained, we establish final correspondences via simple nearest neighbor search in the feature space. Note that the trained network can be used to establish correspondences both between shapes within the given shape collection, but also *across new, never seen shape pairs.* In our evaluation below, and unless otherwise noted, we always evaluate on a test set, never seen during training. We dubbed the resulting algorithm **NCP-UN**, and we summarize it in Algorithm 1.

We argue that it is advantageous to perform the second step for the entire collection at once, instead of doing it for each pair individually for two main reasons: first, it is faster and avoids the need for test-

Algorithm 1: NCP-UN: **N**eural-**C**orrespondence-**P**rior based **UN**supervised Shape Matching

1 **Input:** Collection of shapes $\{\mathcal{N}_i\}$
2 **Output:** p2p maps between shapes of the test set of $\{\mathcal{N}_i\}$
1: Train a variant of an off-the-shelf unsupervised matching method $\mathbf{UM}$ on the train set of $\{\mathcal{N}_i\}$
2: Predict p2p maps $\{T_i\}$ on the train set of $\{\mathcal{N}_i\}$ using trained $\mathbf{UM}$
3: Use $\{T_i\}$ to supervise the training of a randomly-initialized neural network $\mathbf{RN}$ with a p2p loss
4: Predict Feature Embedding $\{\mathbf{RN}(\mathcal{N}_i)\}$ on the test set of $\{\mathcal{N}_i\}$ using the trained $\mathbf{RN}$
5: Use $\{\mathbf{RN}(\mathcal{N}_i)\}$ to establish correspondences either using nearest neighbor or functional map pipeline.

time optimization, and second, because the collection provides an additional layer of regularization. In fact, since we optimize for features on each shape, it is difficult to compute features that would overfit to all $O(N^2)$ noisy maps in a collection of N shapes, as the noise across pairs is typically not consistent. We observe that the noise impedance property is even stronger in the case of a collection, and the produced maps are better than if the refinement is done per pair.

For our applications (see Sec. 5), we used the following design choices. For the $\mathbf{UM}$ method, we used the unsupervised functional map method [15, 38], with a DiffusionNet [19] backbone for 3D meshes, and Point-MLP [73] backbone for point clouds. The same backbone was used as $\mathbf{RN}$ for step 3. in our Algorithm 1. Concerning the p2p loss in step 3, we used the NCE loss [71] for 3D meshes and the LIE loss [74] for point clouds. We also used a variant of our approach, dubbed NCP-UN$_{\text{fmap}}$, that uses the FMAP framework in steps 3 and 5 of Algorithm 1. In this variant, we use the functional maps-based method in [11] to formulate the supervised loss in step 3., and similarly, use the learned features, as in [11], to estimate the final correspondences at test time. We use this variant especially on organic shapes, where the Laplacian basis tends to be of high quality.

We want to emphasize, first, that in all of our applications below, we followed the best practices depending on the nature of shapes (e.g., organic shapes vs. point clouds), and second, our observations regarding NCP are independent from the choice of loss and network architecture. A demonstration of the general nature of NCP, as well as more details about the implementation are provided in the supplementary. Finally, *in all* cases, we observe a significant improvement in results when using the second stage of our NCP-UN pipeline (supervised learning) compared to the initial input maps.

4.2 Test Time Denoising

In addition to the previous setting, we also observe that NCP can be used to denoise a correspondence when given a single shape pair. Specifically, given a noisy initial map, we train a neural network to produce feature embeddings that would induce the given input map, used as supervision. Again, we adapt the exact choice of loss to the nature of the shapes and describe the exact options in Sec. 5 below.

Test time denoising can be advantageous in the absence of a shape collection. At the same time, as mentioned in the original DIP work [22], in a case of a single shape pair, early stopping must be used in order to stop training at the best local minimum. For the shape matching task, we propose to use a cycle loss based on bijectivity as the criterion for stopping. Given a pair of shapes $\mathcal{M}, \mathcal{N}$, and two maps between them $\Pi_{\mathcal{MN}}$ and $\Pi_{\mathcal{NM}}$ computed via nearest neighbors between optimized features, and represented as binary matrices, we compute $L_{cycle} = \|X_{\mathcal{N}} - \Pi_{\mathcal{NM}}\Pi_{\mathcal{MN}}X_{\mathcal{N}}\|_F^2$, where $X_{\mathcal{N}}$ is the matrix of xyz coordinates of shape $\mathcal{N}$. We overfit the network for a number of iterations, and stop when L_{cycle} stops improving after a predetermined patience period. The weights produced that produce in the minimum value of L_{cycle} are used to compute the pointwise features that we then use for extracting the p2p map, using nearest neighbors in the feature space.

Correspondences as a Tool for Downstream Tasks Given computed correspondences, we also use them to to solve multiple downstream tasks. Indeed, since our proposed algorithm is unsupervised, we can use its output, i.e., p2p maps, to transfer multiple quantities between shapes, such as key points. For this, we only need a few labeled examples with the quantities we want to transfer, resulting in a few-shot algorithm. In this paper, we propose to focus on key point detection and shape segmentation.

Note, however, that a special attention must be paid, given the possibility of missing parts due, for example, to acquisition errors, or to individual shape function (e.g., a chair may or may not have an armrest). Having a set of labeled shapes, we want to 1) filter out correspondences that do not exist due to partiality 2) aggregate multiple correspondences if they exist. We propose to use cycle consistency loss as metric to eliminate non-existing correspondences, and weighted spatial aggregation for merging multiple correspondences if they exist. We provide implementation details and an example of such an application in the supplementary.

5 Results & Applications

In this section, we provide results in a wide range of challenging tasks, showing the efficiency and robustness of our paradigm to different types of data and tasks. In particular, we consider the tasks of shape correspondence between man-made shapes 5.1.1, non-rigid shape correspondence 5.1.2 and part segmentation 5.2. The fact that our model does not relies on a geometric prior allows it to provide good results on different types of data, including isometric and non-isometric, organic and man-made data. Our complete implementation will be released after publication.

Datasets We conduct our experiment on four different datasets, both on point clouds and 3D meshes. To the best of our knowledge, there is no point cloud dataset that provide dense point-to-point ground truth correspondences. For this purpose, we follow the setup of [40], and use KeyPoint-Net dataset [56] for man-made shape correspondence experiment 5.1.1. KeypointNet is composed of over 8K models and 100K semantic keypoints labeled by professional humans. We use the same train/validation/test splits as the original paper [56]. For the part segmentation experiment 5.2, we use the ShapeNet Part dataset [75], which is composed of 16881 shapes of 16 categories, where each category has a different number of parts ranging from 2 to 6.

For the 3D meshes, we use the organic, non-isometric non-rigid SMAL-based [76, 36] dataset and the SHREC'20 Non-Isometric benchmark [77] for our non-rigid correspondence experiment 5.1.2, as these are examples of challenging *non-isometric* datasets. The former dataset is composed of 50 animal shapes represented as 3D meshes. 25 shapes are used for training, and the remaining 25 are used for testing. It should be noted that the set of animals and poses seen in training is different from the one used in testing. The SHREC'20 benchmark consists of 14 scans of different animals, some of which contain holes and topological deformations. The ground truth maps are given as a set of sparsely annotated key points, which is why most supervised methods cannot be applied.

5.1 3D Shape Correspondences

5.1.1 Correspondences on Point Clouds

We test the quality of the matches produced by NCP-UN on sparse point cloud using KeyPointNet dataset. We train the first stage of our method using the unsupervised functional map method [15, 38], with a PointMLP backbone [78] and XYZ coordinates as input, and impose the bijectivity loss on the functional map. For testing, we follow the same setup as [40], and generate pairs of shapes between all point clouds in a given category, and remove the pairs that do not share the same set of semantic keypoints. We compute the L2 distance between the transferred keypoint and the ground truth one. Our method is evaluated against state-of-the-art learning-based 3D dense correspondence prediction approaches, including AtlasNetV2 [79], FoldingNet [80], IDC [42], CPAE [40], in addition to the nearest neighbor baseline between the XYZ coordinates of the points, which turns out to be a competitive baseline, not considered by previous methods.

In Fig. 2, we report the percentage of testing pairs where the distances between predicted and ground truth maps are below a given threshold for 8 categories, and report the remaining categories in the supplementary. It can be seen that our method beats all the methods on the 8 categories, and achieves state-of-the-result results.

It should be noted that our method obtains SOTA results although the method used in step 1 is not fully designed to work on point clouds (the quality of the Laplacian is not good, and the used bijectivity loss is weak). We believe that using a more sophisticated and improved method will further improve the results.

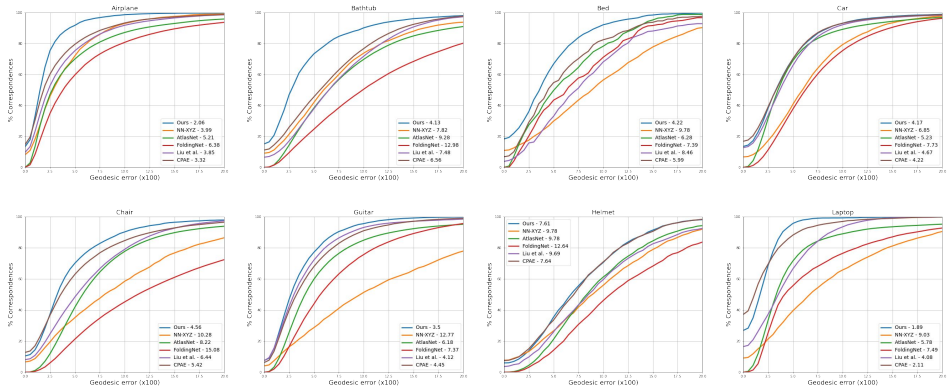

Figure 2: Correspondence accuracy on 8 categories in the KeypointNet dataset.

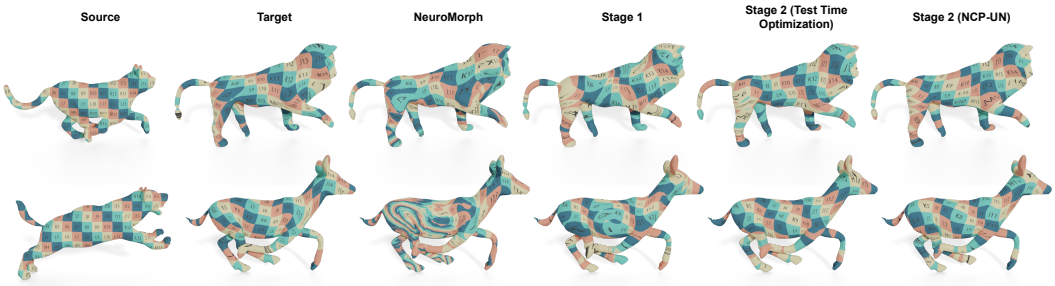

Figure 4: Qualitative results on SMAL dataset, showing the quality of the maps produced by stage 1, and refined by stage2, either using Test Time Optimization, or using NCP-UN.

5.1.2 Correspondences Between Non-Rigid Meshes

Evaluation Protocol We follow the Princeton protocol [81] for evaluating all correspondences, as used in all recent works. Specifically, we compute the pointwise geodesic error between the predicted maps and the ground truth, normalized by the square root of the surface area of the target shape.

For the non-rigid non-isometric application, we trained the unsupervised functional map network from [19], and used NCE loss for the second stage (step 3. in Algorithm 1). P2P maps were constructed by nearest neighbor between feature embeddings. Since the SHREC'20 dataset is small in number of shapes, we follow previous work [82] by training and testing on the same set, whereas for SMAL, the training and testing sets are different. For a fair evaluation, we compared our method to many state-of-the-art supervised and unsupervised methods. For the supervised methods, we compared to FMNet [13], GeomFMaps [11], GeomFMaps + DiffusionNet [19] and LIE [74]. For the unsupervised methods, we comapred to SURFMNet [15], WSupFMNet [38], WSupFMNet + DiffusionNet [19], Smooth Shells [18], Deep Shells [17] and NeuroMorph [82].

As shown in Tab. 2 and Fig. 3, our method achieves state-of-the-art results among unsupervised methods, and is equal or better than many supervised methods. The majority of unsupervised methods and some supervised methods fail because of the near-isometry assumption they make. Since our method makes no assumptions about the topology of the shape, it can perform very well on difficult scenarios such as non-isometric animals. It should be noted that these results would not be possible without the second stage of our algorithm, which is enabled by NCP, an effect that we observed for the first time in this work. For example, for the SMAL dataset, the result of the first stage is **7.8**, and the second stage improves this result to **5.8**, which is a **25%** improvement, and becomes on par or better than some supervised methods.

We provide in Fig. 4 some qualitative results showing the performance of our algorithm against state-of-the-art NeuroMorph method on SMAL dataset. It can be seen that our method produces visually plausible maps, especially after applying the second stage.

Test Time Optimization To show the usefulness of the test time optimization, as well as the early stopping criteria we introduced, the performed the following experiment. After training the first stage of our algorithm, the trained model was used to predict p2p on the test set of **SMAL** dataset. Pairs

Table 2: Matching accuracy on the SMAL test set. Values are mean geodesic error $\times 100$ on unit-area shapes

	Method	SMAL
Unsup	WSupFMNet [38]	13.3
	WSupFMNet + DiffusionNet [19]	7.8
	Smooth Shells [18]	16.3
	DeepShells [17]	15.2
	NeuroMorph [82]	23.1
Sup	FMNet [13]	47.1
	GeomFMaps [11]	8.8
	GeomFMaps + DiffusionNet [19]	6.6
	LIE [74]	20.0
Unsup	Ours	6.0
	Ours (fmap)	**5.8**
	Ours (Test-Time Optimization)	6.4

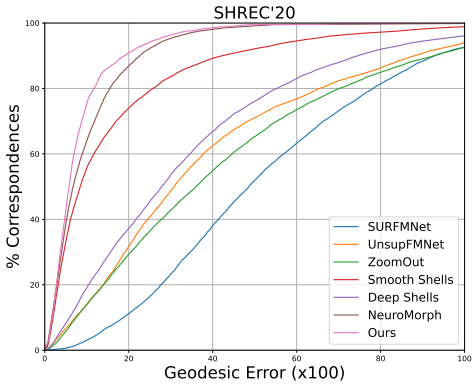

Figure 3: Correspondence quality of different methods on the test set of SHREC'20 Non-Isometric dataset.

Table 3: Part label transfer (in the average IOU(%)) for 16 categories in the ShapeNet part dataset.

	pla.	bag	cap	car	cha.	ear.	gui.	kni.	lam.	lap.	bik.	mug.	pis.	roc.	ska.	tab.	avg.
IDC [42]	60.1	56.2	59.7	-	72.2	45.3	**81.5**	66.4	42.6	88.5	40.5	87.5	**66.4**	37.2	50.7	70.4	61.7
CPAE [40]	61.3	59.3	61.6	-	72.6	55.5	78.9	71.3	53.2	89.9	55.4	86.5	66.2	40.2	61.8	72.5	65.8
Ours	**63.7**	**66.7**	**68.7**	**57.4**	**80.2**	**59**	78.8	**72.5**	**61.9**	**91.4**	**57.2**	**89.5**	61.4	**44.2**	**63.6**	**79.2**	**69.2**

of shapes are then created, and we use our test time optimization method (see Sec. 4.2) to improve upon the maps produced by the first stage. Results are reported in Tab. 2, see row "Ours (Test-Time Optimization)". It can be seen, that similar to the previous case, training a random network on top of artifact-laden maps helps denoise them. We also show in Fig. 4 some qualitative results that show the effect of test time optimization, and compare it to NCP-UN.

The results of this experiment confirm our hypothesis that doing the second stage on a collection, instead of individual pairs, provides a second layer of regularization, as the results are better, in addition to improving computational complexity. In fact, the second stage using a neural network takes around **15 minutes**, meanwhile, doing it on each pair individually takes around **160 minutes**, which represents more than $10\times$ improvement in running time.

5.2 Part Segmentation Transfer

We further validate our approach on the part label transfer task on PartNet dataset following the setup of [40]. Result are summarized in Tab. 3, where the average intersection-over-union IOU is reported. Our method outperforms the baselines on 14 out of 16 categories, achieving a state of the art result by more than 3% IOU in average. In the classes with a large intra-class variations such as Lamps, our method outperforms the baselines up to 8 % average IOU points. In addition to being performant, our method is extremely fast. In fact, training both stage of our method takes in average 1 hour and half, this is more efficient in comparison to the baseline, where CPAE for example take around 24 hours to be trained for one category.

6 Conclusion & Limitations

In this work, we demonstrated that randomly initialized neural networks can act as powerful regularizers for the 3D shape matching task, and we named this effect Neural Correspondence Prior (NCP). Through extensive experiments, we shed light on the properties of NCP and develop a two-stage algorithm for unsupervised 3D shape matching. We demonstrate the effectiveness and generality of our algorithm on a range of shape matching problems, including unsupervised shape matching on both man-made and non-rigid shapes, achieving state-of-the-art results on several benchmarks.

One limitation of our approach is that it assumes that the p2p maps used to train the second stage of the algorithm still have some structure. In fact, if the unsupervised method used in the first stage fails completely and provides random maps, our method will not converge. This problem can be mitigated by using a good application-specific unsupervised method for the first stage, especially with the recent interest and increase in unsupervised 3D learning methods.

References

[1] Federica Bogo, Javier Romero, Matthew Loper, and Michael J. Black. FAUST: Dataset and evaluation for 3d mesh registration. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, June 2014.

[2] Leonid Pishchulin, Stefanie Wuhrer, Thomas Helten, Christian Theobalt, and Bernt Schiele. Building statistical shape spaces for 3d human modeling. *Pattern Recognition*, 67:276–286, July 2017.

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

[4] Robert W. Sumner and Jovan Popović. Deformation transfer for triangle meshes. *ACM Transactions on Graphics*, 23(3):399–405, August 2004.

[5] Ilya Baran, Daniel Vlasic, Eitan Grinspun, and Jovan Popović. Semantic deformation transfer. *ACM Transactions on Graphics*, 28(3):1–6, July 2009.

[6] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Fast global registration. In *Computer Vision – ECCV 2016*, pages 766–782. Springer International Publishing, 2016.

[7] Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. 3d-coded: 3d correspondences by deep deformation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 230–246, 2018.

[8] Jonathan Masci, Davide Boscaini, Michael Bronstein, and Pierre Vandergheynst. Geodesic convolutional neural networks on riemannian manifolds. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 37–45, 2015.

[9] 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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5115–5124, 2017.

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

[11] Nicolas Donati, Abhishek Sharma, and Maks Ovsjanikov. Deep geometric functional maps: Robust feature learning for shape correspondence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8592–8601, 2020.

[12] Souhaib Attaiki, Gautam Pai, and Maks Ovsjanikov. DPFM: Deep partial functional maps. In *2021 International Conference on 3D Vision (3DV)*. IEEE, December 2021.

[13] Or Litany, Tal Remez, Emanuele Rodola, Alex Bronstein, and Michael Bronstein. Deep functional maps: Structured prediction for dense shape correspondence. In *Proceedings of the IEEE international conference on computer vision*, pages 5659–5667, 2017.

[14] Oshri Halimi, Or Litany, Emanuele Rodola, Alex M Bronstein, and Ron Kimmel. Unsupervised learning of dense shape correspondence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4370–4379, 2019.

[15] Jean-Michel Roufosse, Abhishek Sharma, and Maks Ovsjanikov. Unsupervised deep learning for structured shape matching. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1617–1627, 2019.

[16] E. Rodolà, L. Cosmo, M. M. Bronstein, A. Torsello, and D. Cremers. Partial functional correspondence. *Computer Graphics Forum*, 36(1):222–236, February 2016.

[17] Marvin Eisenberger, Aysim Toker, Laura Leal-Taixé, and Daniel Cremers. Deep shells: Unsupervised shape correspondence with optimal transport. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 10491–10502. Curran Associates, Inc., 2020.

[18] Marvin Eisenberger, Zorah Lähner, and Daniel Cremers. Smooth shells: Multi-scale shape registration with functional maps. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12262–12271, 2020.

[19] Nicholas Sharp, Souhaib Attaiki, Keenan Crane, and Maks Ovsjanikov. Diffusionnet: Discretization agnostic learning on surfaces. *ACM Trans. Graph.*, 01(1), 2022.

[20] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E. Sarma, Michael M. Bronstein, and Justin M. Solomon. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics (TOG)*, 2019.

[21] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems*, pages 5105–5114, 2017.

[22] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Deep image prior. *arXiv:1711.10925*, 2017.

[23] Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. In *Proceedings of the 36th International Conference on Machine Learning*, Proceedings of Machine Learning Research, pages 5301–5310. PMLR, 2019.

[24] Yusuf Sahillioglu. Recent advances in shape correspondence. *The Visual Computer*, 36:1705 – 1721, 2019.

[25] S. Biasotti, A. Cerri, A. Bronstein, and M. Bronstein. Recent trends, applications, and perspectives in 3d shape similarity assessment. *Computer Graphics Forum*, 35(6):87–119, October 2015.

[26] Bailin Deng, Yuxin Yao, Roberto M Dyke, and Juyong Zhang. A survey of non-rigid 3d registration. *arXiv preprint arXiv:2203.07858*, 2022.

[27] G. K. L. Tam, Zhi-Quan Cheng, Yu-Kun Lai, F. C. Langbein, Yonghuai Liu, D. Marshall, R. R. Martin, Xian-Fang Sun, and P. L. Rosin. Registration of 3d point clouds and meshes: A survey from rigid to nonrigid. *IEEE Transactions on Visualization and Computer Graphics*, 19(7):1199–1217, July 2013.

[28] Maks Ovsjanikov, Mirela Ben-Chen, Justin Solomon, Adrian Butscher, and Leonidas Guibas. Functional Maps: A Flexible Representation of Maps Between Shapes. *ACM Transactions on Graphics (TOG)*, 31(4):30, 2012.

[29] Emanuele Rodolà, Luca Cosmo, Michael M Bronstein, Andrea Torsello, and Daniel Cremers. Partial functional correspondence. In *Computer Graphics Forum*, volume 36, pages 222–236. Wiley Online Library, 2017.

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

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

[32] Meged Shoham, Amir Vaxman, and Mirela Ben-Chen. Hierarchical functional maps between subdivision surfaces. *Computer Graphics Forum*, 38(5):55–73, August 2019.

[33] Jing Ren, Adrien Poulenard, Peter Wonka, and Maks Ovsjanikov. Continuous and orientation-preserving correspondences via functional maps. *ACM Transactions on Graphics (TOG)*, 37(6), 2018.

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

[35] Dorian Nogneng and Maks Ovsjanikov. Informative descriptor preservation via commutativity for shape matching. *Computer Graphics Forum*, 36(2):259–267, May 2017.

[36] Riccardo Marin, Souhaib Attaiki, Simone Melzi, Emanuele Rodolà, and Maks Ovsjanikov. Why you should learn functional basis, 2021.

[37] Maks Ovsjanikov, Etienne Corman, Michael Bronstein, Emanuele Rodolà, Mirela Ben-Chen, Leonidas Guibas, Frederic Chazal, and Alex Bronstein. Computing and processing correspondences with functional maps. In *ACM SIGGRAPH 2017 Courses*. ACM, July 2017.

[38] Abhishek Sharma and Maks Ovsjanikov. Weakly supervised deep functional maps for shape matching. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 19264–19275. Curran Associates, Inc., 2020.

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

[40] An-Chieh Cheng, Xueting Li, Min Sun, Ming-Hsuan Yang, and Sifei Liu. Learning 3d dense correspondence via canonical point autoencoder, 2021.

[41] Zerong Zheng, Tao Yu, Qionghai Dai, and Yebin Liu. Deep implicit templates for 3d shape representation, 2020.

[42] Feng Liu and Xiaoming Liu. Learning implicit functions for topology-varying dense 3d shape correspondence, 2020.

[43] Reinhard Heckel and Paul Hand. Deep decoder: Concise image representations from untrained non-convolutional networks. *International Conference on Learning Representations*, 2019.

[44] Reinhard Heckel and Mahdi Soltanolkotabi. Compressive sensing with un-trained neural networks: Gradient descent finds the smoothest approximation. *ArXiv*, abs/2005.03991, 2020.

[45] Gary Mataev, Michael Elad, and Peyman Milanfar. Deepred: Deep image prior powered by red. *ArXiv*, abs/1903.10176, 2019.

[46] Jiaming Liu, Yu Sun, Xiaojian Xu, and Ulugbek S. Kamilov. Image restoration using total variation regularized deep image prior. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, May 2019.

[47] Daniel Otero Baguer, Johannes Leuschner, and Maximilian Schmidt. Computed tomography reconstruction using deep image prior and learned reconstruction methods. *Inverse Problems*, 36(9):094004, sep 2020.

[48] Sören Dittmer, Tobias Kluth, Peter Maass, and Daniel Otero Baguer. Regularization by architecture: A deep prior approach for inverse problems. *Journal of Mathematical Imaging and Vision*, 62(3):456–470, oct 2019.

[49] Zezhou Cheng, Matheus Gadelha, Subhransu Maji, and Daniel Sheldon. A bayesian perspective on the deep image prior, 2019.

[50] Francis Williams, Teseo Schneider, Claudio Silva, Denis Zorin, Joan Bruna, and Daniele Panozzo. Deep geometric prior for surface reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[51] Rana Hanocka, Gal Metzer, Raja Giryes, and Daniel Cohen-Or. Point2mesh. *ACM Transactions on Graphics*, 39(4), August 2020.

[52] Tomas Jakab, Ankush Gupta, Hakan Bilen, and Andrea Vedaldi. Unsupervised learning of object landmarks through conditional image generation. In *NeurIPS*, 2018.

[53] Tomas Jakab, Ankush Gupta, Hakan Bilen, and Andrea Vedaldi. Self-supervised learning of interpretable keypoints from unlabelled videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[54] James Thewlis, Hakan Bilen, and Andrea Vedaldi. Unsupervised learning of object landmarks by factorized spatial embeddings. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[55] Olivia Wiles, A. Sophia Koepke, and Andrew Zisserman. Self-supervised learning of a facial attribute embedding from video. In *BMVC*, 2018.

[56] Yang You, Yujing Lou, Chengkun Li, Zhoujun Cheng, Liangwei Li, Lizhuang Ma, Cewu Lu, and Weiming Wang. Keypointnet: A large-scale 3d keypoint dataset aggregated from numerous human annotations. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2020.

[57] Ruoxi Shi, Zhengrong Xue, Yang You, and Cewu Lu. Skeleton merger: an unsupervised aligned keypoint detector. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2021.

[58] Yang You, Wenhai Liu, Yanjie Ze, Yong-Lu Li, Weiming Wang, and Cewu Lu. Ukpgan: A general self-supervised keypoint detector, 2020.

[59] Tomas Jakab, Richard Tucker, Ameesh Makadia, Jiajun Wu, Noah Snavely, and Angjoo Kanazawa. Keypointdeformer: Unsupervised 3d keypoint discovery for shape control. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2021.

[60] Wang Yifan, Noam Aigerman, Vladimir G Kim, Siddhartha Chaudhuri, and Olga Sorkine-Hornung. Neural cages for detail-preserving 3d deformations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 75–83, 2020.

[61] Yeonsik Jo, Se Young Chun, and Jonghyun Choi. Rethinking deep image prior for denoising. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5087–5096, October 2021.

[62] Maks Ovsjanikov, Mirela Ben-Chen, Justin Solomon, Adrian Butscher, and Leonidas Guibas. Functional maps. *ACM Transactions on Graphics*, 31(4):1–11, August 2012.

[63] Federico Tombari, Samuele Salti, and Luigi Di Stefano. Unique signatures of histograms for local surface description. In *European conference on computer vision*, pages 356–369. Springer, 2010.

[64] O. Litany, E. Rodolà, A. M. Bronstein, and M. M. Bronstein. Fully spectral partial shape matching. *Computer Graphics Forum*, 36(2):247–258, May 2017.

[65] Adam Gaier and David Ha. Weight agnostic neural networks. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d' Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[66] Andrew M. Saxe, Pang Wei Koh, Zhenghao Chen, Maneesh Bhand, Bipin Suresh, and Andrew Y. Ng. On random weights and unsupervised feature learning. Omnipress, 2011.

[67] Sunghwan Hong and Seungryong Kim. Deep matching prior: Test-time optimization for dense correspondence. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9907–9917, October 2021.

[68] Jian Sun, Maks Ovsjanikov, and Leonidas Guibas. A concise and provably informative multi-scale signature based on heat diffusion. In *Computer graphics forum*, volume 28, pages 1383–1392. Wiley Online Library, 2009.

[69] Mathieu Aubry, Ulrich Schlickewei, and Daniel Cremers. The wave kernel signature: A quantum mechanical approach to shape analysis. In *2011 IEEE international conference on computer vision workshops (ICCV workshops)*, pages 1626–1633. IEEE, 2011.

[70] Samuele Salti, Federico Tombari, and Luigi Di Stefano. SHOT: Unique signatures of histograms for surface and texture description. *Computer Vision and Image Understanding*, 125:251–264, August 2014.

[71] Saining Xie, Jiatao Gu, Demi Guo, Charles R Qi, Leonidas Guibas, and Or Litany. PointContrast: Unsupervised pre-training for 3d point cloud understanding. In *European Conference on Computer Vision*, pages 574–591. Springer, 2020.

[72] Yuan Cao, Zhiying Fang, Yue Wu, Ding-Xuan Zhou, and Quanquan Gu. Towards understanding the spectral bias of deep learning. In *IJCAI*, 2021.

[73] Xu Ma, Can Qin, Haoxuan You, Haoxi Ran, and Yun Fu. Rethinking network design and local geometry in point cloud: A simple residual mlp framework, 2022.

[74] Riccardo Marin, Marie-Julie Rakotosaona, Simone Melzi, and Maks Ovsjanikov. Correspondence learning via linearly-invariant embedding. *ArXiv*, abs/2010.13136, 2020.

[75] Li Yi, Vladimir G. Kim, Duygu Ceylan, I-Chao Shen, Mengyan Yan, Hao Su, Cewu Lu, Qixing Huang, Alla Sheffer, and Leonidas Guibas. A scalable active framework for region annotation in 3d shape collections. *SIGGRAPH Asia*, 2016.

[76] Silvia Zuffi, Angjoo Kanazawa, David Jacobs, and Michael J. Black. 3D menagerie: Modeling the 3D shape and pose of animals. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[77] Roberto M. Dyke, Yu-Kun Lai, Paul L. Rosin, Stefano Zappalà, Seana Dykes, Daoliang Guo, Kun Li, Riccardo Marin, Simone Melzi, and Jingyu Yang. SHREC'20: Shape correspondence with non-isometric deformations. *Computers & Graphics*, 92:28–43, November 2020.

[78] Xu Ma, Can Qin, Haoxuan You, Haoxi Ran, and Yun Fu. Rethinking network design and local geometry in point cloud: A simple residual MLP framework. In *International Conference on Learning Representations*, 2022.

[79] Theo Deprelle, Thibault Groueix, Matthew Fisher, Vladimir Kim, Bryan Russell, and Mathieu Aubry. Learning elementary structures for 3d shape generation and matching. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[80] Yaoqing Yang, Chen Feng, Yiru Shen, and Dong Tian. FoldingNet: Point cloud auto-encoder via deep grid deformation. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, June 2018.

[81] Vladimir G. Kim, Yaron Lipman, and Thomas Funkhouser. Blended intrinsic maps. *ACM Transactions on Graphics*, 30(4):1–12, July 2011.

[82] Marvin Eisenberger, David Novotný, Gael Kerchenbaum, Patrick Labatut, Natalia Neverova, Daniel Cremers, and Andrea Vedaldi. Neuromorph: Unsupervised shape interpolation and correspondence in one go. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7469–7479, 2021.

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] See Sec. 6
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See supplementary
   (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? [Yes] See supplementary
   (b) Did you include complete proofs of all theoretical results? [Yes] See supplementary

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] Code will be made available after publication
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Sec. 5 and Supp Mat
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] In some experiments when it's not prohibitively expensive
   (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] See Supplementary
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? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [No] The used datasets are publicly available.
   (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]